From A Medial Surface To A Mesh

Thomas Delame, Céline Roudet, Dominique Faudot

Eurographics Symposium on Geometry Processing 2012
Eitan Grinspun and Niloy Mitra
(Guest Editors)

Volume 31 (2012), Number 5

From A Medial Surface To A Mesh

T. Delamé & C. Roudet & D. Faudot

Le2i - Université de Bourgogne, France

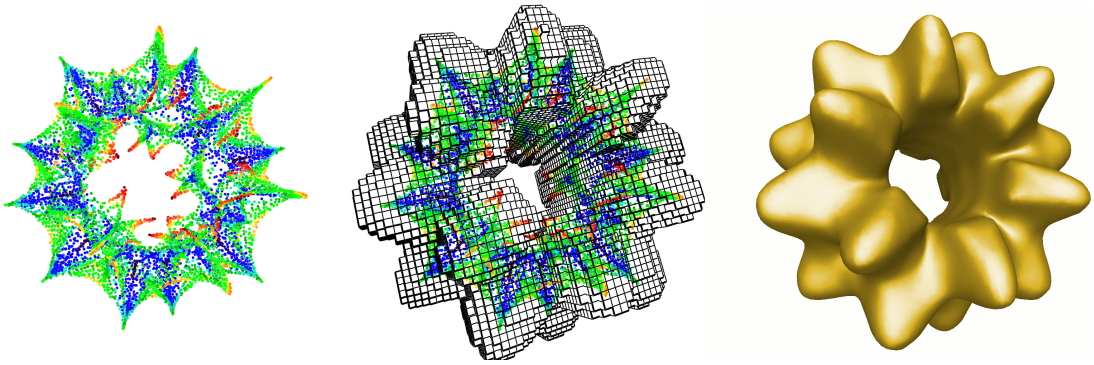


Figure 1: *Our method builds a mesh from a medial surface. Left: the input is the medial surface $\mathcal{S}$. The colors depict the radii of medial atoms, from hot color tones for small values (details) to cold color tones for big values. Middle: a coarse mesh $\mathcal{M}_1$ is first built, using a volumetric approach based on an octree construction. Right: the coarse mesh is refined to produce our final result mesh $\mathcal{M}_2$.*

Abstract

Medial surfaces are well-known and interesting surface skeletons. As such, they can describe the topology and the geometry of a 3D closed object. The link between an object and its medial surface is also intuitively understood by people. We want to exploit such skeletons to use them in applications like shape creation and shape deformation. For this purpose, we need to define medial surfaces as Shape Representation Models (SRMs). One of the very first task of a SRM is to offer a visualization of the shape it describes. However, achieving this with a medial surface remains a challenging problem.
In this paper, we propose a method to build a mesh that approximates an object only described by a medial surface. To do so, we use a volumetric approach based on the construction of an octree. Then, we mesh the boundary of that octree to get a coarse approximation of the object. Finally, we refine this mesh using an original migration algorithm. Quantitative and qualitative studies, on objects coming from digital modeling and laser scans, shows the efficiency of our method in providing high quality surfaces with a reasonable computational complexity.

Keywords: Skeleton, Medial Surface, Octree, Shape Representation Models

Categories and Subject Descriptors (according to ACM CCS): I.3.6 [Computer Graphics]: Methodology and Techniques—Graphics data structures and data types I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Curve, surface, solid, and object representations

1. Introduction

A *skeleton* is a thin structure centered in an object, describing its topology and geometry. An object must then be closed. A skeleton is an intuitive structure, in the sense that people intuitively understand links between a skeleton and the shape it describes.

One way to exploit this interesting property is to use a skeleton for applications like shape creation or deformation. This idea was explored in 2D in [DLCB], with very interesting results for these applications. However, in [DLCB], the skeleton is not a Shape Representation Model (SRM). An implicit surface model was used to render the shape. Main-

taining a coherence throughout the interaction between the skeleton and the implicit surface model was very difficult. Since a skeleton decribes both the topology and the geometry of an object, we could use a skeleton as a SRM to avoid such problems.

There exist many types of skeletons, divided into two main categories:

1. Curvilinear skeletons which are composed of curves. They were successfully used for shape registration [AJWB03], data reconstruction [TZCO09], and mesh segmentation [ATC*08]. Until recently in [SLSK07], curvilinear skeletons had no mathematical definition and they were defined only by procedural formulations. For a comparative study of such skeletons, we refer to [CSM07].
2. Surface skeletons, such as *PISA axes* [Ley87], *Midpoint locus* [BA84] or *medial surface* [Blu67], are composed of curves and surfaces. As exposed in [CSM07], a surface skeleton better captures the geometry of an object than a curvilinear skeleton, which is limited to cylindrical shapes.

A surface skeleton seems a better candidate than a curvilinear one to be a SRM, as it can describe any type of object in theory: in [SKS12], the authors showed we can tightly approximate a shape with a skeleton. Thus, we have chosen to study such a skeleton: the medial surface, introduced by [Blu67], since PISA axes [Ley87] are not well defined. Moreover, it is not clear how we can reach the geometry of the object, described by a midpoint locus [BA84]. We recall that each element of a medial surface, called a *medial atom*, is a maximal inscribed ball inside the studied object.

If we want to use a medial surface $\mathcal{S}$ as a SRM, we should be able to first define an object $\mathcal{G}$ from the geometry encoded within $\mathcal{S}$. Then, we should find a way to display it. We use the term *garbing* to denote this process of building a surface $\mathcal{G}$ from $\mathcal{S}$. Garbing is the very first task of what we can do when using a medial surface as a SRM. Unfortunately, garbing is not a straightforward task, as explained in the Section 2.2. This is why we deal with garbing in the rest of the paper, to make a step forward the use medial surfaces as SRMs.

In this paper, we propose a new garbing algorithm for a medial surface $\mathcal{S}$. From this particular skeleton, we define a shape $\mathcal{G}$ as the surface of a union of simple primitives. This shape $\mathcal{G}$ is first approximated by a coarse mesh, using a volumetric approach which is based on the construction of an octree. Then, we refine this mesh by making each of its vertices converge toward $\mathcal{G}$.

2. Previous Work

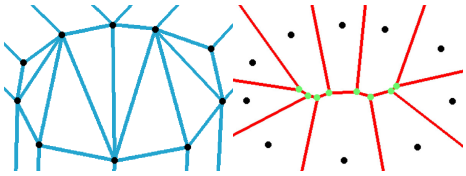
In this section, we first introduce methods that produce a medial surface $\mathcal{S}$ from a known object $\mathcal{O}$. This to explain how we can get some test skeletons for our garbing algorithm. Then, we discuss the other existing garbing techniques.

One thing to keep in mind is that $\mathcal{S}$ is always a discrete set of medial atoms, even if we use a continuous approach for our computation. This is because, in practice, people always use an approximation paradigm to extract medial surfaces [ABE09].

2.1. Continuous Medial Surface Extraction

Medial atom locii can also be seen as points inside $\mathcal{O}$ with at least two closest points on the boundary of $\mathcal{O}$. This alternative definition justifies the three kinds of approaches used to extract a medial surface $\mathcal{S}$:

- Computing bisectors: Given two objects $\mathcal{O}_1$ and $\mathcal{O}_2$, their bisector $\mathcal{B}$ is defined as the locus of equidistant points from $\mathcal{O}_1$ and $\mathcal{O}_2$. If $\mathcal{O}_1 = \mathcal{O}_2$, $\mathcal{B}$ is called an untrimmed self-bisector, and $\mathcal{S}$ is a subset of $\mathcal{B}$. This is the method used in [dOdF03].
- Using Partial Differential Equations (PDEs): Medial atoms can also be defined by an analogy, called the grass-fire analogy. $\mathcal{O}$ is filled with an equally dense grass. A fire is lit at $t = 0$ all around the boundary of $\mathcal{O}$. Quench points of fire fronts along with their time of formation are medial atoms. [KTZ95] has shown this is equivalent to consider the set of shock positions in PDEs of motion at fixed speed, in a direction initially normal to the object boundary.
- Building Voronoi diagrams from a sampling of the boundary of $\mathcal{O}$: A Voronoi diagram is the dual of a Delaunay tetrahedrization. It is composed of cells, each of which contains points located closer to a particular sample point than to any other. Those cells are delimited by Voronoi vertices. All Voronoi vertices have four closer sample points, as they are circumcenter of Delaunay tetrahedron (See Figure 2 for an illustration in 2D). In [AK00], the authors proved that a subset of the Voronoi vertices, called *poles*, converges toward the medial surface as the sampling density increases.



(a) (b)

Figure 2: *Illustration of a medial surface extraction based on a 2D Voronoi diagram. (a) A Delaunay triangulation (in blue) is made on the object surface sampling (black dots). (b) Its dual, the Voronoi diagram is represented in red. Voronoi vertices (gree dots) are approximations of the medial atoms.*

2.2. Medial Surface Garbing

Once we have a medial surface $\mathcal{S}$, there exists three classes of garbing methods: *i)* naively render the union of geometric primitives, *ii)* use an implicit surface formulation, and *iii)* mesh the object by taking into account some properties of the skeleton.

2.2.1. Naive rendering

Naive methods consist in rendering the whole set of geometric primitives, in our case: spheres. This is generally done through a ray caster, such as *POV-Ray* [pov]. A naive rendering is very simple, and do not require to use more than the list of spheres in $\mathcal{S}$. Naive rendering is then used to test the skeleton. However, the produced objects can look "bumpy" and normal discontinuities lead to annoying artifacts (See Figure 3). Moreover, as spheres are not linked in any way, it can be difficult to realize some operations on the surface of their union, such as texturing.

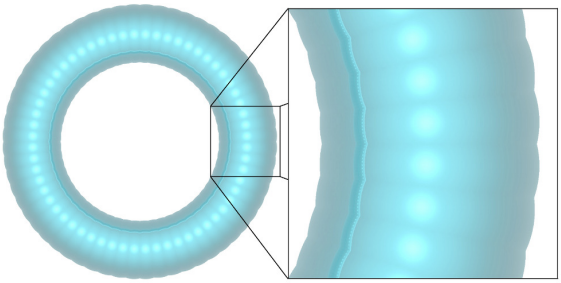


Figure 3: *Naive medial surface garbing of a torus. Bumpy pattern can be seen at each sphere intersection.*

2.2.2. Implicit surface formulation

Implicit surfaces are the level set of a scalar function F. They are known to produce very soft objects and have the ability to blend objects if we combine them. Thus implicit surfaces can be used to remove bumpy patterns. A basic way to do so, is to consider each atom as a blob [Bli82]. Considering radial basis functions centered at medial atoms locii is another possible solution, introduced in [SAAY]. We can also consider the whole skeleton $\mathcal{S}$ as a convolution skeleton; a technique used in [PABME11] for vessels or in 2D in [DLCB].

Meshing an implicit surface is done by sampling the function F. In some cases, it is hard to find a good sampling to capture the correct topology. For example, when two parts of the shape are very close from each other. Thus, sometimes the surface can be missed or unwanted blending between some logical components can be encountered.

2.2.3. Meshing the boundary

Meshing the boundary of the primitive union seems a good alternative. We can do it naively by meshing each primitive $s \in \mathcal{S}$ and make the garbing mesh evolve each time a primitive mesh is added. At each step, one of these meshes is added to the current garbing mesh, which is initially empty. To add a primitive mesh to the current garbing mesh, the intersection between them is computed and a remeshing step is used. This can be very time consuming. The surface obtained from this union of spheres can be more efficiently meshed by a *skin surface* [KV]. A skin surface is a more general formulation than considering the boundary of the union of spheres. A parameter, called the shrink factor, controls the distance of the surface around the sphere centers. However with that technique, bumpy patterns remain.

Instead of doing so, properties enclosed within $\mathcal{S}$ can be used. For example in [YFJ*03], the authors add information to each medial atom, such as object angles and local frames. Provided a quadrilateral mesh of medial atoms is known, the authors build a cubic B-Spline model from the medial surface. The result mesh is very smooth, with no bumpy patterns. However the medial surface should have no branching, which can be very restrictive in terms of available shapes. Moreover, this additional information is obtained during the skeletonization process. That means we need to know the shape described by $\mathcal{S}$ in order to get it. If we use $\mathcal{S}$ as a SRM, $\mathcal{S}$ will be modified, and object angles or local frames should be updated accordingly, but the modified shape will not be available.

The work we present in this paper proposes to mesh the surface with no more information than the medial atom positions and radii, and no restriction on the structure of the medial surface.

Outline

The next section presents our new garbing algorithm. From a medial surface $\mathcal{S}$, we define a shape $\mathcal{G}$ composed of a union of cones and spheres. This shape $\mathcal{G}$ will be approximated by a mesh $\mathcal{M}_2$ in three steps we summarize here:

- producing an octree $\mathcal{T}$ driven by four subdivision criteria which uses the information of $\mathcal{S}$ and the topological properties of the object $\mathcal{O}$ described by $\mathcal{S}$;
- meshing the boundary of $\mathcal{T}$ to obtain a coarse approximation of the surface of the union of spheres contained in $\mathcal{S}$, the result mesh of this step is called $\mathcal{M}_1$;
- refining $\mathcal{M}_1$ by a new migration method which projects elements of $\mathcal{M}_1$ on $\mathcal{G}$, the result mesh is then called $\mathcal{M}_2$ and is our garbing mesh.

Then, we present some results to validate our work. We finish with a conclusion and present some suggestions for future work.

3. Visualizing The Object

We consider an unstructured medial surface $\mathcal{S}$. This skeleton can then be seen as a set of spheres $\mathcal{S} = \{a_i = (p_i, r_i) \in \mathbb{R}^3 \times \mathbb{R}^+\}$.

3.1. Building The Octree

The first step of our algorithm consists in building a discretization of the shape described by $\mathcal{S}$. To do so, we have chosen to build a regular octree $\mathcal{T}$, because it allows the generation of uniformly sized faces, which improves the quality of the final mesh. Moreover, an octree is a spatial hierarchy. Thus, all queries asking for cells that contain point or geometric primitives will be efficient. This octree $\mathcal{T}$ will be used in the next step as a basis to extract a coarse garbing mesh.

Terminal cells of an octree are called leaves, and written $\mathcal{L}$. Each non leaf cell c has exactly eight sons $Down(c) = \{c_1, \ldots, c_8\}$. The first cell, called *Root* is at the subdivision level 0. The sons of a cell of level n are at the level $n+1$.

Each cell $c \in \mathcal{T}$ defines a volume, called an octant: $Octant(c)$. In this paper, all octants are cubes. Moreover, all spheres of $\mathcal{S}$ that intersect the octant of c are written $Spheres(c)$. If $Spheres(c)$ is empty, c is said *inactive*, otherwise c is said *active*.

The axes of $Octant(Root)$ are aligned on the principal axes of $\mathcal{S}$ and *Root* contains all spheres: $Spheres(Root) = \mathcal{S}$. The four criteria we enumerate in the next sections, are used to subdivide each cell, starting with *Root*. When a cell c is subdivided, we cut $Octant(c)$ into eight equal cubes, one for each son.

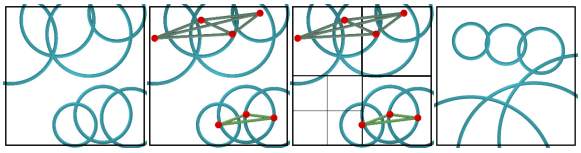
3.1.1. Criterion #1: One Logical Component By Cell

During the subdivision procedure, a leaf c may contain more than one logical component of $\mathcal{O}$. For example, c can contain spheres that belong to two different fingers of a human hand. If we do not subdivide c, those logical components (fingers in our example) would be glued when constructing the approximation of the object. Thus our subdivision process must ensure that distinct logical components are situated in different leaves. This is the purpose of this first criterion.

To implement the Criterion #1, we must face a Level of Detail (LoD) problem. At a close view, two spheres can belong to different parts of an object, e.g. distinct phalanxes of a forefinger. But at a more general view, these spheres seem to belong to the same part, in our example the same forefinger. To solve this LoD problem, we will use $Octant(c)$ as a viewing volume.

We consider that two spheres S_1 and S_2 belong to the same part of $\mathcal{O}$ with respect to c, if and only if: there exists a sequence $a_0, a_1, \ldots, a_n, a_{n+1}$ of spheres in $Spheres(c)$, with $a_0 = S_1$ and $a_{n+1} = S_2$ such that $\forall i \in [\![0, n]\!]$, a_i and a_{i+1} intersect each other.

In our implementation, we build a graph $G(\mathcal{N}, \mathcal{E})$ where the nodes $\mathcal{N}$ are elements of $Spheres(c)$ and $\mathcal{E} = \{(n_i, n_j) \in \mathcal{N} \mid n_i \text{ and } n_j \text{ intersect each other}\}$. If this graph has more than one connected component, c is subdivided (See Figure 4). This simple technique works on every test we have made on practical cases. However, if $Octant(c)$ is not small enough, small logical components can entirely fit inside a cell, and then be missed.



(a) (b) (c) (d)

Figure 4: *Illustration of the Criterion #1. a) A cell filled with 7 atoms. b) This cell does not satisfy the Criterion #1: there are two connected components if we draw the graph $G(\mathcal{N}, \mathcal{E})$. c) Two subdivision steps are necessary to validate this criterion within each cell. But this is not enough to capture the local topology. Here, our Criterion #3 (See Section 3.1.3) is needed. d) A failure case: this criterion does not detect the very small logical component.*

3.1.2. Criterion #2: A Hole Leads To Inactive Leaves

The object described by $\mathcal{S}$ can be of any genus. Consequently each hole should be identified during the octree construction, in order to capture the correct global topology of this object. The second criterion aims at ensuring that every hole leads to inactive leaves inside $\mathcal{T}$.

The main concern of this criterion is to efficiently find every holes. For that purpose, heuristics can be used during the skeletonization process. For example, we could store along with $\mathcal{S}$ a set $\mathcal{S}^-$ composed of *negative atoms*, i.e. maximal tangential balls *outside* the object. If we divide all the radii of those balls by 2, this criterion can be fulfilled by finding the active leaves that intersect one of these reduced balls, and subdivide them. Reduced balls are considered because the sets of spheres $\mathcal{S}$ and $\mathcal{S}^-$ intersect each other. Thus a cell c which intersects both elements of $\mathcal{S}$ and $\mathcal{S}^-$ could lead to an infinite subdivision process.

However, these two sets $\mathcal{S}$ and $\mathcal{S}^-$ are redundant. A consequence is that when we change one of these sets, the other one must be updated accordingly. Maintaining a coherence between those both sets is a tricky task. Instead we focus on a minimal level d that active cells must reach in order to be octree leaves. This level d acts as a LoD factor, and helps to discover every hole with respect to this LoD.

We present here a definition of d which take into account what we call the *characteristic radius* r_c. Ideally, this radius represents at best the local thickness of the object. Experimentally, we found that $r_c = \frac{r_{min} + r_{max}}{2}$ is representative, where r_{min} and r_{max} are the minimal radius and the maximal radius of $\mathcal{S}$. We ensure that every octant has a side less than $\frac{r_c}{8}$, and d is the first octree level that passes this test. See Figure 5 for an illustration of the characteristic radius.

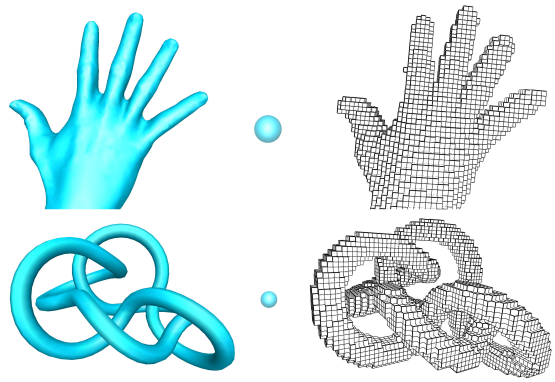


Figure 5: *Illustration of the Criterion #2 and the characteristic radius r_c. Left: original object we skeletonized. Center: a sphere with a radius equal to r_c, r_c gives an idea of the local thickness of the shape. Right: the octree only built with Criterion #2, it captures main parts of the object.*

3.1.3. Criterion #3: Two Neighbor Leaves have Intersecting Sphere Sets

This criterion is very similar to Criterion #1, as it deals with unwanted blending between distinct logical components. But unless Criterion #1 which prevents two logical components from being glued together inside a leaf, Criterion #3 avoid them to be glued by neighbor leaves. Neighbors of a cell c are all cells of the same level whose octant share at least one vertex with $Octant(c)$. Neighbors of c are written $\mathcal{N}(c)$.

Let c and c' be two active neighbor leaves. If no sphere from $Spheres(c)$ intersects a sphere of $Spheres(c')$, then we consider that c and c' have distinct logical components. Thus, those cells are subdivided to ensure that these logical components will not be glued together in the next algorithm step (See Figure 7).

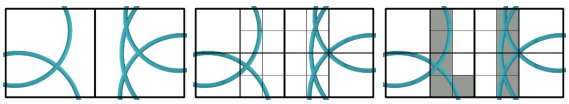


Figure 6: *Illustration of the Criterion #3. a) Two neighbor leaves have disjoint sphere sets. If we do not subdivide them, during the surface extraction, logical components corresponding to those disjoint sets will be connected. b) After one subdivision, the same situation occurs. At the second subdivision step, inactive cells appear and this criterion is valid. c) The surface approximation obtained by boundary active leaves gives now the right local topology (See Section 3.2 for the definition of a boundary leaf).*

3.1.4. Criterion #4: One Level Difference Between Adjacent Active Leaves

We impose the one level difference rule on our octree $\mathcal{T}$. Let c and c' be two active leaves, respectively at level n and n'. This rule says that if $Octant(c)$ and $Octant(c')$ share at least one edge (18-connexity), then $n = n'$, or $n = n' \pm 1$.

This rule is commonly used to control the mesh gradation, i.e. the variation in edge size [BHPF98], and to control element aspect ratios. On a more practical point of view, it eases also the search of the leaves sharing a vertex with $Octant(l)$, for $l \in \mathcal{L}$. Such leaves are called the leaf neighbors of l, $\mathcal{LN}(l)$, and should then be found at level n, $n+1$ or $n-1$ if n is the level of l.

If a leaf l of level n does not satisfy this criterion, all the cells $c \in \mathcal{LN}(l)$ of level $n-1$ are subdivided (See Figure 7).

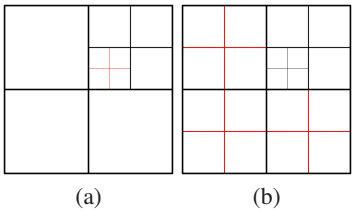


Figure 7: *Illustration of the Criterion #4. a) New cells in red are subdivided to fulfill a criterion. These cells make the fourth criterion invalid. b) Three cells of lower level are subdivided to satisfy this criterion.*

3.2. Extracting A Coarse Approximation Of The Surface

In Section 3.1, we have built an octree $\mathcal{T}$, using information captured by $\mathcal{S}$. In this section, we show how to mesh the boundary of $\mathcal{T}$. The result mesh, called $\mathcal{M}_1$, is a coarse approximation of the surface of the union of spheres contained in $\mathcal{S}$.

First, we must define what we mean by the octree boundary. Boundary faces $\mathcal{BF}$ are faces of leaf octants that separate active leaves from inactive leaves. If a leaf $l \in \mathcal{L}$ has a boundary face, we call it a boundary leaf. The boundary of $\mathcal{T}$ is then the set of boundary leaves.

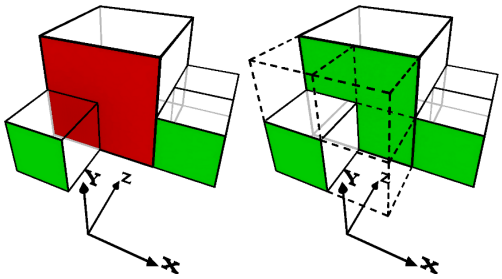
There are basically two ways to mesh boundary leaves:

1. connect their center by faces, using a face quality criterion to choose which centers to interconnect. Such criterion can take into account the face aspect ratio or the angles between edges for instance. It is a hard task to build a face quality measure which is robust enough for all the objects that can be described by $\mathcal{S}$.
2. use a subset of the boundary faces $\mathcal{BF}$, as done in [ISS09]: the technique we have chosen because of its robustness.

Each cell c is a cube composed of the following faces: $F(c) = \{F(c)_{dir},\ dir \in \{X^+,X^-,Y^+,Y^-,Z^+,Z^-\}\}$. The six directions X^+,X^-,Y^+,Y^-,Z^+,Z^- belong to the frame of $\mathcal{T}$, which is aligned along the principal directions of $\mathcal{S}$. The direction of a face gives its outward oriented normal. We write $\widetilde{f}$ the face f with a reversed normal, called the opposite face of f.

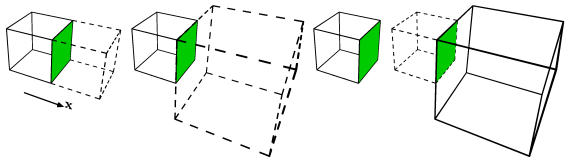
Since leaves are not at the same level, we cannot take all faces of $\mathcal{BF}$ (See Figure 8). To collect faces f in the direction dir, we process all $l \in \mathcal{L}$. We set n: the level of l, and $\mathcal{LN}(l)_{dir}$: the leaf neighbor of l in the direction dir. With these notations, a boundary face f is the mesh $\mathcal{M}_1$ if it matches one of these four configurations (See Figure 9):

1. l is active and $\mathcal{LN}(l)_{dir}$ is an inactive leaf of level n: f is $F(l)_{dir}$
2. l is active and $\mathcal{LN}(l)_{dir}$ is an inactive leaf of level $n-1$: f is $F(l)_{dir}$
3. l is active and $\mathcal{LN}(l)_{dir}$ does not exist: f is $F(l)_{dir}$
4. l is inactive and $\mathcal{LN}(l)_{dir}$ is an active leaf of level $n-1$: f is $\widetilde{\mathcal{F}(l)_{dir}}$



(a) (b)

Figure 8: *The concept of boundary face in $\mathcal{T}$. Plain lines depict active leaves, and dashed ones are for inactive leaves. (a) We want to collect Z^- oriented faces (in green). Among these faces, the red f is not taken into account. This is because it is partially occluded by another active leaf. (b) Instead, we consider the Z^+ faces of inactive cells on the other side of f, and we just take their opposite.*



(a) (b) (c) (d)

Figure 9: *The four different configurations for a X^+ face.*

When all boundary faces for the six directions are collected, $\mathcal{M}_1$ is complete. It gives a coarse but satisfying approximation of the object described by $\mathcal{S}$ (See Figure 10).

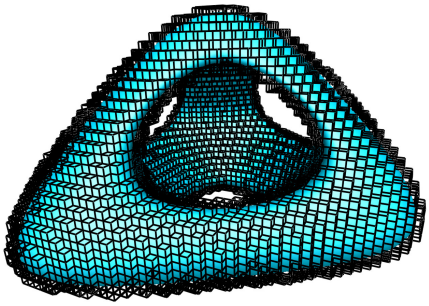


Figure 10: *$\mathcal{M}_1$, the coarse approximation of the object described by $\mathcal{S}$. Here, the original object $\mathcal{O}$ is given in blue, a shape with a genus of four. The mesh $\mathcal{M}_1$ has the same genus, and remains very close to $\mathcal{O}$.*

3.3. Refining Step

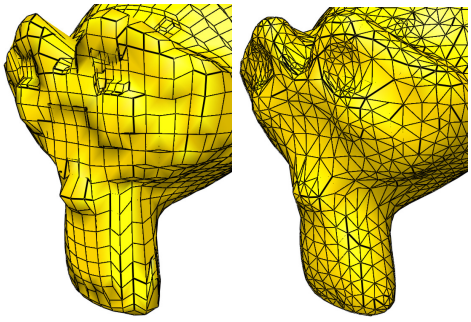
So far, we have a mesh $\mathcal{M}_1$ that is a coarse approximation of the object described by $\mathcal{S}$. However, as we can see on Figure 10, $\mathcal{M}_1$ gives a voxelized aspect to the surface. The last step of our method is a refining one, to improve both the aspect of the surface and the error made in the approximation. The result mesh of this step is $\mathcal{M}_2$.

The target shape of this refining step is $\mathcal{G}$. We cannot define it as the surface of theunion of spheres contained in $\mathcal{S}$, since it would produce the same artifacts as those exposed by Figure 3. Instead, we simply also consider a set of cones: one cone for each pair of intersecting spheres (See Figure 13 a)). A cone C for two intersecting spheres S_1 and S_2 is such that the surface of $\{C,S_1,S_2\}$ forms the convex hull of $\{S_1,S_2\}$. All these cones are inserted inside the leaves of $\mathcal{T}$ if they intersect their octants: for such a leaf l we write them $Cones(l)$. The set of all spheres and cones is written $\mathcal{P}$, and $\mathcal{G}$ is then the surface of the union of all $g \in \mathcal{P}$.

To compute $\mathcal{M}_2$, we first initialize it with the dual of $\mathcal{M}_1$. Hence every vertex v of $\mathcal{M}_2$ is the center of its associated face in $\mathcal{M}_1$. Similarly every face f of $\mathcal{M}_2$ connects vertices associated to three adjacent faces in $\mathcal{M}_1$. If we do not take the dual of $\mathcal{M}_1$, we obtain self intersecting faces after the migration step, as shown in Figure 11.

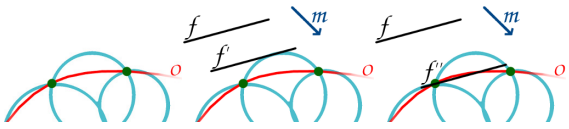
A fact to take into account, is that spheres from $\mathcal{S}$ generally go out of the skeletonized object. This is due to the approximation paradigm used for the skeletonization process: spheres of $\mathcal{S}$ only touch the original object at sampling points. Thus, the union of spheres looks bigger than the original object. If we choose to project faces of $\mathcal{M}_2$ until they touch $\mathcal{G}$: the effect get worse (See Figure 12). Instead, we consider each vertex of $\mathcal{M}_2$, and project it toward $\mathcal{G}$.

Let v be a vertex of $\mathcal{M}_2$. It belongs to an active leaf c. We will project v toward the set of primitives $\mathcal{P}(c) = Cones(c) \cap Spheres(c)$. To this end, for each geometric primitive $g \in \mathcal{P}(c)$, we define a migration vector $\overrightarrow{m}(v)_g$. The migration direction for v is then $\overrightarrow{m}(v) = -\sum_{g\in\mathcal{P}(c)} \overrightarrow{m}(v)_g$.



(a) (b)

Figure 11: *The importance of taking the dual of $\mathcal{M}_1$. (a) The migration step is directly done on $\mathcal{M}_1$. We can see some self intersecting faces. (b) The migration step is done on the dual of $\mathcal{M}_1$, there is not self intersecting faces and the mesh gradation is better.*



(a) (b) (c)

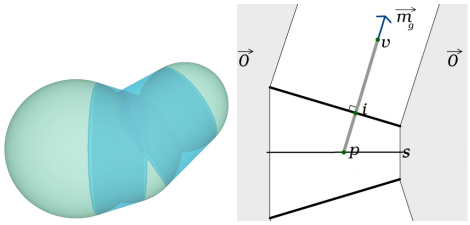
Figure 12: *Principle of our migration method. The green dots depict the boundary samples of $\mathcal{O}$ used during the skeletonization algorithm. The blue circles represent the spheres of $\mathcal{S}$. (a) The union of spheres is bigger than $\mathcal{O}$. (b) If we move a face f from $\mathcal{M}_2$ in a direction m until it touches $\mathcal{G}$, the resulting face is f'. (c) Instead, we choose to stop the migration when all the vertices of f have met an element of $\mathcal{G}$. The resulting face f'' allows to get closer to $\mathcal{O}$ than with f'. It also reduces the effect observed in (a).*

We now explain how to define $\overrightarrow{m}(v)_g$ for a primitive g. Each primitive has a symmetry axis s, which is a line for a cone and is reduced to a point for a sphere. We first compute a line l that passes through the point v and that is normal to the surface of g (See Figure 13 b)). The point i is then the intersection between l and g, and the point p is the intersection between l and s. We set $\overrightarrow{m}(g)_v = \frac{\|p-i\|}{\|v-p\|} \times \frac{\overrightarrow{pv}}{\|v-p\|}$. When we cannot define the line l for a cone, $\overrightarrow{m}(g)_v = \overrightarrow{0}$. The way $\overrightarrow{m}(g)_v$ is defined is such that $\overrightarrow{m}(v)$ is directed toward bigger and closer primitives.

Finally, every vertex v of $\mathcal{M}_2$ is projected toward $\mathcal{G}$ using $\overrightarrow{m}(v)$. $\mathcal{M}_2$ is then our garbing mesh. As such, it approximates the object described by a medial surface $\mathcal{S}$, containing no more information than a set of spheres.

4. Results

To validate our work, we rely on both quantitative and qualitative comparisons with other garbing methods. Those comparisons are made on eight medial surfaces we extracted



a) b)

Figure 13: *The use of cones. a) The set of primitives $\mathcal{P}$ is composed of three spheres from $\mathcal{S}$ and two cones which connect each pair of intersecting spheres. b) Definition of the migration vector of a vertex v for a primitive g which is here a cone.*

from models coming from laser scans or digital modeling. To extract those skeletons, we chose to implement the *powershape* algorithm [ACK01], a technique which computes the set of *poles* (See Section 2.1).

4.1. Quantitative comparison with other methods

We have first computed the error between an original shape $\mathcal{O}$ and the produced garbing mesh $\mathcal{M}_2$. This error is not only related to our work, but also to the skeletonization process: $\mathcal{S}$ is an approximation of the medial surface, and hence it introduces an approximation error. Thus, this measure gives an indication of the quality of both $\mathcal{S}$ and $\mathcal{M}_2$, with respect to $\mathcal{O}$.

We compared our work with an algorithm that produces a *skin surface* [KV]. To do so, We have used the *3D Skin Surface Meshing* package of CGAL [cga]. The computation times are almost the same as the ones obtained for with garbing method (See Table 2).

Regarding this error, we have compared two different shape distances: the Hausdorff distance, and the Root-Mean-Square (RMS) distance. All these computations were obtained with the *MESH* software [ASCE02]. We present in Table 1 these relative distances, with respect to the bounding box diagonal of $\mathcal{O}$.

Considering the Hausdorff distance, we obtain less than 0.85% for simple objects and less than 2.18% for objects with more details. Thus our garbing algorithm produces a good approximation of the shape encoded by a medial surface. Compared to the *skin surface* method [KV], we get better results for simple objects. For complex objects, i.e. with more atoms, we obtain similar results, except for the monkey. When we raise the minimal active leaf depth in $\mathcal{T}$ for that model (lines for Monkey* and Monkey** in Table 1), we manage to get the same RMS distance. Hence we compare very well to the *skin surface* algorithm, while producing meshes with less vertices (up to 27 times less vertices for the Bumpy Torus model).

	Original Object		$\mathcal{S}$	Garbing Mesh				Skin Surface			
	V	F	Atoms	V	F	*H*	*RMS*	V	F	*H*	*RMS*
Sphere	3 224	6 444	1	1 344	2 684	0.13 %	0.04 %	26	48	3.03 %	2.31 %
Torus	1 200	2 400	60	1 628	3 256	0.37 %	0.11 %	2 065	4 130	1.03 %	0.49 %
Knot	1 280	2 562	199	9 540	19 080	0.73 %	0.32 %	21 177	42 354	1.38 %	0.38 %
Eight	766	1 536	676	1 810	3 624	0.71 %	0.28 %	21 193	42 390	0.89 %	0.26 %
Hand	2 518	5 000	2 386	5 710	11 416	0.85 %	0.72 %	66 431	132 858	0.73 %	0.68 %
Monkey	7 790	15 408	6 092	3 362	6 713	2.18 %	0.27 %	165 597	331 142	1.67 %	0.17 %
Monkey*	7 790	15 408	6 092	10 075	20 139	1.64 %	0.19 %	165 597	331 142	1.67 %	0.17 %
Monkey**	7 790	15 408	6 092	27 411	54 813	1.63 %	0.17 %	165 597	331 142	1.67 %	0.17 %
Genus4	6 652	13 312	6 514	14 678	39 324	1.14 %	0.07 %	180 564	361 136	1.36 %	0.06 %
Bumpy Torus	15 876	31 752	15 611	16 296	32 592	0.94 %	0.08 %	441 006	882 012	0.49 %	0.06 %

Table 1: *Statistics for test models. The minimal active leaf depth in $\mathcal{T}$ for the Monkey model is* 5. *In order to show that we can improve the results with finer cells, we have raised the minimal depth to* 6 *for Monkey*, and to* 7 *for Monkey**.*

	Our Method	Skin Surface
Sphere	0.02	0.01
Torus	0.03	0.12
Knot	0.15	0.77
Eight	0.31	1.10
Hand	3.06	3.84
Monkey	12.23	10.84
Genus4	13.32	10.38
Bumpy Torus	28.90	26.65

Table 2: *Computation times in second for our method and the* skin surface *algorithm.*

4.2. Qualitative comparison with other methods

We also validated our work by comparing visually our garbing meshes with their corresponding original shapes. The union of spheres contained in medial surfaces were rendered by the *POV-Ray* tool, to compare them with our method. The *skin surfaces* were not considered in this comparison, because they present some artifacts which are nonexistent on *POV-Ray* images. The Figure 15 presents the results for some of the eight models we have studied.

The approximation errors given by the quantitative comparison in Section 4.1 are confirmed by those results: our garbing meshes are visually very close to the original objects. Compared to the unions of spheres, the garbing meshes are more pleasant. This is due to bumpy pattern removal, which leads to smoother surfaces.

4.3. Limitations

In practice, medial surfaces are not intended to be used for models with sharp edges or planes. Because these parts require a higher atom density to build a garbing mesh with the desired precision. Thus, our method has difficulties to deal efficiently with such models. However, if we subdivide a garbing mesh obtained for such a model and refine that subdivided mesh with our migration step, we can improve the results (See Figure 14 a) and b)).

Another limitation came from the size of the octree leaves. If they are not small enough, details can be missed by the garbing mesh, as the eyes, the nose or the mouth of the monkey in the Figure 15. Experimentally, when we raise the minimal active leaf depth in $\mathcal{T}$, thus reducing the size of octree leaves, we can recover these details, as shown by Figure 14 c), d), and e)).

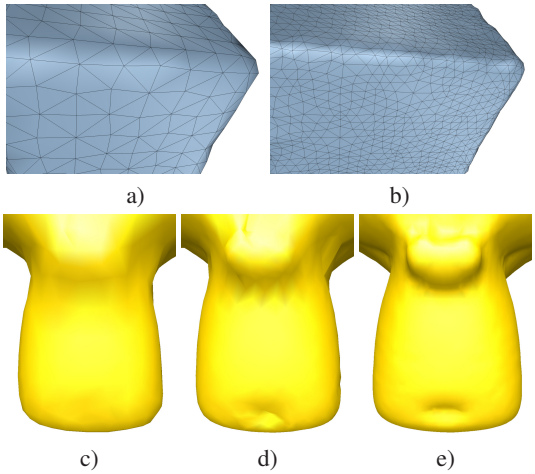


Figure 14: *Limitations of our method. a) Garbing mesh on the skeleton of a cube. The sharp edge is missed. b) If we subdivide the previous mesh one time and refine it, the sharp edge is almost recovered. c), d), and e) present garbing meshes for Monkey, Monkey* and Monkey**.*

5. Conclusion & Future Work

Medial surfaces are interesting structures for applications in computer graphics. Their properties make them good candidates to be efficient SRMs. One of their very basic tasks as SRMs, is to build an approximation of the object they describe: a task called in this paper *medial surface garbing*. This task is essential to visualize shapes only described by medial surfaces. However, medial surface garbing is far from

straightforward: we must take care of unwanted blending, deal with all kind of global topology and avoid bumpy patterns.

In this paper we have proposed a method to achieve the medial surface garbing on an unstructured medial surface $\mathcal{S}$. Our method runs as fast as an efficient meshing of the union of spheres associated to $\mathcal{S}$. Moreover the produced results are more satisfying than those obtained with previous methods, since bumpy patterns have disappeared. Finally the approximation error remains of the same order for the two methods.

The proposed method can be improved, especially on two directions we are already working on. First, the geometric data $\mathcal{P}$ of $\mathcal{S}$ could defined in a different way, to produce a smoother surface. To realize such a goal, we are notably exploring the use of the Dupin cyclides. Second, we should add a fifth criterion during the subdivision of $\mathcal{T}$, to discover small features of $\mathcal{O}$. This will solve the limitation we have when missing details in the garbing mesh. Currently, this criterion is not working in all cases, this is why we do not present it in this paper.

We have made a step forward for the effective use of medial surfaces as SRMs in computer graphics. We are now interested in exploiting those skeletons through this garbing algorithm, for shape creation and shape deformation. Another research field can be addressed to take advantage of this work: the skeletonization process itself. Indeed such a garbing mesh can allow to check if the medial surface remains acceptable, when an approximation or a simplification is sought.

References

[ABE09] ATTALI D., BOISSONNAT J.-D., EDELSBRUNNER H.: Stability and Computation of Medial Axes: a State-of-the-Art Report. In *Mathematical Foundations of Scientific Visualization, Computer Graphics, and Massive Data Exploration*, Mathematics and Visualization. Springer-Verlag, 2009, pp. 109–125. 2

[ACK01] AMENTA N., CHOI S., KOLLURI R. K.: The power crust. In *SMA '01: Proceedings of the sixth ACM symposium on Solid modeling and applications* (2001), pp. 249–266. 7

[AJWB03] AYLWARD S. R., JOMIER J., WEEKS S., BULLITT E.: Registration and analysis of vascular images. *International Journal of Computer Vision 55*, 2-3 (2003), 123–138. 2

[AK00] AMENTA N., KOLLURI R. K.: Accurate and efficient unions of balls. In *Proc. 16th Annu. ACM Sympos. Comput. Geom* (2000), pp. 119–128. 2

[ASCE02] ASPERT N., SANTA-CRUZ D., EBRAHIMI T.: Mesh: Measuring errors between surfaces using the hausdorff distance. In *Proceedings of the IEEE International Conference on Multimedia and Expo* (2002), vol. I, pp. 705 – 708. http://mesh.epfl.ch. 7

[ATC*08] AU O. K.-C., TAI C.-L., CHU H.-K., COHEN-OR D., LEE T.-Y.: Skeleton extraction by mesh contraction. *ACM Trans. Graph. 27* (2008), 44:1–44:10. 2

[BA84] BRADY M. J., ASADA H.: Smooth local symmetries and their implementations. *Int. Journal of Robotic Research* (1984). 2

[BHPF98] BOROUCHAKI H., HECHT F., PASCAL, FREY J.: Mesh gradation control. In *Int. J. Numer. Methods Eng* (1998), pp. 131–141. 5

[Bli82] BLINN J. F.: A generalization of algebraic surface drawing. *ACM Trans. Graph. 1*, 3 (July 1982), 235–256. 3

[Blu67] BLUM H.: A Transformation for Extracting New Descriptors of Shape. In *Models for the Perception of Speech and Visual Form*. MIT Press, 1967, pp. 362–380. 2

[cga] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org. 7

[CSM07] CORNEA N. D., SILVER D., MIN P.: Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics 13* (2007), 530–548. 2

[DLCB] DELAMÉ T., LÉON J. C., CANI M. P., BLANCH R.: Gesture-based design of 2d contours: an alternative to sketching? In *Proc. of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*, SBIM '11, pp. 63–70. 1, 3

[dOdF03] DE OLIVEIRA J. B. S., DE FIGUEIREDO L. H.: Robust approximation of offsets, bisectors, and medial axes of plane curves. *Reliable Computing* (2003), 161–175. 2

[ISS09] ITO Y., SHIH A. M., SONI B. K.: Octree-based reasonable-quality hexahedral mesh generation using a new set of refinement templates. *International Journal for Numerical Methods in Engineering 77*, 13 (2009), 1809–1833. 5

[KTZ95] KIMIA B. B., TANNENBAUM A. R., ZUCKER S. W.: Shapes, shocks, and deformations i: The components of two-dimensional shape and the reaction-diffusion space. *International Journal of Computer Vision 15* (1995), 189–224. 2

[KV] KRUITHOF N. G. H., VEGTER G.: Meshing skin surfaces with certified topology. In *Proceedings of the Ninth International Conference on Computer Aided Design and Computer Graphics*, CAD-CG '05, IEEE Computer Society, pp. 287–294. 3, 7

[Ley87] LEYTON M.: Symmetry–curvature duality. *CVGIP 38* (1987), 327–341. 2

[PABME11] PIZAINE G., ANGELINI E. D., BLOCH I., MAKRAM-EBEID S.: Vessel geometry modeling and segmentation using convolution surfaces and an implicit medial axis. In *ISBI* (2011), pp. 1421–1424. 3

[pov] Persistence of vision raytracer. [online] http://www.povray.org. 3

[SAAY] SAMOZINO M., ALEXA M., ALLIEZ P., YVINEC M.: Reconstruction with voronoi centered radial basis functions. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, SGP '06, Eurographics Association, pp. 51–60. 3

[SKS12] STOLPNER S., KRY P., SIDDIQI K.: Medial Spheres for Shape Approximation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 6 (2012), 1234–1240. 2

[SLSK07] SHARF A., LEWINER T., SHAMIR A., KOBBELT L.: On-the-fly curve-skeleton computation for 3d shapes. *Computer Graphics Forum 26*, 3 (2007), 323–328. 2

[TZCO09] TAGLIASACCHI A., ZHANG H., COHEN-OR D.: Curve skeleton extraction from incomplete point cloud. *ACM Trans. Graph. 28*, 3 (2009), 71:1–71:9. 2

[YFJ*03] YUSHKEVICH P., FLETCHER P. T., JOSHI S., THALL A., PIZER S. M.: Continuous medial representations for geometric object modeling. In *2D and 3D, Image and Vision Computing* (2003), pp. 17–27. 3

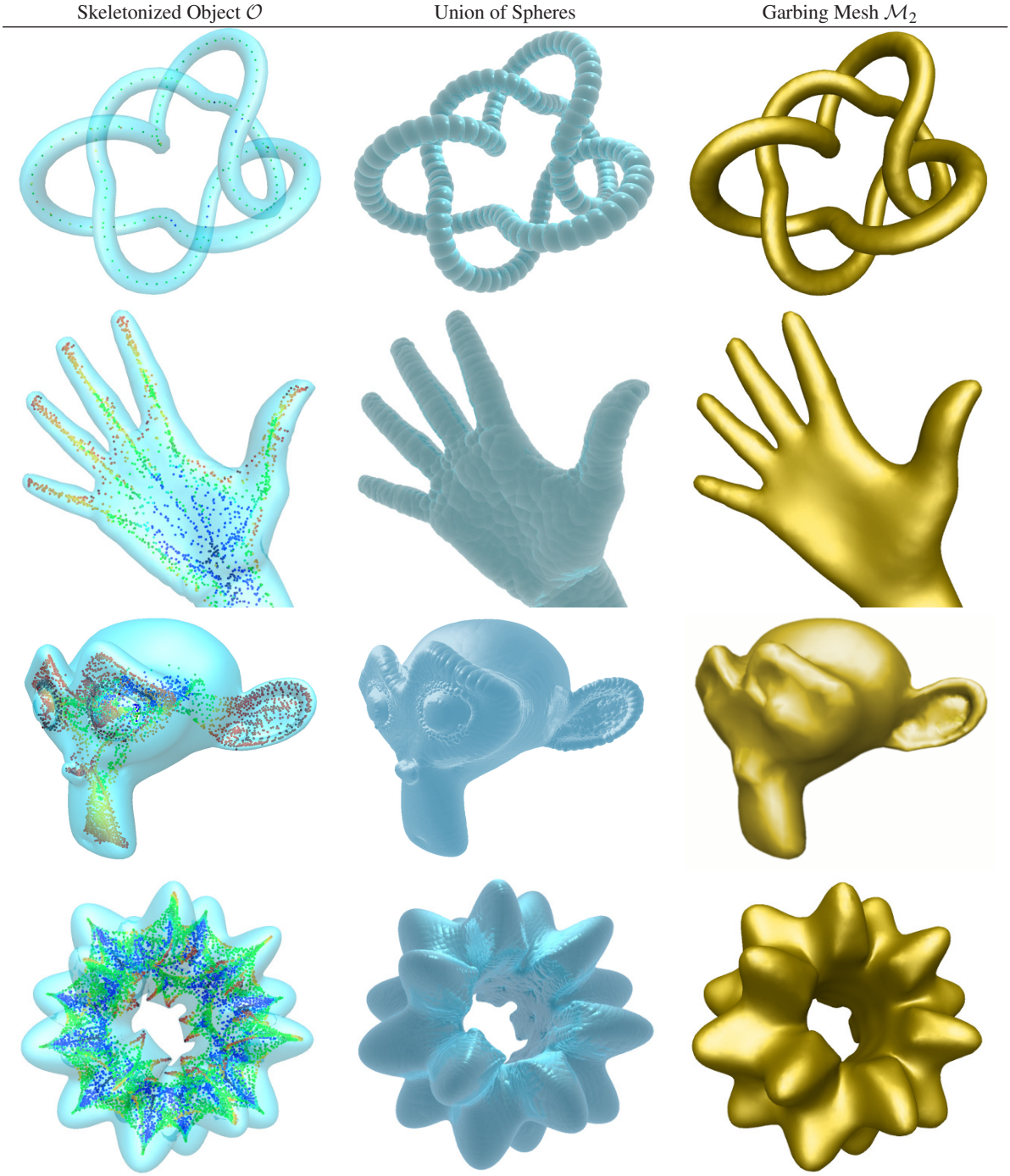


Figure 15: *Graphical Results. The first column displays the original object and its medial surface. The second column depicts the union of spheres contained in the medial surface. The third column shows our garbing mesh, which approximates the original object.*