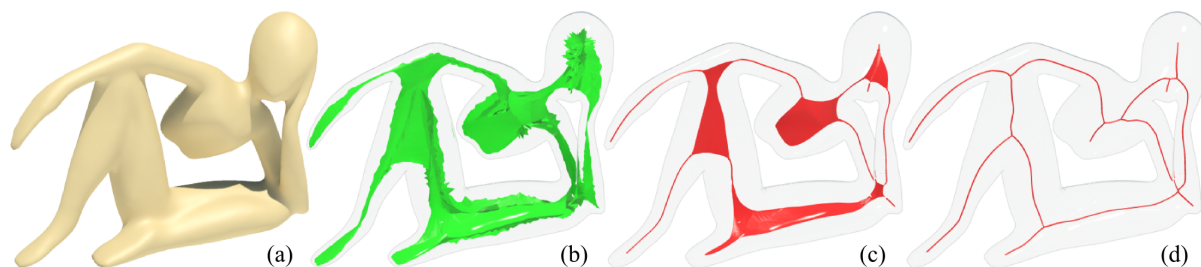# Mean Curvature Skeletons

Andrea Tagliasacchi[†], Ibraheem Alhashim, Matt Olson, and Hao Zhang

GrUVi Lab, School of Computing Science, Simon Fraser University, Canada

Figure 1: *Given a watertight surface (a), the well-known medial axis transform (b) often produces too complex of a structure to be of practical use. Our skeletonization algorithm can produce intermediate* meso-skeletons *(c), which contain medial sheets where needed and curves where appropriate, while converging to a* medially centered *curve skeleton output (d).*

Abstract

Inspired by recent developments in contraction-based curve skeleton extraction, we formulate the skeletonization problem via mean curvature flow (MCF). While the classical application of MCF is surface fairing, we take advantage of its area-minimizing characteristic to drive the curvature flow towards the extreme so as to collapse the input mesh geometry and obtain a skeletal structure. By analyzing the differential characteristics of the flow, we reveal that MCF locally increases shape anisotropy. This justifies the use of curvature motion for skeleton computation, and leads to the generation of what we call "mean curvature skeletons". To obtain a stable and efficient discretization, we regularize the surface mesh by performing local remeshing via edge splits and collapses. Simplifying mesh connectivity throughout the motion leads to more efficient computation and avoids numerical instability arising from degeneracies in the triangulation. In addition, the detection of collapsed geometry is facilitated by working with simplified mesh connectivity and monitoring potential non-manifold edge collapses. With topology simplified throughout the flow, minimal post-processing is required to convert the collapsed geometry to a curve. Formulating skeletonization via MCF allows us to incorporate external energy terms easily, resulting in a constrained flow. We define one such energy term using the Voronoi medial skeleton and obtain a medially centred curve skeleton. We call the intermediate results of our skeletonization motion meso-skeletons; *these consist of a mixture of curves and surface sheets as appropriate to the local 3D geometry they capture.*

1. Introduction

Skeletonization is a powerful and well-adopted means of shape abstraction. A variety of applications including shape segmentation, matching, retrieval, and reconstruction have utilized skeletal representations of 3D shapes. The best known skeletal structure is the Blum medial axis [Blu67] and its variants have played a central role in fields such as med-

ical shape analysis [SP09]. One characteristic of the medial axis is that it is formed at the singularities of the Euclidean distance transform. Correspondingly, one can obtain the medial axis of a 3D shape by an inward surface evolution which tracks the distance transform, resulting in a complex composition of surface sheets in the general case.

Due to the topological complexity and sensitivity to surface perturbations of the medial axes, however, a more popular choice for skeletal representations in computer graphics has been curve skeletons [HSKK01, DS06, ATC*08,

† andrea.tagliasacchi@gmail.com
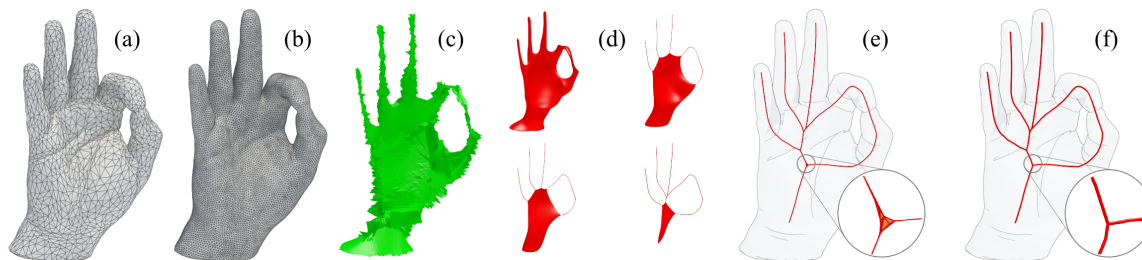
Figure 2: *An overview of our skeletonization algorithm. The input mesh (a) is first remeshed (b) to obtain better-quality Voronoi poles, from which a medial skeleton is computed (c) to constrain our construction of a medially centered skeleton. The algorithm performs iterative mesh contraction to generate a series of meso-skeletons in intermediate steps (d). Note that mesh connectivity is simplified throughout the contraction process. The last iteration produces a thin skeletal structure (e) which is finally converted into a curve skeleton (f) via edge collapse. The inserts in (e) and (f) show close-ups at the center of the palm.*

TZCO09, LGS12]. A natural and interesting question to ask then is whether a curve skeleton of a 3D shape can also be obtained through a surface evolution. A good candidate seems to be the mean curvature flow (MCF) [DMSB99], owing to its area minimizing characteristic [ATC*08, CK11].

In this paper, we closely examine MCF as a means for curve skeleton extraction from a 3D shape. Noting that area minimization alone does not explain the tendency of MCF to produce 1D structures, we analyze the differential geometry of the flow and show that MCF favors contractions of a surface along the direction of the largest curvature. In other words, MCF induces a surface motion that accentuates *local shape anisotropy*. This motivates the use of MCF for computing curve skeletons since a curve skeleton represents the anisotropy of a shape. One practical implication in the discrete setting is that MCF generates high aspect ratio triangles along the low principal curvature directions.

We formulate the skeletonization problem using MCF and call the resulting skeletal structures *mean curvature skeletons*. Given an input triangle mesh, we drive the flow towards the extreme so as to collapse the geometry and obtain a skeletal structure; see Fig. 2. Our algorithm is entirely mesh-based and we adopt the discrete form of MCF for triangle meshes [DMSB99, BKP*10]. To ensure faster convergence, we rely on an implicit scheme which allows large time steps. This scheme is realized by a constrained Laplacian smoothing process, but with two important modifications to achieve stable and efficient processing of the collapsed geometry.

1. First, since the tendency of MCF to create skinny triangles leads to numerical issues for the implicit solver, we perform local remeshing via edge splits and edge collapses concurrently with the geometry flow. This not only produces a more regular tessellation of the collapsed shape and eliminates an otherwise-required post-processing step, but also leads to more efficient computation, as mesh connectivity is simplified with the flow.
2. Second, we take advantage of the connectivity updates to control the discrete flow. The flow stops as soon as

further local contraction would lead to a non-manifold edge collapse, corresponding to a pinch in the surface.

Our work is not the first to connect MCF to curve skeleton extraction. The most representative method of this kind, which inspired our investigation, is the mesh contraction scheme of Au et al. [ATC*08]. Their work also relies on an implicit scheme to execute a constrained Laplacian smoothing. To prevent the contraction from shrinking the input mesh into a point, the contraction force is counterbalanced by attraction forces directed towards the vertices from previous time steps. A balance between the two forces is controlled heuristically by tuning several parameters. Throughout the contraction, both the topology of the shape and connectivity of the mesh are preserved. However, there is no control to center the thinned structure. Both curve skeleton extraction and centering are performed as post-processes.

Our analysis of the MCF reveals a more intricate connection to skeletonization. The insight gained leads to a better understanding of the tuned parameters, fewer of which are required, and they are associated with precise mathematical interpretations and therefore more predictable behaviour. Like that of Au et al. [ATC*08], our flow is also topology-preserving. However, the mesh connectivity is simplified throughout the flow, leading to simplified control of the contraction and skeleton formation. In contrast to Au et al. [ATC*08], we do not require boosting contraction forces at each iteration to combat stiffness in the Laplacian system. As well, we pin already-contracted vertices, to prevent the boosted contraction from pulling vertices past the point of full contraction, without requiring an heuristic to do so.

A key advantage of our simplified flow control is that we can easily incorporate new energy terms. We define such a term using the Voronoi medial skeleton and obtain a medially centered skeleton as a result of the geometry flow itself rather than a separate post-process [ATC*08]. Intermediate results of our skeletonization process are what we call *meso-skeletons*, which consist of a mixture of curves and surface sheets as appropriate to the local 3D geometry they capture; see Fig. 1. Indeed, insisting only on curve skeletons might
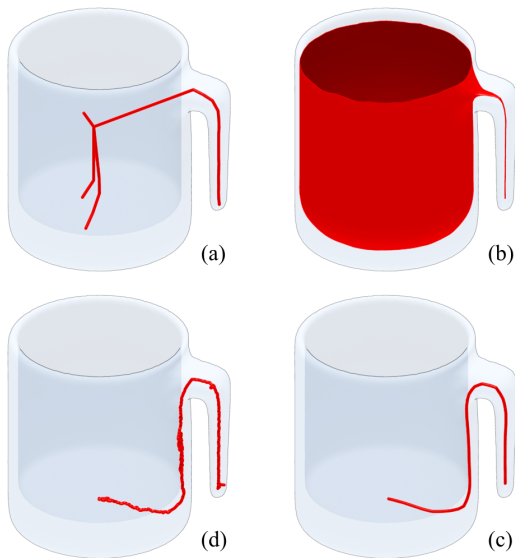
Figure 3: *Meso-skeletons produced by our algorithm on a cup model. The container part obviously does not admit a natural curve skeleton. We can produce a range of results from medial axis-dominant (b) to curve-dominant (c). Note that in all these results, the skeleton stays interior to the model. In contrast, the result from Au et al. [ATC*08] (a) is outside the shape and does not abstract the cup well. We obtain a result similar to Dey et al. [DS06] (d) , but with greater smoothness and reduced computational effort.*

not always provide us with the most appropriate shape abstraction. For example, a meso-skeleton is a more appropriate representation of a coffee mug; see Fig. 3.

The main contributions of our work are:

- A differential geometric analysis of the MCF to reveal its intricate connection to curve skeleton extraction, further motivating the use of MCF for the task and making the formulation cleaner and more mathematically justified.
- An efficient and robust 3D skeletonization algorithm that is topology-preserving and produces medially centered curve skeletons in a unified control framework.
- A simplified MCF flow control via local dynamic remeshing for efficiency and numerical stability.
- The ability to generate meso-skeletons to more faithfully abstract shapes of more general varieties.

The implementation of our skeletonization algorithm is already made available to researchers and practitioners. Fully documented code can be found at the following site: http://code.google.com/p/starlab-mcfskel.

2. Background and related works

Skeletons are effective shape abstractions which augment the information conveyed by traditional three dimensional

representations. They are commonly employed across a wide range of domains and in a number of applications including reconstruction [ABK98], segmentation [ATC*08], shape matching [HSKK01], and virtual navigation [WDK01]. It is important to note that for three dimensional shapes the term "skeleton" is ambiguous as it is used to refer to both medial axis skeletons [SP09], which are composed of 2D structures (surfaces), and curve skeletons [CMS07], which are composed of 1D structures (curves). While any object can be *exactly* represented by a medial axis skeleton, the same is not true for curve skeletons. Only a shape which has has a generalized cylindrical cross-section can be correctly *approximated*. The advantages of both approaches can be combined by considering what we call "meso-skeletons". Meso-skeletons use a compact curve skeleton in area of the shape which are approximately cylindrical, with the ability to represent local volume with medial axes where needed. An algorithm for the generation of meso-skeletons on digital volumetric objects was introduced in [LCLJ10]. The approximation quality provided by a skeletal representation is fundamental whenever they are used for studying the local volumetric structure of a shape like in segmentation [SSCO08] or statistical volumetric analysis [PFJ*03].

Medial axis skeletons. Medial axis skeletons, based on the *medial axis transform* or *MAT* [Blu67], provide a true dual shape representation. For a 3D solid, the medial axis skeleton is defined as the locus of the centres of all maximally-inscribed spheres, and is composed of two-dimensional components referred to as medial surfaces or sheets. The MAT is inherently sensitive to perturbations in the object's boundary (see Fig. 1 for an example) and additional local [DZ03] or global [MGP10] post-processing is needed to handle boundary noise. Medial surfaces, together with the medial radius function, provide a powerful volumetric representation of the shape. This volumetric interpretation is fundamental in tasks like shape statistical analysis [PFJ*03], the reconstruction of incomplete shapes [TOZ*11] and the definition of volumetric diffusion distances [BS12].

The key advantage of medial axis skeletons is that they are indeed *medial*: Their surfaces are found inside the object, in the middle of the shape. Other skeletal structures which aim to provide a similarly meaningful volumetric representation must satisfy this medial requirement. The computation of medial axis skeletons has received much attention in the literature and we refer the reader to [SP09] for an extensive coverage. When the input data is in the form of surfaces embedded in $\mathbb{R}^3$, a common approach for medial axis skeleton extraction lies in the computation of the Voronoi diagram of a sampling of the surface. Given a sufficiently good sampling, the Voronoi poles [ACK00] form a provably convergent sampling of the medial axis.

Curve skeletons. Curve skeletons provide an alternative, more compact representation of a shape, as they are formed by curves embedded in the 3D space. Having a smaller di-

mensionality than medial axis skeletons, the curve skeleton nevertheless captures the essential topology of the shape. The cost of this simpler representation is the loss of invertibility: Given only a curve skeleton and a radius function defined thereon, only an approximate shape approximation can be generated. Here again a meso-skeleton can provide necessary precision by including medial surfaces where the aspect ratio of the shape's cross-section is too large.

While a formal definition of curve skeletons has been proposed by Dey et al. [DS06], in practice, curve skeletons are better defined in a domain-specific fashion. Given a set of required properties like robustness, smoothness, and centeredness (see [CMS07] for a complete list) the chosen *skeletonization* algorithm is that which is most efficient while satisfying these requirements. Thus, because of its high computational cost and lack of smoothness, the algorithm of Dey et al. [DS06] is impractical in most applications.

It is possible to classify other skeletonization methods beyond a simple consideration of whether they work on a volumetric grid or on a discretized surface. Reeb-graph based solutions [BFS00] focus on capturing the shape topology. Methods based on the medial axis [LKC94, DS06] attempt to prune its surfaces down to a graph of curves. Other solutions attempt to generalize the concept of medialness to that of centeredness: These methods generate a potential field inside the shape's volume whose critical points can be traced to define a skeleton [HF08]. Other methods focus on more extrinsic properties like rotational symmetry [TZCO09] or shape convexity [LKA06]. There are also skeletonization algorithms which consider space-time data [SY07,ZST*10] or template fitting [BP07,LLW12].

However, from a geometric point of view, the most successful methods for building curve skeletons are based on *contracting* the shape to a degenerate surface (i.e. curves or points in space) [ATC*08, CTO*10, CK11] by applying a controlled mean curvature flow process. Our work is closely related to that of [ATC*08], in which the surface is contracted by a Laplacian force while collapsed features are anchored to avoid convergence to a degenerate solution. We extend the work of [ATC*08] by performing a thorough analysis of the contraction process in which we interpret it in the differential setting as a mean curvature motion. This justifies the use of this motion for accentuating shape anisotropy and producing a high-quality skeletonization.

Given this improved understanding, we are also able to understand the issues [ATC*08] has when attempting to generate a truly well-centred (or even medial) curve skeleton. By comparison, we provide a simpler control structure, which allows us to merge the contraction and local remeshing processes into a unique framework that produces higher quality skeletons. In addition, this simplified control system lets us augment the optimization with energies that attempt to perform medial embedding *directly* within the evolution and permits the generation of meso-skeletons as a by-product.

Mean curvature flow. Given a surface *S*, *mean curvature flow* or *MCF* is a motion that iteratively moves each surface point along its anti-normal with a speed proportional to the local average curvature:

$$\dot{S} = -H\mathbf{n} \quad H = (k1+k2)/2 \tag{1}$$

If we measure surface fairness in terms of its *membrane energy*, which measures the area of the surface *S*, by applying variational calculus we see that a fair surface is associated with a vanishing Laplacian [BKP*10]. Given that $H\mathbf{n} = \Delta p$, the flow described by Eq. (1) can be interpreted as a progressive fairing operation [DMSB99]. We draw a connection between this flow and the skeletonization problem by interpreting curve skeletons as zero-area, zero-volume degenerate manifolds with infinitesimal cross-section. As a solid object has non-zero area, any flow that performs skeletonization must progressively reduce surface area.

It is important to note that drastically different results can be obtained depending on the discretization of *S*. By discretizing the problem in a level-set framework [OF02], the flow could result in changes of shape topology. Instead, here we consider a version of curvature flow which does not change shape topology even if the surface becomes arbitrarily thin; rather, we detect and avoid potential topology changes as part of our algorithm.

3. Overview

The input to our skeletonization algorithm is a watertight manifold triangle mesh. Once the algorithm reaches full convergence, it produces a medially centered curve skeleton. After a preprocessing step, the algorithm iteratively invokes an implicit constrained Laplacian solver and optimizes the triangulation by local remeshing, until the volume of the shape vanishes. The algorithm can be halted before full convergence to produce a meso-skeleton, composed of curves in regions with circular cross-section and sheets elsewhere. Pseudocode is provided in Fig. 4.

Preprocessing. We use *QHull* to compute the voronoi diagram of the mesh vertices and process its output to compute medial poles. As discussed in Section 2, these poles are known to lie on the medial axis when the surface is sufficiently well-sampled. We resample the shape according to [BK04] to obtain a uniform sampling that produces higher quality Voronoi poles.

Iterative mesh contraction via MCF. As discussed in Sec. 2, the area-reducing properties of the MCF make it a good candidate for the core of a skeletonization algorithm, but several modifications are required to obtain a mean curvature motion that performs effective skeletonization. We justify the use of MCF in Sec. 4.1, describe the basis for our implicit solver in Sec. 4.2, and follow with details of the implementation in Sec. 4.5.

```
function COMPUTE_SKELETON(shape S_in)
    S ← Resample(S_in)
    vor ← Voronoi_poles(S)
    while has_volume(S') do
        update_laplacian_and_weights(S, vor)
        perform_mesh_contraction(S, vor)
        update_connectivity(S)
    end while
    collapse_shortest_edges(S)
    return S
end function
```

Figure 4: *Pseudocode for our algorithm.*

Updating Laplacian and weights. As the input surface is contracted towards the resulting meso-skeleton, the local curvature of the surface changes, and thus the Laplacian operator must be updated. Control of the velocity of the motion is obtained using a regularization scheme introduced in Sec. 4.2 and refined in Sec. 4.4 and Sec. 4.5.

Local dynamic remeshing. Since our constrained Laplacian solver increases local anisotropy by contracting edges aligned with directions of high curvature, it tends to produce triangles with high aspect ratio and hinders solver performance. In Sec. 4.3, we present a method which splits badly-shaped triangles and collapses short edges to mitigate this problem. Performing these steps at each iteration speeds up the solver and avoids a post-processing step.

Mean curvature skeleton formation. When the iteration is completed, we perform a simple edge collapse operation to reduce the degenerate manifold surface to a set of curves. We repeatedly collapse the shortest edge on a vertex that has two neighbouring faces. Edges in medial sheet portions of the meso-skeleton are not collapsed.

4. Mean curvature skeletons via MCF

We now present our algorithm for constructing mean curvature skeletons. In Section 4.1, we argue that MCF is particularly suitable for skeletonization, first by relating skeletonization to curve parameterization, then by studying the changes in local surface geometry throughout MCF motion. In Section 4.2, we discuss flow discretization and the practical implications of such a model. Section 4.4 shows how we control the discrete flow to obtain a curve skeleton as the steady state of the motion. In Section 4.5, we compute medially embedded skeletons via a slight variation of the MCF motion; we also discuss how the intermediate states of our evolution provide a suitable meso-skeletal structure.

Given a solid object $\mathcal{O}$ with manifold watertight boundary surface $S = \partial\mathcal{O}$, we consider its triangle mesh approximation $G = (V, E)$ where $V = [v_1; v_2; ...; v_n]$ are the vertices and E are the edges. There exist several possible discretizations of the Laplace-Beltrami operator [WMKG08]; Due to the positive-definiteness requirements imposed by

the Cholesky solver, we adopt the *cotangent* Laplacian discretization from [DMSB99] and we will refer to its matrix representation as L.

4.1. Mean curvature flow for skeletonization

As discussed in Sec. 2, the area-reducing properties of the MCF only make it a good *candidate* for skeletonization. We now provide further insight by interpreting skeletonization as a curve parametrization, then studying the anisotropic effects such a motion causes via a differential analysis.

Curve parameterization. Recall that by exploiting the fact that $H\mathbf{n} = \Delta p$, we can relate MCF to Tutte's embedding [Tut60]. In Tutte's parameterization, we can fix a set of vertex positions to a convex polygon $\{p_i = p_i^0\}_{i \in 1..n}$ and then look at the steady steady state of Eq. (1) $\Delta p = 0$. This observation allows us to interpret the problem of skeletonization as one of *curve parameterization*. A simple skeleton could be produced by marking two extremities on a shape and solving Tutte's embedding using the marked points as boundary constraints.

However, this curve parameterization is fundamentally different from computing the flow described by Eq. (1): In the former, we only need to compute the Laplacian matrix L *once* to obtain our final solution; in the latter, the mean curvature of the surface changes over time and the matrix must be updated. In addition, skeletonization algorithms are often expected to compute extremities rather than take them as input. This is problematic as Eq. (1), in the limit, will contract the surface to a point rather than a linear skeleton. We address these issues by first analyzing the differential properties of the curvature motion and by then constructing a suitable control scheme in Section 4.2.

Anisotropic flow. Consider a curve $\alpha(s)$ lying on the input surface $S^{t=0}$ and parametrized by arc length s. As mean curvature motion takes place, we obtain a family of curves $\alpha_t : I \to S_t$. We take $p = \alpha_t(0) \in S_0$ with $v = \frac{d}{ds}\alpha_0(s)$ as the unit tangent vector. Now consider the evolution of $v_t = \frac{\partial}{\partial t}\alpha_t(0)$ with respect to time t:

$$\begin{aligned}
\frac{\partial}{\partial t} v_t|_{t=0} &= \frac{\partial}{\partial t}\frac{\partial}{\partial s}\alpha_t(s)|_{(0,0)} = \frac{\partial}{\partial s}\frac{\partial}{\partial t}\alpha_t(s)|_{(0,0)} \\
&= -\frac{\partial}{\partial s}H(\alpha_0(s))n(\alpha_0(s))|_{s=0} \\
&= -\bar{D}_v(H(p)\vec{n}(p)) \\
&= -dH(v)_p n(p) - H(p)\bar{D}_v n_p
\end{aligned} \tag{2}$$

where $\bar{D}_v$ is the covariant derivative in $\mathbb{R}^3$. We can now use Eq. (2) to examine the change of a tangent vector with respect to t by evaluating $\langle \dot{v}, v \rangle$. As v is a tangent vector, the first component of the last line of Eq. (2) vanishes:

$$\langle \dot{v}, v \rangle = -H(p)\langle \bar{D}_v n_p, v \rangle = -H(p)\Pi_p(v) \tag{3}$$

where $\Pi_p(v)$ is the second fundamental form.

Equation (3) helps us understand the anisotropic distortion caused by Eq. (1) around a point $p \in S$. If we consider an infinitesimal geodesic disk around the point p, this disk will be distorted in the direction of v according to the orientation of v with respect to the principal curvature directions. Consequently Eq. (3) demonstrates the utility of MCF as a tool for skeletonization, as it favours a contraction of the surface in the direction of largest curvature. That is, it creates a motion which accentuates local shape anisotropy.

4.2. Discrete mean curvature flow

A simple discretization of the MCF in Eq. (1) can be obtained by noting that $H\mathbf{n} = \Delta p$ and by applying a forward Euler integration: $v_i^{t+1} = v_i^t + h\Delta v_i^t$. Explicit discretizations of this type are typically used for surface smoothing where the limitations on small time-steps imposed by its elliptic structure are not a problem [DMSB99]. However, in skeletonization, we are not interested in a fine discretization of the the flow movement. We prefer implicit schemes that allow us to discretize Eq. (1) with much larger time steps.

We obtain an implicit scheme for MCF by using the discretized Laplacian operator L. Directly solving the Laplacian equation $LV = 0$ is problematic, as L is singular and the system admits a trivial solution $V = 0$. To address this problem, we solve the regularized version:

$$\begin{bmatrix} L \\ W_H \end{bmatrix} V^{t+1} = \begin{bmatrix} 0 \\ W_H V^t \end{bmatrix} \qquad (4)$$

Where by W_H is a diagonal matrix such that $W_{H,i} = w_H$. The solution of this linear system minimizes the following quadratic energy:

$$E = \left\| LV^{t+1} \right\|^2 + w_H^2 \sum_i \left\| v_i^{t+1} - v_i \right\|^2 \qquad (5)$$

Examining the right hand side of Eq. (4) shows that this regularizer works to bound the *velocity* of vertices undergoing curvature motion. Slowing down the motion is essential, not only from a numerical point of view, but also from a geometrical one. As the surface evolves, the local curvature of the surface changes, and thus the Laplacian operator must be updated to construct an approximation of Eq. (1). This is demonstrated in Fig. 5.

4.3. Local dynamic remeshing

Due to the anisotropic behaviour discussed in Section 2, particular care must be taken when solving an optimization problem like Eq. (5). For example, consider an highly anisotropic shape, where where $k_1 >> k_2 = 0$ (e.g. an infinitely extended cylinder of small radius, where $k_1 = \text{const}$). The anisotropy boosting effects of the mean curvature flow described by Eq. (3) will cause edges aligned with the principal curvature direction $\vec{k}_1$ to vanish rapidly, while edges

aligned with $\vec{k}_2$ retain their length. This results in the formation of very high aspect ratio triangles which hinder the performance of the finite element solver by stiffening the system. In [ATC*08,CK11] this problem was overcome by employing a multi-grid solver. In this paper we perform a simple local remeshing at each iteration which *collapses* edges deemed too short, provided that the collapse retains the manifoldness of the shape, and *splits* badly shaped triangles. In particular we split an edge e whenever one of its incident triangles has an angle greater than $\theta_0 = 110°$ at the opposite vertex. We heuristically position the new vertex by perpendicularly projecting the third vertex of the split face onto e.

Since the mean curvature flow reduces edge length in proportion to the edge's curvature, this process allows us to reduce the vertex count of the skeleton, which is already significantly smaller than the mesh, as the iteration progresses rather than maintain a dense skeleton which is later simplified in a post-processing step. In addition, it permits us to *maintain* a correspondence between points on the skeleton and associated vertices on the mesh, rather than recomputing such a correspondence as a post-processing step. This facilitates the use of the skeleton for mesh-editing operations.
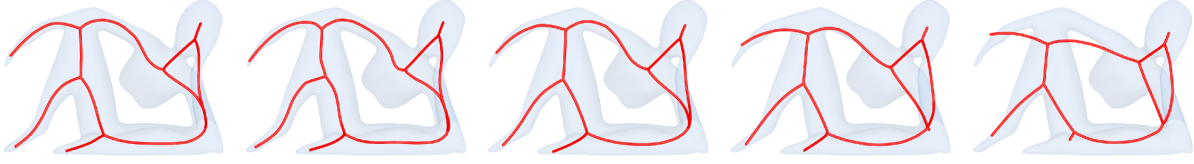
4.4. Discrete skeletonization flow

As we maintain the manifold properties of the surface during evolution, it is important to note that solving Eq. (5) as presented would result in a straightening of already collapsed branches of the shape, pulling them away from the medial axis. Furthermore, the tips of skeletal branches exhibit a very large curvature and consequently they would be very strongly contracted back into the shape. We would like to stop the movement of a point on a branch of the skeleton as soon as the creation of such a branch is detected. As we identify a branch as an infinitesimal cross-section manifold, the problem reduces to measuring whether the local shape cross-section is below a certain threshold (i.e. close to zero).

In the continuous setting, determining whether a surface has a locally vanishing cross-section can be achieved by looking at an infinitesimal geodesic disk centred at a point p. If the local surface has disk topology then the local geometry is non-degenerate. Our dynamic topology regularization allows us to adapt this process directly to the discrete setting. We can test for degeneracy by monitoring the Euler characteristic of an infinitesimal (w.r.t. the edge collapse threshold) neighborhood of a vertex. Whenever this neighborhood does not exhibit disk topology we set the corresponding diagonal element in W_H to a sufficiently large number, effectively fixing its position in space throughout evolution.

4.5. Medial skeletonization flow

One of the fundamental flaws of geometric contraction for skeletonization [ATC*08] is that it does not guarantees centeredness of the skeleton. To compute a well-centered curve

Figure 5: *The effect of changing the ratio W_L/W_H and its effect on the embedding. We apply our method, setting $W_H = 20$ and $W_L = \{1.5, 3.0, 6.0, 12.0, 24.0\}$. This results in the skeletonization flow converging in (resp.) $\{26, 12, 6, 3, 2\}$ iterations. Note that while increasing W_L results in faster convergence, it has a severe effect on the quality of the embedding.*

skeleton (i.e. one that lies on the medial axis) we modify the energy of Eq. (5) as follows:

$$E = E_{smooth} + E_{velocity} + E_{medial}$$

where the individual energies are define as:

$$E_{smooth} = \left\| W_L L V^{t+1} \right\|^2$$

$$E_{velocity} = \sum_i w_H \left\| \left(v_i^{t+1} - v_i \right) \right\|^2 \qquad (6)$$

$$E_{medial} = \sum_i w_M \left\| \left(v_i^{t+1} - \mu(v_i) \right) \right\|^2$$

The energy term E_{medial} pulls the evolving surface toward the medial axis. This is done by defining $\mu(v_i)$ as a map which corresponds a vertex v_i to a chosen medial axis Voronoi pole. At $t = 0$ we simply correspond each surface vertex to its associated medial pole. As the motion progresses we apply a simple update: whenever an edge is collapsed, we retain the Voronoi pole which is closest to the resulting vertex.

Note that when the motion associated with Eq. (6) has converged, our skeleton lies along the medial axis in an approximating (as opposed to interpolating) sense. The energy $E_{velocity}^{t=\infty}$ will have vanished, while E_{smooth} and E_{medial} will balance each other, keeping the skeletal curves smooth yet medially embedded. This variational interpretation of medially embedded skeleton produces noticeably smoother skeletal curves than those typical of other medial axis-based algorithms. At the same time, it provides robustness against possible under-sampling of the medial axis.

Performing a motion like the one associated with Eq. (6) is not only interesting because of its steady state. It is clear from the non-converged iterations in Fig. 2 that an informative decomposition of the shape is obtained before the skeleton converges to a curve. This flow is able to generate a *meso-skeleton*, composed of curves in approximatively cylindrical regions but retaining a surface representation in regions associated with strong reflectional symmetry.

5. Results and discussions

In this section, we show mean curvature skeletons generated by our algorithm and compare them to curve skeletons ob-

tained from two state-of-the-art approaches: the mesh-based contraction scheme of Au et al. [ATC*08] and the volumetric, medial geodesic based approach of Dey et al. [DS06]. In Figure 6, we show the performance of our skeletonization algorithm on a set of heterogeneous shapes, and we showcase the novel meso-skeletons in Figure 7.

Input, preprocessing, solvers. We applied our algorithm to a range of meshes of varying resolution. In our examples the mesh sized ranged from $50k$ to $200k$ faces. As our algorithm is based on a Voronoi approximation of the medial axis, it was important to re-mesh the surfaces in such a way that the sampling was adequate to resolve the local feature size. Optimally, an ε-sampling would have been preferable, but with no code publicly available we opted for a simple uniform sampling which, if dense enough, provides poles which approximate the medial axis just as well. To achieve such a sampling we employed a tangential remeshing based on Laplacian smoothing and local connectivity changes [BK04]. To solve the large sparse linear systems with a structure similar to Eq. (4), we use a direct solver based on Cholesky factorization (LL^t). It is noted in [BKP*10] that such a method not only outperforms other types of solvers in general, but also that the observed runtime of factorization grows linearly with the size of the mesh.

Parameters. Our algorithm requires a total of four parameters $\{w_L, w_H, w_M, \varepsilon\}$; of these, however, *only three* require tuning. First, the w_L, w_H, and w_M parameters have a partition of unity property over multiplication; thus, only two of them need to be set. Also, ε controls the resolution of our skeleton, as throughout the process we collapse any edge $\|e\| < \varepsilon$ provided that such a collapse is manifold.

The triplet of parameters $\{w_L, w_H, w_M\}$ offers a clean mathematical interpretation. The ratio w_L/w_H controls the velocity of the movement and the approximation quality with which we resolve the motion as illustrated in Figure 5. The ratio w_L/w_M controls the smoothness of the medial approximation, and w_M controls the degree to which the surface is attracted to the medial axis. In our experience, this parameter is closely influenced by the number of features, particularly *medial* features, to be retained in our skeleton. An example is provided in Figure 6, where varied w_M obtained different scales of features for the *Elephant* model.

Mesh		Timings			Distances	
Name	Size	[DS06]	[ATC*08]	This technique	[ATC*08]	This technique
Cuboid	37k	10' + 0.009"	16.61" + 1'77.71"	1.57" + 8.05" + 0.032"	0.02820	0.00449
Dragon	72k	30' + 0.013"	31.83" + 5'01.93"	2.54" + 12.97" + 0.048"	0.01589	0.00769
Fertility	36k	9' + 0.007"	13.77" + 1'11.96"	1.30" + 4.24" + 0.054"	0.01202	0.00447
Hand	35k	11' + 0.008"	09.61" + 1'15.85"	1.44" + 2.78" + 0.152"	0.01027	0.00476
SGP	60k	25' + 0.011"	11.13" + 3'28.89"	2.19" + 5.24" + 0.049"	0.03067	0.00381

Table 1: *Quantitative comparison of our method to the state of the art for the models in Fig. 8. In columns 3–5 we show recorded execution times; we split execution times as [DS06]: ($t_{geodesic} + t_{erosion}$); [ATC*08]: ($t_{contraction} + t_{surgery}$); for our technique we report ($t_{voronoi} + t_{contraction} + t_{surgery}$). In columns 6–7 we measure average asymmetric Euclidean distances to the skeleton produced by [DS06], normalized by the length of the bounding box diagonal. To compute this distance, the skeletons were re-sampled such that no edge is longer than 0.1% of the bounding box diagonal.*
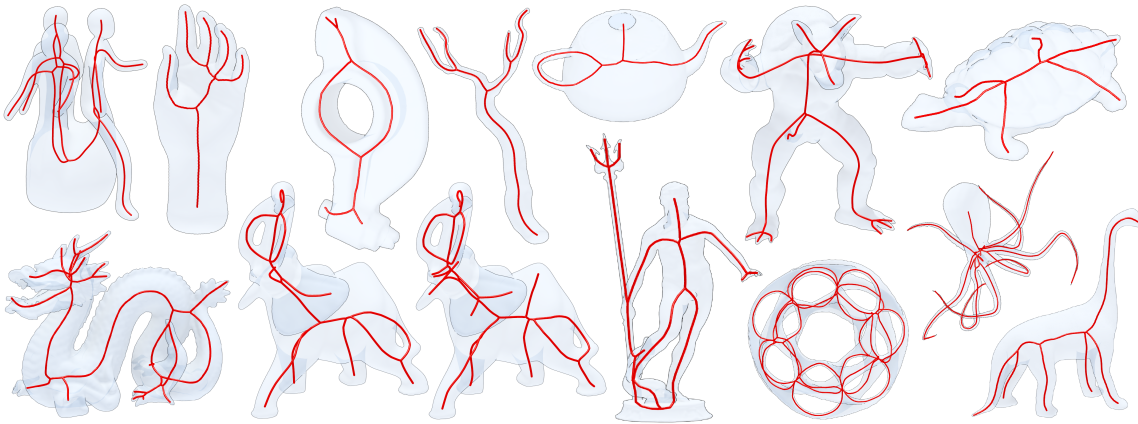

Figure 6: *A gallery of models that have been skeletonized by our algorithm. On each model the algorithm took less than a minute to produce a skeleton, which was produced with the parameters $\{w_L = 1, w_H = 20, w_M = 40, \varepsilon = .002 * bbox.diag()\}$, with the exception of the elephant model, where we illustrate how lowering the parameter $w_M = 20$ results in a coarser skeleton.*

Comparison to Au et al. Figure 8 shows that the necessity for post-processing and re-centering in [ATC*08] generally results in a severe under-sampling of the skeleton. Re-centering exploits correspondences between nodes on the skeleton and the surface and re-locates a node to the centre of mass of its corresponding surface points. This re-centering is often unable to position the curves on the medial axis (see the distance measures reported in Table 1) or even inside the shape (see the last row of Figure 8). Furthermore, in order to make this re-centering effective, a large set of surface points must be taken in order to form a uniform radial distribution; this causes the coarseness in the resulting structure. In addition, the topology surgery algorithm required by [ATC*08] is based on quadric edge collapses, and is far more complex than the shortest edge collapse in our solution. This is notable in light of the lengthy running times we recorded for the surgery step, illustrated in Table 1. As the method of [ATC*08] emphasizes efficiency over preci-

sion, it doubles the contraction weights at every iteration $W_L^{t+1} = 2W_L^t$. Another important difference lies in the fact that our contraction is highly tied to mean curvature flow, while in [ATC*08] the use of the ratio A_i^t/A_i^0 to control attraction constraints eliminates this connection. Taken as a whole, these improvements allow our algorithm to produce better results with more intuitive parametric control.

Comparison to Dey et al. The algorithm of [DS06] relies on the computation of geodesic distances in between every pair of points, consequently requiring $O(n^2 \log n)$ computational time, resulting in long running times even for models of moderate complexity as shown in Table 1. While we are not aware of an equivalent result for three dimensions, it is known that curvature flow in 2D is associated with a constant decrease in area: $\dot{A} = -2\pi$ [Gra87]. We believe a similar argument applies to 3D curvature flow, which implies that the complexity of our algorithm is a constant factor of the volume of the model, times the complexity of a single iteration
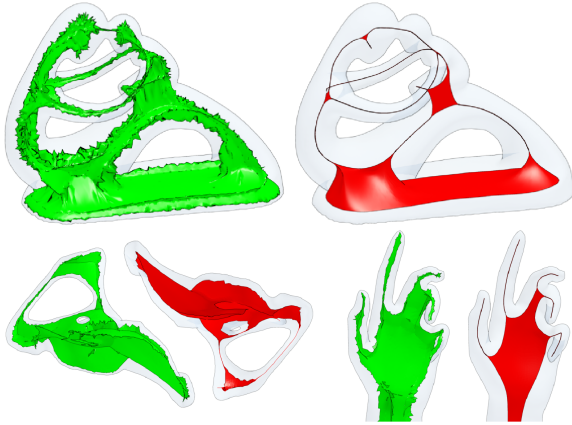
Figure 7: *Not all shapes are well abstracted by curves. Meso-skeletons can characterize regions with circular cross-sections as well as those exhibiting local reflectional symmetry. Note how the meso-skeletons produced by our algorithm (red) are not hindered by noisy medial axes (green).*

of the solver – which, for the solver we employed, is linear. A minor disadvantage is that, by computing a skeleton as a strict subset of the medial axis, the method of [DS06] produces branches that are quite noisy (see for example Figure 6). However, one *significant* advantage of [DS06] is the ability to change the *resolution* of the skeleton quickly once the initial geodesic computation is complete; in comparison, the presented method would need to construct a new skeleton for each tuple of input parameters.

6. Conclusion and future work

We present a mesh-based contraction algorithm for curve skeleton extraction. Our work was inspired by the mesh contraction scheme of Au et al. [ATC*08] but offers several marked improvements, both in its formulation as well as the results it achieves. Qualitatively speaking, our curve skeleton results are more similar to those obtained by the algorithm of Dey et al. [DS06]. In retrospect, this latter algorithm also takes a flow approach: it involves two distance transforms, one on the input 3D shape and one on the medial sheets.

While [DS06] was the first to define curve skeletons formally, we believe that mean curvature flow is also an excellent candidate for an alternate definition. In this paper, we employed Voronoi poles to guide the surface toward the medial axis. Nevertheless, as the MAT offers a direct interpretation as the first order discontinuities of the Euclidean distance field, we believe that it is possible to provide a formal definition of the medially-centred curve skeletons of Sec. 4.5 in the continuous setting.

In future work, we plan to investigate mean curvature meso-skeletons further and produce a flow that generates them as the steady state of a properly constructed differential equation. Performance-wise, we find highly detailed
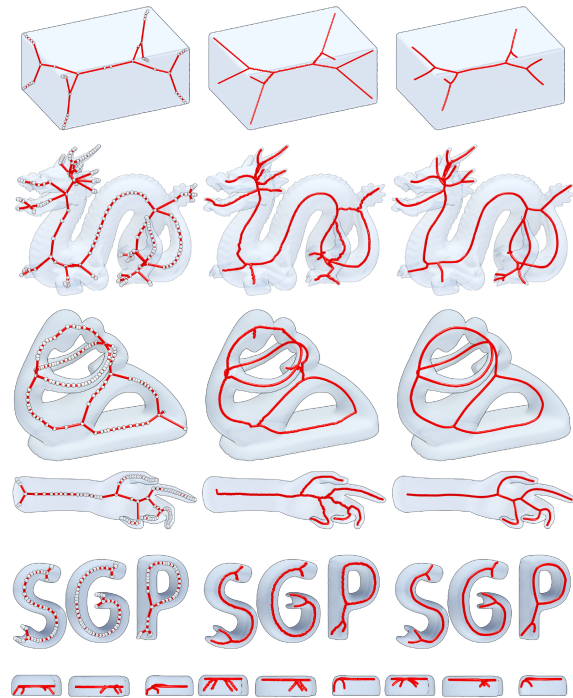
Figure 8: *Curve skeletons computed from mesh contraction [ATC*08] (left column), medial geodesics [DS06] (middle), and our method (right). The last two rows show the side and top views of an "SGP" model. The cross-sectional profile of each letter along the curve skeleton is not circular, but a concave shape resembling the letter 'U'. Note how the skeletons generated from [ATC*08] are not only far from being medial, but also lie exterior to the shape (seen as being centred in the top view of the SGP in contrast to the other two methods). Note also the sparsity and coarseness of the generated skeletal vertices, and that medial geodesic skeletons are often noisy. Our method produces smoother, interior skeletons that lie close to the medial axes.*

surfaces to cause issues for the evolution as the resulting Voronoi pole structure is dense and noisy. Simply removing medial poles with small spoke angles does not work well, as internal energies are known to be stiffer than external ones, resulting in an unnatural motion. We would like to find a solution to this problem within the presented framework. Finally, we would also like to consider integrating our technique into the highly optimized curvature flow solver recently presented by Chuang and Kazhdan [CK11].

References

[ABK98] AMENTA N., BERN M., KAMVYSSELIS M.: A new Voronoi-based surface reconstruction algorithm. *ACM Transactions on Graphics (Proc. of SIGGRAPH)* (1998), 415–421. 3

[ACK00] AMENTA N., CHOI S., KOLLURI R. K.: The power crust, unions of balls, and the medial axis transform. *Computational Geometry: Theory and Applications 19* (2000), 127–153. 3

[ATC*08] AU O. K.-C., TAI C.-L., CHU H.-K., COHEN-OR D., LEE T.-Y.: Skeleton extraction by mesh contraction. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 27*, 3 (2008), 44(1–10). 1, 2, 3, 4, 6, 7, 8, 9

[BFS00] BIASOTTI S., FALCIDIENO B., SPAGNUOLO M.: Extended Reeb graphs for surface understanding and description. In *Discrete Geometry for Computer Imagery* (2000), Springer, pp. 185–197. 4

[BK04] BOTSCH M., KOBBELT L.: A remeshing approach to multiresolution modeling. In *Computer Graphics Forum (Proc. ACM/EG Symposium on Geometry Processing)* (2004), ACM, pp. 185–192. 4, 7

[BKP*10] BOTSCH M., KOBBELT L., PAULY M., ALLIEZ P., LÉVY B.: *Polygon Mesh Processing*. Ak Peters Series. A K Peters, 2010. 2, 4, 7

[Blu67] BLUM H.: *A transformation for extracting new descriptors of shape*. Models for the perception of speech and visual form. MIT Press, 1967. 1, 3

[BP07] BARAN I., POPOVIĆ J.: Automatic rigging and animation of 3D characters. *ACM Transactions on Graphics (Proc. of SIGGRAPH) 26*, 3 (2007), 72(1–8). 4

[BS12] BERGER M., SILVA C.: Medial kernels. *to appear in Computer Graphics Forum (Proceedings of Eurographics) 31*, 2 (2012). 3

[CK11] CHUANG M., KAZHDAN M.: Fast mean-curvature flow via finite-elements tracking. In *Computer Graphics Forum* (2011). 2, 4, 6, 9

[CMS07] CORNEA N. D., MIN P., SILVER D.: Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics 13*, 3 (2007), 530–548. 3, 4

[CTO*10] CAO J., TAGLIASACCHI A., OLSON M., ZHANG H., SU Z.: Point cloud skeletons via Laplacian based contraction. In *Proc. of the IEEE Conference on Shape Modeling and Applications* (2010), pp. 187–197. 4

[DMSB99] DESBRUN M., MEYER M., SCHRÖDER P., BARR A.: Implicit fairing of irregular meshes using diffusion and curvature flow. In *ACM Transactions on Graphics (Proc. of SIGGRAPH)* (1999), pp. 317–324. 2, 4, 5, 6

[DS06] DEY T., SUN J.: Defining and computing curve-skeletons with medial geodesic function. *Computer Graphics Forum (Proc. ACM/EG Symposium on Geometry Processing)* (2006), 143–152. 1, 3, 4, 7, 8, 9

[DZ03] DEY T. K., ZHAO W.: Approximating the Medial Axis from the Voronoi Diagram with a Convergence Guarantee. *Algorithmica 38*, 1 (2003), 179–200. 3

[Gra87] GRAYSON M.: The heat equation shrinks embedded plane curves to round points. *J. Diff. Geom 26*, 2 (1987), 285–314. 8

[HF08] HASSOUNA M., FARAG A.: Variational curve skeletons using gradient vector flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2008), 2257–2274. 4

[HSKK01] HILAGA M., SHINAGAWA Y., KOHMURA T., KUNII T. L.: Topology matching for fully automatic similarity estimation of 3D shapes. *ACM Transactions on Graphics (Proc. of SIGGRAPH)* (2001), 203–212. 1, 3

[LCLJ10] LIU L., CHAMBERS E., LETSCHER D., JU T.: A simple and robust thinning algorithm on cell complexes. *Computer Graphics Forum 29*, 7 (2010), 2253–2260. 3

[LGS12] LIVESU M., GUGGERI F., SCATENI R.: Reconstructing the curve-skeletons of 3d shapes using the visual hull. *IEEE Transactions on Visualization and Computer Graphics* (2012). 1

[LKA06] LIEN J., KEYSER J., AMATO N.: Simultaneous shape decomposition and skeletonization. In *Proc. of the ACM Symposium on Solid and Physical Modeling* (2006), pp. 219–228. 4

[LKC94] LEE T., KASHYAP R., CHU C.: Building skeleton models via 3D medial surface/axis thinning algorithms. *CVGIP: Graphical Model and Image Processing 56*, 6 (1994), 462–478. 4

[LLW12] LU L., LÉVY B., WANG W.: Centroidal voronoi tesselation of line segments and graphs. *to appear in Computer Graphics Forum (Proceedings of Eurographics) 31*, 2 (2012). 4

[MGP10] MIKLOS B., GIESEN J., PAULY M.: Discrete scale axis representations for 3d geometry. *ACM Transactions on Graphics (TOG) 29*, 4 (2010), 101. 3

[OF02] OSHER S., FEDKIW R.: *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2002. 4

[PFJ*03] PIZER S., FLETCHER P., JOSHI S., THALL A., CHEN J., FRIDMAN Y., FRITSCH D., GASH A., GLOTZER J., JIROUTEK M., ET AL.: Deformable m-reps for 3D medical image segmentation. *International Journal of Computer Vision 55*, 2 (2003), 85–106. 3

[SP09] SIDDIQI K., PIZER S.: *Medial Representations: Mathematics, Algorithms and Applications*. Springer, 2009. 1, 3

[SSCO08] SHAPIRA L., SHAMIR A., COHEN-OR D.: Consistent mesh partitioning and skeletonization using the shape diameter function. *The Visual Computer 24*, 4 (2008), 249–259. 3

[SY07] SCHAEFER S., YUKSEL C.: Example-based skeleton extraction. *Computer Graphics Forum (Proc. ACM/EG Symposium on Geometry Processing)* (2007), 153–162. 4

[TOZ*11] TAGLIASACCHI A., OLSON M., ZHANG H., HAMARNEH G., COHEN-OR D.: VASE: Volume-Aware Surface Evolution for surface reconstruction from incomplete point clouds. *Computer Graphics Forum (Proc. ACM/EG Symposium on Geometry Processing) 30*, 5 (2011), 1563–1571. 3

[Tut60] TUTTE W.: Convex representations of graphs. *Proceedings London Mathematical Society 10*, 38 (1960), 304–320. 5

[TZCO09] TAGLIASACCHI A., ZHANG H., COHEN-OR D.: Curve skeleton extraction from incomplete point cloud. *ACM Transactions on Graphics, (Proceedings SIGGRAPH 2009) 28*, 3 (2009), 71:1–71:9. 1, 4

[WDK01] WAN M., DACHILLE F., KAUFMAN A.: Distance-field based skeletons for virtual navigation. In *Proc. of the IEEE Conference on Visualization* (2001), p. 246. 3

[WMKG08] WARDETZKY M., MATHUR S., KÄLBERER F., GRINSPUN E.: Discrete laplace operators: no free lunch. In *ACM SIGGRAPH ASIA 2008 courses* (2008), ACM, p. 19. 5

[ZST*10] ZHENG Q., SHARF A., TAGLIASACCHI A., CHEN B., ZHANG H., SHEFFER A., COHEN-OR D.: Consensus skeleton for non-rigid space-time registration. *Computer Graphics Forum (Proc. of Eurographics) 29*, 2 (2010), 635–644. 4