EUROGRAPHICS 2012 / P. Cignoni, T. Ertl
(Guest Editors)

Volume 31 (2012), Number 2

Centroidal Voronoi Tessellation of Line Segments and Graphs

Lin Lu[†], Bruno Lévy[‡] and Wenping Wang[§]

Abstract

Centroidal Voronoi Tessellation (CVT) of points has many applications in geometry processing, including remeshing and segmentation, to name but a few. In this paper, we generalize the CVT concept to graphs via a variational characterization. Given a graph and a 3D polygonal surface, our method optimizes the placement of the vertices of the graph in such a way that the graph segments best approximate the shape of the surface. We formulate the computation of CVT for graphs as a continuous variational problem, and present a simple, approximate method for solving this problem. Our method is robust in the sense that it is independent of degeneracies in the input mesh, such as skinny triangles, T-junctions, small gaps or multiple connected components. We present some applications, to skeleton fitting and to shape segmentation.

Categories and Subject Descriptors (according to ACM CCS): Computer Graphics [I.3.5]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems; Algorithms

1. Introduction

In this paper, we propose a generalization of Centroidal Voronoi Tessellation (CVT). CVT is a fundamental notion that has a wide spectrum of applications in computational science and engineering, including geometry processing. Intuitively, CVT optimizes the placement of points in a domain (for instance the interior of a 3D surface), in such a way that the set of points evenly samples the domain. In this paper, our goal is to show that the main idea of CVT can be generalized to more complicated settings. Specifically, we show that a given set of possibly interconnected segments can be optimized to obtain the best fitting to the interior of a surface. Our generalization of CVT is obtained by using the variational characterization (minimizer of Lloyd's energy), adapted to line segments. To optimize the objective function, we propose a new algorithm, based on an approximation of Voronoi diagrams for line segments and an efficient yet accurate clipping algorithm.

As an application of the method, we show how a template skeleton can be fitted to a mesh model. Our method is simple, automatic, and requires only one parameter (regularization weight). We also show examples of mesh segmentations computed by our method.

This paper proposes the following contributions :

- A generalization of Centroidal Voronoi Tessellations for line segments, based on a variational characterization. Our formalization can take structural constraints into account, such as a graph of interconnected segments that share vertices;
- a method for computing the CVT of line segments and graphs in 3D, that is both simple (see algorithm outline page 4) and robust to most degeneracies encountered in 3D meshes (cracks, T-junctions, degenerate triangles . . .);
- a segmentation method that does not depend on the initial discretization. Part boundaries can pass through triangles, and parts may group several connected components (e.g. trousers with the legs, hair with the head . . .);
- *Limitations:* Our skeleton fitting should be considered as a demonstration of our technical contribution rather than our major contribution. We aknowledge that some state-of-the-art skeleton detection methods may give better results.

2. Previous Work

Centroidal Voronoi Tessellation

CVT for points A complete survey about CVT is beyond the scope of this paper. The reader is referred to

[†] Hong-Kong University and Shandong University
[‡] INRIA Nancy Grand-Est and LORIA
[§] Hong-Kong University

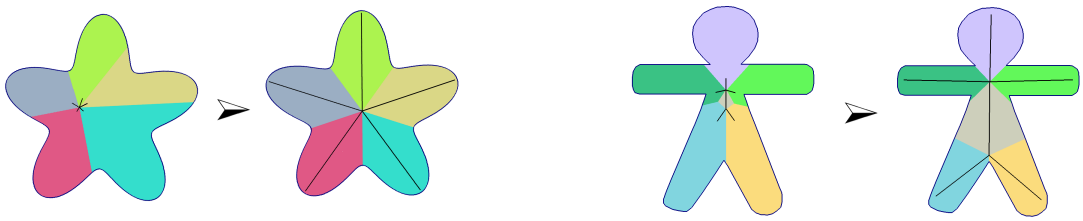


Figure 1: *Principle of our method: two examples of Centroidal Voronoi Tessellations for connected line segments in 2D.*

the survey in [DFG99]. We will consider here the context of Geometry Processing, where CVT was successfully applied to various problems. Alliez *et al.* developed methods for surface remeshing [AMD02], surface approximation [CSAD04] and volumetric meshing [ACSYD05]. Valette *et al.* [VC04, VCP08] developed discrete approximations of CVT on mesh surfaces and applications to surface remeshing. We note also some recent applications of CVT to generate scale-like patterns on objects [LS10] .

In all these works, the mathematical formulation of CVT resulted in elegant algorithms that are both simple and efficient. However, they use some approximations, that make them dependent on the quality of the initial mesh. For instance, applications of these methods to mesh segmentation are constrained to follow the initial edges, and applications to 3D meshing need to approximate the clipped Voronoi cells using quadrature samples.

The computation of CVT can also be characterized as a smooth variational problem, and solved with a quasi-Newton method [LWL*09]. This allows a generalization to the L_p norm [LL10] for hex-dominant meshing.

In this paper, we use the smooth variational approach, and replace the approximations used in previous works with an accurate computation of the clipped 3D Voronoi cells, thus making our algorithm independent of the initial discretization.

CVT for line segments A first algorithm to compute segment CVT was proposed in 2D, in the domain of Non-Photorealistic Rendering [HHD03]. The approach moves each segment to the centroid of its Voronoi cell, according to the original definition of CVT. In this setting, it is unclear to see how to handle a connected graph (more on this below). In contrast, we propose a variational characterization of CVT together with a general algorithm to solve the variational problem, that makes incorporating structural constraints easier.

Skeleton Extraction

Numerous methods have been proposed to extract the skeleton of a 3D shape. We refer the reader to the survey [CM07]. Based on the underlying representation, they can be classified into two main families :

Discrete volumetric methods These methods resample the interior of the surface, using for instance voxel grids [JBC07, WL08]. Baran *et al.*[BP07] construct a discretized geometric graph and then minimize a penalty function to penalize differences in the embedded skeleton from the given skeleton. Brunner *et al.*[BB08] build a force field based on the body-centered cubic lattice and then integrate the critical points of the force field into thinning to achieve the skeleton. The advantage of volumetric methods is that, since they resample the object, they are insensitive to poorly shaped triangles in the initial mesh. However, they are limited by the discretization and may lack precision, especially when the mesh has thin features.

Continuous surface methods These methods work on a polygonal mesh directly. It is possible to obtain the skeleton as the Reeb graph of some smooth function [AHLD07, PSBM07]. An alternative was proposed [ATC*08], based on Laplacian smoothing and mesh contraction. However, since it relies on a differential operator on the mesh, it fails to give good results for meshes with bad quality or with unwanted shape features like spikes, hair or fur. Using dense sample of the geometry, [DS06] first compute an approximate medial surface and then prune it to obtain a curve skeleton based on geodesic distance. Moreover, their method cannot extract a continuous skeleton from a mesh with multiple connected components. Methods that extract the skeleton from a segmentation of the model [KT03], [SY07],[dATTS08] suffer from the same limitation.

Some methods do not belong to the above general classes. In [SLSK07], the skeleton points are tracked while a deformable model evolving inside the objects which can be point clouds or polygonal meshes. A post-processing is needed to filter the noisy branches. In [TZCO09], a skeleton is automatically constructed from an incomplete point set, supposing that the object has some generalized rotational symmetries.

In this paper, we propose a volumetric method that directly uses the initial representation of the surface. The interior of the input model is (implicitely) represented by a set of tetrahedra. Therefore our approach shares the advantage of volumetric methods (robustness) and the advantage of surface-based ones (accuracy).

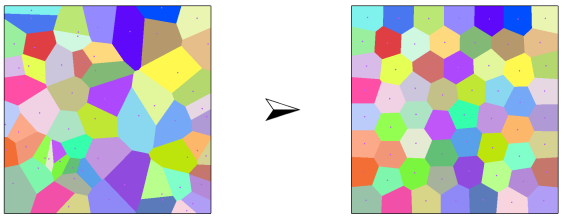


Figure 2: *CVT for points in 2D.*

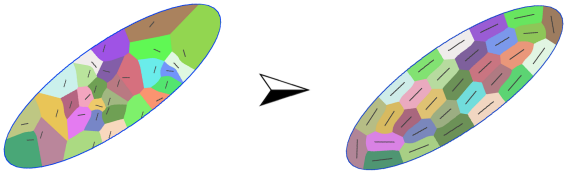


Figure 3: *CVT for segments in an ellipse (with the regularization term, see Section 3.4 below).*

3. Centroidal Voronoi Tessellation for line segments and graphs

We now present our approach to generalizing Centroidal Voronoi Tessellation (CVT) to line segments and graphs. Such a generalization of CVT to line segments has several applications, such as skeleton fitting and segmentation demonstrated here, and also vector field visualization or image stylization. The idea is illustrated in Figure 1. This section is illustrated with 2D examples, but note that the notions presented here are dimension independent. Our results in 3D are shown later in the paper. As shown in Figure 1, starting from an initial configuration of the skeleton, we minimize in each Voronoi cell (colors) the integral of the squared distance to the skeleton (generalized Lloyd energy). This naturally fits the edges of the skeleton into the protrusions of the mesh.

We first recall the usual definition of CVT for points (Section 3.1), then we extend the definition to line segments and graphs (Section 3.2), and introduce the approximation that we will use (Section 3.3). Section 3.4 introduces the regularization term, and Section 4 our solution mechanism.

3.1. CVT for points

CVT has diverse applications in computational science and engineering, including geometry processing [DFG99, ACSYD05] and has gained much attention in recent years. In this paper we extend the definition of CVT to make it applicable for sets of inter-connected line segments (or graphs). We first give the usual definition of Voronoi Diagram and Centroidal Voronoi Tessellation.

Let $\mathbf{X} = (\mathbf{x}_i)_{i=1}^n$ be an ordered set of n seeds in $\mathbb{R}^N$. The Voronoi region $Vor(\mathbf{x}_i)$ of $\mathbf{x}_i$ is defined by :

$$Vor(\mathbf{x}_i) = \{\mathbf{x} \in \mathbb{R}^N \mid \|\mathbf{x}-\mathbf{x}_i\| \leq \|\mathbf{x}-\mathbf{x}_j\|, \forall j \neq i\}.$$

The Voronoi regions of all the seeds form the Voronoi diagram (VD) of $\mathbf{X}$. Let us now consider a compact region $\Omega \subset \mathbb{R}^N$ (for instance, the interior of the square around Figure 2-left). The clipped Voronoi diagram is defined to be the set of clipped Voronoi cells $\{Vor(\mathbf{x}_i) \cap \Omega\}$.

A Voronoi Diagram is said to be a Centroidal Voronoi Tessellation (CVT) if each seed $\mathbf{x}_i$ coincides with the barycenter of its clipped Voronoi cell (geometric characterization).

An example of CVT is shown in Figure 2-right. This definition leads to Lloyd's relaxation [Llo82], that iteratively moves all the seeds to the barycenter of their Voronoi cells. Alternatively, CVT can be also characterized in a variational way [DFG99], as the minimizer of Lloyd's energy F :

$$F(\mathbf{X}) = \sum_{i=1}^{n} f(\mathbf{x}_i) \quad ; \quad f(\mathbf{x}_i) = \int_{Vor(\mathbf{x}_i)\cap\Omega} \|\mathbf{x}-\mathbf{x}_i\|^2 \, d\mathbf{x} \tag{1}$$

Using this latter variational formulation, a possible way of computing a CVT from an arbitrary configuration consists in minimizing F [LWL*09]. Now we explain how this variational point of view leads to a more natural generalization to line segments as compared to the geometric characterization used in Lloyd's relaxation.

3.2. CVT for line segments and graphs

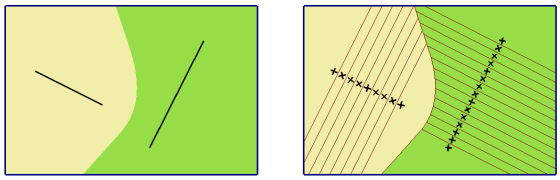
Besides points, it is well known that general objects such as line segments can also be taken as the generators of Voronoi diagrams (see Figure 3). For instance, Hiller *et al.* [HHD03] have used line segments Voronoi diagrams to generalize the notion of stippling used in Non-Photorealistic Rendering. Their method uses a discretization on a pixel grid, and a variant of Lloyd's relaxation to move the segments. They translate each segment to the centroid of its Voronoi cell, and then align it with the cell's inertia tensor. In our case, it is unclear how to apply this method to a set of segments that share vertices (i.e. a graph).

For this reason, we consider the variational characterization of CVT, that we generalize to line segments. Let E be the set of line segments with the end points in $\mathbf{X}$. We define the CVT energy for segment $[\mathbf{x}_i,\mathbf{x}_j]$ in Ω as:

$$g([\mathbf{x}_i,\mathbf{x}_j]) = \int_{Vor([\mathbf{x}_i,\mathbf{x}_j])\cap\Omega} d(\mathbf{z},[\mathbf{x}_i,\mathbf{x}_j])^2 \, d\mathbf{z}, \tag{2}$$

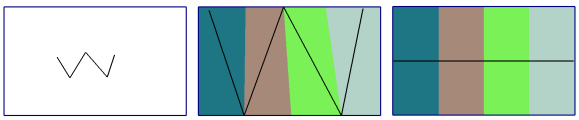
where $Vor([\mathbf{x}_i,\mathbf{x}_j]) = \{\mathbf{z} \in \mathbb{R}^N \mid d(\mathbf{z},[\mathbf{x}_i,\mathbf{x}_j]) \leq d(\mathbf{z},[\mathbf{x}_k,\mathbf{x}_l]), \forall k,l \in E\}$ and $d(\mathbf{z},[\mathbf{x}_i,\mathbf{x}_j])$ denotes the Euclidean distance from a point to the segment.

Particularly, several line segments sharing the same vertices define a graph. We denote this graph by $G := (\mathbf{X}, E)$.



(a) (b)

Figure 4: *Voronoi diagram of two line segments.(a)Accurate VD; (b)approximated VD.*



(a) (b) (c)

Figure 5: *A chained graph with 5 vertices and 4 edges.(a)Input; (b)minimizer of CVT energy without regularization; (c)minimizer of CVT energy with regularization energy* $\gamma = 0.15$.

Definition 1 A CVT for a graph $G = (\mathbf{X}, E)$ is the minimizer $\mathbf{X}$ of the objective function G defined by :

$$G(\mathbf{X}) = \sum_{i,j \in E} g([\mathbf{x}_i, \mathbf{x}_j]). \tag{3}$$

In practice, minimizing G is non-trivial, due to the following two difficulties :

- Although the general abstract algorithm for Voronoi diagram applies in theory to line segments [For86], the problem is in practice very complicated, since the bisector of two segments are curves (resp. surfaces in 3D) of degree 2. Some readily available software solve the $2D$ case (e.g., VRONI [Hel01] and CGAL), but the problem is still open in $3D$. Note that the sub-problem of computing the intersection between two generic quadrics is extremely difficult [WJG02, LMPP06] ;
- supposing the VD of line segments is known, integrating distances over cells bounded by quadrics is non-trivial.

3.3. Approximated CVT for segments and graphs

For these two reasons, we use an approximation. As shown in Figure 4, we replace the segment $[\mathbf{x}_i, \mathbf{x}_j]$ with a set of samples $(\mathbf{p}_k)$:

$$\mathbf{p}_k = \lambda_k \mathbf{x}_i + (1 - \lambda_k)\mathbf{x}_j \quad , \quad \lambda_k \in [0,1].$$

Then the Voronoi cell of the segment can be approximated by the union of all intermediary points' Voronoi cells, and its energy g can be approximated as follows :

$$Vor([\mathbf{x}_i, \mathbf{x}_j]) \simeq \bigcup_k Vor(\mathbf{p}_k) \quad ; \quad g([\mathbf{x}_i, \mathbf{x}_j]) \simeq \sum_k f(\mathbf{p}_k)$$

where f denotes the point-based energy (Equation 1).

This yields the following definition :

Definition 2 An approximate CVT for a graph $G = (\mathbf{X}, E)$ is the minimizer $\mathbf{X}$ of the objective function $\widetilde{G}$ defined by :

$$\widetilde{G}(\mathbf{X}) = \sum_{i,j \in E} \sum_k f(\mathbf{p}_k) = \sum_{i,j \in E} \sum_k f(\lambda_k \mathbf{x}_i + (1 - \lambda_k)\mathbf{x}_j). \tag{4}$$

where f denotes the point-based energy (Equation 1).

Since it is based on a standard Voronoi diagram and Lloyd energy, the so-defined approximated objective function $\widetilde{G}$ is much simpler to optimize than the original G given in Equation 3. Note that G and $\widetilde{G}$ depend on the same variables $\mathbf{X}$. The intermediary samples $\mathbf{p}_k = \lambda_k \mathbf{x}_i + (1 - \lambda_k)\mathbf{x}_j$ are not variables, since they depend linearly on $\mathbf{x}_i$ and $\mathbf{x}_j$.

To avoid degenerate minimizers, we now introduce a stiffness regularization term, similarly to what is done in variational surface design.

3.4. Regularization, stiffness

The CVT energy tends to maximize the compactness of the dual Voronoi cells, which is desired in general. However, some particular configurations may lead to unwanted oscillations. For instance, the (undesired) configuration shown in Figure 5(b) has a lower energy than the one shown in Figure 5(c). Intuitively, the long thin cells in (c) have points that are far away from the skeleton. The problem is also encountered in data analysis and statistical learning theory when computing the *principal curve* that approximates a point cloud [HS89]. To avoid unwanted oscillations, principal curves use a regularization term, defined as the squared length of the curve. To regularize curve and surface fitting, other approaches such as DSI (Discrete Smooth Interpolation) [Mal89] use a criterion that they call "roughness", defined as the Dirichlet energy (squared Laplacian). In a certain sense, it is dual to edge length (that minimizes coordinate's variation along each edge) since it minimizes coordinate's variations on each 1-ring neighborhood. Both criteria can be used by our approach. We observed that the Dirichlet energy results in (slightly) better joint placement in the skeleton fitting application. Our regularization term $R(\mathbf{X})$ is then defined as follows :

$$R(\mathbf{X}) = \sum_{\mathbf{x}_i, v(\mathbf{x}_i) > 1} \|\mathbf{x}_i - \frac{1}{v(\mathbf{x}_i)} \sum_{\mathbf{x}_j \in N(\mathbf{x}_i)} \mathbf{x}_j\|^2, \tag{5}$$

where $v(\mathbf{x}_i)$ denotes the valence, $N(\mathbf{x}_i)$ the neighbors of $\mathbf{x}_i$.

We can now define the objective function F minimized by our approach :

$$\mathcal{F}(\mathbf{X}) = \widetilde{G}(\mathbf{X}) + \gamma |\Omega| R(\mathbf{X}). \tag{6}$$

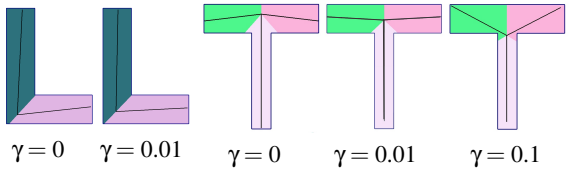


Figure 6: *Influence of the regularization parameter* γ.

where $\widetilde{G}(\mathbf{X})$ is the approximated Lloyd energy of the segments (Equation 4) and $R(\mathbf{X})$ is the regularization term (Equation 5). $\gamma \in \mathbb{R}^+$ denotes the influence of the regularization term. Note that the regularization term R is multiplied by the volume of the object $|\Omega|$ so that $\widetilde{G}$ and R have compatible dimensions.

Figure 6 shows the influence of the parameter γ. For valence 2 nodes, a high value of γ tends to straighten the joints. For branching nodes, a high value of γ tends to homogenize the segment angles and lengths. In our experiments, good results were obtained with $\gamma = 0.01$ for $2D$ and 0.02 for $3D$ shapes. (We used the two values in all the experiments.)

4. Solution mechanism

To minimize the objective function F, we use an efficient quasi-Newton solver. A recent work on the CVT energy showed its C^2 smoothness [LWL*09] (except for some seldom encountered degenerate configurations where it is C^1). In our objective function F, the term $\tilde{G}$ composes linear interpolation with the Lloyd energy, and the regularization term R is a quadratic form. Therefore, F is also of class C^2, which allows us to use second-order optimization methods. As such, Newton's algorithm for minimizing functions operates as follows :

(1) while $\|\nabla F(\mathbf{X})\| > \varepsilon$
(2) solve for $\mathbf{d}$ in $\nabla^2 \mathcal{F}(\mathbf{X})\mathbf{d} = -\nabla F(\mathbf{X})$
(3) find a step length α such that
$F(\mathbf{X} + \alpha\mathbf{d})$ sufficiently decreases
(4) $\mathbf{X} \leftarrow \mathbf{X} + \alpha\mathbf{d}$
(5) end while

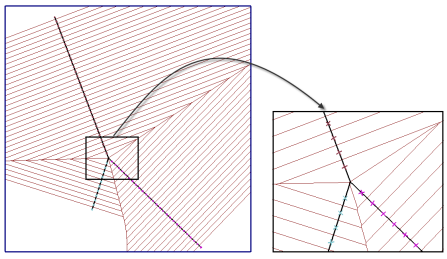


Figure 7: *Illustration of the sampling on the segments.*

where $\nabla F(\mathbf{X})$ and $\nabla^2 F(\mathbf{X})$ denote the gradient of F and its Hessian respectively. Computing the Hessian (second-order derivatives) can be time consuming. For this reason, we use L-BFGS [LN89], a quasi-Newton method that only needs the gradient (first-order derivatives). The L-BFGS algorithm has a similar structure, with a main loop that updates $\mathbf{X}$ based on evaluations of F and ∇F. The difference is that in the linear system of line (2), the Hessian is replaced with a simpler matrix, obtained by accumulating successive evaluations of the gradient (see [LN89] and [LWL*09] for more details).

In practice, one can use one of the readily available implementations of L-BFGS (e.g. TAO/Petsc[BMM*07]).

Thus, to minimize our function F with L-BFGS, what we need now is to be able to compute $F(\mathbf{X})$ and $\nabla F(\mathbf{X})$ for a series of $\mathbf{X}$ iterates, as in the following outline, detailed below (next three subsections).

Algorithm outline - CVT for line segments and graphs
For each Newton iterate $\mathbf{X}$
1. For each segment $[\mathbf{x}_i, \mathbf{x}_j]$, generate the samples $\mathbf{p}_k$;
2. For each $\mathbf{p}_k$, compute the clipped Voronoi cell of $\mathbf{p}_k$, i.e. $Vor(\mathbf{p}_k) \cap \Omega$ where Ω denotes the interior of the surface ;
3. Add the contribution of each $\mathbf{p}_k$ to F and ∇F.

4.1. Generating the samples $\mathbf{p}_k$

We choose a sampling interval h. In our experiments, $2/100^{th}$ of the bounding box's diagonal gives sufficient precision. We then insert a sample every h along the segments. Terminal vertices (of valence 1) are inserted as well.

Vertices $\mathbf{x}_i$ of valence greater than 1 are skipped, in order to obtain a good approximation of the bisectors near branching points (see Figure 7). Another option is to sample the line segment by a fixed number of points so that shorter lines may achieve more sampling density during the evolution. However, our experiments showed little difference between the results obtained with the two methods. Results and timings with different sampling densities are compared in Figure 10.

4.2. Computing the clipped Voronoi cells

We now compute the Delaunay triangulation of the samples $\mathbf{p}_k$ (one may use CGAL for instance), and then the Voronoi diagram is obtained as the dual of the Delaunay triangulation. Then we need to compute the intersections between each Voronoi cell $Vor(\mathbf{p}_k)$ and the domain Ω, defined as the interior of a triangulated surface. Since Voronoi cells are convex (and not necessarily the surface), it is easier (though equivalent) to consider that we clip the surface by the Voronoi cell. To do so, we use the classical re-entrant clipping algorithm [SH74], depicted by in Figure 8-A, that considers a convex window as the intersection of half-spaces applied one-by-one to the clipped object. In our 3D case, when clipping the triangulated surface with a half-space,

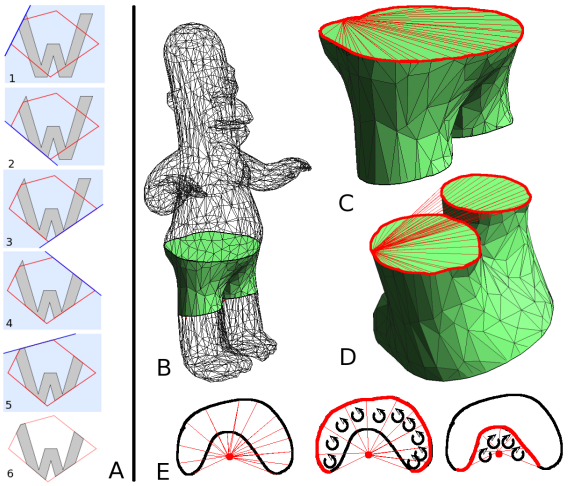


Figure 8: *Sutherland-Hogdman re-entrant clipping.*

each triangle can be considered independently. We show an example in Figure 8-B, where two bisectors generate Homer's "trousers". Since it processes the triangles one by one, the algorithm is extremely simple to implement, and does not need any combinatorial data structure. Moreover, it can be applied to a "triangle soup", provided that the polygons have correct orientations (i.e., coherent normals).

However, it is important to mention that the surface needs to be closed after each half-space clipping operation. This is done by connecting each intersection segment (thick red in Figure 8-C) with the first intersection point, thus forming a triangle fan. Note that when the intersection line is non-convex, this may generate geometrically incorrect configurations, such as the sheet of triangles between the legs in Figure 8-D. However, this configuration is correct from a computational point of view, if we keep the orientation of the triangles, as explained in Figure 8-E. Suppose we want to compute the area of the "bean" shape, by summing triangles connected to the red vertex. With the orientation of the triangles, the extraneous area appears twice, with a positive and negative orientation that cancel-out. After the clipping operation, the clipped Voronoi cell of the point $\mathbf{p}_k$ is represented by its boundary, as a list of triangles $(\mathbf{q}_i, \mathbf{q}_j, \mathbf{q}_k)$. The interior of the Voronoi cell is obtained as a set of oriented tetrahedra $(\mathbf{p}_k, \mathbf{q}_i, \mathbf{q}_j, \mathbf{q}_k)$, created by connecting $\mathbf{p}_k$ to each triangle. Note also that $\mathbf{p}_k$ may be outside its clipped Voronoi cell, but again, with the orientation of the tetrahedra, this still gives the correct result for Lloyd's energy, barycenter and mass computed in the next section.

4.3. Adding the contributions to F and ∇F

Following the variational framework for CVT [LWL*09, LL10], we use the L-BFGS quasi-Newton solver, that needs to compute the value and gradient of the objective function F at each iteration.

We first consider the approximated segment Lloyd energy $\widetilde{G}$, that can be decomposed into a sum of terms. Each term corresponds to one of the tetrahedra computed during the clipping (previous section). The CVT energy associated with a tetrahedron $T = (\mathbf{p}_k, \mathbf{q}_1, \mathbf{q}_2, \mathbf{q}_3)$ is given by

$$\frac{|T|}{10}(\mathbf{U}_1^2 + \mathbf{U}_2^2 + \mathbf{U}_3^2 + \mathbf{U}_1.\mathbf{U}_2 + \mathbf{U}_2.\mathbf{U}_3 + \mathbf{U}_3.\mathbf{U}_1),$$

where $\mathbf{U}_i = \mathbf{q}_i - \mathbf{p}_k$ and $|T|$ denotes the oriented volume of T. To compute the gradients, we first recall the gradient of the point-based Lloyd energy [DFG99], given by :

$$\begin{aligned} &\tfrac{\partial F}{\partial \mathbf{p}_k} = 2m_k(\mathbf{p}_k - \mathbf{c}_k) \\ &\text{where:} \\ &m_k = \int_{Vor(\mathbf{p}_k)\cap\Omega} d\mathbf{x} \quad ; \quad \mathbf{c}_k = \tfrac{1}{m_k}\int_{Vor(\mathbf{p}_k)\cap\Omega} \mathbf{x}d\mathbf{x} \end{aligned} \tag{7}$$

By applying the chain rule to the expression of the intermediary points $\mathbf{p}_k = \lambda_k \mathbf{x}_i + (1-\lambda_k)\mathbf{x}_j$, we obtain the gradient of $\widetilde{G}$ for the edge $[\mathbf{x}_i, \mathbf{x}_j]$:

$$\begin{aligned} \frac{\partial \widetilde{G}}{\partial \mathbf{x}_i} &= \sum_k \frac{\partial f(\mathbf{p}_k)}{\partial \mathbf{x}_i} \\ &= \sum_k \frac{\partial f(\mathbf{p}_k)}{\partial \mathbf{p}_k}\frac{\partial \mathbf{p}_k}{\partial \mathbf{x}_i} = 2\sum_k m_k(\mathbf{p}_k - \mathbf{c}_k)\lambda_k; \\ \frac{\partial \widetilde{G}}{\partial \mathbf{x}_j} &= \sum_k \frac{\partial f(\mathbf{p}_k)}{\partial \mathbf{x}_j} = 2\sum_k m_k(\mathbf{p}_k - \mathbf{c}_k)(1-\lambda_k), \end{aligned}$$

where m_k and $\mathbf{c}_k$ denote the mass and the centroid of $Vor(\mathbf{p}_k)\cap\Omega$ respectively (see Equation 7). Note that m_k and $\mathbf{c}_k$ can be easily computed from the clipped Voronoi cell, represented by a union of oriented tetrahedra (see Section 4).

The gradient of $\widetilde{G}$ with respect to the graph vertex $\mathbf{x}_i$ gathers the contributions of all the gradients of the sampling points $\mathbf{p}_k$ in the segments incident to $\mathbf{x}_i$.

The contribution of the regularization term to the gradient ∇R is given by :

$$\begin{aligned} \frac{\partial R}{\partial \mathbf{x}_i} &= 2\left(\mathbf{x}_i - \frac{1}{v(\mathbf{x}_i)}\sum_{\mathbf{x}_j\in N(\mathbf{x}_i)}\mathbf{x}_j\right) \\ &- 2\sum_{\mathbf{x}_j\in N(\mathbf{x}_i), v(\mathbf{x}_j)>1}\frac{1}{v(\mathbf{x}_j)}\left(\mathbf{x}_j - \frac{1}{v(\mathbf{x}_j)}\sum_{\mathbf{x}_k\in N(\mathbf{x}_j)}\mathbf{x}_k\right). \end{aligned}$$

5. Results and discussion

We have experimented with our method using several datasets. We used $\gamma = 0.01$ for the 2D examples and $\gamma = 0.02$ for the 3D examples. The stopping criterion is $\|\nabla F(\mathbf{X})\| \leq$

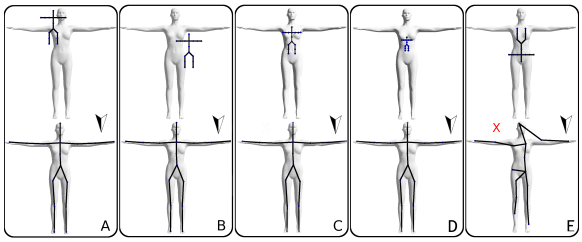


Figure 9: *Influence of the initialization on the Victoria model.*

10^{-7} or the improvement of $F(\mathbf{X})$ between two iterations is less than 10^{-10}.

Initialization

The initial skeleton is assigned manually; however, in most cases we can use a skeleton template for humans or quadruped. Once it is assigned, its structure will not change along the optimization. As demonstrated in Figure 9, our method is reasonably independent of the initial position (A,B) and initial scale (C,D), but may fail for more extreme initially mismatched configurations (E). In subsequent tests, initialization is provided by aligning the bounding box of the skeleton, as done in [BP07].

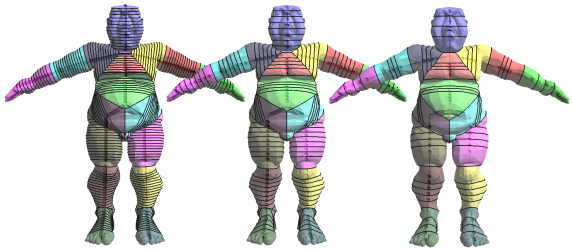
Sampling

Formally, analyzing the influence of the sampling requires an in-depth mathematical study, as done in [Bra94]. Here we limit ourselves to give some experimental data. Figure 10 shows the influence of the edge sampling interval h, used to approximate the segment Voronoi cells. As can be seen, a coarse sampling is sufficient to obtain a result similar to the one obtained with a finer sampling. We use $h = 2/100^{th}$ of the bounding box diagonal (upper middle) to achieve the efficiency, and increase the sampling to $h = 1/100^{th}$ of the bounding box diagonal (upper left). Note that we use the interval h in generating the samples; therefore, the number of total samples may change during the optimization process.

Robustness

Our method is robust to mesh degeneracies (Figure 11), such as T-Junctions, and small gaps (red lines), thanks to the accurate computation of clipped Voronoi cells (thick black lines) and the triangle-by-triangle (or tet-by-tet) integration. Figure 14 shows that noisy meshes with skinny triangles can be processed without any numerical instability (data acquired by the Visual Hull technique). To our knowledge, this is the first method that achieves this degree of robustness.

We show in Figure 12 some examples from the SHREK database. More examples are shown in Figure 13. Except for a small number of failure cases (red crosses), satisfying results were obtained. Our method was successfully applied to meshes with degeneracies (multiple components, holes, T-junctions). In addition, a segmentation is obtained, that can be used for matching features between meshes and morphing. Our method failed in some cases. One example of failure is shown in Figure 12 (red cross), where the arms are close to the body.



h	#smpl	#iter	time (s)
0.03	90	3	3.8
0.025	106	3	6.3
0.02	126	3	6.9
0.015	165	3	12.1
0.01	236	3	17.2

Figure 10: *Influence of the skeleton sampling. The sampling interval for the cartoon human model is $h = 0.01, 0.02, 0.03$ from left to right. (The bounding box diagonal is normalized to 1. The model has 7624 vertices and 18 nodes in the skeleton.) In the table, #smpl denotes the final number of sample points; #iter is the number of Newton iterations.*

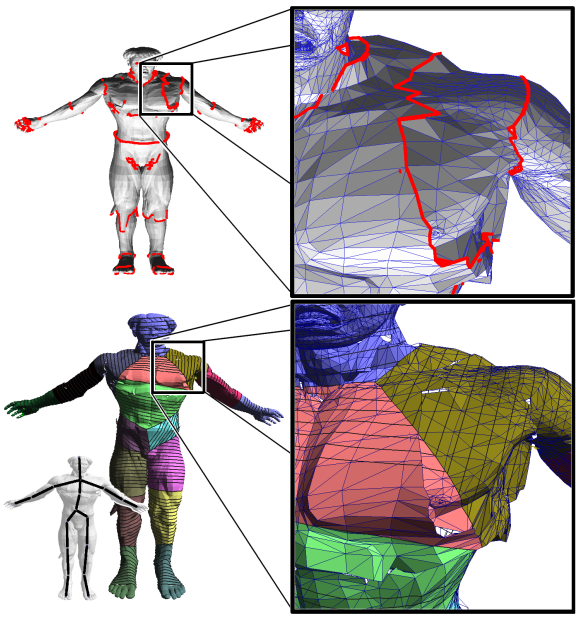


Figure 11: *Robustness to mesh degeneracies (small holes) for model* Hulk.

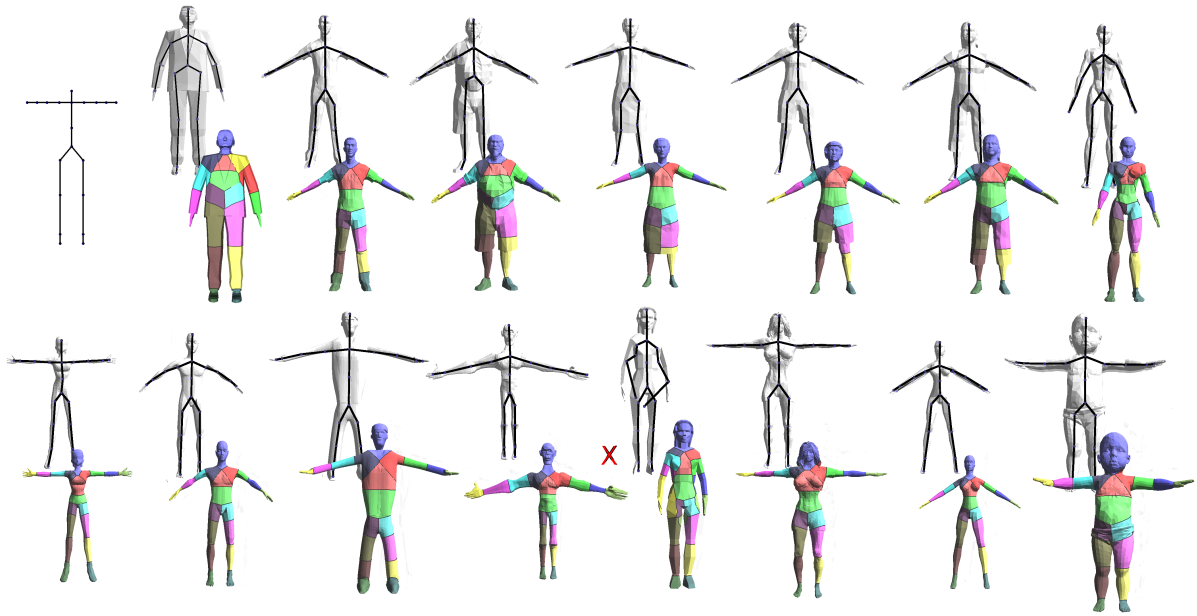


Figure 12: *Humans from the SHREK database.The initial skeleton is shown on the upper left.*

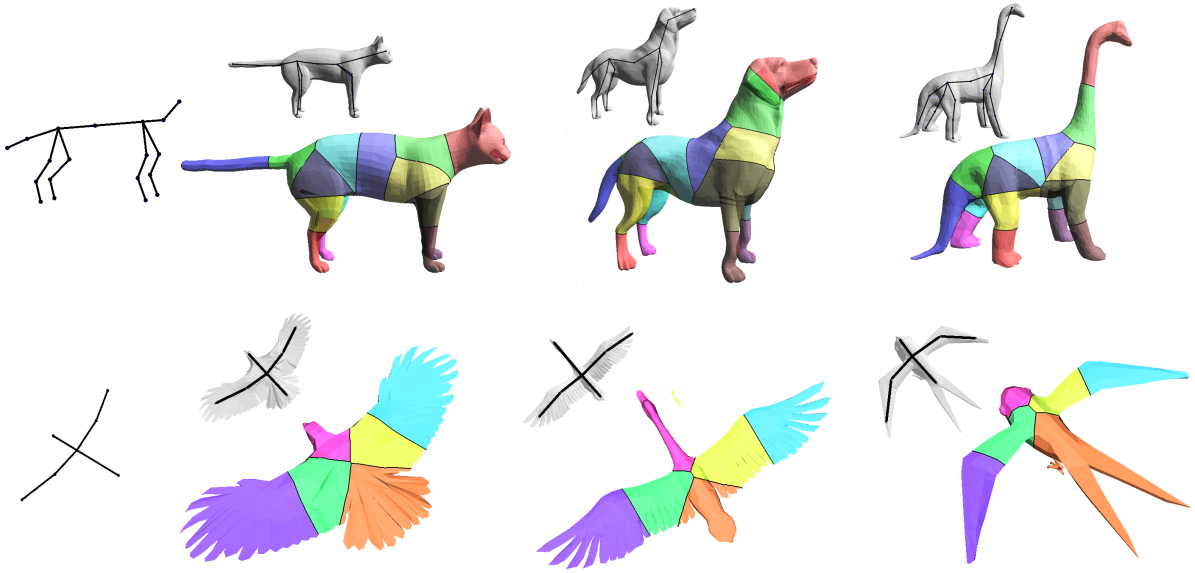


Figure 13: *More results on quadrupeds and birds of our method. The initial skeletons are shown on the left.*

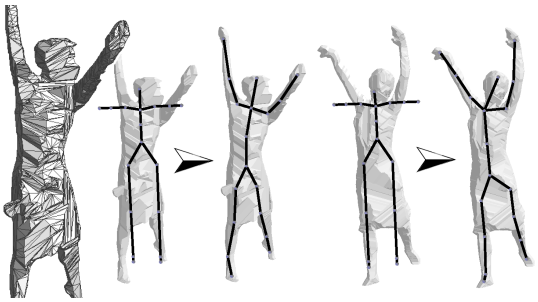


Figure 14: *Robustness to mesh degeneracies (skinny triangles) for model* Dancer.

In Figure 15, we show the results from mesh contraction, medial geodesic function [DS06] and force field method [BB08], and then apply our resulted skeleton to animation in the framework of Pinocchio [BP07]. In our case, we acknowledge that the problem is easier since our method makes use of the template skeleton as a "prior".

Performance

Some timing statistics are given in Table 5. Results obtained with a multithreaded implementation, on a quad-core 2.0 GHz machine.

The performance of models in our experiments

model	#vert	#nodes	#smpl	#iter	time(s)
Dancer	2581	18	94	7	7.9
Victoria	1024	16	107	6	12.5
	5024	16	103	5	7.8
	10024	16	106	7	27.9
Hulk	15089	18	113	7	16.2
Dinosaur[1]	5002	19	108	5	10.8
Dinosaur[2]	2500	19	107	5	4.32
Cat	7207	19	84	3	4.37
Dog	5435	19	133	6	14.2

#vert refers to the number of vertices; #smpl is the final number of sample points; #iter is the number of Newton iterations; the column of time gives the total time of the optimization in seconds.

6. Conclusion

In this paper, we have introduced a simple way of generalizing the notion of Centroidal Voronoi Tesselation (CVT). We think that considering CVT in terms of energy minimization rather than centroids is a more general point of view, leading to interesting generalizations. In particular, we showed that primitives that are more general than points can be used. Beyond the graph of segments considered here, in future work we will consider CVT-based shape optimization by defining CVT with other types of primitives (i.e. cylinders, plates . . .).

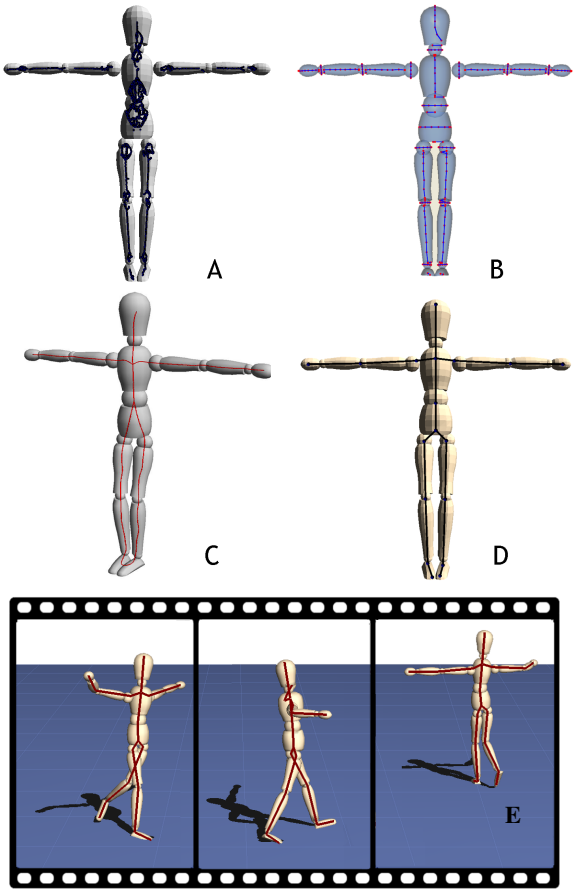


Figure 15: *Comparison with Dey* et al.*'s method (A), Au* et al.*'s mesh-contraction (B), Brunner* et al.*'s force field method (C) and our method (D), that can be used to transfer an animation with the Pinnochio system [BP07] (E). Note that the original method in [BP07] requires a single-component mesh and thus does not process this example.*

Acknowledgements

The authors wish to thank the anonymous reviewers for their remarks that helped us improving the paper. This work is partly supported by the European research council (GOODSHAPE ERC-StG-205693), by the ANR (MORPHO), by the China National Natural Science Foundation (U1035004) and Innovation Fund of Shandong University (2010HW010).

References

[ACSYD05] ALLIEZ P., COHEN-STEINER D., YVINEC M., DESBRUN M.: Variational tetrahedral meshing. *ACM Transactions on Graphics 24*, 3 (2005), 617–625. 2, 3

[AHLD07] AUJAY G., HÉTROY F., LAZARUS F., DEPRAZ C.: Harmonic skeleton for realistic character animation. In *SCA* (2007), pp. 151–160. 2

[AMD02] ALLIEZ P., MEYER M., DESBRUN M.: Interactive geometry remeshing. *ACM Transactions on Graphics 21* (July 2002), 347–354. 2

[ATC*08] AU O. K.-C., TAI C.-L., CHU H.-K., COHEN-OR D., LEE T.-Y.: Skeleton extraction by mesh contraction. In *ACM TOG, SIGGRAPH conf. proc.* (2008), pp. 1–10. 2

[BB08] BRUNNER D., BRUNNETT G.: Fast force field approximation and its application to skeletonization of discrete 3d objects. *Computer Graphics Forum 27*, 2 (2008), 261–270. 2, 9

[BMM*07] BENSON S., MCINNES L. C., MORÉ J., MUNSON T., SARICH J.: TAO user manual (revision 1.9). http://www.mcs.anl.gov/tao. 5

[BP07] BARAN I., POPOVIĆ J.: Automatic rigging and animation of 3d characters. *ACM Trans. Graph. 26*, 3 (2007), 72. 2, 7, 9

[Bra94] BRANDT J. W.: Convergence and continuity criteria for discrete approximations of the continuous planar skeleton. *CVGIP: Image Understanding 59*, 1 (1994), 116–124. 7

[CM07] CORNEA N. D., MIN P.: Curve-skeleton properties, applications, and algorithms. *IEEE Transactions on Visualization and Computer Graphics 13*, 3 (2007), 530–548. Member-Silver, Deborah. 2

[CSAD04] COHEN-STEINER D., ALLIEZ P., DESBRUN M.: Variational shape approximation. In *ACM TOG (SIGGRAPH conf. proc.)* (New York, NY, USA, 2004), ACM, pp. 905–914. 2

[dATTS08] DE AGUIAR E., THEOBALT C., THRUN S., SEIDEL H.-P.: Automatic conversion of mesh animations into skeleton-based animations. vol. 27, pp. 389–397. 2

[DFG99] DU Q., FABER V., GUNZBURGER M.: Centroidal Voronoi tessellations: applications and algorithms. *SIAM Review* (1999). 2, 3, 6

[DS06] DEY T. K., SUN J.: Defining and computing curve-skeletons with medial geodesic function. In *SGP* (2006), Eurographics Association, pp. 143–152. 2, 9

[For86] FORTUNE S.: A sweepline algorithm for voronoi diagrams. In *Symp. Comp. Geom. proc.* (1986), ACM. 4

[Hel01] HELD M.: VRONI: An engineering approach to the reliable and efficient computation of voronoi diagrams of points and line segments. *Computational Geometry: Theory and Applications 18*, 2 (2001), 95 – 123. 4

[HHD03] HILLER S., HELLWIG H., DEUSSEN O.: Beyond stippling- methods for distributing objects on the plane. *Computer Graphics Forum 22*, 3 (2003), 515–522. 2, 3

[HS89] HASTIE T., STUETZLE W.: Principal curves. *Journal of the American Statistical Association 84*, 406 (1989). 4

[JBC07] JU T., BAKER M. L., CHIU W.: Computing a family of skeletons of volumetric models for shape description. *Comput. Aided Des. 39*, 5 (2007), 352–360. 2

[KT03] KATZ S., TAL A.: Hierarchical mesh decomposition using fuzzy clustering and cuts. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers* (New York, NY, USA, 2003), ACM, pp. 954–961. 2

[LL10] LÉVY B., LIU Y.: Lp centroidal voronoi tesselation and its applications. *ACM TOG (SIGGRAPH conf. proc.)* (2010). 2, 6

[Llo82] LLOYD S. P.: Least squares quantization in PCM. *IEEE Transactions on Information Theory 28*, 2 (1982), 129–137. 3

[LMPP06] LAZARD S., MARIANO PENARANDA L., PETITJEAN S.: Intersecting Quadrics: An Efficient and Exact Implementation. *Computational Geometry 35* (2006), 74–99. 4

[LN89] LIU D. C., NOCEDAL J.: On the limited memory BFGS method for large scale optimization. *Mathematical Programming: Series A and B 45*, 3 (1989), 503–528. 5

[LS10] LANDRENEAU E., SCHAEFER S.: Scales and scale-like structures. *Computer Graphics Forum 29*, 5 (2010), 1653–1660. 2

[LWL*09] LIU Y., WANG W., LÉVY B., SUN F., YAN D.-M., LU L., YANG C.: On centroidal voronoi tessellation—energy smoothness and fast computation. *ACM Trans. Graph. 28*, 4 (2009), 1–17. 2, 3, 5, 6

[Mal89] MALLET J.-L.: Discrete smooth interpolation. *ACM Trans. Graph. 8* (April 1989), 121–144. 4

[PSBM07] PASCUCCI V., SCORZELLI G., BREMER P.-T., MASCARENHAS A.: Robust on-line computation of reeb graphs: simplicity and speed. *ACM Trans. Graph. 26*, 3 (2007), 58. 2

[SH74] SUTHERLAND I. E., HODGMAN G. W.: Reentrant polygon clipping. *Commun. ACM 17*, 1 (1974), 32–42. 5

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

[SY07] SCHAEFER S., YUKSEL C.: Example-based skeleton extraction. In *SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2007), Eurographics Association, pp. 153–162. 2

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

[VC04] VALETTE S., CHASSERY J.-M.: Approximated centroidal Voronoi diagrams for uniform polygonal mesh coarsening. *Computer Graphics Forum (Proc. Eurographics)* (2004). 2

[VCP08] VALETTE S., CHASSERY J. M., PROST R.: Generic remeshing of 3d triangular meshes with metric-dependent discrete voronoi diagrams. *IEEE Transactions on Visualization and Computer Graphics 14*, 2 (2008), 369–381. 2

[WJG02] WANG W., JOE B., GOLDMAN R.: Computing quadric surface intersections based on an analysis of plane cubic curves. *Graph. Models 64*, 6 (2002), 335–367. 4

[WL08] WANG Y.-S., LEE T.-Y.: Curve-skeleton extraction using iterative least squares optimization. *IEEE Transactions on Visualization and Computer Graphics 14*, 4 (2008), 926–936. 2