Variational Shape Approximation

David Cohen-Steiner
Duke U.

Pierre Alliez
INRIA

Mathieu Desbrun
U. of So. Cal.

Abstract

A method for concise, faithful approximation of complex 3D datasets is key to reducing the computational cost of graphics applications. Despite numerous applications ranging from geometry compression to reverse engineering, efficiently capturing the geometry of a surface remains a tedious task. In this paper, we present both theoretical and practical contributions that result in a novel and versatile framework for geometric approximation of surfaces. We depart from the usual strategy by casting shape approximation as a variational geometric partitioning problem. Using the concept of geometric proxies, we drive the distortion error down through repeated clustering of faces into best-fitting regions. Our approach is entirely discrete and error-driven, and does not require parameterization or local estimations of differential quantities. We also introduce a new metric based on normal deviation, and demonstrate its superior behavior at capturing anisotropy.

Keywords: surfaces, geometric approximation, geometric error metrics, Lloyd's clustering algorithm, anisotropic remeshing.

1 Introduction

Finding a concise, yet geometrically-faithful digital representation of a surface is at the core of several research themes in graphics. Given the excessive verbosity of many 3D datasets (and in particular, of scanned meshes), reducing the number of mesh elements (triangles, quads, or polygons) of a surface mesh while maintaining its geometric fidelity is crucial for subsequent geometry processing. Ideally, each element should be made as *efficient* as possible by stretching it locally in order to fit a large area of the shape we wish to approximate while minimizing geometric error. This quest for geometric efficiency naturally raises the following question: given a 3D surface, a target number of face elements, and an error metric, what is the best geometric approximation of the object that one can find with this face budget? Or similarly, given a distortion tolerance, what is the smallest polygonal mesh approximant with a distortion lesser than the tolerance? Despite the fundamental nature of this question, its cry for an NP-hard optimization problem has led most researchers to shy away from the search for "optimal" meshes. In this paper, we present a novel approach where shape approximation is tackled as a discrete, variational partitioning problem for which provably-good heuristics are readily available.

1.1 Related Work

Many techniques have been specifically designed to exploit an object's local planarity, symmetry and features in order to optimize its geometric representation. While most simplification approaches try to provide an ϵ-approximation with respect to various metrics, rare are the methods that target a minimum distortion error for a given budget of linear mesh elements.

Partitioning A powerful solution to mesh simplification is to greedily cluster geometric elements, creating in effect a partition of the original object. Mesh decimation provides an elegant approach
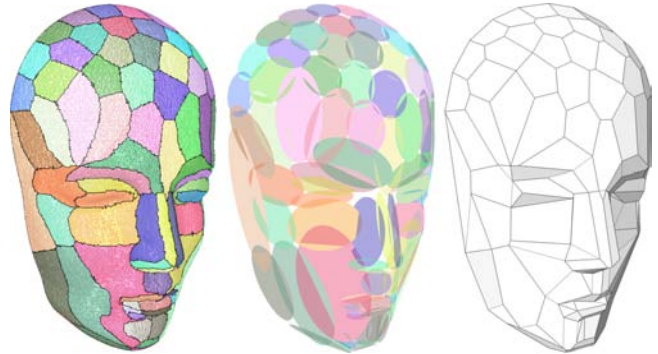
Figure 1: *Variational Shape Approximation: Through repeated error-driven partitioning (left), we find a set of geometric proxies (represented as ellipses, center) providing a concise geometric description of an input surface (62K triangles) by efficiently capturing the anisotropy of the initial model; notice the presence of disks on near-spherical regions, and stretched ellipses on near-parabolic regions. These proxies are then used to construct an approximating polygonal mesh (right). No user interaction, parameterization, or differential estimates have been used; total processing time: 3s.*

to such a partitioning, through greedy and repeated collapsing of mesh elements [Hoppe 1996; Klein et al. 1996; Garland and Heckbert 1998; Lindstrom and Turk 1998]. However, and although some of the metrics used for clustering can be proven asymptotically optimal (i.e., for infinitesimal triangles) for the $\mathcal{L}^2$ metric [Heckbert and Garland 1999], the greedy nature of decimation leads to suboptimal meshes. A similar statement is true for another (almost dual) family of approaches [Maillot et al. 1993; Kalvin and Taylor 1996; Inoue et al. 1999; Sheffer 2001; Sander et al. 2001; Garland et al. 2001; Grinspun and Schröder 2001; Lévy et al. 2002] which gather faces in a set of characteristic regions to provide a succinct, higher-level description of the geometry. Even when this process is iterated to improve the results [Shlafman et al. 2002; Katz and Tal 2003], no attempt is made at minimizing a well-defined geometric error.

Global optimization Contrasting sharply with the previous greedy techniques, Hoppe *et al.* [Hoppe et al. 1993] proposed to cast mesh simplification as an optimization problem. With an energy functional measuring deviation from the input mesh, they showed that optimizing the number of vertices, as well as their positions and their connectivity, captures the curvature variations and features of the original geometry. Although their functional is mostly a point-to-surface Euclidean distance, they report excellent results for mesh simplification. This method was extended later on to also use an image metric, in order to optimize the mesh not only through its geometry, but using its texture and normals [Lindstrom and Turk 2000]. Despite a spring force restricting the anisotropy of the results, such optimization techniques often result in irregular meshes for which geometric efficiency (i.e., how many faces are needed to capture geometry) is particularly good. While other methods use some form of local mesh optimization (see, for instance, [Balmelli et al. 2002; Ohtake et al. 2003b]), this subject remains marginally studied to date, most certainly because the mere size of the search space hampers efficiency.

Anisotropy Remeshing techniques [Turk 1992; Lee et al. 1998; Kobbelt et al. 1999; Guskov et al. 2000] are often much less concerned by approximation efficiency than by the quality of the mesh elements. For instance, the new vertices are very often left on the

original manifold, resulting in rather poor visual results for extreme simplification. However, when a succinct and accurate geometric representation is needed, a strategic placement and aspect ratio of the mesh elements is crucial, and leads to a strikingly significant pay-off in terms of the ratio of mesh size to geometric precision [Simpson 1994; Borouchaki and Frey 1998]. This is a consequence of the natural *anisotropic nature* of most surfaces: as brought to light in recent graphics work [Interrante et al. 1996; Girshick et al. 2000; Rössl and Kobbelt 2000; Hertzmann and Zorin 2000], the main traits of an originally oversampled mesh can be extracted from a close inspection of the curvature tensor field. Aligning either strokes (as done by caricaturists) or mesh edges (as done in anisotropic remeshers [Botsch and Kobbelt 2001; Alliez et al. 2003]) along these curvature lines results in a particularly effective way to describe the geometry of a surface by respecting local symmetries and key features that govern lighting effects. Although such a strategy increases the mesh efficiency by matching the conditions of optimality for the $\mathcal{L}^2$ metric in the limit (see Section 2.1), there is no theoretical guarantee of its efficiency of approximation at coarse scales; additionally, local approximations of differential curvatures, known to be arguable on discrete meshes, render these methods more prone to suboptimal results. In this paper, we argue that a direct, discrete approximation technique should exhibit such an adaptation to anisotropy asymptotically (as it is a good sanity check), *not* be artificially guided by it at coarse scales.

Although good practical approaches for shape approximation have been proposed in the past, only marginal work has been devoted to global minimization of approximation error with respect to a chosen metric. This is most regrettable given the slew of applications that could benefit from an efficient shape approximation algorithm—animation, automatic normal map or geometry image layout (see [Sander et al. 2003]), optimized coarse meshing for multiresolution analysis, remeshing, and progressive compression to name a few. Contrary to most techniques proposed so far that were either based on greedy approaches or guided by results valid only asymptotically, we develop a theoretical and practical framework to help with the difficult problem of finding a provably good trade-off between conciseness and geometric distortion.

1.2 Contributions

We propose a new strategy for the design of succinct and efficient shape approximations. Our approach is entirely error-driven, and uses a novel discrete, variational method that does not resort to any estimation of differential quantities or parameterization. To achieve our goal, we introduce novel geometric concepts: we define *geometric proxies* as a best-fit geometric surrogate to effectively circumvent topological issues (Section 2.2); we then define proper *shape error metrics* to measure how well a proxy fits a piece of geometry (Section 2.3.2); finally, we cast the approximation problem as a *variational partitioning* problem (Section 3.1). The resulting algorithm (Section 3), an extension of fast clustering techniques, generates efficient geometric approximations of arbitrary triangulated geometry. We also introduce a polygonal remeshing algorithm based on our contributions (Section 4), and demonstrate the interest of our discrete, variational approach through multiple results in Section 5.

2 Shape Approximation

In this section, we start with a brief background on approximation theory applied to surfaces, and motivate our new approach to shape approximation through variational partitioning.

2.1 Approximation Theory

Approximation theory deals with the problem of replacing complicated mathematical objects with simpler ones while keeping the primal information content. As we are about to see, a lot of work has been done on best approximations of smooth functions—yet, less is known when it comes to approximating geometry.

Figure 2: *Lion-vase (400K-face model): Our shape approximation algorithm distributes mesh elements according to local surface complexity. (Right) Flat shaded comparison between original model and its 5K-polygonal $\mathcal{L}^{2,1}$-approximation (5 min, no user interaction); notice the good preservation of shape and highlights.*

Functional Setting Given a class of functions and a metric (usually $\mathcal{L}^p$ or $\mathcal{L}^\infty$), approximation theory has provided strong results on the best approximations with n elements, be they piecewise-constant elements or higher order ones. Such results have given rise, for example, to optimal image encoders that give the Kolmogorov entropy bounds of the problem at hand [Cohen et al. 2001]. However, most of these results cannot be easily extended to surfaces: the functional setting relies on a parameterization when comparing two functions. In the general case of two arbitrary surfaces, with no mapping known from one to the other, the functional metrics cannot be used directly.

Height Field Approximation and Notion of Efficiency For the special case of height fields where an obvious parameterization can be readily used, a few results are known about the optimality of piecewise-linear approximation *at the asymptotic limit when the areas of the approximating elements (typically, triangles) vanish*. It has been proven that with respect to the $\mathcal{L}^2$ metric, the triangulation that minimizes the error of piecewise linear interpolation for a given *large* number of triangles must have an optimal triangle's orientation aligned with the eigenvectors of the Hessian of the function, and an optimal size in each principal direction given by the reciprocal square root of the absolute value of the corresponding eigenvalue [Nadler 1986][1]. Such an alignment and stretching of the triangles optimizes the *efficiency* of the mesh, i.e., minimizes the error per surface area. Results mostly identical are also proven for an arbitrary $\mathcal{L}^p$ metric [Simpson 1994]. A few results are also known for optimal approximation of the gradient error [D'Azevedo and Simpson 1991], or for bilinear approximation [D'Azevedo 2000], but again, they are only asymptotically valid. These different results are fairly narrow in scope: first, they are restricted to height fields; second, the triangulations are assumed to be *interpolating* the height field at vertices, which does not seem like the optimal way to closely approximate geometry at a coarse level; and third, the asymptotical case does not help in designing a concrete surface approximation for a small number of triangles. Recent progress on concrete bounds for the interpolation error and the gradient interpolation error for a non-infinitesimal triangle [Shewchuk 2002] offers much better insights, but still does not provide, to date, a better approach for practical mesh generation. In fact, it is known that finding the piecewise-linear triangulation with a given number of vertices that optimally approximates a height field with respect to the $\mathcal{L}^\infty$ metric is a NP-hard problem [Agarwal and Suri 1998].

[1]Note that there is a subtle twist on hyperbolic regions, where there is not a unique optimal shape and direction, but a whole family of them; we will come back to this impediment in Section 2.3.1

Arbitrary Geometry   Aside from the asymptotic results mentioned above, theoretical knowledge on optimally efficient piecewise linear approximation of arbitrary surfaces is mostly unchartered territory despite the considerable amount of practical work on digital geometry. This lack of foundations and the intrinsic complexity of this problem explains the overwhelming usage of greedy algorithms, that can reduce the number of triangles but at the expense of an uncontrollable approximation error, or conversely, can guarantee a given approximation error criterion but at the expense of an uncontrollable number of triangles (with the noticeable exception of a few computational geometry papers proposing algorithms for convex and bivariate surfaces [Agarwal and Suri 1998], or about optimally-sparse ϵ-sampling for accurate surface reconstruction [Amenta and Bern 1999; Boissonnat and Oudot 2003]).

The notion of *distance* between two surfaces is, however, routinely used. Probably the most used metric in graphics, the $\mathcal{L}^p$ distance between a surface X and an approximating surface Y is the extension of the traditional $\mathcal{L}^p$ metric for the functional setting, and is often defined as:

$$\mathcal{L}^p(X,Y) = \left(\frac{1}{|X|} \iint\limits_{x \in X} \|d(x,Y)\|^p dx \right)^{\frac{1}{p}}$$

$$\text{with: } d(x,Y) = \inf_{y \in Y} \|x - y\|$$

where $\|.\|$ is the Euclidean distance, while $|.|$ is the surface area. The extension of the $\mathcal{L}^\infty$ metric, called the *Hausdorff distance*, is naturally expressed as: $\mathcal{H}(X,Y) = \max_{x \in X} d(x,Y)$, but can be quite delicate to compute [Aspert et al. 2002]. Notice that these definitions are *sided*: a true distance measure should add the symmetrized version. However, in the context of surface approximation, the symmetric counterpart increases the complexity significantly as it contains an integral over the unknown surface. It is thus discarded in practice, as in [Hoppe et al. 1993]) for instance.

2.2   Variational Partitioning and Proxies

Given the theoretical difficulty in finding a piecewise-linear approximation of geometry with optimal efficiency, we propose to reformulate the problem of surface approximation by introducing the notions of shape proxies and variational partitions.

Removing Topology from the Search   A best geometric approximation has no obvious reason to preserve the topology of the input surface. Imagine a square-like 2-manifold flat almost everywhere, except for a high and thin fin in the middle: a best approximation with two piecewise-linear elements is most likely a non-manifold surface made out of a quad (for the flat part) and a triangle (for the fin). Similarly, a nearly flat surface with multiple tiny holes is very well approximated with a single quad: the initial object and its best 1-element approximant do not have the same genus. The topology of the approximant should therefore be *automatically induced* from the best placements of a given budget of elements, possibly producing a non-simplicial, polygonal mesh. Thus, disregarding topological considerations of the resulting mesh seems not only reasonable, but highly desirable. Moreover, the search space gets considerably simplified, avoiding the delicate simultaneous (or alternating) optimization of vertices' positions *and* connectivity at the same time [Hoppe et al. 1993]. However, we now face the issue of defining an approximation quality *not knowing* the topology of the resulting approximant.

Approximation Through Partitioning   Agarwal and Suri [1998] mentioned that the problem of functional approximation can be cast as a *geometric partitioning* one. This idea of clustering points or faces of a 3D objects into a partition to help approximate the geometry has already been used many times in graphics [Kalvin and Taylor 1996; Heckel et al. 2001; Pauly et al. 2002], and particularly for mechanical parts [Inoue et al. 1999; Sheffer 2001], where clear-cut features make the partitioning easier. After all, an approximating face is nothing but a surrogate, linear approximant for a set of original clustered faces that share,

on average, similar geometric characteristics. Therefore, clustering faces into a partition with k regions appears to be a natural way to efficiently resample geometry (see Figure 3). However, clustering is traditionally achieved in a greedy fashion. Although we base our geometric approximation on partitioning through clustering too, we will see in the next section that we iteratively seek a *partition that minimizes a given error metric* (hence the name "variational partitioning"). We start by defining our terminology.
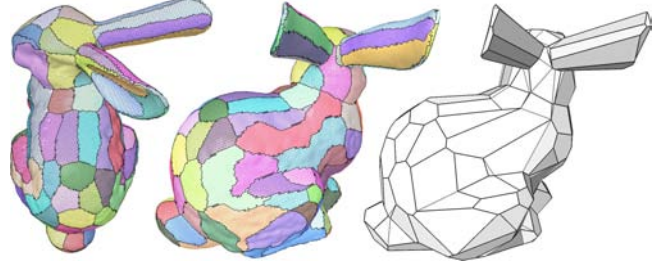

Figure 3: *Bunny: (left and center) $\mathcal{L}^{2,1}$-optimized geometric partitioning; (right) Anisotropic polygonal mesh deduced from the partition. Notice the stretching of the elements on the ears.*

Partition and Proxies   Each region $\mathcal{R}_i$ of a partition $\mathcal{R}$ can be summarily represented to first order as an "average" point X_i and an "average" normal $\mathbf{N}_i$ (the word *average* is here used in a broad sense; it will be made clear in Section 2.3 when we define a metric with respect to which these averages will represent the best local linear fit). We will denote such a local representative pair $P_i = (X_i, \mathbf{N}_i)$ a *shape proxy* of the associated region. Thus, for any given partition of a surface in k regions, we associate a set P of shape proxies $\{P_i\}_{i=1..k}$ that approximate the whole geometry.

At this point, it is worthwhile to point out that a k-partition, in effect, defines a *dual meta-mesh* of the original: the proxies define k dual meta-faces (obtained through clustering of original faces), and the connectivity of the k regions of the partition *induces* the topology of this dual mesh: as we claimed earlier, focusing on partitioning drastically simplifies our task since the final connectivity of the approximant is handled implicitly. Note that the topology of the approximant and the original one do not have to match: the approximant will automatically filter geometrically-irrelevant topological details, such as the tiny handles often present on scanned meshes [Wood et al. 2004]. Now, for this approximant to be relevant, we need to evaluate the quality of the partition—in order to find a partition with near-optimal quality.

2.3   Metrics on Proxies

Defining an appropriate error metric is a key ingredient in approximation. As mentioned earlier, the $\mathcal{L}^2$ or Hausdorff metrics are often used when comparing two triangulated surfaces. In our case, we want to measure the geometric relevance of a proxy set for a given surface: new definitions of error metrics are thus presented next. It will allow us to "score" a partition in terms of how well it approximates a surface.

2.3.1 $\mathcal{L}^2$ Metric for Proxies

We can easily extend the notion of $\mathcal{L}^2$ distance to our framework. Given a region, $\mathcal{R}_i$, and its associated proxy, $P_i = (X_i, \mathbf{N}_i)$, we denote $\Pi_i(.)$ the orthogonal projection of the argument on the "proxy" plane going through X_i and normal to $\mathbf{N}_i$; the $\mathcal{L}^2$ metric is then:

$$\mathcal{L}^2(\mathcal{R}_i, P_i) = \iint\limits_{x \in \mathcal{R}_i} \|x - \Pi_i(x)\|^2 dx. \qquad (1)$$

This formula (from which we have removed the usual square root and area normalization, irrelevant for optimization purposes) measures the integral of the squared error between the region $\mathcal{R}_i$ and its linear proxy P_i. Notice that we integrate the real $\mathcal{L}^2$ distance over the surface, not just the distance evaluated at the vertices such as

in [Garland et al. 2001]; in doing so, we make the optimization robust to irregular sampling rate of the input geometry (see Figure 4). As we will explain in Section 3.4, finding the $\mathcal{L}^2$-optimal proxy for a given region is fairly simple, since it only requires computing a barycenter and a covariance matrix.
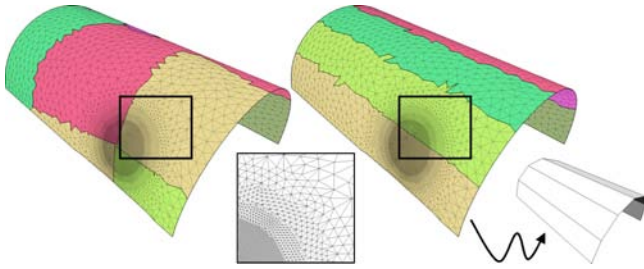

Figure 4: $\mathcal{L}^2$-optimized partition for a highly non-uniform input mesh (notice the disk-shaped region with refined triangles). The sampling irregularity severely distorts the partitioning if point-based covariance matrices are used (left), while our triangle-based covariance matrices (right) provide the expected polygonal approximation, capturing the true geometry.

As proven for elliptic areas in the asymptotic limit [Nadler 1986], an $\mathcal{L}^2$-optimal approximation of a surface will tend to create elements efficiently taking advantage of local anisotropy by being stretched in the minimum curvature direction with an aspect ratio of $\sqrt{|\kappa_{max}/\kappa_{min}|}$. This stretching along the minimum curvature direction makes very good use of the local shape of the object. However, in the hyperbolic case, there is no unique optimal shape and alignment. Since we are targeting a variational approach, this non-unique optimality is worrisome: a minimization algorithm can randomly jump around in the null space of the functional, resulting in undesired oscillations. To circumvent this issue, we look for a novel metric next.

2.3.2 Introducing $\mathcal{L}^{2,1}$ as a Shape Metric

The $\mathcal{L}^2$ metric tries to match geometry through approximation of the geometric position of the object in space. However, the normal field is fundamental in the way the visual system interprets the object's *shape*: normals govern lighting effects such as diffusion, specularity, as well as curvature lines and silhouettes; a smooth normal field defines a smooth shape, and normal discontinuities indicate features. Moreover, there is evidence that our visual perception is actually *more* sensitive to changes in normals rather than in changes in positions: this remarkable property has been used in compression for instance, where quantization noise can be better hidden in the low-frequency errors [Sorkine et al. 2003]. As already noted in [Garland et al. 2001],a metric based on the error in normal approximation may therefore be more appropriate than $\mathcal{L}^2$.

In addition to these considerations on visual perception, there is also strong evidence that correctly approximating the normal field is an altogether better approach. In the functional setting, Shewchuk [2002] advocates that one should focus on getting good bounds on the gradient interpolation error, as these are much more difficult to control: the functional interpolation errors can always be improved through refinement, whereas such a refinement may not improve the gradient interpolation quality. In fact, approximating a function well does not mean that its gradient will also get approximated [Fu 1993]: there are known examples (Schwarz's Chinese lantern for instance) of triangulated surfaces converging to a smooth surface for the Hausdorff metric, but with a surface area diverging, and a non-converging normal field. However, as hinted by the Poincaré-Wirtinger-Sobolev inequality, controlling the upper bound of the norm of the gradient interpolation error allows to also bound the norm of the interpolation error.

Given the cogent body of evidence in favor of a normal-based measure of distortion, we introduce a new *shape metric* that we

denote $\mathcal{L}^{2,1}$, as it is based on a $\mathcal{L}^2$ measure of the normal field:

$$\mathcal{L}^{2,1}(\mathcal{R}_i, P_i) = \iint\limits_{x \in \mathcal{R}_i} \|\mathbf{n}(x) - \mathbf{n}_i\|^2 dx. \tag{2}$$

We show in Appendix A that this metric is numerically superior to $\mathcal{L}^2$ in several ways:

◇ The anisotropy of the surface is better captured, since the asymptotic aspect ratio of an optimal element is in $|\kappa_{max}/\kappa_{min}|$, therefore largely superior to the asymptotic $\mathcal{L}^2$ behavior. This advantage is already confirmed at coarse scale, as shown in Figure 6. Moreover, we prove that there is a *unique* optimal shape and alignment in the limit for any (non-isotropic) surface type, be it parabolic, elliptic, or hyperbolic. The difference in results with the $\mathcal{L}^2$ metric is very noticeable (see Figures 5 and 16): although the two metrics have their own advantages, $\mathcal{L}^{2,1}$ consistently gives equal or better visual results according to our tests. Further results exhibiting the good behavior of this new metric can be found in Section 5.

◇ Finding the best normal proxy is as simple as averaging the normals over the associated region (see Section 3.4). We do not have to compute a covariance matrix, and thus, we save a significant amount of computations compared to $\mathcal{L}^2$ (or even compared to the normal-based metric used in [Garland et al. 2001]).

Finally, note that our asymptotical results are in agreement with the optimal case (super-convergence) of the gradient approximation mentioned in [Shewchuk 2002; D'Azevedo and Simpson 1991].
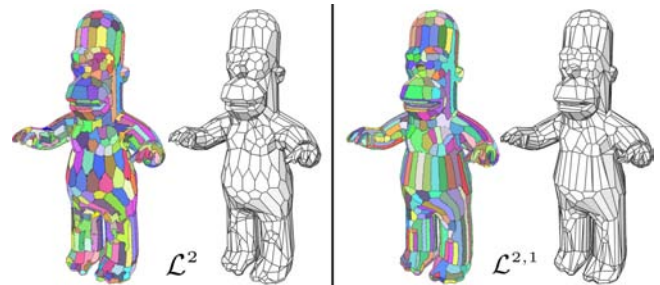

Figure 5: *Homer: This character illustrates the effect that an error metric can have on approximation. While $\mathcal{L}^2$ (left) and $\mathcal{L}^{2,1}$ (right) behave similarly on near-spherical regions such as the top of the head, the belly and mouth regions are very different in each case.*

2.4 Optimal Shape Proxies

We now have everything we need to define what we mean by an optimal partitioning of an arbitrary surface:

*Given an error metric E (either $\mathcal{L}^2$ or $\mathcal{L}^{2,1}$), a desired number k of proxies, and an input surface S, we call **optimal shape proxies** a set P of proxies P_i associated to the regions R_i of a partition $\mathcal{R}$ of S that minimizes the total distortion:*

$$E(\mathcal{R}, P) = \sum_{i=1..k} E(\mathcal{R}_i, P_i).$$

In other words, the set of proxies is optimal with respect to an error metric if it minimizes the total approximation error over the possible sets of proxies of same cardinality. Of course, in practice we cannot hope to find the global minimum in a reasonable time. However, we set up our shape approximation as a discrete, variational partitioning of the initial faces such that we can now apply simple and powerful discrete clustering algorithms that achieve very good and on some simple cases near-optimal results.

3 Optimizing Shape Proxies

Given an error metric E, a number k of proxies, and an input geometry S of arbitrary size and topology, we wish to efficiently find a partitioning $\mathcal{R}$ of S in k disjoint, connected regions and its respective set P of optimal proxies that minimizes (or nearly minimizes) $E(\mathcal{R}, P)$. Because in practice the input geometry is triangulated, we can consider this mesh as a discrete collection of faces: the problem is then cast into optimal discrete clustering, for which simple algorithms have been proven extremely efficient.
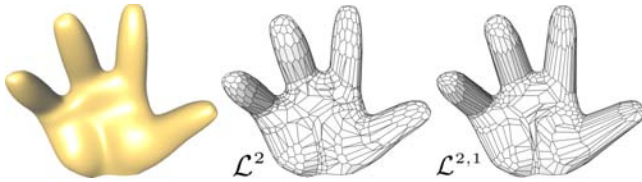
Figure 6: *Four-fingered hand (left): Polygonal shape approximation using $\mathcal{L}^2$ (center) and $\mathcal{L}^{2,1}$ (right), both for 200 proxies. Notice the greater anisotropy in the $\mathcal{L}^{2,1}$ case.*

3.1 Background on Lloyd's Clustering Algorithm

Clustering a set of discrete points involves dividing them into non-overlapping regions (or clusters), where points belonging to a region are closer by some measure of proximity to one another than to points in other clusters. Every region can be characterized by a single, "average" center, and the set of all k regions is called a k-partition. The Lloyd algorithm is a deterministic, fixed point iteration that provides such a partitioning [Lloyd 1982]. Conceptually, the idea is simple: after defining k random centers, all the data points are partitioned into k regions by assigning each point to its nearest center. Then, the algorithm updates the centers to be the barycenters (centroids) of their associated regions before starting a new partition with these new centers. This process is repeated until a stopping criterion is met. It can be proven that such an algorithm (sometimes referred to as *k-means clustering*) aims at minimizing a cost function E based on how tightly each region is packed and how well separated the different clusters are: the functional E defined by a set of N points $\{X_j\}$ and k centers $\{c_i\}$ is: $E = \sum_{i \in 1..k} \sum_{X_j \in \mathcal{R}_i} \|X_j - c_i\|^2$. For such a functional, Lloyd's algorithm always converges in a finite number of steps, since each step reduces the energy E: the partitioning stage minimizes E for a fixed set of centers c_i, while the fitting stage minimizes E for a fixed partition. Notice that the optimal fixed point is strongly linked to the notion of *centroidal Voronoi diagram* [Du et al. 1999] in the continuous case, for which centers are exactly the centroids of their associated Voronoi cell.

Because of its simplicity and ease of implementation, Lloyd's algorithm is widely used even for higher-order functionals as it manages to find a very good (though not guaranteed to be global) minima. Moreover, Lloyd's technique also suffers from fewer oscillations than other physically-based particle-spreading techniques, explaining why this algorithm and its variants are used in many different contexts, including graphics [Hausner 2001; Ohtake et al. 2003a; Katz and Tal 2003; Surazhsky et al. 2003; Sander et al. 2003]. Therefore, if we are able to adapt Lloyd's algorithm and use it as a minimizing tool to drive the distortion error down instead of optimizing compactness, we should be able to quickly produce a low-distortion partitioning and a set of geometric proxies that closely approximate any input geometry.

3.2 Our Algorithm At a Glance

Lloyd's method hinges on the two phases of *partitioning* and *fitting*, repeated alternately to drive the total energy down. Paralleling this process, we present a simple and efficient extension of Lloyd's algorithm to variational, geometry-driven partitioning that includes the following steps:

◇ **Geometry Partitioning** In order to create a partition of an arbitrary non-flat triangulation, we use an error-minimizing region growing algorithm that will segment the object in non-overlapping connected regions.

◇ **Proxy Fitting** Once a partition is found, we compute for each region an optimal local representative, the proxy (see Section 2.2). These geometric proxies, that minimize the distortion error for a given partition, are an extension of the centroids in the original Lloyd's algorithm.

Nomenclature We now describe the algorithm, and will refer to the input surface as S, its current partition as $\mathcal{R}$, its k regions as $\mathcal{R}_i$, and their current respective proxy as $P_i = (X_i, \mathbf{N}_i)$. The distortion

error will be referred to as E, and can represent either the $\mathcal{L}^2$- or $\mathcal{L}^{2,1}$-based error defined in Section 2.3.
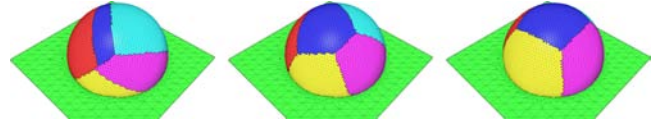

Figure 7: *Half-sphere on plane: (left) random initialization of a 6-partitioning; (center) after one iteration of our optimization, the regions self-organize; (right) after 5 iterations, the regions settle.*

3.3 Geometry Partitioning

Knowing a current set of proxies P, we wish to update the partition $\mathcal{R}$ while trying to minimize the distortion error $E(\mathcal{R}, P)$ in the process. We perform this k-proxy clustering as follows.

Initial Seeding For each region of the previous partition, we first find the triangle T_i of S that is the *most similar* to its associated proxy. This is easily achieved by visiting each current partition region $\mathcal{R}_i$, and by going once through all its triangles to find the one with the smallest distortion error $E(T_i, P_i)$ (computed using Equations 3 or 4 in Appendix B). In order to bootstrap the algorithm, the very first geometry partitioning picks k triangles at random on the object, and each of these triangles are assigned a proxy defined as the triangle's barycenter and its normal. Once these k triangles are found, we assign them to their respective proxies, while we remove the proxy assignment of all the other triangles in order to start a new partition from scratch.

Distortion-minimizing Flooding Once these seed triangles are found, we wish to "grow" a region out from them, in order to find a new, better partition. Just like in Lloyd's algorithm, we wish to cluster together only faces that are "close" (i.e., with a low error distortion) to the proxy. Therefore, for each seed triangle T_i, we insert its three adjacent triangles T_j in a global priority queue, with *a priority equal to their respective distortion error* $E(T_j, P_i)$, and we add an additional tag indicating the label i of the proxy they are being tested against (a triangle can therefore appear up to three times in the queue, with different tags and priorities). The region-growing process is then performed by repeatedly popping triangles with least distortion until the priority queue is empty. For each triangle popped out from the queue, we check its proxy assignment: if it has already been assigned to a proxy, we do nothing and go to the next triangle in the queue; otherwise, we assign it to region of the proxy indicated by the tag, and push the (up to two) unlabeled incident triangles into the queue along with the same tag. When the priority queue has been emptied, each triangle has been assigned to a proxy: therefore we have a new partition. Notice that our growing process ensures connected and non-overlapping regions as required, and that this flooding procedure is extremely rapid ($Nlog(N)$ complexity). Note also that this partitioning method is quite different from previous clustering techniques: we use an integrated distortion error instead of a term based on local geometric criteria such as in [Katz and Tal 2003].

3.4 Proxy Fitting

Once we have found a new partition $\mathcal{R}$ over the surface S, we now wish to update the respective proxies $P_i = (X_i, \mathbf{N}_i)$ in order for them to be the best representatives of their associated newly-updated region $\mathcal{R}_i$ (iterative partitioning is exemplified by Figure 7). Note that, for the given partition $\mathcal{R}$, this procedure will find the set of proxies that *minimizes* the total distortion error $E(\mathcal{R}, P)$. This minimization is easily done using the equations given in Appendix B. For the $\mathcal{L}^2$ metric, X_i is simply the barycenter of the region $\mathcal{R}_i$ while $\mathbf{N}_i$ is the direction (the sign does not matter) indicated by the eigenvector associated with the smallest eigenvalue of the covariance matrix of the region—i.e., the proxy is the least-square fitting plane traditionally found with Principal Component

Analysis. For the $\mathcal{L}^{2,1}$ metric, the best fit procedure is even simpler. The proxy normal is the area-weighted average of the triangles' normals of the region; the point X_i, although irrelevant to the $\mathcal{L}^{2,1}$ minimization, is chosen to be the barycenter of the region for display and remeshing purposes.

3.5 Improvements and Details

Several enhancements over the basic algorithm we just described are easily implemented, dramatically improving the efficiency and the controllability of our basic technique.

Choosing the Number of Proxies In order to make the variational partitioning more flexible, we have added a number of possible interactions that the user can utilize if necessary. We let the user not only pick the desired number of proxies at any time, but we also allow interactive, incremental insertion and deletion of proxies. The insertion is done by finding the current region with maximum total distortion, and within it, we pick the triangle with worst distortion error as the initial seed for the next flooding (this is, in spirit, a farthest-point sampling heuristic); this will add a new region and proxy in the most needed part of the object (see example Figure 8). Similarly, we allow the incremental deletion of a region. We select the region to be deleted as follows: for each pair (or even a random set of pairs, if efficiency is an issue) of adjacent regions, we *simulate* a merging of the two regions and compute the resulting distortion with the new best fitting proxy; the pair of regions achieving the smallest distortion when merged are then replaced with a single one, deleting a proxy in effect. With these options, it becomes very easy to obtain a good partition in a matter of a few seconds.

Region Teleportation It is no surprise that, in the course of finding a lower distortion, the algorithm can find itself stuck in a local minimum. Typically, this can happen on a very flat region: if a region happens to be encircled by other regions with similar proxies, it may be locally stuck in this minimum configuration as this position prevents it to roam on the surface and find more efficient positions. Yet it is clearly suboptimal to leave this region as is. We have therefore implemented a *region teleportation* procedure to give a region the chance to tunnel out of a local minima, similar in spirit to [Bossen and Heckbert 1996; Lindstrom and Turk 2000]. At regular intervals during the clustering process, we simply force a region deletion as described above, immediately followed by a region insertion: the effect of this simple two-step operation is to remove a region stuck in a local minimum, and "teleport" it where it is most needed. In practice, it is better to first test if the teleportation is worth it: we use a heuristic that tests whether the error added by a (simulated) deletion is smaller than half of the error of the worst region. If this test fails, no teleportation is necessary. These local operations drastically improve the final quality of our partition. This good behavior is most likely due to the similarity of our strategy with a variant of the Lloyd's algorithm [Kanungo et al. 2002], known for having tight bounds on the optimality of the results.

Farthest-point Initialization A good initial seeding of the regions goes a long way in getting a good final minimum fast. Although we have found in all our tests that the naïve initializa-
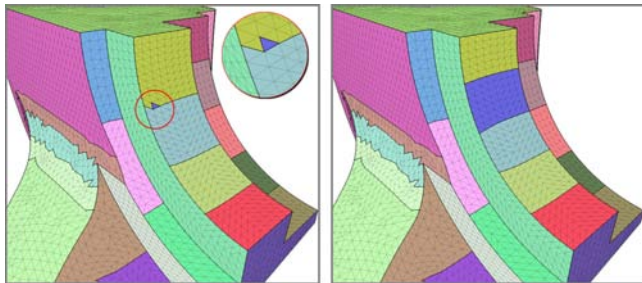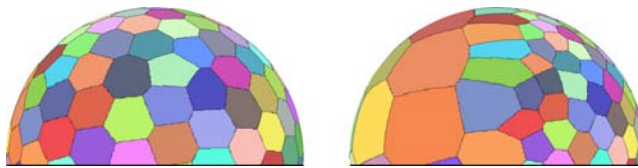

Figure 9: *Partitioning without (left) and with (right) user-designed area-weighting: the left-hand side of the sphere has been painted to force a much coarser discretization.*

tion with randomly chosen seeds as described above is just fine in practice, we have tailored a specific initialization for non-organic shapes: meshes of mechanical parts for instance can be dealt with in a particularly efficient fashion through a simple alternative initialization. We simply add one region at a time, perform a partitioning, then proceed by adding a new region at the triangle of maximum error with respect to the region it belongs to (again, this is reminiscent of the usual farthest point strategy); no fitting between two floodings is necessary, as the proxy values are directly picked from the seed triangles' barycenters and normals. This initialization works very well on non-smooth objects, but does not have a significant pay-off on other objects, particularly if a lot of noise is present.

Convergence Although we cannot guarantee global convergence of our variational approach, a very good behavior is observed in practice: the proxies start settling down after a few iterations, or oscillate around extremely similar distortion errors. Note that there is no need to wait for the regions to settle: in a matter of two to twenty iterations, results are already visually pleasing (see the error as a function of the number of iterations in Figure 10). Convergence is, however, *guaranteed* for convex objects for the $\mathcal{L}^{2,1}$ norm, since it amounts to the well-known k-means (area-weighted) clustering of the discrete normals on the image of the Gauss map. Furthermore, convergence would also be guaranteed for arbitrary surfaces *if* one was to relax the connectedness of the regions in the partition; however, having proxies that correspond to disconnected patches of surface is less geometrically relevant.
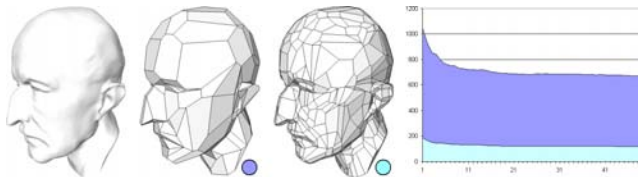
Figure 10: *Max Planck: For the two optimized approximations (130 and 300 proxies resp.), we show the associated curves of the $\mathcal{L}^{2,1}$-distortion error as a function of the number of Lloyd's iterations; as expected, a few iterations suffice to reach a much reduced distortion error.*

Tailoring Refinements Another valuable interactive tool is to allow the user to paint regions that require more (resp., less) details despite their lesser (resp., larger) geometric importance, as in [Kho and Garland 2003]. For instance, when approximating a face with very few proxies, the eyes may not be very apparent since they are not a significant geometric feature: by artificially scaling up or down the area of the faces painted, the error metric will weight these regions differently with no change to the algorithm. The effect of such a forced refinement can be seen on Figure 9.

Smoothing the Normal Field If extreme simplification is desired, it may also be relevant to artificially smooth the vector field for the $\mathcal{L}^{2,1}$ metric, i.e., the normal associated to each face: this helps embellish the final result at a low cost—but is by no means necessary. In order not to introduce a bias in the shape, we use a local non-iterative anisotropic smoothing, easily implemented as it reuses a part of the partitioning procedure. For each face, we initialize a proxy with the triangle's barycenter and normal, then launch a flooding algorithm similar in spirit to the one explained in Section 3.3; but we perform the flooding only for this seed, with additional thresholds on both the maximum deviation angle with

Figure 8: *Fandisk: When the user interactively adds a proxy, a seed triangle is placed in the worst-approximated region (left), and the next iteration allows a new region to quickly grow (right).*

the seed's normal (typically, ten degrees) and the maximum number of triangles to collect. A local connected region of most similar geometry is therefore found and spread anisotropically around the initial seed. We then artificially change the normal of the seed triangle to the area-weighted average normal, which is a smoothed version of the initial normal. This is very similar in spirit to WENO techniques [Osher and Fedkiw 2002], but for geometry: such an anisotropic averaging preserves normal discontinuities perfectly.

4 Application to Meshing

Now that we found a nearly-optimal partitioning, its proxies capture the essence of the input geometry. They form a perfect draft for remeshing: if proxies are thought of as local faces (local tangent planes), we already have the blueprint of a mesh. Additionally, the adjacency graph of the partition defines the connectivity of a mesh as well. Therefore, if a mesh output is needed, a set of vertices and edges can easily be added to obtain a mesh that closely approximates the original object, in the spirit of [Kalvin and Taylor 1996].

Anchor Vertices Since the proxies can be seen as approximate faces of the final mesh, we must put vertices at the intersection of the proxies. Thus, we create an *anchor vertex* at every original vertex where three or more regions meet. In order to account for every region, we then check whether each region boundary has at least three anchor vertices attached to it; if not, we simply add anchor vertices accordingly as it will guarantee the presence of at least one polygon per region. The spatial position of these anchor vertices is determined as follows: for each neighboring proxy of an anchor, we compute the projection of the associated vertex from which the anchor was created onto the proxy (i.e., its ideal position for this proxy); we then simply average these projections.

Edge Extraction Since every anchor vertex has a pointer to its originating vertex on the input surface, it is easy to now add edges between the anchor vertices by simply visiting each region boundary. These so-constructed edges may approximate the region's boundary rather coarsely, inducing geometric inaccuracies later on during the triangulation of the approximant mesh faces. Thus, we use a simple recursive chord-length subdividing algorithm. If **a** and **b** are two anchor vertices linked by an edge separating proxy P_i and P_j, we visit all the original vertices of the associated boundary arc, find the largest distance d from these vertices to the edge $(\mathbf{a}, \mathbf{b})$, and add an anchor vertex there. However, if the angle between proxy P_i and P_j is rather small, even a coarse approximation of the boundary will do: it does not add geometric information on the shape. On the contrary, when there is a large angle in between the two proxies, a more accurate discretization is desirable. We thus proceed as follows: if the criterion $d \cdot \sin(\mathbf{N}_i, \mathbf{N}_j)/\|(\mathbf{a}, \mathbf{b})\|$ is larger than a given threshold, we recursively add anchor vertices and edges to better approximate the boundary until the criterion is met.
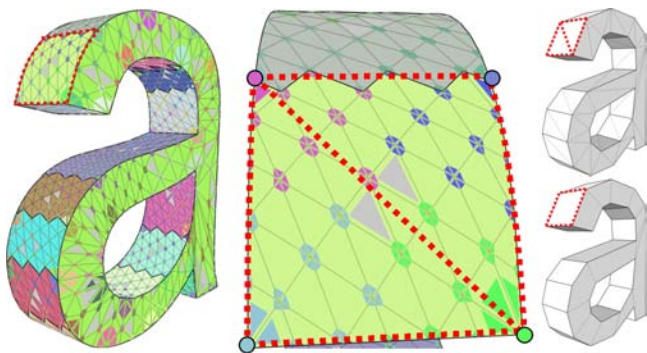

Figure 11: *Discrete Constrained Delaunay Triangulation: Flooding the mesh from the anchor vertices (solid dots) creates triangles (light grey, left and center) whose three corners have different colors. Each of these triangles generates a meta-triangle during meshing. A final edge-removal pass provides a $\mathcal{L}^{2,1}$-polygonal model (right, bottom).*

Triangulation With the anchor vertices and edges defined, we already have a polygonal mesh. However, when the number of proxies is fairly small, the polygons have no guarantee of being almost flat or convex. Thus, we need to triangulate this initial graph in order to be able to truly call it a mesh. This is done through a "discrete" Constrained Delaunay triangulation (CDT) to make the process robust to any sort of extreme approximation: indeed, we will create Delaunay-like triangles within each region, while constraining the existing anchor-based edges to be part of the final triangulation. To achieve this pseudo-CDT, we resort once again to a flooding algorithm, very similar to the multi-source Djisktra's shortest path algorithm with an edge weight equal to its length, and for which the sources are the anchor vertices: it will, in spirit, construct discrete Voronoi cells from which the triangulation is easily extracted. In a first step, we only flood the boundary of a region so that every vertex on it is colored depending on the closest anchor vertex: this will enforce the *constrained* triangulation by forcing the boundary to be in it. We then start a flooding of the interior of the region, coloring the vertices also according to their closest anchor vertex.

The extraction of the final triangles is now straightforward. We look at every triangle of the input mesh whose three vertices have distinct colors: each of them corresponds to a triangle in the final triangulation, emanating from the anchor vertices indicated by the three colors. A final pass can be done on these newly-added edges: swapping some of them may locally improve the compacity of the triangles in the rare occasions when the discrete approximation of the Delaunay triangulation is imperfect. A simple test followed by a swap if it shows relevant is therefore done on each internal edges. A summary of this process is depicted in Figure 11.
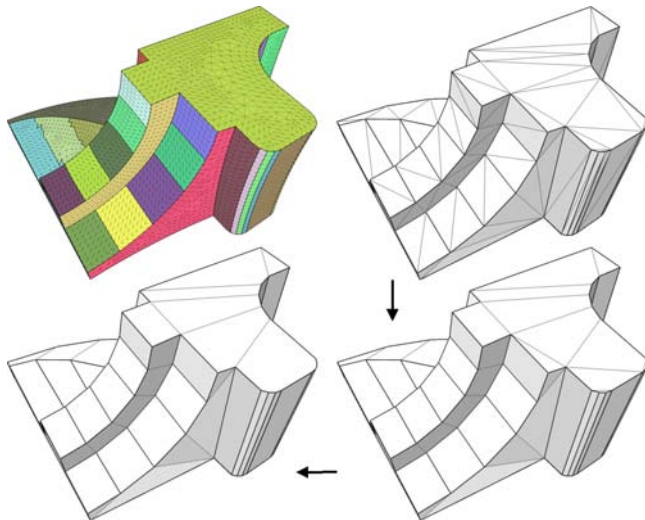

Figure 12: *Generation of a polygonal model: triangle mesh obtained via CDT (top-right); creation of well-shaped quads (bottom-right), then polygons (bottom-left) by iterative edge removal.*

Polygons Due to the very nature of our partitioning, the newly-triangulated mesh may have useless edges: since every region should result in nearly flat geometry, there are sometimes no reason to have edges within these polygonal faces, except to avoid concave polygons, detrimental to the graphics pipeline. We therefore perform a final pass over the triangulation to remove the edges that do not contribute to the shape. First, we try to make as many nicely-shaped quads as possible: we look at edges that can be safely removed (i.e., that produce no normal flips); we sort them by a score linked to the well-shapedness of every candidate quad [Pebay 2002]; finally, we go down the list and remove the edges creating the best quads first, until the list is empty. Second, we perform a second pass in order to create larger polygons if possible; for each

remaining edge, we first test if removing it would not induce a concavity in the mesh, and if it would not change the normal by more than a given threshold (typically, 20 degrees); for the eligible edges, we compute a score based on the area of the potential polygon; we finally go over the edges in order of decreasing score, to create the largest polygons first. A close-up on the remeshed Fandisk model in Figure 12 exhibits the type of polygonalization we finally obtain.

The meshing of the proxies is fast, and never took more than one second on all the meshes shown in this paper. Other meshing techniques could be used, using local parameterization of the regions for instance, but our discrete parameterization-free approach has proved satisfactory.
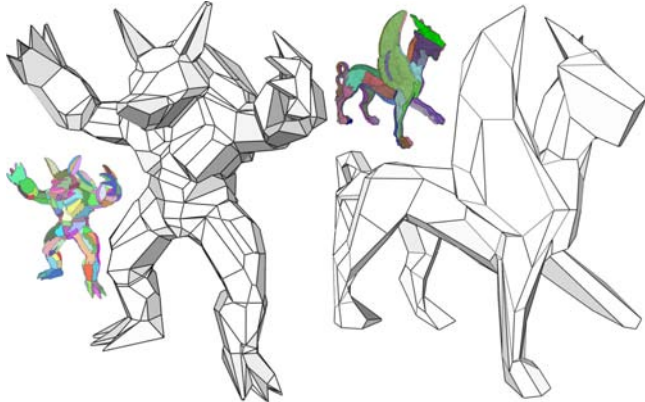
Figure 13: $\mathcal{L}^{2,1}$-Approximation: our versatile framework optimizes the efficiency of geometric representations. (Left) Armadillo (300 proxies, initially 346K triangles), (Right) Feline (50 proxies, initially 100K triangles).

5 Results and Discussion

We have tested our variational partitioning technique extensively, on geometry varying from organic shapes to mechanical parts, and from toy examples to large, noisy scanned meshes (see Figures 2 and 13). Although the $\mathcal{L}^2$ metric provides good approximations in general, the $\mathcal{L}^{2,1}$ results are in agreement with what we would have intuitively *expected* from a good segmentation of geometry, and often capture more details. In both cases, our variational approach allows the symmetries to be quickly found, the anisotropy to be automatically detected and exploited, and the regions to line up with the features. Finally, we insist on the fact that, while the "canons of beauty" for graphics meshes usually involve nicely-shaped triangles with a smooth sampling gradation, our concise, optimized meshes sharply depart from the norm; but they gain in efficiency by respecting features and symmetries (see Figure 17).

The application of this new type of approach are numerous. Such an automatic segmentation/polygonalization of redundant datasets can be, for instance, directly used for reverse engineering [Várady et al. 1997; Botsch and Kobbelt 2001] and scanned meshes. The proxy optimization also seem to offer interesting alternatives to existing methods, such as the greedy selection of representative planes for billboard clouds [Décoret et al. 2003]. Moreover, using anisotropic ellipses (computed from the eigenvalues of the regions' covariant matrices) could further optimize surface splatting. The WENO-type face clustering has been surprisingly effective at smoothing geometry, and could be explored further. Lastly, various optimizations are likely to further improve the efficiency of the minimization procedure, such as lazy evaluations for instance.

Limitations Being based on iterative optimization, our technique cannot compete with greedy methods such as [Garland and Heckbert 1998] in terms of computational time: improving mesh efficiency can be three to twenty times slower when compared with simplex removal. Although it remains fairly interactive (between 3s for Figure 1, to 10 minutes for Figure 17), it should be reserved for offline computations: greedy approaches perform really well given
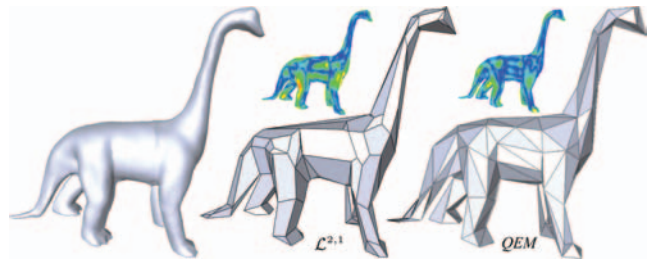
Figure 14: (Left) Dinosaur model; (Center) $\mathcal{L}^{2,1}$-approximation; (Right) Results for QEM [Garland and Heckbert 1998] with same number of edges. Note that our approach reproduces the "highlights" (see neck), with a symmetric Hausdorff error 18% better (as measured by [Aspert et al. 2002]).

the processing time (see Figures 14 and 15). In fact, our tests show that Qslim [Garland and Heckbert 1998] often outperforms our $\mathcal{L}^2$ results if an $\mathcal{L}^2$-optimized mesh with a given number of triangle is sought, as our actual meshing procedure is not error-driven and is intrinsically designed for *polygonal* outputs. Similarly, we are only handling piecewise-linear 2-manifolds, although an extension to point clouds with local approximation of connectivity is feasible. At the algorithmic level, our meshing technique can still be improved. For instance, we could allow the final mesh to be non-manifold, resulting in even more concise meshes for extreme simplifications. Indeed, in the case of a fin-like feature, we currently create a pyramid-type fin during the discretization even if there is only one region for the whole fin. Lastly, the Voronoi-like cells obtained on spherical regions (see top of Homer's head on Figure 5) indicate that we could locally extract a *dual* mesh, leading to nicely-shaped triangles in round regions and still elongated elements in anisotropic regions.
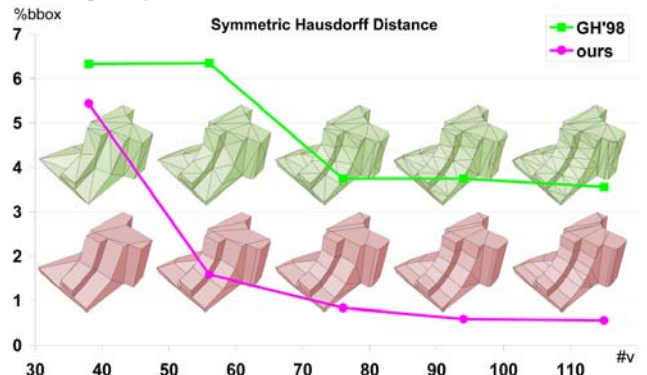
Figure 15: Comparison of the Hausdorff error for QEM [Garland and Heckbert 1998] and for our $\mathcal{L}^{2,1}$ technique, for equal number of vertices (a comparison using equal number of edges leads to an extremely similar curve).

6 Conclusions

By breaking away from the traditional approximation paradigm that consists in directly optimizing a piecewise-linear approximant of an original surface, we have proposed a simple and efficient variational shape approximation approach. Through mutual and repeated error-driven optimizations of a partition and a set of local proxies, our method provides concise geometric representations either in the form of local best-fit geometric representatives or in the form of a polygonal mesh. We have also presented a novel shape metric, allowing the capture of more subtle details than the traditional $\mathcal{L}^2$ metric.

The versatility of our framework paves the way to a multitude of future work. We plan to try a Sobolev metric (H^1) next, since it would simply consist of the sum of the $\mathcal{L}^2$ and $\mathcal{L}^{2,1}$ energies and would only require a low-order polynomial root solver to compute the best fit. Other metrics, incorporating color and texture information, can also be easily tried. Variational motion approximation, i.e., a 4D (3D + time) approximation using a space-time metric,
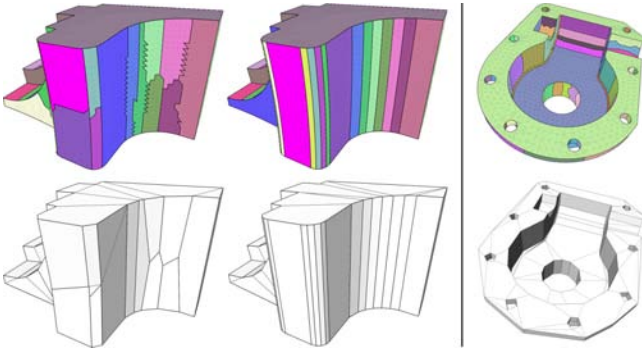
Figure 16: *Mechanical parts: 50 proxies on the Fandisk using $\mathcal{L}^2$ (left) vs $\mathcal{L}^{2,1}$ (center). Notice that the two metrics are adopting two different approximation strategies. Approximating the high-genus casting model with 172 $\mathcal{L}^{2,1}$-proxies (right).*

could also be used to simplify large scientific simulation sequences, making the best of both spatial *and* temporal components. Geometry compression and higher-order proxies are two other obvious avenues to explore. Finally, we wish to study the notion of shape complexity (see [King and Rossignac 1999]) and how it relates to the choice of a metric, as it could help making a few steps towards a sampling theory for shapes.

Acknowledgments We wish to thank Peter Schröder, Zoë Wood, and Eitan Grinspun for early feedback, and Laura Lee-Chin for help on the video. The Lion-vase model (Figure 2) is courtesy of SensAble Technologies, Inc.; the Dinosaur model is courtesy of Michael Garland, and the David model is courtesy of Marc Levoy and the Stanford graphics group.

A Asymptotic Behavior of the $\mathcal{L}^{2,1}$ Metric

(A.1) (A.2) (A.3)
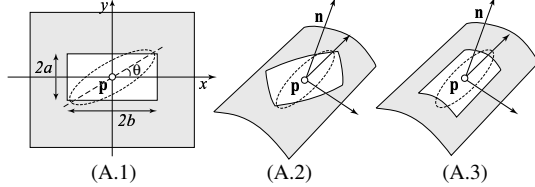
In this section, we prove that asymptotically, an element is most efficient if: (i) the eigenvectors of its inertia matrix are along the principal curvature directions, and (ii) the eigenvalues' ratio is equal to the principal curvatures' ratio. Although this property holds for arbitrary elements, we only consider rectangular elements for the sake of simplicity. Consider an arbitrary surface $\mathcal{S}$. Let $\mathcal{R}$ be a small rectangle of dimension $2a \times 2b = |\mathcal{R}|$, and such like $\mathcal{R}$ is tangent in its center to the surface $\mathcal{S}$ at a point $\mathbf{p}$. The normal $\mathbf{n_p}$ at $\mathbf{p}$ is therefore also normal to $\mathcal{R}$. The only parameters that are not determined are a, b, and the angle θ between the minimum curvature direction and the side of $\mathcal{R}$ (see Figures A.1 and A.2). Then we have [Gray 1998]:

$$\mathbf{n}(x,y) \simeq \mathbf{n_p} + \mathbf{H}\binom{x}{y}$$

$\mathbf{H}$ is the (symmetrical) Hessian matrix. Since $\mathbf{p}$ is the barycenter of R, the average normal $\mathbf{N}$ is $\mathbf{n_p}$. Therefore, the $\mathcal{L}^{2,1}$-based error E is:

$$E = \iint_{\mathcal{R}} \|\mathbf{n}(x,y) - \mathbf{n_p}\|^2 \mathrm{d}x\,\mathrm{d}y = \iint_{\mathcal{R}} (x\ y)\ \mathbf{H}^t\ \mathbf{H}\ \binom{x}{y}\ \mathrm{d}x\,\mathrm{d}y$$

$$= \iint_{\mathcal{R}} (x\ y)\ \mathbf{Q}\ \binom{x}{y}\ \mathrm{d}x\,\mathrm{d}y = \int_{-a}^{+a}\int_{-b}^{+b}(x\ y)\begin{pmatrix} Q_{11} & Q_{12} \\ Q_{12} & Q_{22} \end{pmatrix}\binom{x}{y}\ \mathrm{d}x\,\mathrm{d}y$$

$$= \frac{4ab}{3}(Q_{11}\ a^2 + Q_{22}\ b^2)$$

Notice that $\mathbf{Q} = \mathbf{H}^t\mathbf{H} = \mathbf{H}^2$ is by definition always symmetric positive, even if $\mathbf{H}$ is not positive (like in hyperbolic regions). We now define the efficiency $f = E/|\mathcal{R}|$ as the ratio of error covered by area unit [Simpson 1994]. Obviously, we wish f to be minimum. In our case, we can rewrite:

$$f = \frac{1}{3}(Q_{11}\ a^2 + Q_{22}\ b^2)$$

Now if we try to optimize, using a Lagrange multiplier λ, the efficiency as a function of a and b under the constraint that the area ab is constant, we get the following linear system:

$$\frac{2}{3}\binom{Q_{11}\ a}{Q_{22}\ b} = \lambda\binom{b}{a}$$

We then find that the optimal dimensions of $\mathcal{R}$ is: $a = \frac{\mu}{\sqrt{|Q_{11}|}}$, $b = \frac{\mu}{\sqrt{|Q_{22}|}}$, μ being a constant.

For this optimal rectangle, we have: $f = \frac{2|\mathcal{R}|}{12}\sqrt{Q_{11}Q_{22}}$. However, notice that $det\ \mathbf{Q} = Q_{11}Q_{22} - Q_{12}^2 \geq 0$ for any θ. The efficiency f is therefore best when $Q_{12} = 0$: $\mathbf{Q}$ is then diagonal, which means that $\mathbf{H}$ is also diagonal and thus, $\theta = 0$. As a consequence, the optimal quadrangle is aligned with the principal curvature (since $\theta = 0$ - see Figure A.3); and has a side ratio of $a/b = \sqrt{\frac{Q_{11}}{Q_{22}}} = |\frac{H_{11}}{H_{22}}|$, i.e., of ratio $|\kappa_2/\kappa_1|$ (since $\mathbf{H}$ is diagonal in the optimal configuration). This result is particularly strong as it is valid in all regions, be them elliptic or hyperbolic.

B Formulas for Error Metrics

In this last section, we provide the reader with the equations needed to compute the distortion errors and the best-fitting (error minimizing) proxies.

Formulas for $\mathcal{L}^2$ Let $T_i = (v_1, v_2, v_3)$ be a triangle of area $|T_i|$, and let P_i be a proxy $(X_i, \mathbf{N}_i)$—here considered as a plane passing through X_i, of normal $\mathbf{N}_i$. Let d_1, d_2, and d_3 the orthogonal distances of the vertices v_1, v_2, and v_3 to the plane P_i. The $\mathcal{L}^2$ distance between T_i and P_i is:

$$\mathcal{L}^2(T_i, P_i) = \frac{1}{6}(d_1^2 + d_2^2 + d_3^2 + d_1 d_2 + d_1 d_3 + d_2 d_3)|T_i|. \quad (3)$$

As for the minimization, we find the best-fit X_i of the region $\mathcal{R}_i$ simply using: $X_i = \left(\sum_{T_i \in \mathcal{R}_i} g_i |T_i|\right) / \left(\sum_{T_i \in \mathcal{R}_i} |T_i|\right)$, where g_i stands for the barycenter of triangle T_i, i.e. $g_i = (v_1 + v_2 + v_3)/3$.

The best-fit normal $\mathbf{N}_i$ of the region $\mathcal{R}_i$ is equal to the eigenvector corresponding to the smallest eigenvalue of the following matrix:

$$\sum_{T_i \in \mathcal{R}_i}\left(\frac{2|T_i|}{72}M_i\begin{pmatrix} 10 & 7 & 0 \\ 7 & 10 & 0 \\ 0 & 0 & 0 \end{pmatrix}M_i^t + |T_i|\ g_i\ g_i{}^t\right) - \left[\sum_i |T_i|\right]X_i\ X_i^t$$

where $M_i = \begin{pmatrix} v_2 - v_1 \\ v_3 - v_1 \\ 0\ 0\ 0 \end{pmatrix}$.

Formulas for $\mathcal{L}^{2,1}$ For a triangle T_i of area $|T_i|$, of normal $\mathbf{n}_i$, and of associated proxy $P_i = (X_i, \mathbf{N}_i)$, the $\mathcal{L}^{2,1}$ error is computed as follows:

$$\mathcal{L}^{2,1}(T_i, \mathcal{P}) = \|\mathbf{n}_i - \mathbf{N}_i\|^2 |T_i|. \quad (4)$$

Now for region $\mathcal{R}_i$, the optimal proxy normal $\mathbf{N}_i$ is simply equal to the vector: $\sum_{T_i \in \mathcal{R}_i} |T_i|\mathbf{n}_i$, after normalization to make it unit.

References

AGARWAL, P., AND SURI, S. 1998. Surface Approximation and Geometric Partitions. *SIAM J. Comput. 19*, 1016–1035.

ALLIEZ, P., COHEN-STEINER, D., DEVILLERS, O., LÉVY, B., AND DESBRUN, M. 2003. Anisotropic Polygonal Remeshing. *ACM Trans. on Graphics 22*, 3, 485–493.

AMENTA, N., AND BERN, M. 1999. Surface Reconstruction by Voronoi Filtering. *GEOMETRY: Discrete and Computational Geometry 22*.

ASPERT, N., SANTA-CRUZ, D., AND EBRAHIMI, T. 2002. MESH: Measuring Errors between Surfaces using the Hausdorff Distance. In *IEEE Multimedia*, 705–708.

BALMELLI, L., VETTERLI, M., AND LIEBLING, T. M. 2002. Mesh Optimization Using Global Error with Application to Geometry Simplification. *Graphical Models 64*, 3-4 (May), 230–257.

BOISSONNAT, J. D., AND OUDOT, S. 2003. Provably good surface sampling and approximation. In *Proc. of Symp. on Geo. Processing*, 9–18.

BOROUCHAKI, H., AND FREY, P. 1998. Adaptive Triangular-Quadrilateral Mesh Generation. *Intl. J. Numer. Methods Eng. 41*, 915–934.

BOSSEN, F., AND HECKBERT, P. 1996. A Pliant Method for Anisotropic Mesh Generation. In *5th Intl. Meshing Roundtable*, 63–76.

BOTSCH, M., AND KOBBELT, L. 2001. Resampling Feature and Blend Regions in Polygonal Meshes for Surface Anti-Aliasing. In *Eurographics*, 402–410.

COHEN, A., DAHMEN, W., DAUBECHIES, I., AND DEVORE, R. 2001. Tree Approximation and Optimal Encoding. *Appl. Comput. Harmon. Anal. 11*, 2, 192–226.

D'AZEVEDO, E., AND SIMPSON, B. 1991. On Optimal Triangular Meshes for Minimizing the Gradient Error. *Numer. Math. 59*, 321–348.

D'AZEVEDO, E. F. 2000. Are Bilinear Quadrilaterals Better Than Linear Triangles? *SIAM Journal on Scientific Computing 22(1)*, 198–217.

DÉCORET, X., DURAND, F., SILLION, F., AND DORSEY, J. 2003. Billboard Clouds for Extreme Model Simplification. *ACM Trans. on Graphics 22*, 3, 689–696.

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

FU, J. H. G. 1993. Convergence of curvatures in secant approximations. *Journal of Differential Geometry 37*, 177–190.

GARLAND, M., AND HECKBERT, P. 1998. Simplifying Surfaces with Color and Texture using Quadric Error Metrics. In *IEEE Visualization Proceedings*, 263–269.

GARLAND, M., WILLMOTT, A., AND HECKBERT, P. 2001. Hierarchical Face Clustering on Polygonal Surfaces. In *ACM Symp. on Interactive 3D Graphics*, 49–58.

GIRSHICK, A., INTERRANTE, V., HAKER, S., AND LEMOINE, T. 2000. Line Direction Matters: Argument for the Use of Principal Directions in 3D Line Drawings. In *Int. Symp. on Non-Photoreal. Anim. and Rend.*, 43–52.

GRAY, A., Ed. 1998. *Modern Differential Geometry of Curves and Surfaces*. Second edition. CRC Press.

GRINSPUN, E., AND SCHRÖDER, P. 2001. Normal Bounds for Subdivision-Surface Interference Detection. In *IEEE Visualization '01*, 333–340.

GUSKOV, I., VIDIMCE, K., SWELDENS, W., AND SCHRÖDER, P. 2000. Normal Meshes. In *Proceedings of ACM SIGGRAPH*, 95–102.

HAUSNER, A. 2001. Simulating Decorative Mosaics. In *Proceedings of ACM SIGGRAPH*, 573–578.

HECKBERT, P., AND GARLAND, M. 1999. Optimal Triangulation and Quadric-Based Surface Simplification. *Journal of Comp. Geo.: Theory and Applications 14(1-3)*, 49–65.

HECKEL, B., UVA, A. E., HAMANN, B., AND JOY, K. I. 2001. Surface Reconstruction Using Adaptive Clustering Methods. In *Geometric Modelling: Dagstuhl 1999*, vol. 14, 199–218.

HERTZMANN, A., AND ZORIN, D. 2000. Illustrating Smooth Surfaces. In *ACM SIGGRAPH Proceedings*, 517–526.

HOPPE, H., DEROSE, T., DUCHAMP, T., MCDONALD, J., AND STUETZLE, W. 1993. Mesh Optimization. In *ACM SIGGRAPH Proceedings*, 19–26.

HOPPE, H. 1996. Progressive Meshes. In *ACM SIGGRAPH Proceedings*, 99–108.

INOUE, K., ITOH, T., YAMADA, A., FURUHATA, T., AND SHIMADA, K. 1999. Clustering Large Number Of Faces For 2-Dimensional Mesh Generation. In *8th Int. Meshing Roundtable*, 281–292.

INTERRANTE, V., FUCHS, H., AND PIZER, S. 1996. Illustrating Transparent Surfaces with Curvature-Directed Strokes. In *IEEE Visualizaton Proceedings*, 211–218.

KALVIN, A. D., AND TAYLOR, R. H. 1996. Superfaces: Polygonal mesh simplification with bounded error. *IEEE Comp. Graphics and App. 16*, 3 (May), 64–77.

KANUNGO, T., MOUNT, D. M., NETANYAHU, N. S., PIATKO, C. D., SILVERMAN, R., AND WU, A. Y. 2002. A local search approximation algorithm for k-means clustering. In *Proceedings of the Symp. on Computational Geometry*, 10–18.

KATZ, S., AND TAL, A. 2003. Hierarchical Mesh Decomposition Using Fuzzy Clustering and Cuts. *ACM Trans. on Graphics 22*, 3, 954–961.

KHO, Y., AND GARLAND, M. 2003. User-Guided Simplification. In *Proceedings of ACM Symp. on I3D*, 123–126.

KING, D., AND ROSSIGNAC, J. 1999. Optimal Bit Allocation in Compressed 3D Models. *Comput. Geo. Theory & Appl. 14*, 1-3, 99–118.

KLEIN, R., LIEBICH, G., AND STRASSER, W. 1996. Mesh Reduction with Error Control. In *IEEE Visualization Proceedings*, 311–318.

KOBBELT, L., VORSATZ, J., LABSIK, U., AND SEIDEL, H.-P. 1999. A Shrink Wrapping Approach to Remeshing Polygonal Surfaces. *Eurographics Proceedings (CGF) 18*, 119–130.

LEE, A., SWELDENS, W., SCHRÖDER, P., COWSAR, L., AND DOBKIN, D. 1998. MAPS: Multiresolution Adaptive Parameterization of Surfaces. In *ACM SIGGRAPH Proceedings*, 95–104.

LÉVY, B., PETITJEAN, S., RAY, N., AND MAILLOT, J. 2002. Least Squares Conformal Maps for Automatic Texture Atlas Generation. In *ACM SIGGRAPH Proceedings*, 362–371.

LINDSTROM, P., AND TURK, G. 1998. Fast and Memory Efficient Polygonal Simplification. In *IEEE Visualization Proceedings*, 279–286.

LINDSTROM, P., AND TURK, G. 2000. Image-driven simplification. *ACM Transactions on Graphics 19*, 3 (July), 204–241.

LLOYD, S. 1982. Least square quantization in PCM. *IEEE Trans. Inform. Theory 28*, 129–137.

MAILLOT, J., YAHIA, H., AND VERROUST, A. 1993. Interactive Texture Mapping. In *ACM SIGGRAPH Proceedings*, 27–34.

NADLER, E. 1986. Piecewise-Linear Best $\mathcal{L}^2$ Approximation On Triangulations. In *Approximation Theory V*, C. K. Chui, L. L. Schumaker, and J. D. Ward, Eds. Academic Press, 499–502.

OHTAKE, Y., BELYAEV, A., ALEXA, M., TURK, G., AND SEIDEL, H.-P. 2003. Multi-level Partition of Unity Implicits. *ACM Trans. on Graphics 22*, 3, 463–470.

OHTAKE, Y., BELYAEV, A., AND PASKO, A. 2003. Dynamic Mesh Optimization for Implicit Surfaces with Sharp Features. *Visual Computer 19(2)*, 115–126.

OSHER, S., AND FEDKIW, R., Eds. 2002. *Level Set Methods and Dynamic Implicit Surfaces*. Applied Mathematics Sciences 153. Springer-Verlag.

PAULY, M., GROSS, M., AND KOBBELT, L. P. 2002. Efficient Simplification of Point-Sampled Surfaces. In *Proc. of IEEE Visualization*, 163–170.

PEBAY, P. 2002. Planar Quadrangle Quality Measures: Is There Really A Choice? In *11th Intl. Meshing Roundtable*, 53–62.

RÖSSL, C., AND KOBBELT, L. 2000. Line-art Rendering of 3D Models. In *Proceedings of Pacific Graphics*, 87–96.

SANDER, P. V., SNYDER, J., GORTLER, S. J., AND HOPPE, H. 2001. Texture Mapping Progressive Meshes. In *ACM SIGGRAPH Proceedings*, 409–416.

SANDER, P., WOOD, Z. J., GORTLER, S., SNYDER, J., AND HOPPE, H. H. 2003. Multi-chart Geometry Images. In *Proc. of Symp. on Geo. Processing*, 146–155.

SHEFFER, A. 2001. Model Simplification for Meshing Using Face Clustering. *Computer Aided Design 33*, 925–934.

SHEWCHUK, J. R., 2002. What Is a Good Linear Finite Element? Interpolation, Conditioning, Anisotropy, and Quality Measures. Preprint.

SHLAFMAN, S., TAL, A., AND KATZ, S. 2002. Metamorphosis of polyhedral surfaces using decomposition. *Eurographics Proceedings (CGF) 22*, 3 (Sept.), 219–228.

SIMPSON, R. B. 1994. Anisotropic Mesh Transformations and Optimal Error Control. *Appl. Num. Math. 14(1-3)*, 183–198.

SORKINE, O., COHEN-OR, D., AND TOLEDO, S. 2003. High-pass Quantization for Mesh Encoding. In *Proc. of Symp. on Geo. Processing*, 42–51.

SURAZHSKY, V., ALLIEZ, P., AND GOTSMAN, C. 2003. Isotropic Remeshing of Surfaces: a Local Parameterization Approach. In *Proceedings of 12th International Meshing Roundtable*, 215–224.

TURK, G. 1992. Re-Tiling Polygonal Surfaces. In *ACM SIGGRAPH Proceedings*, 55–64.

VÁRADY, T., MARTIN, R. R., AND COX, J. 1997. Reverse Engineering of Geometric Models - An Introduction. *CAD 29*, 4, 255–268.

WOOD, Z., HOPPE, H., DESBRUN, M., AND SCHRÖDER, P. 2004. Removing Excess Topology in Isosurfaces. *ACM Trans. on Graphics 23*, 2 (Apr.).

Figure 17: *David (700K triangles): at two different resolutions for the $\mathcal{L}^{2,1}$ metric (resp. 5 min, and 10 min), front and back views. Edges inside each region are removed to improve clarity.*