Recent Advances in Remeshing of Surfaces

Pierre Alliez[1], Giuliana Ucelli[2], Craig Gotsman[3], and Marco Attene[4]

[1] INRIA, France pierre.alliez@sophia.inria.fr
[2] IGD / GraphiTech, Italy giuliana.ucelli@graphitech.it
[3] Technion, Israel gotsman@cs.technion.ac.il
[4] CNR, Italy jaiko@ge.imati.cnr.it

Summary. Remeshing is a key component of many geometric algorithms, including modeling, editing, animation and simulation. As such, the rapidly developing field of geometry processing has produced a profusion of new remeshing techniques over the past few years. In this paper we survey recent developments in remeshing of surfaces, focusing mainly on graphics applications. We classify the techniques into five categories based on their end goal: structured, compatible, high quality, feature and error-driven remeshing. We limit our description to the main ideas and intuition behind each technique, and a brief comparison between some of the techniques. We also list some open questions and directions for future research.

1 Introduction

Surface meshes are commonly used as a representation of shape in many computer graphics applications. Many of these meshes are generated by scanning devices or by isosurfacing implicit representations. Unfortunately such processes - especially if automated - are error-prone, and the resulting "raw" meshes are rarely satisfactory. Often they are oversampled and contain many redundant vertices. Besides the reduction of this complexity, which has stimulated a considerable amount of work in automated mesh simplification [LRC+02], there is frequently a need to improve the *quality* of the mesh, in terms of vertex sampling, regularity and triangle quality. This improvement process is called *remeshing* (see example Fig.1). It is useful to ease not only the display process, but also the editing, animation, processing, storing and transmission. For these reasons, remeshing of surfaces has received considerable attention over the past few years.

We invite the reader interested in related topics to read several comprehensive courses and tutorials on subdivision surfaces [Sch98, ZS00], geometric modeling [KBB+00], digital geometry processing [Tau00, SS01] morphing [Ale02], simplification and compression [LRC+02, GGK02, AG03] and parameterization [FH04].

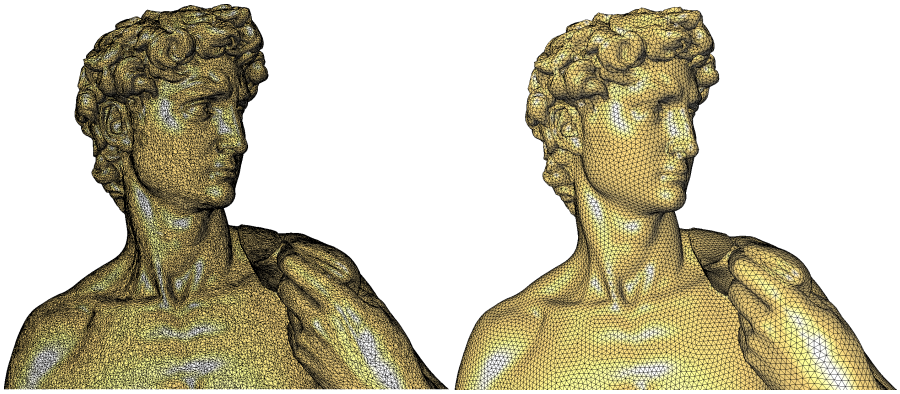

Fig. 1. Uniform remeshing of the Digital Michelangelo David model. Figure reproduced from [SAG03].

1.1 Remeshing

There is no precise definition of remeshing, since it often varies according to the targeted goal or application. Nonetheless, one possible definition could be: "Given a 3D mesh, compute another mesh, whose elements satisfy some quality requirements, while approximating well the input". "Quality" has several meanings. It can be related to the sampling, grading, regularity, size and shape of elements. Often a combination of these criteria is desired in real applications. Some remeshing techniques proceed by altering the input, and some generate a new mesh from scratch.

1.2 Applications

Remeshing of surfaces is beneficial to a wealth of applications which take as input a meshed surface. These range from modeling to visualization through reverse engineering and simulation: creation and editing, animation, metamorphosis, approximation, simulation, denoising, smoothing and fairing, efficient rendering, compression, feature recovery and levels of detail.

1.3 Main Issues

We begin by listing briefly some general issues that arise during the remeshing process:

- **Validity.** The mesh has to be a *valid* mesh. This usually means that it should be a simple manifold. Typically it will also be closed, namely not contain boundaries.
- **Quality.** The quality of mesh elements is crucial for robustness and numerical stability, required for numerical simulation as well as for geometry processing. Numerical computations, such as finite element analysis, require fairly regular meshes, both in terms of geometry and connectivity. These meshes are used to

compute mechanical stress or solve heat and other differential equations. A high-quality mesh is required to minimize numerical errors and singularities that might otherwise arise (see [She02]).

- **Fidelity.** The newly generated mesh has to best approximate the original shape geometry, while keeping the mesh complexity below a given budget. Ideally, "just enough" resolution for the problem being solved is sought. This involves choosing an error metric, as well as to decide between interpolation and approximation.
- **Discrete input.** The input is given as a discrete mesh, which is usually only an approximation of some (unknown) continuous shape. Having just this discrete approximation hampers most shape interrogation operations (*e.g.* normal, tangent plane, curvature estimations) when the discretization is not ideal. Moreover, meshes generated from sampled point clouds by reconstruction algorithms may be contaminated by aliasing artifacts and lack important features present in the original.
- **Large data sets.** Modern 3D scanners generate very large datasets when the sampling rate is increased to ensure that no details are missed. As a result, the sampling and tessellation is insensitive to the shape, and there is much redundancy in the data.
- **Uncertainty.** Data obtained by an acquisition process such as laser scanning is often contaminated by electronic, mechanical or even optical noise present in the scanning pipeline.
- **Correspondence.** A central issue common to all remeshing techniques is to find the corresponding location of a new vertex on the input mesh surface. Such a correspondence is typically found by computing a parameterization of the input mesh. This is a complex problem which is either computationally expensive, suffers from accuracy issues, or imposes restrictions on the mesh. It is particularly problematic when performing the remeshing operations on a 2D parametric domain: the mapping of a nontrivial 3D structure (possibly a 3D mesh with arbitrary genus and holes) to a 2D parametric domain inevitably introduces some metric distortion, and may lead to the loss of important information. Furthermore, if the parameterization is combined with mesh segmentation, it is likely to encounter difficulties near the patch boundaries. Other parameter-free approaches work directly on the surface, and perform local modifications on the mesh (like adding, removing, or relocating vertices). During these adaptations, the mesh vertices are forced to remain on the input mesh. This type of approach can be found in several different techniques [Fre00, FB98, HRD$^+$93, Hop96, RVSS00, Tur92, SG03]. The optimizations are either performed in 3D (which is computationally expensive), or in a tangent plane (which is faster, but less accurate). By using local operations, this approach may avoid the pitfalls of the techniques based on global operations; and by performing the remeshing operations on a 2D plane, it can be considerably faster than 3D optimizations. The distortion caused by mapping a 3D mesh to a 2D parametric domain can be considerably reduced by using optimizations such as overlapping patches [SG03]; and error accumulation (as often caused by local operations) can be minimized by constantly comparing

to a reference smooth approximation of the original geometry (*e.g.* by using triangular cubic Bézier patches such as PN triangles [VPBM01] or continuous patches [WM96]).

We now list some general desirable algorithmic functionalities of a remeshing algorithm:

- **Levels of detail.** Support for continuous Levels-of-Detail (*i.e.*, continuous-resolution representations) is often desirable for rendering and transmission applications. This poses a major challenge to remeshing algorithms.
- **Complexity.** The computer graphics community is mainly interested in interactive algorithms, thus the speed of the remeshing algorithm is important. So the main focus is on the tradeoff between the quality of the result and the speed of the remeshing operation. Typically close-to-linear runtime complexity is required.
- **Theoretical guarantees.** Algorithms that provide guarantees on the topology, matching of constraints, bounds on the distortion error such as geometry and normals, or bounds on the shape of elements are highly desirable for applications where certified results are required.

2 State of the Art

To keep the structure of this survey as clear as possible, we classify the remeshing techniques by their end goal rather than by the technique they employ. We classify the techniques into five main categories: *structured* remeshing (Section 2.1), *compatible* remeshing (Section 2.2), *high quality* remeshing (Section 2.3), *feature* remeshing (Section 2.4) and *error-driven* remeshing (Section 2.5). Some techniques appear in several of these categories when they achieve several goals simultaneously.

2.1 Structured Remeshing

Definition

Structured remeshing replaces an unstructured input mesh with a structured one. In a structured mesh, sometimes called a regular mesh, all internal vertices are surrounded by a constant number of elements. A semi-regular mesh is obtained by regular subdivision of an irregular mesh (see [SS01]). All the vertices are regular except for a small number of *extraordinary* vertices (see Fig.2). A highly regular mesh is one in which the vast majority of vertices are regular, yet the mesh has not necessarily been generated by subdivision.

Motivation

Structured meshes offer certain advantages over unstructured ones. Their connectivity graph is significantly simpler, hence allows for efficient traversal and localization

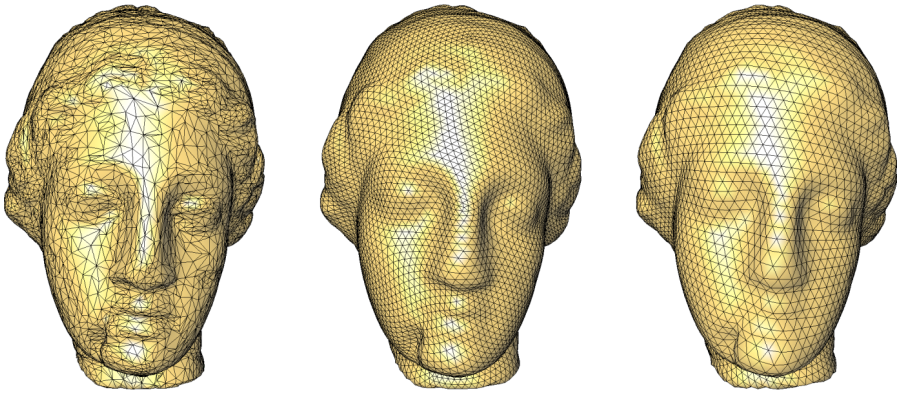

Fig. 2. Meshes: Irregular, semi-regular and regular.

in the algorithms. Semi-regular meshes, which are essentially piecewise-regular, offer a tradeoff between the simplicity of structured meshes and the flexibility of unstructured meshes.

Semi-Regular

Semi-regular meshes are obtained by recursive subdivision of an initial base mesh (Fig.3). Their hierarchical structure makes them ideal for multiresolution analysis (coarsification by down sampling and smoothing) and synthesis (subdivision and adding of details). They have proven useful for modeling smooth or piecewise smooth surfaces, reverse engineering, multiresolution analysis and modeling, morphing, editing and visualization with levels of detail applications.

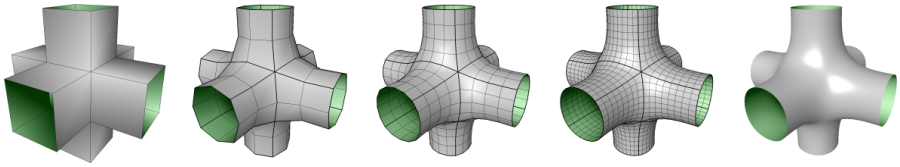

Fig. 3. Semi-regular mesh obtained by recursive subdivision of an initial base mesh.

The emerging field of geometry processing [SS01] has made significant use of semi-regular meshes. A fundamental question of geometry processing is the following: is it possible to extend the methods of classical digital signal processing (e.g. the discrete Fourier transform and wavelets), usually applied on regular uniform structures, to the irregular non-uniform setting ? This question still remains only partially solved, and the solution of choice consists of semi-regular remeshing of the original shape so that the geometric “signal” is resampled onto regular and uniformly sampled patches. One example of geometry processing is given by a set of discrete

operators used for smoothing and fairing, applicable only in the regular and uniform setting.

The main techniques for semi-regular remeshing can be classified into two main categories according to the way they find correspondences between the input and output meshes. The first class of techniques uses a parameterization to find a bijective correspondence. The techniques in this class differ mainly by the type of parameterization:

- Techniques that parameterize the input mesh on a global planar domain [HG00]. The parameter domain is then resampled, and the new mesh connectivity is projected back into 3D space, resulting in an improved version of the input (Fig.4). The main drawbacks of the global parameterization methods are the sensitivity to the specific parameterization used, and the metric distortion that may arise (due to the fact that the 3D structure is forced into a parameter plane). Furthermore, many of these techniques involve the solution of a large set of equations, resulting in substantial computation. Sander *et al.* [SGSH02] use a hierarchical approach based on multigrid methods, which can accelerate the process to almost linear time even for large meshes; but numerical precision issues may arise for meshes with severe isoperimetric distortion.
- Techniques that parameterize the original model onto a set of base triangular domains, the latter obtained either by simplification, or by partitioning the original mesh into regions using a discrete analogue of the notion of a Voronoi tile. This technique, used by [ERD$^+$95, LSS$^+$98, GVSS00], yields excellent results while being sensitive to the patch structure (see example Fig.5). The vertex sampling is also delicate to control.

The second class of techniques does not rely on any parameterization but instead uses ray shooting [KVLS99] to find correspondences for shrink wrapping the new mesh onto the input mesh (Fig.6).

Shape compression techniques employing *semi-regular remeshing* are among the best reported to date. The main idea behind these techniques [KSS00, GVSS00, KG03, PA02] is the observation that a mesh representation has three components: geometry, connectivity and parameterization, of which the latter two (*i.e.* connectivity and parameterization) are not important for the representation of the geometry. The goal is therefore to reduce the "volume" of these two components as much as possible by semi-regular remeshing (see [AG03] for a more detailed description of this shape compression technique).

Discussion

In all mapping-based methods, the parameterization plays a critical role, and any deficiencies in it will be amplified in the output. In particular, building a globally smooth parameterization is notoriously difficult [KLS03]). Having a subdivision

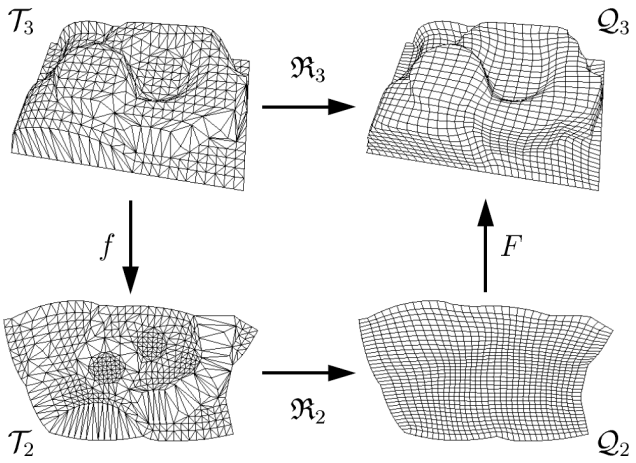

Fig. 4. Quadrilateral remeshing: The main idea of the algorithm is to circumvent the three-dimensional remeshing problem by flattening the 3D mesh T_3 to a 2D version T_2, and solving the two-dimensional problem instead. The deflation function f is then defined by linearly mapping each triangle of T_3 to the corresponding triangle in T_2 while the inverse inflation function F enables to get back from 2D to 3D. Figure reproduced from [HG00].

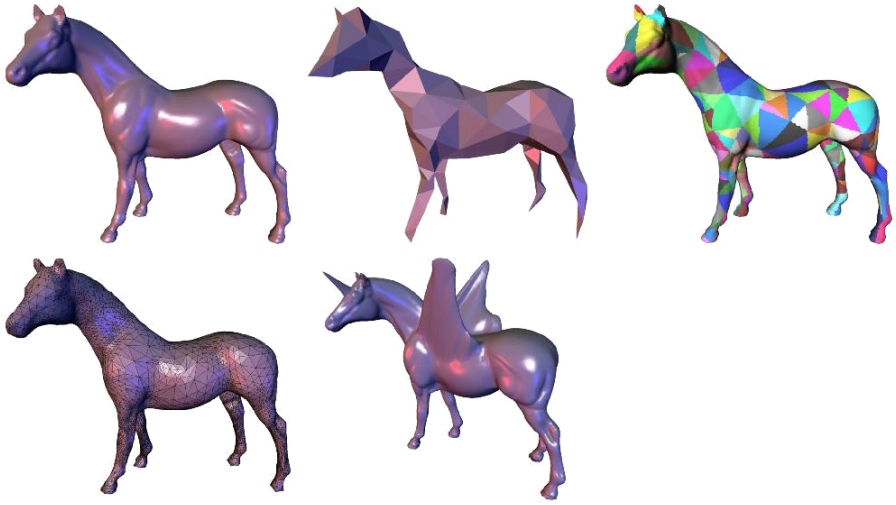

Fig. 5. Multiresolution adaptive parameterization of surfaces. Overview of the algorithm. Top left: a scanned input mesh (courtesy Cyberware). Next the parameter or base domain, obtained through mesh simplification. Top right: regions of the original mesh colored according to their assigned base domain triangle. Bottom left: adaptive remeshing with subdivision connectivity. Bottom middle: multiresolution edit. Figure reproduced from [LSS$^+$98].

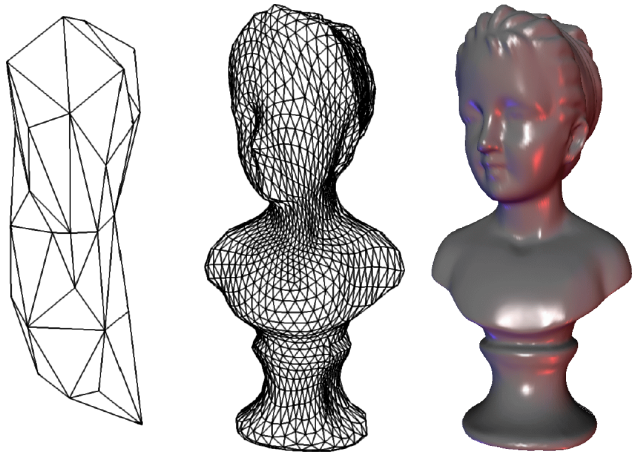

Fig. 6. Remeshing by shrink wrapping. The original bust model has 61K triangles. The base mesh with 72 triangles is subdivided three times to generate the center mesh and 5 times to generate the right image. Figure reproduced from [KVLS99].

connectivity is still necessary for multiresolution analysis, which has proven a powerful tool for many geometric modeling and processing applications. The challenge remains to handle irregular meshes directly. This will remain difficult while current geometry processing approaches are designed in analogy to their continuous counterpart.

Completely Regular

In a regular mesh (a grid, triangle or hexagonal tessellation) the connectivity is implicit, the compactness and regularity of the data structure improves the efficiency and facilitates the implementation of many algorithms. Regular remeshing has proven successful for efficient rendering (no cache indirection), texture and other modulation mapping (*e.g.* normal, transparency maps).

Gu et al. [GGH02] remesh irregular triangle meshes using a regular rectangular grid. The input mesh of arbitrary genus is initially cut to reduce it to a single topological disc. It is then parameterized on the unit 2D square while minimizing a geometric-stretch measure. This is then represented as a so-called *geometry image* that stores the geometry as well as any modulation map required for visualization purposes (Fig.7). Such a compact grid structure drastically simplifies the rendering pipeline since all cache indirections found in usual irregular mesh rendering are eliminated. Despite its obvious importance for efficient rendering, this technique has a few drawbacks due to the inevitable surface cutting: each geometry image has to be homeomorphic to a disk, therefore closed or genus> 0 models have to be cut along a cut graph. In particular, it introduces unacceptably high parameterization distortion for high genus models or shapes with high isoperimetric ratios (*e.g.* long

extremities). To alleviate these drawbacks, Sander *et al.* [SWG+03] use an atlas construction to map the input mesh onto charts of arbitrary shape. Those charts are then packed in a geometry image in parameter space, and a zippering algorithm is used to remove the discontinuities across chart boundaries and create a watertight surface. Another way to minimize seams due to cutting is to first parameterize the mesh to a sphere [GGS03], which is then mapped in a highly structured way to the square.

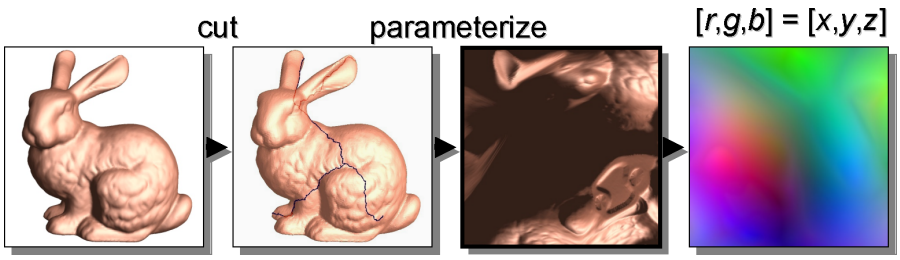

Fig. 7. Construction of a Geometry Image: Original mesh (70k faces, genus 0), original mesh with cut, parameterization and Geometry Image (257×257). Figure reproduced from [GGH02].

Discussion

The concept of geometry images follows the recent trends in graphics that represent all surface modulation signals as "texture images" (normal maps, bump maps, transparency maps, color maps, light maps, reflection maps) instead of using a fine mesh with attributes at each vertex. The key idea is to represent the shape geometry itself using regular grids, assuming the cost of 3D transformations to be negligible with respect to the cost of "decorating" the mesh using a complex multi-texturing process. Research on geometry images, mainly driven by Hoppe and co-workers, anticipates the unification of vertex and image buffers.

Highly Regular

In [SRK03] a remeshing method for the creation of piecewise regular meshes is described. Based on their orientation, this algorithm partitions the triangles into six sets. The set of triangles whose normal is closest to the positive x-direction is sampled using a regular grid in the y-z plane. The other five sets are sampled similarly using the appropriate grids. Finally, these re-sampled pieces are connected into one valid mesh. The result typically contains a large fraction of regular vertices; specifically, all the internal vertices of each piece are regular by construction, while some irregular vertices may appear along the seams.

Surazhsky and Gotsman [SG03] perform local modifications directly on the mesh surface in order to obtain a highly regular mesh. One key feature of their method is

the use of overlapping patches to locally parameterize the surface (which overcomes both the problems of global parameterization and the remeshing problems that usually arise near the patch boundaries when parameterizing based on mesh segmentation). Another key feature is a series of edge-collapse and edge-flip operations combined with area-based mesh optimization to improve regularity and to produce well-shaped triangles (without the problem of long and skinny triangles typically created if mesh generation is based on triangle areas). As the overlapping parameterization allow to apply 2D mesh optimization methods to 3D meshes (while minimizing the distortion problem, typical of mapping a 3D mesh to a 2D parametric domain), this algorithm is fast as well as robust (see an example in Fig.8).

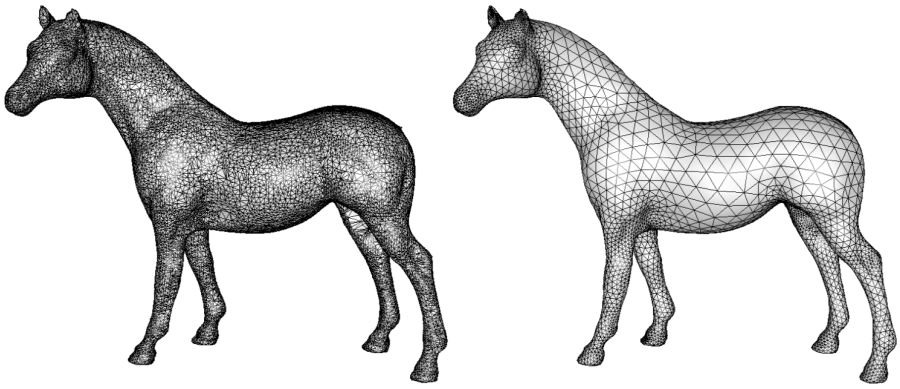

Fig. 8. Highly regular remeshing. Figure reproduced from [SG03].

Discussion

Highly regular meshes are frequently obtained by tessellating on a regular grid. Surazhsky and Gotsman [SG03] demonstrate that highly regular meshes cannot be generated simply by local mesh adaptation, unless some semi-global operations, such as drifting edges, are performed. One challenge is to obtain semi-regular meshes with a prescribed number of irregular vertices (up to that required by the Euler formula) by semi-global adaptation instead of subdivision.

2.2 Compatible Remeshing

Definitions

Given a set of 3D meshes with a partial correspondence between them, *compatible remeshing* amounts to generating a new set of meshes which are remeshes of the input set, such that they have a common connectivity structure, well-shaped polygons, approximate well the input, and respect the correspondence.

Motivation

Motivating applications are morphing between shapes and attributes, multi-model shape blending, synchronized model editing, fitting template models to multiple data sets and principal component analysis. In these applications the common connectivity is usually more important than the mesh element quality.

Joint Parameterization

Much of the work done on compatible meshing focuses on morphing as the target application. This requires first the computation of a *joint parameterization* (sometimes called *cross parameterization*), namely, a bijective mapping between the two meshes, possibly subject to some constraints. Alexa [Ale02] gives a good review of joint parameterization and compatible remeshing techniques developed for morphing. Joint parameterization is typically computed by parameterizing the models on a common base domain. One popular choice is the sphere. There are a number of algorithms for spherical parameterization, e.g. [Ale99, GGS03, PH03]. Of those, only Alexa's method addresses feature correspondence (see Fig.9). However, it does not guarantee a bijective mapping and is not always capable of matching the features. An inherent limitation of a spherical parameterization is that it can only be applied to closed, genus zero surfaces.

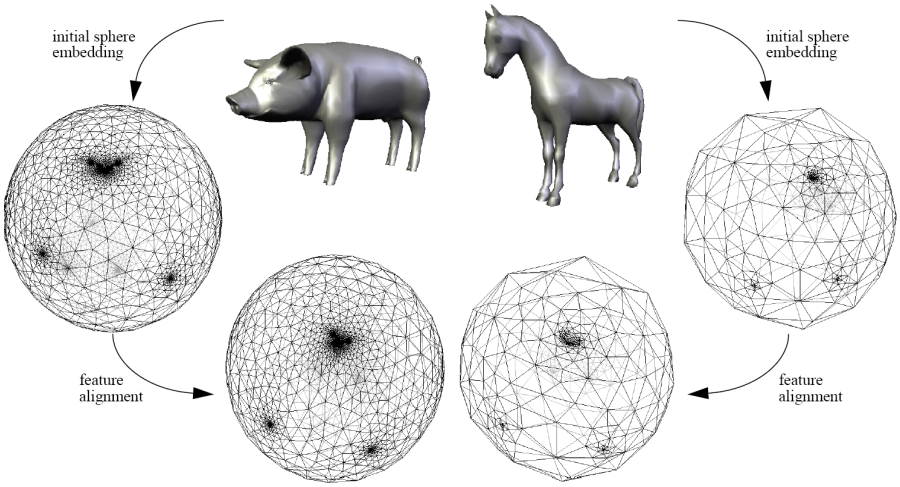

Fig. 9. Joint spherical parameterization: First, an initial sphere embedding is computed for each mesh. Second, the initial subdivision is deformed such that the common features coincide on the spheres. The two connectivities are then merged. Figure reproduced from [Ale99].

A more general approach is to parameterize the models over a common base mesh [LDSS99, LCLC03, MKFC01, PSS01]. This approach splits the meshes into matching patches with an identical inter-patch connectivity. After the split, each set

of matching patches is parameterized on a common convex planar domain. An advantage of this approach is that it naturally supports feature correspondence by using feature vertices as corners of the matching patches. The main challenge in mapping the models to a single base mesh is to construct identical inter-patch connectivities. The vast majority of the methods use heuristic techniques that work only when the models have nearly identical shape. Praun et al. [PSS01] provide a robust method for partitioning both meshes into patches given user-supplied base mesh connectivity. A common disadvantage of existing techniques to construct base meshes is that the patch structure severely restricts the freedom of the parameterization. As a result, the shape of the patches has a huge influence on the amount of mapping distortion.

Given the joint parameterization, many techniques [Ale99, KSK00] generate the common connectivity for the models by overlaying the meshes in this parameter domain and computing a common intersection mesh. The new mesh captures the geometry of the models. However, the new mesh is typically much larger than the input meshes and has very badly shaped triangles. The overlaying algorithm is also extremely tricky to implement, as it requires multiple intersection and projection operations. An alternative is to remesh the models using a regular subdivision connectivity derived from the base mesh [LDSS99, MKFC01, PSS01]. Due to the rigid connectivity structure, the shape of the mesh triangles reflects the shape of the base mesh. Thus, if the shape of the triangles is poor (because, for example,the user picked unevenly spaced feature vertices) the shape of the mesh triangles will reflect this. More importantly, a model that contains features interior to the base mesh triangles will require a very dense subdivision mesh over the entire model.

Inter-Surface Mapping

Kraevoy and Sheffer [KS04] developed a technique for joint parameterization and compatible remeshing of two genus-0 meshes with a partial correspondence (Fig.10). The input of the algorithm is a pair of triangle meshes and a set of corresponding feature vertices. The first stage of the algorithm constructs a common base domain by incrementally adding pairs of matching shortest edge paths. Care is taken to avoid intersections and blocking, as well as to preserve cyclic orders in order to obtain matching patch layouts. Face paths are then added until all patches are triangulated, and an additional path flip procedure improves the connectivity of the patch layout. The second stage computes a shape preserving parameterization with smooth transitions between patches using the mean-value parameterization followed by an adjacency preserving smoothing procedure. The last stage constructs compatible meshes by alternating vertex relocation to attract vertices towards areas of higher error, and error-driven mesh refinement. The approximation of normals is improved by an additional pseudo edge-flip refinement procedure. The meshes generated by this procedure contain significantly fewer elements than those generated by simple overlaying methods, while approximating the geometry and normals of the input model.

Schreiner et at. [SAPH04] use a procedure similar to that of Kraevoy and Sheffer for base mesh construction, handling models of arbitrary genus more robustly. To generate a smooth joint parameterization, they use a symmetric, stretch based relaxation procedure, which trades off high computational complexity for quality of

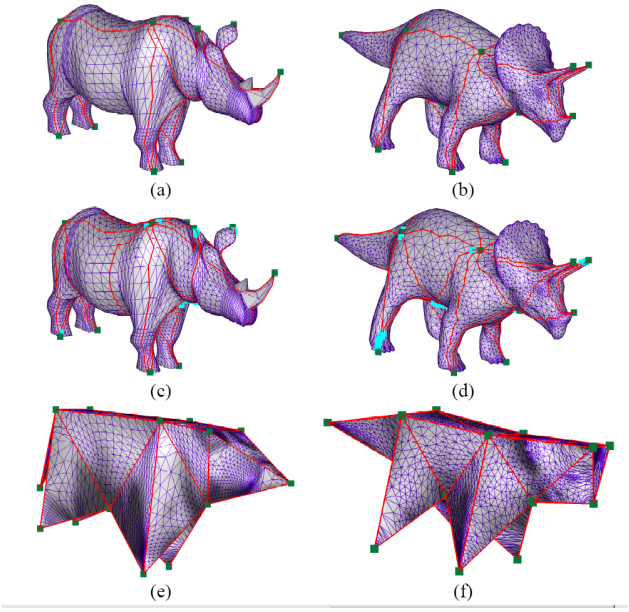

Fig. 10. Base domains construction for joint parameterization and compatible remeshing of two genus-0 meshes (feature vertices are dark green): (a),(b) edge paths; (c),(d) face paths, new vertices are highlighted (turquoise); (e),(f) base meshes. Figure reproduced from [KS04].

the mapping. The common mesh is generated using an overlay of the input meshes, as described above. To avoid artifacts, the method has to relax the feature vertex correspondence in some cases.

Discussion

While compatible remeshing is becoming increasingly important in computer graphics animation applications, where a sequence of meshes is available, it is still plagued by a number of problems. The selection of pairs of correspondent feature points is still manual. Very few existing methods extend easily to arbitrary genus surfaces and long animation sequences. Lastly, the results are still highly dependent on the parameterization method used to perform the joint parameterization.

2.3 High Quality Remeshing

Definitions

In our taxonomy *high quality remeshing* means to generate a new discretization of the original geometry with a mesh that exhibits the three following properties: well-shaped elements, uniform or isotropic sampling and smooth gradation sampling. A well-shaped triangle has aspect ratio as close to 1 as possible, and a well-shaped

quadrilateral contains angles between two consecutive edges as close to $\pi/2$ as possible. Isotropic sampling means that the sampling is locally uniform in all directions. Requiring uniform sampling is even more restricting by dictating the sampling to be uniform over the entire mesh. Smooth gradation means that if the sampling density is not uniform - it should vary in a smooth manner [BHP97].

Motivation

High quality remeshing is motivated by numerical stability and reliability of computations for simulation. Efficient rendering, interactive free-form shape modeling, as well as a few geometry processing algorithms such as compression, fairing or smoothing also benefit from high quality meshes. The shape of mesh elements [PB01] has a direct impact on the numerical stability of numerical computations for finite element analysis, as well as for efficient rendering. For the widely used triangle meshes, it is desirable to have no small angles and/or no large angles, depending on the targeted computations (see [She02]).

We restrict our description to *point-based* sampling techniques, although other primitives can be evenly distributed on surfaces for meshing (*e.g.* bubble packing [YS04], square cell packing [SL98], placement of streamlines [ACSD$^+$03]). Uniform (resp. isotropic) point sampling for remeshing amounts to globally (resp. locally) distributing a set of points on the input model in as even a manner as possible. We may distinguish between *greedy* sample placement methods which insert one point at a time to refine the newly generated model, and *relaxation-based* methods which improve an initial placement either locally or globally through point relocation.

Farthest point sampling.

The *farthest point* paradigm [LPZE96] advocates inserting one sample point at a time, as far as possible from previously placed samples, *i.e.* at the center of the biggest void. Its main advantage is in retaining the uniformity while increasing the density. In contrast to stochastic approaches, it can guarantee some uniformity by bounding the distance between samples [BO03]. This paradigm, also called Delaunay refinement [Che93, Rup95, Mil04] or sink insertion [EG01] has proven particularly effective in producing uniform as well as isotropic sample placements. It has been recently extended using the geodesic distance estimated on the input mesh to find the center of the biggest voids [PC03, MD03]. From an initial point set sampled on the input mesh, a Delaunay-like triangulation is created by taking the dual of a geodesic-based Voronoi diagram constructed using the Fast Marching method of Sethian and Kimmel [Set99].

Advancing front.

A popular method for evenly-spaced placement is the advancing front paradigm commonly used for meshing [AFSW03, Har98, TOC98]. This method has been recently extended using an approximation of the geodesic distance for remeshing by

Sifri *et al.* [SSG03]. A more general approach was recently introduced by Dong et al. [DKG05], who compute two orthogonal harmonic Morse functions on the mesh surface. Drawing contours of each results in a good quad remesh (Fig.11). Another quasi-uniform remeshing approach based on an advancing front is implicit in the SwingWrapper compression scheme [AFSR03]. In order to reduce the number of bits to encode the vertex locations, SwingWrapper partitions the surface into geodesic triangles that, when flattened, constitute a new mesh which is strongly compressible. The remeshing is performed so that for each vertex of the new mesh there is at least one incident isosceles triangle having a prescribed height. Though not optimally uniform, the remeshing performed by SwingWrapper might effectively be used as an initial guess for iterative processes which try to optimize uniformity.

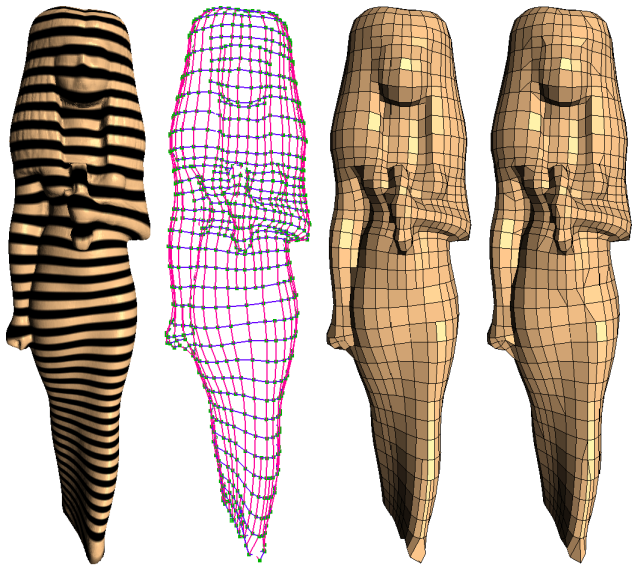

Fig. 11. Quadrilateral remeshing of arbitrary manifolds: (a) A harmonic function is computed over the manifold. (b) A set of crossings along each flow line is constructed. (c) A non-conforming mesh is extracted from this net of flow crossings. (d) A post-process produces a conforming mesh composed solely of triangles and quadrilaterals. Figure reproduced from [DKG05].

Attraction-repulsion.

One of the first remeshing techniques to surface in the graphics community was described by Turk [Tur92]. It places a (user defined) number of new vertices on the input mesh, and arranges the new vertices with the help of an attraction-repulsion particle relaxation procedure, followed by an intermediate mutual tessellation that

contains both the vertices of the original mesh and the new vertices. This simple approach produced quite remarkable results, although it had several limitations. Most notably, it is not suitable for models that have sharp edges and corners, so does not precisely approximate such a surface.

Umbrella operator.

Another popular method commonly used for even placement of samples consists of repeatedly moving each sample point to the barycenter of its neighbors, and updating the mesh connectivity. This procedure tends to generate globally uniform edges in the simple case, and locally uniform edges (*i.e.* isotropic sampling) if weights are assigned to edges [VRS03].

The interactive remeshing technique introduced by Alliez *et al.* [AMD02] is based on global parameterization. It represents the original mesh by a series of 2D maps in parameter space, and allows the user to control the sampling density over the surface patch using a so-called control map, the latter created from the 2D maps. First an initial isotropic resampling is performed using an error-diffusion sampling technique originally designed for image half-toning [Ost01], followed by relaxation using the umbrella operator. This method is a hybrid between a greedy and a variational method since the coeffi cients used for error diffusion are optimized during an offline procedure which seek a placement with a so-called blue-noise profi le, related to the notion of isotropic sampling. The initial sample placement is then performed in a single pass at run time. See example Fig.12.

Local area equalization.

Precise uniform sampling can be achieved through local area equalization. Assuming the one-ring of the vertex to be relocated is fi xed, the new position is computed by solving a linear system in order to minimize the dispersion of area among all incident triangles [SG03]. This technique has been recently extended to local equalization of the Voronoi areas of the vertices in order to symmetrize a linear system used for multiresolution modeling [BK04]. The system is solved effi ciently using a Cholesky-based solver that takes advantage of symmetric band-limited matrices. Although effi cient and robust, these area equalization techniques do not provide an easy way to globally distribute a set of samples in accordance to a density function.

Lloyd relaxation.

Precise isotropic sample placement can be achieved through the use of the Lloyd clustering algorithm [Llo82], which consists of alternating Voronoi partitioning with relocation of the generators to the centroid of their respective Voronoi cell (Fig.13). Such a relaxation procedure generates centroidal Voronoi diagrams [DFG99], where the generators coincide with the centroid of their respective cells. Lloyd relaxation minimizes an energy related to the compactness of the Voronoi cells (and hence

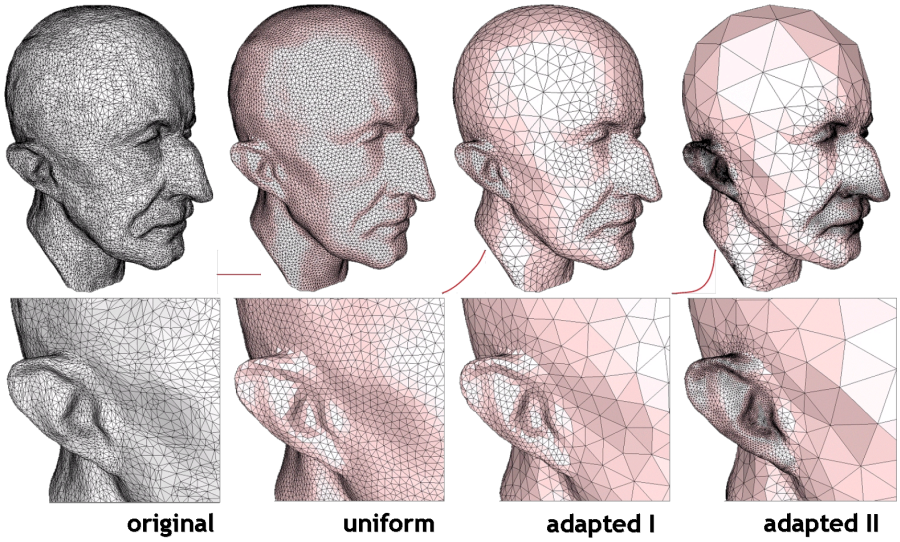

Fig. 12. Interactive geometry remeshing: Remeshing of the MaxPlanck model with various distribution of the sampling with respect to the curvature. The original model (left) is remeshed uniformly and with an increasing importance placed on highly curved areas (left to right) as the magnified area shows. Figure reproduced from [AMD02].

isotropic sampling) while equi-distributing the energy within each cluster, as shown by Gersho in the late seventies [Ger79]. Contrary to other methods, this method allows the definition of a density function related to the desired size of each Voronoi cell. It will then generate a distribution of energy which globally matches the local size while achieving precise isotropic sampling.

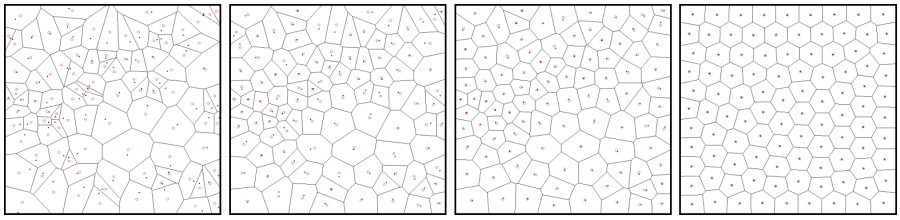

Fig. 13. Lloyd relaxation: A set of generators (black dots) are randomly generated (the centroid of each Voronoi area is depicted as a red circle). Each iteration of the Lloyd algorithm moves each generator to its associated centroid, and updates the Voronoi diagram.

Alliez *et al.* [AdVDI03], and Surazhsky *et al.* [SAG03] proposed two remeshing techniques based on Lloyd relaxation. The first uses a global conformal planar parameterization and then applies relaxation in the parameter space using a density

function designed to compensate for the area distortion due to flattening (Fig.14). To alleviate the numerical issues for high isoperimetric distortion, as well as the artificial cuts required for closed or genus models, the second approach applies the Lloyd relaxation procedure on a set of local overlapping parameterizations (Fig.15). More recently, the Lloyd-based isotropic remeshing approach has been extended in two directions: one uses the geodesic distance on triangle meshes to generate a centroidal geodesic-based Voronoi diagram [PC04], while the other is an efficient discrete analog of the Lloyd relaxation applied on the input mesh triangles [VC04].

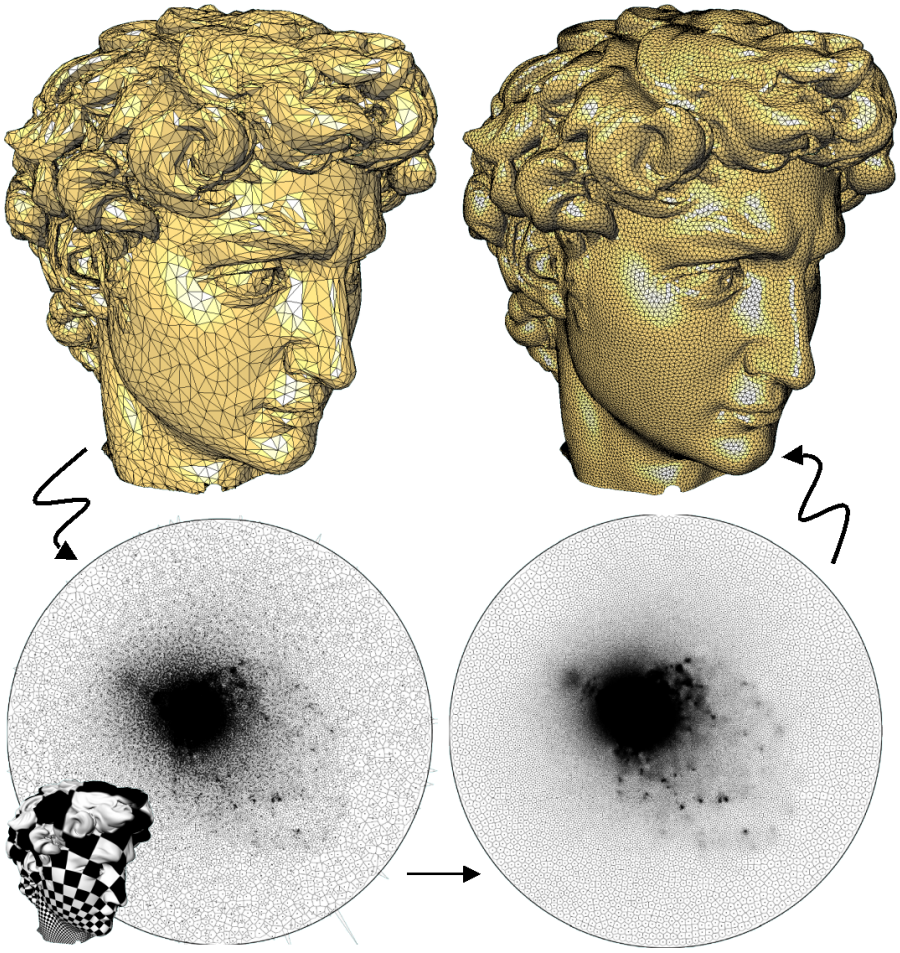

Fig. 14. Uniform remeshing of the David head: a planar conformal parameterization is computed (bottom left). Then Lloyd relaxation is applied in parameter space in order to obtain a weighted centroidal Voronoi tessellation, with which the mesh is uniformly resampled. Figure reproduced from [AdVDI03].

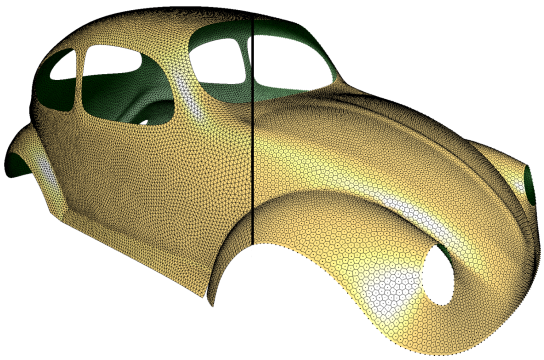

Fig. 15. Uniform remeshing of the Beetle: Lloyd relaxation is applied over local overlapping parameterizations as described in [SAG03].

Discussion

As expected, relaxation-based sample placement methods achieve better results than greedy methods, at the price of lengthier computations. Nevertheless, the only methods that provide certified bounds on the shape of elements are the greedy approaches based on Delaunay refinement. The Lloyd-based isotropic sampling method combined with local overlapping parameterization have recently proven successful for isotropically distributing a point set in accordance with a density function [SAG03]. Two remaining challenges related to the Lloyd relaxation method are to prove or to give sufficient conditions for achieving convergence to a global optimum, and to accelerate convergence. Another promising direction for efficient isotropic sampling is the hierarchical Penrose-based importance sampling technique developed by Ostromoukhov [ODJ04], which is deterministic and several orders of magnitude faster than relaxation methods.

2.4 Feature Remeshing

Definitions

Assume that a triangle mesh is an approximation of a curved shape, possibly with sharp edges and corners. We call the process that takes such a triangle mesh and generates a new tessellation in which the original sharp features are better approximated *feature remeshing*. In this context, the quality of the approximation may be measured either using a purely geometric metric (the L^∞ norm, for example, is strongly affected by badly-approximated sharp edges), or on a metric which reflects visual-quality (*e.g.*, normal deviation), or a combination of both.

Motivation

Most acquisition techniques, as well as several recently developed remeshing algorithms [RCG+01, SRK03, GGH02, AFSR03], restrict each sample to lie on a specific

line or curve whose position is completely defined by a pre-established pattern. In most cases, such a pattern cannot be adjusted to coincide with sharp edges and corners of the model, and almost none of the samples will lie on such sharp features. Thus, the sharp edges and corners of the original shape are removed by the sampling process and replaced by irregularly triangulated chamfers, which often result in a poor-quality visualization and high L^∞ distortion.

Feature-preserving

When the original shape is available, the error between such a shape and the approximating triangle mesh may be reduced by dense sampling. Over-sampling, however, will significantly increase the number of vertices, and thus the associated complexity, transmission and processing cost. Furthermore, as observed by Kobbelt *et al.* [KBSS01], the associated aliasing problem will not be solved by over-sampling, since the surface normals in the reconstructed model will not converge to the normal field of the original object. To cope with such a problem, an extended marching cubes algorithm has been proposed in [KBSS01]. The input shape is first converted into a signed distance field. This representation is then polygonized using a variant of the marching-cubes [LC87] algorithm in which vertex normals are derived from the distance field and used to decide whether a voxel contains a sharp feature or not. If so, additional vertices are created within the voxel and placed at intersections between the planes defined by the vertices and their associated normal. Another feature-preserving approach was proposed in [JLSW02], able to accurately polygonize models with sharp features using adaptive space subdivision (an octree), resulting in polygonal models with fewer faces. In a different setting, an original triangulation may be remeshed without converting it into a scalar distance field, and the aliasing problem may be avoided by snapping some of the evenly distributed vertices onto sharp creases, as proposed in [VRKS01].

Feature-enhancing

When the original shape is not available, the EdgeSharpener method [AFRS03] provides an automatic procedure for identifying and sharpening the chamfered edges and corners. In a first step, the mesh is analyzed and the average dihedral angle at edges is computed. Based on this value, “smooth” regions are grown on the mesh, and the strips of triangles separating neighboring smooth regions are considered “aliasing artifacts ” made of chamfer triangles. The growing process results in a number of smooth regions in which all the internal edges have a nearly flat dihedral angle. EdgeSharpener infers the original sharp edges and corners by intersecting planar extrapolations of the smooth regions. Then, each chamfer triangle is subdivided, and the newly inserted vertices are moved to the intersections, which are assumed to better approximate the original sharp features (see Fig.16). Unless the input contains significant amounts of noise, EdgeSharpener does not introduce undesirable side-effects, and limits the modifications to the portions of the mesh which are actually

chamfer artifacts. Furthermore, EdgeSharpener has been tested on results of several feature-insensitive remeshing algorithms [AFSR03, SRK03, RCG+01], and has been shown to significantly reduce the L^∞ distortion introduced by the remeshing process.

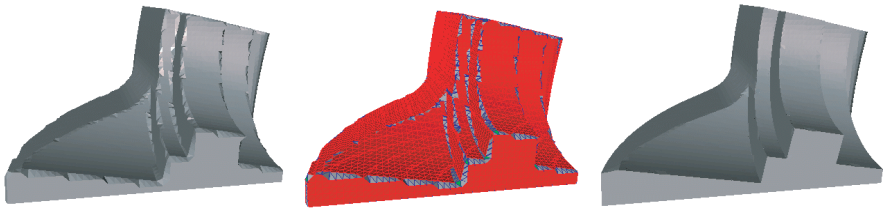

Fig. 16. EdgeSharpener: A triangle mesh reconstructed from a point cloud (left) is improved by EdgeSharpener [AFRS03]. Smooth regions are identified (red) and chamfer triangles (gray and green with blue edges) are sharpened (right).

To give the designer more flexibility, an interactive remeshing approach has been proposed in [KB03] for restoring corrupted sharp edges. The user is required to construct a number of fishbone structures (spine and orthogonal ribs) which will be automatically tessellated to replace the original chamfers. Though not automatic, this method is particularly suitable for simple models with few sharp edges, and allows to sharpen the chamfers as well as to modify the swept profiles to produce blends or decorated edges.

One may argue that an application of the extended marching cubes [KBSS01] to a polygonal mesh may be used to infer and hence reconstruct the sharp features. In [KBSS01], the application to remeshing is discussed and, in fact, it is useful to improve the quality of meshes having degenerate elements or other bad characteristics. In some cases, the information at the edge-intersections makes it possible to reconstruct sharp features in an Edge-Sharpener like manner. For example, if a cell contains an aliased part that does not intersect the cell's edges, the normal information at the intersections is used to extrapolate planes and additional points are created on the inferred sharp feature. If, on the other hand, the cell's edges do intersect the aliased part, the normal information is contaminated, and nothing can be predicted about any possible feature reconstruction. Moreover, remeshing the whole model through the extended marching cubes approach can introduce an additional error on the regions without sharp features, while the local remeshing produced by EdgeSharpener only affects the aliased zones by subdividing the triangles that cut through the original solid (or through its complement) near sharp edges.

Discussion

Being able to preserve or reconstruct sharp features is undoubtedly important. Methods that do not assume the availability of the original surface, however, must necessarily rely on heuristics to infer and restore sharp edges and corners in an aliased

model. Thus one of the main challenges in this context is the definition of a formal framework for sampling non-smooth surfaces. Although such a framework has been defined for smooth models [ABK98, BO03], the problem of dealing with tangential discontinuities remains open, even for the 2D case [DW01].

2.5 Error-driven Remeshing

Definitions

Error-driven remeshing amounts to generating a mesh which maximizes the tradeoff between complexity and accuracy. The complexity is expressed in terms of the number of mesh elements, while the geometric accuracy is measured relative to the input mesh and according to a predefined distortion error measure. The efficiency of a mesh is qualified by the error per element ratio (the smaller, the better). One usually wants to minimize the approximation error for a given budget of elements, or conversely, minimize the number of elements for a given error tolerance. Another challenging task consists of optimizing the efficiency tradeoff at multiple levels of detail.

Motivation

Efficient representation of complex shapes is of fundamental importance, in particular for applications dealing with digital models generated by laser scanning or isosurfacing of volume data. This is mainly due to the fact that the complexity of numerous algorithms is proportional to the number of mesh primitives. Examples of related applications are modeling, processing, simulation, storage or transmission. Even for most rendering algorithms, polygon count is still the main bottleneck. The main need is to automatically adapt the newly generated mesh to the local shape complexity.

Mesh simplification or refinement methods are obvious ways of generating efficient meshes. In this survey we will not pretend to survey the plethora of polygonal simplification techniques published in the last few years, and instead refer the interested reader to the comprehensive course notes and surveys [HG97, Gar00, Lue01, LRC+02, GGK02]. We complement these documents by focusing on techniques that proceed by optimization or by recovering a continuous model from the input mesh. This includes techniques specifically designed to exploit a shape's local planarity, symmetry and features in order to optimize its geometric representation. We focus in more detail on techniques that construct efficient meshes by extracting, up to a certain degree, the "semantical content" of the input shape.

Hoppe *et al.* [HRD+93] formulate the problem of efficient triangle remeshing as an optimization problem with an energy functional that directly measures the $\mathcal{L}^2$ error deviation from the final mesh to the original one. They showed that optimizing the number of vertices, as well as their geometry and connectivity, captures the

curvature variations and features of the original geometry. Despite a spring force restricting the anisotropy of the results and an approximate point-to-surface Euclidean $\mathcal{L}^2$ distance measure, this technique results in particularly efficient meshes. Alliez *et al.* [ALSS99] describe another optimization method which minimizes the volume between the simplified mesh and the input mesh using a gradient-based optimization algorithm and a finite-element interpolation model implicitly defined on meshes. The volume-based error metric is shown to accurately fit the geometric singularities on 3D meshes by aligning edges appropriately, without any distinction required between smooth and sharp areas.

Following previous work on feature remeshing (see Section 2.4), the remeshing technique introduced by Alliez *et al.* [ACSD+03] pushes the idea of aligning edges on features further by generalizing it to the entire surface. They generate a quad-dominant mesh that reflects the symmetries of the input shape by sampling the input shape with curves instead of the usual points. The algorithm consists of three main stages. The first stage recovers a continuous model from the input triangle mesh by estimating one 3D curvature tensor per vertex. The normal component of each tensor is then discarded and a 2D piecewise linear curvature tensor field is built after computing a discrete conformal parameterization. This field is then altered to obtain smoother principal curvature directions. The singularities of the tensor field (the umbilics) are also extracted. The second stage consists of resampling the original mesh in parameter space by building a network of lines of curvatures (a set of "streamlines" approximated by polylines) following the principal curvature directions. A user-prescribed approximation precision in conjunction with the estimated curvatures is used to define the local density of lines of curvatures at each point in parameter space during the integration of streamlines. The third stage deduces the vertices of the newly generated mesh by intersecting the lines of curvatures on anisotropic areas and by selecting a subset of the umbilics on isotropic areas (estimated to be spherical). The edges are obtained by straightening the lines of curvatures in-between the newly extracted vertices on anisotropic areas, and simply deduced from the Delaunay triangulation on isotropic areas. The final output is a polygon mesh with mostly elongated quadrilateral elements on anisotropic areas, and triangles on isotropic areas. Quads are placed mostly on regions with two (estimated) axis of symmetry, while triangles are used to either tile isotropic areas or to generate conforming convex polygonal elements. On flat areas the infinite spacing of streamlines will not produce any polygons, except for the sake of convex decomposition (see example Fig.17). This approach has been recently extended so as not to rely on any parameterization [MK04].

Although the edge sampling strategy described above increases the mesh efficiency by matching the conditions of optimality for the $\mathcal{L}^2$ metric in the limit, there is no guarantee of its efficiency at coarse scales. Moreover, this technique involves local estimation of curvatures, known to be difficult on discrete meshes. The estimator itself requires the definition of a scale which remains elusive (intuitively, the scale itself should depend on the approximation tolerance). These observations mo-

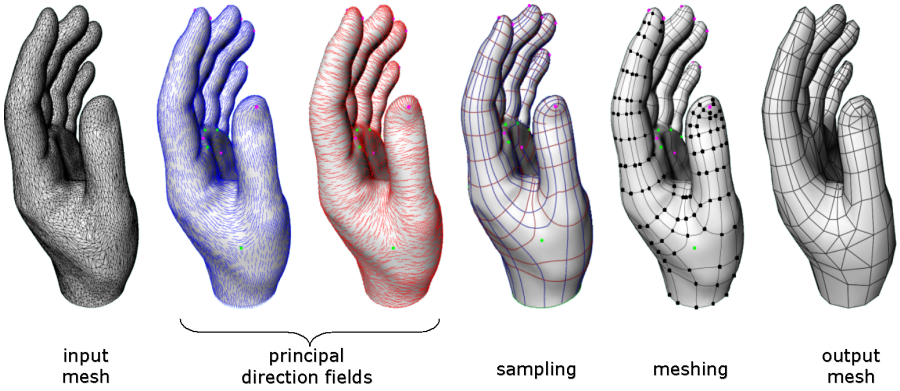

Fig. 17. Anisotropic remeshing: From an input triangulated geometry, the curvature tensor field is estimated, then smoothed, and its umbilics are deduced (colored dots). Lines of curvatures (following the principal directions) are then traced on the surface, with a local density guided by the principal curvatures, while usual point-sampling is used near umbilic points (spherical regions). The final mesh is extracted by subsampling, and conforming-edge insertion. The result is an anisotropic mesh, with elongated quads aligned to the original principal directions, and triangles in isotropic regions. Figure reproduced from [ACSD+03].

tivate an effi cient remeshing approach based exclusively on the approximation error. Thus Cohen-Steiner *et al.* [CSAD04] propose an error-driven clustering approach that does not resort to any estimation of differential quantities nor parameterization. Error-driven remeshing is now cast as a variational partitioning problem where a set of planes (so-called proxies) are iteratively optimized using Lloyd's heuristic to minimize a predefi ned approximation error (Fig.18). As in the original Lloyd algorithm, the key idea hinges on alternating partitioning and moving each representative to the centroid of its region. The partitioning is generated triangle by triangle using a region growing procedure driven by a global priority queue. The queue is sorted by the error between each new triangle candidate for expansion and the proxy (representative) of the corresponding region. The analog of the centroid in the metric space is now simply the best fi t proxy for each region. Closed forms for the errors between one triangle and one proxy, as well as for the best fi t proxy are given for regions consisting of a set of triangles, both for the $\mathcal{L}^2$ and $\mathcal{L}^{2,1}$ ($\mathcal{L}^2$ deviation of normals) error metric. A polygonal remeshing technique is proposed based on a discrete analog of a Voronoi diagram implemented with a two-pass partitioning algorithm over the input triangle mesh. The elements of the resulting polygonal meshes will then exhibit orientation and elongation guided by the minimization of the approximation error instead of being the result of a curvature estimation process as in [ACSD+03].

Discussion

In this section we narrowed our scope to the study of methods that best preserve the shape geometry during the remeshing stage of the geometry processing pipeline. Despite the considerable amount of work done for mesh approximation through error-

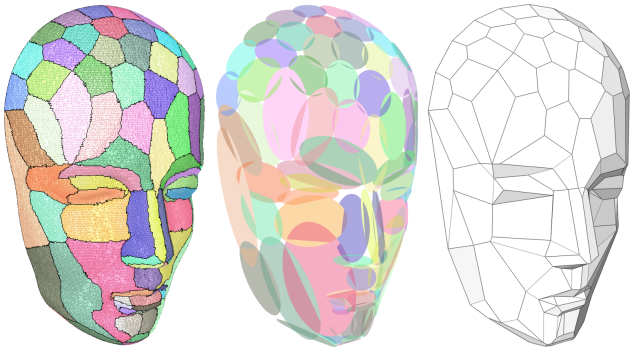

Fig. 18. Error-driven remeshing: Through repeated error-driven partitioning (left), a set of geometric proxies (represented as ellipses, center) is optimized. These proxies are then used to construct an approximating polygonal mesh (right). Figure reproduced from [CSAD04].

driven simplifi cation or refi nement, there is much less work on approximating shapes by using geometric analysis to guide the remeshing process.

Observations have shown that for sketching, artists implicitly exploit the symmetry of a shape when sketching strokes that best convey the desired model. Simple symmetric primitives such as planes, spheres, ellipses, saddles, cylinders and cones are also exploited by artists as basic components for modeling a shape. For reverse engineering, anisotropic remeshers such as [ACSD+03, MK04] help, to a certain degree, to automatically capturing the "semantical" structure of a measured shape by inferring a smooth model and extracting its main traits. The local symmetries and main traits of the shape should ideally be deduced from the elements of the mesh, facilitating structuring and analysis.

References

[ABK98] Nina Amenta, Marshall Bern, and Manolis Kamvysselis. A new voronoi-based surface reconstruction algorithm. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, pages 415–422, Jul 1998.

[ACSD+03] Pierre Alliez, David Cohen-Steiner, Olivier Devillers, Bruno Levy, and Mathieu Desbrun. Anisotropic polygonal remeshing. *ACM.Transactions on Graphics*, 22:485–493, 2003.

[AdVDI03] Pierre Alliez, Eric Colin de Verdiere, Olivier Devillers, and Martin Isenburg. Isotropic surface remeshing. In M.S. Kim, editor, *SMI '03: Proceedings of Shape Modeling International 2003*, pages 49–58, Los Alamitos, 2003. IEEE Computer Society.

[AFRS03] Marco Attene, Bianca Falcidieno, Jarek Rossignac, and Michela Spagnuolo. Edge-sharpener: recovering sharp features in triangulations of non-adaptively remeshed surfaces. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing 2003*, pages 62–69. ACM Press, 2003.

[AFSR03] M. Attene, B. Falcidieno, M. Spagnuolo, and J. Rossignac. Swingwrapper: Retiling triangle meshes for better edgebreaker compression. *Acm Transactions on Graphics*, 22(4):982–996, 2003.

[AFSW03] M. Attene, B. Falcidieno, M. Spagnuolo, and G. Wyvill. A mapping-independent primitive for the triangulation of parametric surfaces. *Graphical Models*, 65(5):260–273, 2003.

[AG03] Pierre Alliez and Craig Gotsman. Recent advances in compression of 3d meshes. In *Proceedings of the Symposium on Multiresolution in Geometric Modeling*, 2003.

[Ale99] M. Alexa. Merging polyhedral shapes with scattered features. In *Proceedings of the International Conference on Shape Modeling and Applications (SMI-99)*, pages 202–210, 1999.

[Ale02] Marc Alexa. Recent advances in mesh morphing. *Computer Graphics Forum*, 21(2):173–196, 2002.

[ALSS99] Pierre Alliez, Nathalie Laurent, Henri Sanson, and Francis Schmitt. Mesh approximation using a volume-based metric. In *Proceedings of the 7th Pacifi c Conference on Computer Graphics and Applications 1999*, pages 292–301, Los Alamitos, 1999. IEEE Computer Society.

[AMD02] P. Alliez, M. Meyer, and M. Desbrun. Interactive geometry remeshing. *Acm Transactions on Graphics*, 21(3):347–354, 2002.

[BHP97] Houman Borouchaki, Frederic Hecht, and J.Frey Pascal. Mesh gradation control. In *Proceedings of 6th International Meshing Roundtable*, pages 131–141. Sandia National Laboratories, 1997.

[BK04] Mario Botsch and Leif Kobbelt. A remeshing approach to multiresolution modeling. In R. Scopigno and D. Zorin, editors, *Proceedings of 2nd Eurographics Symposium on Geometry Processing*, pages 189–196. Eurographics, 2004.

[BO03] J. D. Boissonnat and S. Oudot. Provably good surface sampling and approximation. In *Proc. of Symp. on Geo. Processing*, pages 9–18, 2003.

[Che93] L. Paul Chew. Guaranteed-quality mesh generation for curved surfaces. In *Proceedings of the ninth annual symposium on Computational geometry*, pages 274–280. ACM Press, 1993.

[CSAD04] D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, 2004.

[DFG99] Qiang Du, Vance Faber, and Max Gunzburger. Centroidal Voronoi Tesselations: Applications and Algorithms. *SIAM review*, 41(4):637–676, 1999.

[DKG05] S. Dong, S. Kircher, and M. Garland. Harmonic functions for quadrilateral remeshing of arbitrary manifolds. *Computer Aided Geometric Design*, 2005. To appear.

[DW01] T.K. Dey and R. Wenger. Reconstructing curves with sharp corners. *Computational Geometry Theory and Applications*, 19:89–99, 2001.

[EG01] Herbert Edelsbrunner and Damrong Guoy. Sink-insertion for mesh improvement. In *Proceedings of the seventeenth annual symposium on Computational geometry*, pages 115–123. ACM Press, 2001.

[ERD+95] Matthias Eck, Tony De Rose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, pages 173–182, 1995.

[FB98] Pascal J. Frey and Houman Borouchaki. Geometric surface mesh optimization. *Computing and Visualization in Science*, 1:113–121, 1998.

[FH04] M.S. Floater and K. Hormann. *Surface Parameterization: a Tutorial and Survey*. Springer, 2004.

[Fre00] Pascal J. Frey. About surface remeshing. In *Proceedings of the 9th International Meshing Roundtable*, pages 123–136. Sandia National Laboratories, 2000.

[Gar00] M. Garland. Multiresolution modeling: Survey & future opportunities. In *Eurographics '99, State of the Art Report (STAR)*, pages 111–131. Eurographics, 2000.

[Ger79] Allen Gersho. Asymptotically optimal block quantization. *IEEE Transactions on Information Theory*, IT-25(4):373–380, July 1979.

[GGH02] X. Gu, S.J. Gortler, and H. Hoppe. Geometry images. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, pages 355–361, 2002.

[GGK02] C. Gotsman, S. Gumhold, and L. Kobbelt. *Simplifi cation and compression of 3D-meshes*. 2002.

[GGS03] C. Gotsman, X.F. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3d meshes. *Acm Transactions on Graphics*, 22(3):358–363, 2003.

[GVSS00] I. Guskov, K. Vidimce, W. Sweldens, and P. Schroeder. Normal meshes. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, pages 95–102, 2000.

[Har98] Erich Hartmann. A marching method for the triangulation of surfaces. *the Visual Computer*, 14(3):95–108, 1998.

[HG97] P. Heckbert and M. Garland. Survey of polygonal surface simplification algorithms, 1997.

[HG00] K. Hormann and G. Greiner. Quadrilateral remeshing. In *Proceedings of Vision, Modeling, and Visualization 2000*, pages 153–162, 2000.

[Hop96] H. Hoppe. Progressive meshes. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, pages 99–108, 1996.

[HRD+93] Hugues Hoppe, Tony De Rose, Tom Duchamp, John McDonald, and Werner Stuetzle. Mesh optimization. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, pages 19–26, 1993.

[JLSW02] Tao Ju, Frank Losasso, Scott Schaefer, and Joe Warren. Dual contouring of hermite data. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, pages 339–346, 2002.

[KB03] Leif Kobbelt and Mario Botsch. Feature sensitive mesh processing. In *SCCG '03: Proceedings of the 19th Spring Conference on Computer Graphics*, pages 17–22. ACM Press, 2003.

[KBB+00] Leif Kobbelt, Stefan Bischoff, Mario Botsch, Kolja Kahler, Christian Rossl, Robert Schneider, and Jens Vorsatz. Geometric modeling based on polygonal meshes. In *Euroraphics 2000 Tutorial*, 2000.

[KBSS01] L.P. Kobbelt, M. Botsch, U. Schwanecke, and H.P. Seidel. Feature sensitive surface extraction from volume data. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, pages 57–66, Aug 2001.

[KG03] A. Khodakovsky and I. Guskov. *Compression of Normal Meshes*. Springer-Verlag, 2003.

[KLS03] Andrei Khodakovsky, Nathan Litke, and P. Schroeder. Globally smooth parameterizations with low distortion. *ACM.Transactions on Graphics*, 22:350–357, 2003.

[KS04] V. Kraevoy and A. Sheffer. Cross-parameterization and compatible remeshing of 3d models. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, 2004.

[KSK00] Takashi Kanai, Hiromasa Suzuki, and Fumihiko Kimura. Metamorphosis of arbitrary triangular meshes. *Ieee Computer Graphics and Applications*, 20:62–75, 2000.

[KSS00] A. Khodakovsky, P. Schroeder, and W. Sweldens. Progressive geometry compression. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, pages 271–278, 2000.

[KVLS99] L. Kobbelt, J. Vorsatz, U. Labsik, and H.P. Seidel. A shrink wrapping approach to remeshing polygonal surfaces. *Computer Graphics Forum*, 18:119–130, 1999.

[LC87] W.E. Lorensen and H.E. Cline. Marching cubes: a high resolution 3d surface reconstruction algorithm. *Computer Graphics*, 21:163–169, 1987.

[LCLC03] Jian Liang Lin, Jung Hong Chuang, Cheng Chung Lin, and Chih Chun Chen. Consistent parametrization by quinary subdivision for remeshing and mesh metamorphosis. In *GRAPHITE '03: Proceedings of the 1th International Conference on Computer Graphics and Interactive Techniques in Austalasia and South East Asia 2003*, pages 151–158. ACM Press, 2003.

[LDSS99] Aaron Lee, David Dobkin, Wim Sweldens, and Peter Schroeder. Multiresolution mesh morphing. In *Siggraph 1999, Computer Graphics Proceedings*, pages 343–350, 1999.

[Llo82] S. Lloyd. Least square quantization in PCM. *IEEE Trans. Inform. Theory*, 28:129–137, 1982.

[LPZE96] Michael Lindenbaum, Moshe Porat, Yehoshua Y. Zeevi, and Yuval Eldar. The farthest point strategy for progressive image sampling, 1996.

[LRC+02] D. Luebke, M. Reddy, J. Cohen, A. Varshney, B. Watson, and R. Huebner. *Level of Detail for 3D Graphics*. Morgan-Kaufmann, San Francisco, 2002.

[LSS+98] A.W.F. Lee, W. Sweldens, P. Schroeder, L. Cowsar, and D. Dobkin. Maps: Multiresolution adaptive parameterization of surfaces. *Computer Graphics*, 32:95–104, 1998.

[Lue01] David Luebke. A developer's survey of polygonal simplification algorithms. *Ieee Computer Graphics and Applications*, 2001.

[MD03] Carsten Moenning and Neil A. Dodgson. Fast marching farthest point sampling. Technical Report UCAM-CL-TR-562, University of Cambridge, Computer Laboratory, 2003.

[Mil04] Gary L. Miller. A time efficient delaunay refinement algorithm. In *Proceedings of the fi fteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 400–409. Society for Industrial and Applied Mathematics, 2004.

[MK04] Martin Marinov and Leif Kobbelt. Direct anisotropic quad-dominant remeshing. In *Proceedings of the 12th Pacifi c Conference on Computer Graphics and Applications*, pages 207–216, 2004.

[MKFC01] T. Michikawa, T. Kanai, M. Fujita, and H. Chiyokura. Multiresolution interpolation meshes. In *Proceedings of the 9th Pacifi c Conference on Computer Graphics and Applications 2001*, pages 60–69, Los Alamitos, 2001. IEEE Computer Society.

[ODJ04] Victor Ostromoukhov, Charles Donohue, and Pierre Marc Jodoin. Fast hierarchical importance sampling with blue noise properties new york, ny, usa. *ACM.Transactions on Graphics*, 23, Aug 2004.

[Ost01] Victor Ostromoukhov. A Simple and Efficient Error-Diffusion Algorithm. In *Proceedings of SIGGRAPH*, pages 567–572, 2001.

[PA02] F. Payan and M. Antonini. 3d mesh wavelet coding using efficient model-based bit allocation. In *Proceedings of the 1st International Symposium on 3D Data Processing Visualization and Transmission 2002*, pages 391–394, 2002.

[PB01] Philippe P. Pebay and Timothy J. Baker. A comparison of triangle quality measures. In *Proceedings, 10th International Meshing Roundtable*, pages 327–340, 2001.

[PC03] Gabriel Peyre and Laurent Cohen. Geodesic remeshing using front propagation. In *Proceedings of 2nd IEEE Workshop on Variational, Geometric and Level Set Methods in Computer Vision 2003*, pages 33–40, Los Alamitos, 2003. IEEE Computer Society.

[PC04] Gabriel Peyré and Laurent Cohen. Surface Segmentation Using Geodesic Centroidal Tesselation. In *Proceedings of 2nd International Symposium on 3D Data Processing, Visualization, and Transmission*, pages 995–1002, 2004.

[PH03] Emil Praun and Hugues Hoppe. Spherical parametrization and remeshing. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, pages 340–349, 2003.

[PSS01] Emil Praun, Wim Sweldens, and Peter Schroeder. Consistent mesh parameterizations. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, pages 179–184, 2001.

[RCG+01] C. Rocchini, P. Cignoni, F. Ganovelli, C. Montani, P. Pingi, and R. Scopigno. Marching intersections: an efficient resampling algorithm for surface management. In *Proceedings of the International Conference on Shape Modeling and Applications*, pages 296–305, 2001.

[Rup95] Jim Ruppert. A delaunay refinement algorithm for quality 2-dimensional mesh generation. *Journal of Algorithms*, 18(3):548–585, 1995.

[RVSS00] A. Rassineux, P. Villon, J.M. Savignat, and O. Stab. Surface remeshing by local hermite diffuse interpolation. *International Journal for Numerical Methods in Engineering*, 49:31–49, 2000.

[SAG03] Vitaly Surazhsky, Pierre Alliez, and Craig Gotsman. Isotropic remeshing of surfaces: a local parameterization approach. In *Proceedings of 12th International Meshing Roundtable*, pages 215–224. Sandia National Laboratories, 2003.

[SAPH04] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe. Inter-surface mapping. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, 2004.

[Sch98] P. Schröder. Subdivision for modeling and animation, 1998.

[Set99] J. Sethian. *Level Sets Methods and Fast Marching Methods*. Cambridge University Press, 2nd edition, 1999.

[SG03] V. Surazhsky and C. Gotsman. Explicit surface remeshing. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing 2003*, pages 20–30. ACM Press, 2003.

[SGSH02] P. Sander, S. Gortler, J. Snyder, and H. Hoppe. Signal-specialized parametrization. In *EGWR '02: 13th Eurographics Workshop on Rendering 2002*. Eurographics, 2002.

[She02] Jonathan Richard Shewchuk. What is a good linear element? interpolation, conditioning, and quality measures. In *Proceedings of 11th International Meshing Roundtable*, 2002.

[SL98] K. Shimada and J. Liao. Quadrilateral Meshing with Directionality Control through the Packing of Square Cells. In *7th Intl. Meshing Roundtable*, pages 61–76, oct 1998.

[SRK03] A. Szymczak, J. Rossignac, and D. King. Piecewise regular meshes: Construction and compression. *Graphical.models.*, 2003.

[SS01] W. Sweldens and P. Schroeder, editors. *Digital Geometry Processing*. SIGGRAPH Conference course notes, 2001.

[SSG03] Oren Sifri, Alla Sheffer, and Craig Gotsman. Geodesic-based surface remeshing. In *Proceedings of 12th International Meshing Roundtable*, pages 189–199. Sandia National Laboratories, 2003.

[SWG+03] P. Sander, Z. Wood, S. Gortler, J. Snyder, and H. Hoppe. Multi-chart geometry images. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing 2003*, pages 246–255. ACM Press, 2003.

[Tau00] G. Taubin. Geometric signal processing on polygonal meshes. In *Euroraphics 2000: State of the Art Report (STAR)*. Eurographics, 2000.

[TOC98] J.R. Tristano, S.J. Owen, and S.A. Canann. Advancing Front Surface Mesh Generation in Parametric Space Using a Riemannian Surface Definition. In *Proceedings of the 7th Int. Meshing Roundtable*, 1998.

[Tur92] Greg Turk. Re-tiling polygonal surfaces. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, pages 55–64, 1992.

[VC04] Sebastien Valette and Jean Marc Chassery. Approximated centroidal voronoi diagrams for uniform polygonal mesh coarsening. *Computer Graphics Forum*, 2004.

[VPBM01] Alex Vlachos, Jorg Peters, Chas Boyd, and Jason L. Mitchell. Curved PN triangles. In *Symposium on Interactive 3D Graphics*, pages 159–166, 2001.

[VRKS01] Jens Vorsatz, Christian Rossl, Leif Kobbelt, and Hans Peter Seidel. Feature sensitive remeshing. *Computer Graphics Forum*, pages 393–401, 2001.

[VRS03] J. Vorsatz, C. Rossl, and H.P. Seidel. Dynamic remeshing and applications. In *SMA '03: Proceedings of the 3th ACM Symposium on Solid Modeling and Applications 2003*, pages 167–175. ACM Press, 2003.

[WM96] D.J. Walton and D.S. Meek. A triangular G^1 patch from boundary curves. *Computer Aided Design*, 28(2):113–123, 1996.

[YS04] Soji Yamakawa and Kenji Shimada. Triangular/quadrilateral remeshing of an arbitrary polygonal surface via packing bubbles. In *Proceedings of Geometric Modeling and Processing 2004*, Los Alamitos, 2004. IEEE Computer Society.

[ZS00] D. Zorin and P. Schroeder. Subdivision for modeling and animation. *Computer graphics proceedings, annual conference series: SIGGRAPH conference proceedings*, 2000.