An iterative approach for global triangular mesh regularization

Vincent Vidal, Christian Wolf, Florent Dupont, Guillaume Lavoué

An iterative approach for global triangular mesh regularization

Vincent Vidal Christian Wolf Florent Dupont
Guillaume Lavoué

Technical Report LIRIS

Laboratoire d'informatique en images et systèmes d'information - UMR 5205
Université de Lyon
INSA-Lyon, Bât. J.Verne; 20, Av. Albert Einstein
69621 Villeurbanne cedex, France
Tel.: 0033 4 72 43 63 65 Fax.: 0033 4 72 43 71 17
Email: `forename.surname@liris.cnrs.fr`
Web: `http://liris.cnrs.fr/forename.surname`

July 23, 2009

Abstract

This paper presents a global mesh optimization framework for 3D triangular meshes of arbitrary topology. The mesh optimization task is formulated as an energy minimization problem including data attached terms measuring the fidelity to the original mesh as well as a shape potential favoring high quality triangles. Since the best solution for vertex relocation is strongly related to the mesh connectivity, our approach iteratively modifies this connectivity (edge and vertex addition/removal) as well as the vertex positions. Good solutions for the energy function minimization are obtained by a discrete graph cut algorithm examining global combinations of local candidates. Results on various 3D meshes compare favorably to recent state-of-the-art algorithms regarding the trade-off between triangle shape improvement and surface fidelity. Applications of this work mainly consist in regularizing meshes for numerical simulations, for improving mesh rendering or for improving the geometric prediction in mesh compression techniques.

1 Introduction

Nowadays, 3D triangular meshes are commonly used in many fields, as the hundreds of thousands of existing 3D triangular models can attest. They are of interest in visual effects, video games, scientific visualization, 3D animation and medical surgery simulation based on finite element methods, to say a few. Most of the existing triangular meshes are of unsatisfying quality because of their inappropriate vertex sampling, which is responsible for inequilateral triangles and irregular connectivity. The origin of this irregular sampling may be due to the scanning device, to 3D interactive solid modelling software or to simplification algorithms.

This poor quality causes instability and divergence of various mesh processing applications [1]. Basically, general triangle quality criteria include triangle angles (min and max), area and aspect ratio. Precisely, interpolation and conditioning quality measures have been proposed to deal with shape, size and conditioning of mesh elements. Such element quality criteria implicitly suppose that the ideal mesh element is nearly the same over the whole mesh. An ideal mesh should therefore have a nearly isotropic vertex sampling such that triangle features vary slowly over the domain. This sampling would ideally be curvature-adapted (or/and adapted to numerical simulation accuracy), since curved regions need more primitives in order to better represent the initial surface.

In that context, several remeshing techniques have been introduced in the literature, they consist of improving some quality requirements under constraint conditions, where constraints are soft, not hard. The targeted goals vary according to the application [2]. *Simplification techniques* tend to preserve the overall shape of the mesh while removing as many triangles as possible. *Mesh smoothing methods* consist in removing high frequency noise so as to fair the mesh. Finally, *Mesh optimization* aims at improving the mesh quality, i.e. the regularity of the sampling and of the connectivity, which is precisely the objective of our algorithm.

1.1 Related work

Two classes of approaches have been identified in the vast mesh optimization literature. The first group consists of methods which completely discard the initial geometry and connectivity so as to sample new points on the surface. *Variational partitioning frameworks* [3, 4, 5] are based on vertex/triangle clustering and are often used to produce a coarser mesh with a high approximation quality. This coarse mesh is then usually refined using subdivision techniques. *Parameterization methods* [6] optimize 2D patches instead of a

3D surface, but they generally suffer either from distortion produced by the global parameterization, or from patch boundary post-processing resulting from local parameterization. *Semi-regular remeshing* [7] uses an initial coarse mesh partition and treats each patch separately using subdivision rules. This produces a few irregular vertices (those of the coarse mesh), well-shaped triangles, and a small geometric error. However, it is sensitive to the patch structure and the resulting vertex sampling is difficult to control. *Geodesic front propagation techniques* [8, 9] consider geodesic equidistant curves over the surface and allow to get well-shaped triangles (vertices can be distributed according to the local curvature). However, some pre-processing steps are needed to avoid artifacts when the curve topology is complex. All these methods give an overall good triangle shape, but they suffer from a lack of geometric error control when the new points are sampled on the surface.

The second group of methods works directly on the initial mesh simplices (vertices, edges and resulting triangles), which allows a better control of the geometric fidelity to the original 3D surface. *Local approaches* to mesh optimization consist of using a set of local legal moves (e.g. towards barycenters or angle-bisectors) and connectivity modifications (topological operators) to decrease an energy function. Such an approach may lead to a local minima configuration, especially if used with a greedy optimization (e.g. gradient descent). Surazhsky and Gotsman [10] use local operations such as edge-collapse, edge-split, edge-flip and vertex relocation to regularize the mesh connectivity. Surazhsky et al. [10, 11] apply area-based smoothing to control both triangle quality and vertex sampling over the mesh. To achieve a precise isotropic vertex placement, Surazhsky et al. [11] use a centroidal Voronoi tessellation. *Global approaches* to mesh optimization attempt to resolve the optimization problem in a global way, most of the time solving a sparse linear system [12, 13] or using least squares approximation [14]. The main idea in the global approaches using a so-called *Laplacian global operator* [12, 13, 14], is to infinitely apply a Laplacian operator such that applying it one more time will not change the current vertex positions. That allows a direct formulation as a linear system. Then other constraints are added to take into account invariant vertex positions or to avoid the shrinking effect due to Laplacian smoothing. All these techniques depend on the initial sampling and connectivity, and the triangle shape may therefore be difficult to improve in case of vertex configurations that are not adapted to the surface.

1.2 Our approach and contribution

The mesh optimization problem in this paper can be seen as follow: given an initial mesh M, we try to find another mesh M' with improved triangle qual-

ity (a high amount of almost equilateral triangles) as well as high fidelity to the initial 3D surface (same topology and a small geometric error). The key idea of the proposed algorithm is to globally optimize this trade-off between quality improvement and surface fidelity, using discrete optimization techniques. The optimization is separated into two different iteratively executed steps:

- Finding new vertex positions which globally minimize a well-chosen energy while keeping sharp features.
- Smoothly improving the mesh connectivity.

During the remeshing process, corners and sharp edges are preserved (see section 4.2). The mesh connectivity is smoothly improved using local moves. After each mesh connectivity modification, relaxation of the vertex positions allows to speed up convergence towards better global configurations. The remeshing pipeline is illustrated in figure 1.

Our contributions are:

- An optimization algorithm which searches the globally optimal combination of the locally proposed vertex positions based on the calculation of the minimum *st*-cut/maximum flow in a graph.
- A relaxation technique of vertex positions for better global convergence.
- A simple and fast detection method for sharp features based on a two thresholds hysteresis.

This paper is organized as follows: in section 2, we present our global energy minimization framework for optimizing mesh vertices and connectivity as well as the global optimization algorithm. Section 3 deals with connectivity optimization and convergence improvement through relaxation. Section 4 integrates the global method and explains how the method copes with sharp features. In section 5, we give experimental results and compare them with other recent algorithms from the state of the art. In section 6, we conclude and discuss some future work.

2 Global energy minimization problem

In this work the initial mesh M is copied and kept as reference geometry during the remeshing process. The optimized mesh M' thus starts with the same vertices and connectivity as M. The two 3D triangular meshes we deal with, i.e. the original as well as the optimized one, are considered as a single

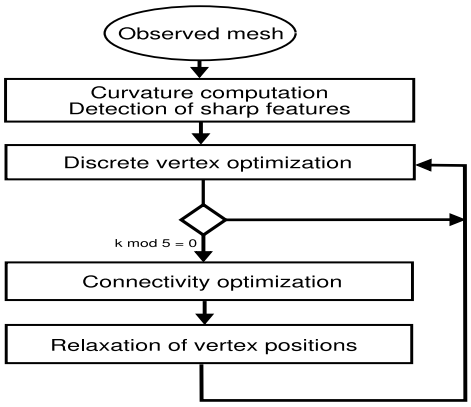


Figure 1: The remeshing pipeline: at every iteration, new vertex positions are chosen using a good global energy minimum approximation; every five iterations the mesh connectivity is modified and relaxed to favor better global vertex configurations

undirected graph $\mathcal{G} = \{V, E\}$, where V is a set of vertices and E is a set of edges. The vertices V are partitioned into a set X of vertices of the new (optimized) mesh, and a set y of positions of the original mesh. All variables are vector valued continuous variables taking values from $\Lambda = \mathbb{R}^3$. y remains constant during the remeshing process, but X may evolve (addition/removal of elements) since the initial sampling is not necessarily adapted to the surface being represented. x thus denotes the current values of the whole set of remeshed model vertices X. Each vertex of the remeshed model is linked to a position of the original mesh (the closest one). This link is usefull to keep track of the geometric distance between a vertex of the optimized mesh and the surface represented by the original mesh (cf. figure 2).

In this paper the mesh optimization is presented as an energy minimization problem, where the energy is expressed in terms of a scalar combination of positive energy potentials which measure the criteria (quality, surface approximation) that we aim to improve. Potentials may depend on one, two, or three vertices of the optimized mesh M' and on vertices of the original mesh M (reference geometry). In the following, S will denote the set of simplices of the remeshed model, which are in this paper limited to points, edges and triangles (0,1 and 2-simplices).

The specific form of our energy function $U(x, y)$ is defined through feature functions (each of them expressing the capacity to improve a property):

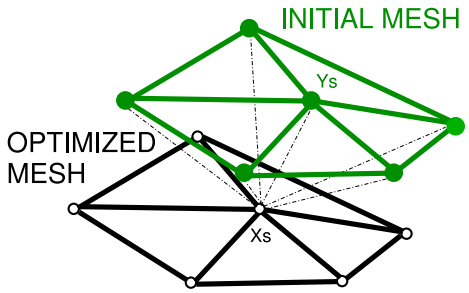


Figure 2: Representation of the graph used for constructing the remeshed model: each vertex of the optimized mesh x_s is linked to its current closest initial mesh vertex y_s and has access to whole initial mesh if needed.

$$\begin{aligned} U(x,y) &= \sum_{(r,s,t)\in\mathcal{S}_2} \lambda_{shape}\psi_{shape}(x_r,x_s,x_t) \\ &+ \sum_{s\in X} \lambda_{data}\psi_{data}(x_s,y) \end{aligned} \tag{1}$$

where $\mathcal{S}_2$ is the set of triangles (2-simplices) of the remeshed model. λ_{shape} and λ_{data} are positive scalars.

$U(x,y)$ evaluates the global configuration of a mesh, and it depends on the vertex position and on the mesh connectivity. Therefore at each vertex relocation or at each mesh connectivity modification, $U(x,y)$ may vary.

2.1 The feature functions

Our model contains two different kinds of positive feature functions: shape functions encoding the prior knowledge on the quality of the mesh connectivity as well data attached functions encoding the approximation quality of the new surface. The role of the former is to favor equilateral mesh triangles, therefore they are calculated on triangles:

$$\psi_{shape}(x_r,x_s,x_t) = \frac{R(x_r,x_s,x_t)}{min(||x_r-x_s||,||x_r-x_t||,||x_s-x_t||)} \tag{2}$$

where R denotes the cicumradius associated to the triangle (x_r,x_s,x_t) and $||.||$ is the usual Euclidean norm. Note that these feature functions do not depend on the initial vertices y and that they are scale invariant.

The data attached feature functions are calculated between a new vertex x_s and the (potentially) whole set of initial vertices:

$$\psi_{data}(x_s, y) = F(x_s, y) \tag{3}$$

where $F(x_s, y)$ is the square of the shortest distance between x_s and a patch of a continuous surface approximated by a subset of the initial vertices y close to the label x_s (geometric distance).

The function F is hard to calculate but may be approximated up to second order [15, 16]:

$$F(x_s, y) \approx \frac{d}{d + |\varrho_{min}|}\tilde{x}_{s1}^2 + \frac{d}{d + |\varrho_{max}|}\tilde{x}_{s2}^2 + \tilde{x}_{s3}^2 \tag{4}$$

where $\tilde{x}_s$ is the point x_s expressed in coordinates of the Frenet frame on a surface point $y_{s'}$ closest to x_s. The Frenet frame is defined through the following coordinate axis: the surface normal at point $y_{s'}$ (z-axis) as well as the two orthogonal vectors defining the principal curvature directions (x and y-axis). ϱ_{min} and ϱ_{max} are the local principal curvature radii at $y_{s'}$, and $d > 0$ is the z-coordinate (in the Frenet frame) of the reference point $p = [\ 0\ 0\ d\ ]^T$ used for the Taylor expansion of the quadratic distance, which we set in our case to $\min(\varepsilon,\ | < x_s - y_{s'}, \vec{z} > |)$, where ε is a small constant (usually 0.01) and $<,>$ denotes the usual scalar product. This function is scale invariant if the curvatures and the coordinates are scaled. We experimentally found that geodesic-based principal curvature extraction and normal computation using the one-ring triangle normals weighted by triangle area give the best results.

The location of the point $y_{s'}$ is efficiently found by local search. For each vertex x_s of the remeshed model, the current closest initial point $y_{s'} \in y$ to x_s is stored (at the beginning it is initialized to the initial vertex y_s position), and at each iteration it is checked whether one of the neighbors of $y_{s'}$ is closer to x_s and the reference updated accordingly.

2.2 Iterative vertex repositioning

The goal of this step is to find out a good estimate for the new vertex positions, given the initial variables y, and the current mesh connectivity.

Unfortunately, the function $U(x, y)$ is not convex and standard gradient descent methods will most likely return a suboptimal solution. In addition, a least-square or linear system approach will return an oversmooth mesh, even if some constraints are added. This suggests a discrete approach for obtaining high fidelity meshes. Our work benefits from recent advances in optimization theory for discrete Markov Random Fields (MRFs) [17] by transforming the continuous problem into a discrete problem, similar to the technique proposed

for optical flow by Lempitsky et al. [18]. However, instead of employing a global discrete optimizer to merge several solutions obtained by existing techniques applied with different parameters, in our case the global discrete optimizer takes decisions on candidates calculated at each step. At each iteration and for each vertex of the remeshed model, a new candidate position is proposed and the optimal decision for the whole set of vertices is calculated, i.e. the decision minimizing (1).

2.3 Candidate proposals

Each candidate position is forced to stay within a small freedom sphere to prevent too much geometric error and to avoid creating a fold-over. The more the region around a vertex x_s is curved, the smaller its freedom radius $\sigma_{freedom}$ is forced to be. Moreover, like the stepwidth in gradient descent, this radius decreases with time to avoid big moves at the end of the optimization process. Similarly to simulated annealing techniques [19], we introduce a temperature parameter T which decreases at each iteration. The radius $\sigma_{freedom}$ is related to this temperature T through a sigmoidal function, as well as to the minimum geodesic radius of the x_s one-ring, and to the local maximal curvature radius measured on the initial mesh:

$$\sigma_{freedom}(x_s, x_{N_s}, y) \quad = \quad \frac{0.5}{1+e^{-T}}.||x_s, x_{N_s}||_g.min(1, \rho_{max}) \tag{5}$$

where ρ_{max} is the maximum curvature radius at y'_s. $0.5||x_s, x_{N_s}||_g$ is half the length of minimum geodesic radius from the considered vertex, it allows to globally combine local candidates without creating a fold-over. For the cooling schedule we used the suggestions in [20] (page 356), setting the temperature T to $T^{(i)} = T^{(1)} \cdot K^{i-1}$ where K is a constant controlling the speed of the cooling process and i denotes the current iteration. Since the initial mesh is normalized before processing (and unnormalized at the end), the maximal mesh dimension does not appear in this equation for curvature radius normalization purpose.

A candidate must decrease the energy, which can be computed quickly from the vertex x_s and its one-ring neighborhood N_s:

$$\Delta = U\Big(\big(x \setminus \{x_s\}\big) \cup \{x_s^{new}\}, y\Big) - U(x, y) = U(x_s^{new}, N_s, y) - U(x_s, N_s, y) \tag{6}$$

Where x_s^{new} is the new candidate position and $U(x_s, N_s, y)$ is the so-called *local evidence* given as:

$$
\begin{aligned}
U(x_s, x_{N_s}, y) &= \sum_{(r,s,t)\in\mathcal{S}_2 : r,t\in N_s} \lambda_{shape}\psi_{shape}(x_r, x_s, x_t) \\
&+ \lambda_{data}\psi_{data}(x_s, y_{s'}, y_{N_{s'}})
\end{aligned}
\tag{7}
$$

Our global optimization technique, which will be given below, admits a maximum number of two candidate positions, i.e. one new candidate position for each vertex additionally to the current vertex position. Furthermore, the convergence of the algorithm depends on the quality of these new candidate positions, so we employ different techniques to calculate them; then instead of choosing the best candidate among these methods, we ranked them by empirical convergence quality and for each vertex x_s we choose the first candidate position which satisfies the constraints given above (in freedom sphere and $\Delta < 0$, cf. equations (5) and (6)). If no candidate is found, the old position is kept. The techniques are, in decreasing convergence quality, angle-based smoothing, Laplacian smoothing, and guided random candidates (cf. algorithm 1).

The angle-based and Laplacian candidates' displacement vectors are multiplied by a small constant $\gamma = 0.06$, which allows to accept these candidates more often, especially in some specific configurations where the vertex is close to its one-ring boundary and the freedom sphere is therefore rather small. This process also reduces the distance between the current position and the proposed one, and thus the geometric error introduced in curved regions.

The smoothing candidates are projected on the tangent plane. The guided random candidates are circularly chosen on the tangent plane at the beginning of the optimization process, and then along the gradient direction of the energy function (1) for the last few "greedy" iterations. That allows us to stick more to the initial surface once the triangle shapes have been well-optimized. Special care is taken for label positions on sharp edges, for which candidates along the sharp edge are preferred. Vertices on corners remain unchanged.

2.4 Candidate decisions

The global decision (i.e. keeping the current position or choosing the new candidate) on the whole set of vertices is taken by Kolmogorov et al.'s graph cut technique [17]. This involves constructing an *st*-graph representing the energy function (1) such that the minimum cut/maximum flow on this graph will give the solution which globally minimizes the energy. Since quality is calculated on triangles, we use the graph construction for the class $\mathcal{F}^3$ of

third order functionals given in [17]. This optimization algorithm is able to optimize third order functions on binary random variables, i.e. variables taking values in $\{0, 1\}$. For that purpose, it is necessary to associate a binary labelling to $\{x_s,\, x_s^{new}\}$, for instance the label *zero* could denote the current vertex position x_s and *one* the proposed position x_s^{new}.

Furthermore, the energy function needs to satisfy the so-called submodularity criterion [17]. This criterion requires that, taking one of the vertices of a given triangle (x_r, x_s, x_t) and fixing the decision to either 0 or 1, the projection onto the two other vertices satisfies the following constraint (without loss of generality we suppose that x_r has been fixed):

$$\begin{aligned}\psi_{shape}(x_r, 0, 0) + \psi_{shape}(x_r, 1, 1) \quad &< \\ \psi_{shape}(x_r, 0, 1) + \psi_{shape}(x_r, 1, 0) &\end{aligned}$$

In our case, the energy function is generally not *graph representable*, i.e. this criterion is not satisfied for every triangle. We ressort to a common technique, namely truncating the energy function: the shape of the non-submodular triangles is not optimized. We can realistically assume that the set of submodular triangles changes with each iteration such that most of the triangles get optimized several times during the optimization process.

The number of submodular triangles depends on the mapping from $\{0, 1\}$ to $\{x_s,\, x_s^{new}\}$. Indeed, the turnout can be increased by choosing a different mapping for each vertex. Two mappings are possible for a single vertex and $2^3 = 8$ mappings are possible for each triangle, but since vertices are shared between triangles, calculating the optimal mapping is a non trivial problem in itself. A solution has been proposed by Schlesinger for energy functions with 2^{nd} order terms [22] using graph cuts on a graph derived from the original graph of the problem. Since our function involves 3^{rd} order terms, we opted for a greedy technique assigning the mappings in a single pass through the mesh. All possible 8 mappings are checked for the first triangle, and the combinations on the not yet fixed vertices are checked for subsequent triangles.

3 Iterative mesh connectivity optimization

Our goal is to optimize the remeshed model connectivity, which will allow us to significantly improve the final results (greater min and smaller max angles) as well as to speed up convergence of the vertex repositioning algorithm. In this paper, the purpose is neither to generate fully regular meshes (those which are a topologically part of a torus) nor highly regular meshes, since

such a processing is very time consuming [21] and not needed for high quality isotropic remeshing.

Our mesh connectivity optimization is divided into two parts:

1. Smooth local connectivity modifications
2. Relaxation of vertex positions

3.1 Smooth local changes

We perform smooth edge-flips/splits/collapses plus smooth vertex-splits (in that order), i.e. those which improve the quality of the triangulation. A smooth local operation consists in keeping the new local configuration obtained after applying a local operator, only if it has improved a given quality criterion. The chosen quality criteria are the minimum angle involved in the local operation for an edge-flip (six angles) or an edge-collapse, the harmonic mean of two scale invariant triangle quality measures (interpolation quality and conditioning quality from [1]) for an edge-split, and the global improvement of vertex valence for a vertex-split.

This is a complex optimization problem plagued by suboptimal local minima since there exist configurations with inequilateral triangles for which no direct operation can improve the minimum angle. To decrease the number of edge and vertex candidates to test for edge-flip, edge-collapse, and vertex-split, we just select those for which the vertex valence can be directly improved. Edge-collapse is restricted to short edges (smaller than a threshold e_{min}) and to flat regions (dihedral angle below 10^o) to avoid introducing too much geometric error. Similarly edge-split is applied only if the edge length is greater than a threshold e_{max}. e_{min} and e_{max} depend on the local maximal curvature and on the size of the mesh bounding box. We use a priority queue based method for the operations on the edges to execute them in the following order: edge-flips/splits/collapses and also to select the operation which best optimizes the associated quality criterion. Afterwards, vertex-splits are applied without any specific order (because their goal is to improve the global vertex valence).

Note that this set of local operators is not minimal, because the vertex-split can be done using edge-split and edge-flip. However, it should go through intermediate unwanted vertex configurations and that is forbidden for the time being. In fact a more complex combinatorial framework could be used to authorize intermediate unwanted vertex configurations for better final results.

3.2 Relaxation of vertex positions

It consists of applying a modified smoothing operator to the vertex positions not adjacent to a sharp edge. More precisely, the angle-based smoothing displacement vector is locally computed, then projected onto the local tangent plane in order to avoid shrinkage. If angle-based smoothing would create a fold-over, the Laplacian smoothing displacement vector is computed and projected. If this creates a fold-over, the position remains unchanged.

Relaxation improves the triangle shape and the local configurations. It thus improves the algorithm's convergence. Mesh connectivity optimization is not needed for all vertex repositioning iterations. One pass for only every five vertex repositioning iterations allows vertices to be correctly relocated before modifying their adjacency relationships.

4 Outline of the method

The global scheme of our method for optimizing 2-manifold meshes of arbitrary genus is presented in figure 1 and in algorithm 1. Even if it does not explicitly appear in the presented algorithm, sharp features of the original mesh are detected and kept at their place during all remeshing iterations.

4.1 Description

The general global optimization algorithm consists of using algorithm 1 twice: the first call is for the non-greedy optimization part when local moves are longer (high initial temperature); the second call for the greedy optimization part (gradient descent), for which smoothing candidates and mesh connectivity optimization are discarded. The first part can be seen as a relaxation part and the second part as the final geometry fitting. The first part attempts to regularize the mesh connectivity and the vertex position. The second part is needed, because the majority of proposed candidates are on local tangent planes, which may introduce a small geometric error. To limit this geometric error, a few greedy iterations at the end of the non-greedy optimization step are sufficient.

4.2 Robust sharp feature detection

Sharp features are the mesh corners and sharp edges where a vertex is considered as a corner if it has at least three adjacent sharp edges. Mesh edges are sharp if the dihedral angle between their two adjacent triangles (2-manifold meshes) is greater than a user-specified threshold θ_{true}.

Input: y (initial vertex positions), $T^{(1)}$ (start temperature), C (cooling speed), k_{max} (number of iterations), greedy (boolean)
Output: x (new vertex positions)

$x \leftarrow y$
$T \leftarrow T^{(1)}$
for $k \leftarrow 0$ **to** $k_{max} - 1$ **do**
  if *not greedy and* k *mod* $5 = 0$ **then**
    Regularize connectivity
    Relax vertex positions
  end
  for $s \in X$ **do**
    $\sigma_{freedom} \leftarrow \frac{0.5}{1 + e^{-T}} . ||x_s, x_{N_s}||_g . min(1, \rho_{max})$
    $x_s^{new} \leftarrow$ first with $\Delta < 0$ and in sphere, out of the following: { angle based smoothing; Laplacian smoothing; guided random }
  end
  $x \leftarrow$ minimum-cut/maximum flow (x, x^{new}, y)
  $T \leftarrow T \cdot C$
end

Algorithm 1: The whole method, including the continuous-discrete solution to vertex relocation. The graph cut algorithm takes the global decision for the whole set of vertices, considering for each vertex of the remeshed model the current positions x_s and a new candidate position x_s^{new}

During the vertex optimization step, corners are kept unchanged and vertices along one or two sharp edges can move only along these edges. During the connectivity optimization step, sharp edges cannot be flipped and edges having at least one sharp neighboring edge cannot be collapsed.

A single threshold leads to noisy decisions, more robust detection can be achieved with two thresholds defining an interval of indetermination, $[\theta_{false}, \theta_{true}]$ which allows us to regularize the decisions for each edge:

$$\text{edge} \leftarrow \begin{cases} \text{not sharp} & \text{if} & \theta_d < \theta_{false} \\ \text{sharp} & \text{if} & \theta_d > \theta_{true} \\ \text{sharp} & \text{if} & \theta_{false} \leq \theta_d \leq \theta_{true} \\ & & \text{and } \exists \text{ at least 2 neighboring} \\ & & \text{edges having } \theta_d > \theta_{true} \\ \text{not sharp} & \text{else} & \end{cases} \tag{8}$$

We experimentally fixed $\theta_{false} = 20^o$ and $\theta_{true} = 35^o$.

5 Experimental results

In order to demonstrate the efficiency of our method, we applied it to several mesh models, with unadapted sampling and very irregular connectivity, which contain smooth parts and sharp features. Some visual results are given in figures 3, 4 and 5.

We experimentally fixed the quality/fidelity trade-off parameters to $\lambda_{shape} = 1$ and $\lambda_{data} = 10^7$. We use 150 non-greedy iterations and 20 greedy iterations in our experiments. 150 iterations are sufficient for obtaining high-quality mesh triangles, and 20 iterations are sufficient for the final gradient descent. We used V. Kolmogorov's code for the implementation of minimum cut in a graph/maximum flow [23, 17], which is available online[1]. For the presented models, the algorithm running time is between 30 seconds and 2 minutes (depending on the model size) on an Intel Core 2 Duo P8400 (2,26 GHz) with 4 GB RAM.

To illustrate the mesh quality, average triangle minimum and maximum angles are presented in table 1. For high-quality meshes, the average minimum (resp. maximum) angle should be greater than 30^o (resp. less than 90^o). The closer to 60^o the average minimum and maximum angles are, the better are the results.

To evaluate the surface fidelity of the remeshed models, the Hausdorff distance and the maximum of the two RMS (Root Mean Square) distances normalized to the bounding box diagonal are presented in table 1. These distances have been obtained using the Metro tool [24].

Our method preserves high frequency mesh features (cf. figure 5), while considerably improving triangle quality. In addition, the vertex positions tend to adapt themselves to the mesh curvature as it can be observed on the hand model in figure 4. Let's note that the number of vertices does not need to be chosen. It adapts itself to the geometry while maintaining the same order of magnitude.

We compare our method to those of Valette et al. [5], Surazhsky et al. [10, 11], and Liu et al. [13]. As can be seen in table 1, our method gives better results in terms of triangle shape and surface fidelity when compared to Valette et al. and Liu et al. . Surazhsky et al.'s methods generate more regular triangles (better averaged min and max angles), but our method better approximates the original surface, even when the number of used vertices is lower. For instance, in reference [10] their proposed method smoothes the triceratops eye (cf. figure 5) resulting in a significant loss of details.

We have tested our method on CAD (Computer-Aided Design) models

[1] http://www.cs.ucl.ac.uk/staff/V.Kolmogorov/software.html

Model	#v	Irreg (%)	Amin (deg)	Amax (deg)	Er_{Haus} (10^{-3})	Er_{RMS} (10^{-3})
Fandisk (init)	6495	20	43.4	86.1	-	-
Fandisk (Liu)	6495	20	44.7	82.0	3.3	0.8
Fandisk (our)	6361	**14**	**48.2**	**77.4**	**1.3**	**0.03**
Cow (init)	2892	53	30.8	92.4	-	-
Cow (Liu)	2892	53	35.1	88.2	5.3	0.9
Cow (our)	2743	**36**	**40.7**	**83.8**	**5.0**	**0.7**
Shark (init)	2560	31	20.8	97.4	-	-
Shark (Liu)	2560	31	26.2	107.5	**3.0**	**0.3**
Shark (Sur1)	2560	30	**50.6**	**71.1**	6.8	0.8
Shark (our)	2345	**23**	32.4	96.3	3.8	**0.3**
Hand (init)	7950	58	32.3	94.1	-	-
Hand (Liu)	7950	58	34.3	92.2	8.8	0.4
Hand (Val)	6802	45	46.1	77.5	2.6	**0.2**
Hand (our)	6800	**28**	**48.1**	**76.5**	**2.0**	**0.2**
Bimba (init)	8857	62	34.2	92.7	-	-
Bimba (Liu)	8857	62	38.1	87.0	4.9	0.5
Bimba (Sur1)	8857	**20**	**53.6**	**67.6**	6.0	0.5
Bimba (Val)	8143	48	45.2	78.1	6.0	0.4
Bimba (our)	8143	37	46.9	77.0	**3.4**	**0.2**
Egea (init)	8268	74	34.7	93.5	-	-
Egea (Liu)	8268	74	38.2	88.3	2.6	0.2
Egea (Sur2)	8705	**6.7**	**52.4**	**69.1**	2.7	**0.2**
Egea (our)	7229	42	48.9	74.9	**2.4**	0.4
Triceratops (init)	2832	59	29.6	95.5	-	-
Triceratops (Sur2)	2758	**13**	**42.2**	**82.5**	8.4	1.1
Triceratops (our)	2693	37	41.0	83.9	**3.9**	**0.5**

Table 1: Statistics on the remeshed models: number of vertices, percentage of irregular vertices, averaged minimal angle, averaged maximal angle, Hausdorff distance and maximum between the 2 RMS distances measured by Metro normalized to the bounding box diagonal. Liu, Val, Sur1, and Sur2 correspond respectively to [13], [5], [11], and [10].

(cf. figure 6) and we obtained better min and max angles for the remeshed models.

6 Conclusion and future work

This paper has presented an original way of computing vertex repositioning using global combination of locally proposed candidates. Its main advantages are the feature sensitiveness and its ability to improve triangle shapes while preserving the original surface fidelity. The convergence speed towards a better global configuration is enhanced using smooth local changes followed by vertex relaxation. The obtained results are better than other methods in terms of surface fidelity. Our method also offers a good surface fidelity/triangle quality trade-off.

As future work, we will investigate quadrangular and anisotropic remeshing. We will also tackle combinatorial optimization (when computationally tractable) in the connectivity processing to improve mesh connectivity configuration.

Acknowledgments

This work has been supported by the French National Research Agency (ANR) through MDCO program (project MADRAS No. ANR-07-MDCO-015).

We wish to thank Vitaly Surazhsky and Ligang Liu for their kind help in applying their respective remeshing methods on our 3D models and for sharing their results with us.

References

[1] Shewchuk J. R.: What is a good linear element? Interpolation, conditioning, and quality measures. In 11th International Meshing Roundtable (2002), pp. 115–126.

[2] Alliez P., Ucelli G., Gotsman C., Attene M.: Recent Advances in Remeshing of Surfaces. Part of the state-of-the-art report, AIM@SHAPE EU network of excellence, 2005.

[3] Cohen-Steiner D., Alliez P., Desbrun M.: Variational shape approximation. In ACM Siggraph (2004), pp. 905–914.

[4] Valette S., Chassery J.-M.: Approximated centroidal voronoi diagrams for uniform polygonal mesh coarsening. Computer Graphics Forum (Eurographics 2004 proceedings) 23, 3 (2004), pp. 381–389.

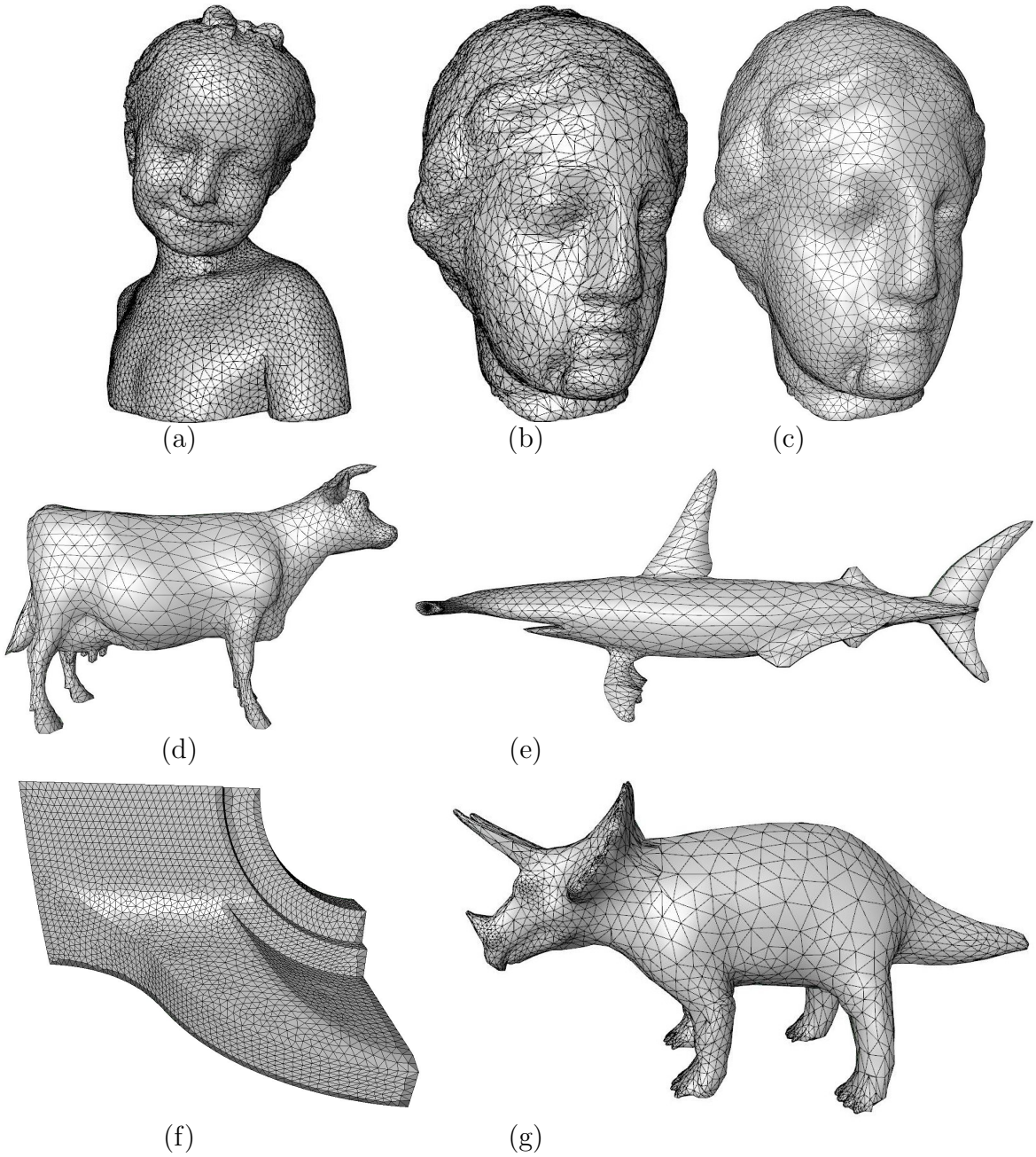
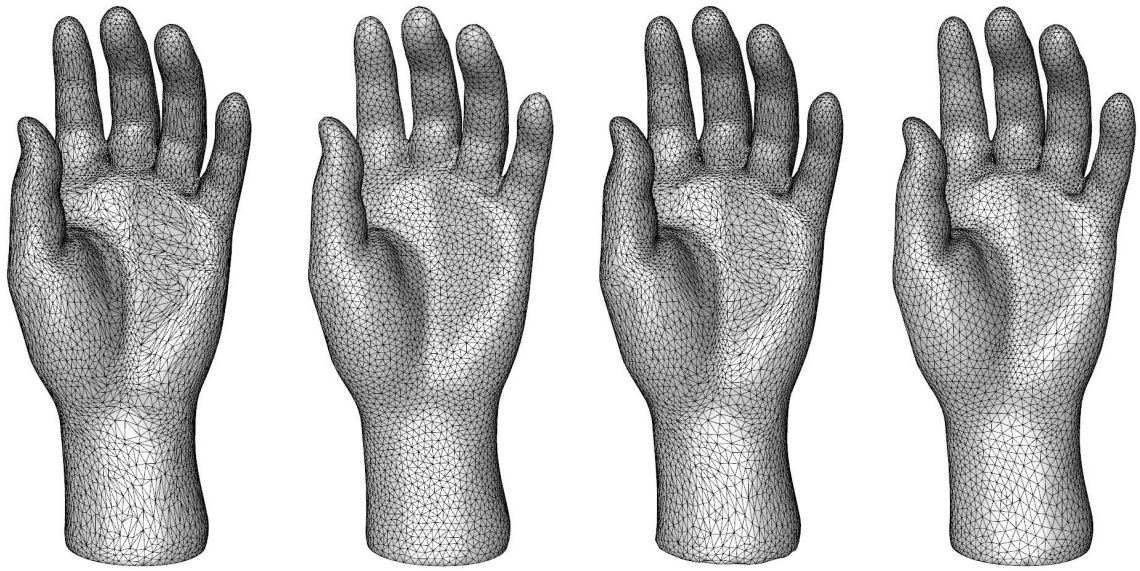
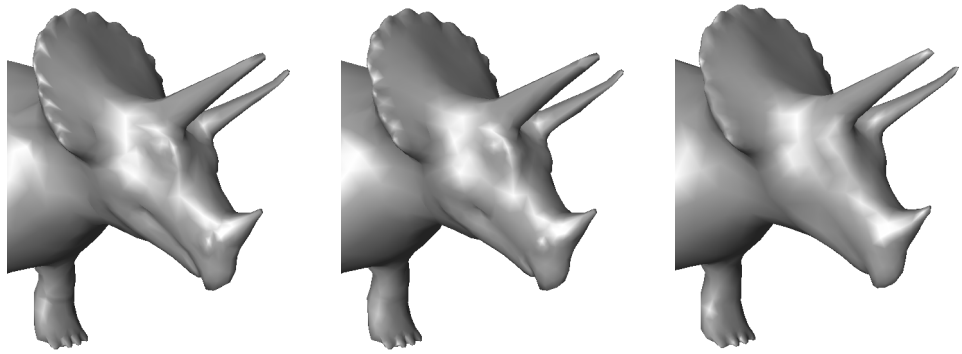


Figure 3: Results obtained with our method: (a) remeshed bimba model; (b) original egea model; (c) remeshed egea model; (d) remeshed cow model; (e) remeshed shark model; (f) remeshed fandisk model; (g) remeshed triceratops model.



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

Figure 4: Comparisons between (a) the original hand model, (b) the Valtette et al. [5] version, (c) the Liu et al. [13] version and (d) our version.

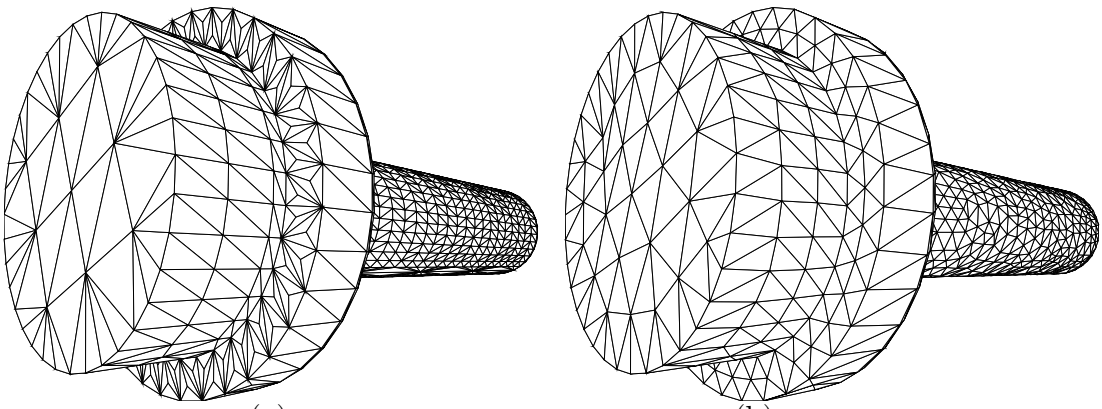


(a) (b) (c)

Figure 5: Comparisons between **(a)** the original triceratops model, **(b)** our remeshed version and **(c)** the Surazhsky and Gotsman [10] remeshed version. The features surrounding the triceratops eye are well-preserved by our method.

[5] Valette S., Chassery J.-M., Prost R.: Generic remeshing of 3d triangular meshes with metric-dependent discrete voronoi diagrams. IEEE Trans Visu Comp Grap 14, 2 (2008), pp. 369–381.

[6] Alliez P., Verdière E.C.D , Devillers O., Isenburg M.: Isotropic surface



(a) (b)

	(a)
#v	1243
Irreg (%)	46
Amin (deg)	25.1
Amax (deg)	95.4
Er_{Haus} (10^{-3})	-
Er_{RMS} (10^{-3})	-

	(b)
#v	1080
Irreg (%)	26
Amin (deg)	41.0
Amax (deg)	81.4
Er_{Haus} (10^{-3})	2.2
Er_{RMS} (10^{-3})	0.3

Figure 6: Comparisons between **(a)** the original CAD model and **(b)** our remeshed version. Note that the vertex valence irregularity percentage that we obtained is almost half the one obtained in the original model, we have also a better Amin and Amax angle convergence towards 60^o. A lower vertices number denotes an optimized triangle distribution since less triangles are required to cover the same surface while introducing an insignificantly small geometric error.

remeshing. In IEEE Shape Modeling International (2003), pp. 49–58.

[7] Guskov I., Khodakovsky A., Schröder P., Sweldens W.: Hybrid meshes: multiresolution using regular and irregular refinement. In Symposium on Computational geometry (2002), pp. 264–272.

[8] Sifri O., Sheffer A., Gotsman C.: Geodesic-based surface remeshing. In 12th International Meshing Roundtable (2003), pp. 189–199.

[9] Peyré G., Cohen L.: Geodesic remeshing using front propagation. International Journal of Computer Vision 69, 1 (2006), pp. 145–156.

[10] Surazhsky V., Gotsman C.: Explicit surface remeshing. In Eurographics/ACM SIGGRAPH symposium on Geometry processing (2003), pp. 20–30.

[11] Surazhsky V., Alliez P., Gotsman C.: Isotropic remeshing of surfaces: a local parameterization approach. In 12th International Meshing Roundtable (2003), pp. 215–224.

[12] Winkler T., Hormann K., Gotsman C.: Mesh massage: A versatile mesh optimization framework. Visual Computer 24, 7 (2008), pp. 775–785.

[13] Liu L., Tai C.-L, Ji Z., WANG G.: Non-iterative approach for global mesh optimization. Computer Aided Design 39, 9 (2007), pp. 772–782.

[14] Nealen A., Igarashi T., Sorkine O., Alexa M.: Laplacian mesh optimization. In ACM Graphite (2006), pp. 381–389.

[15] Pottmann H., Hofer M.: Geometry of the squared distance function to curves and surfaces. In Visualization and Mathematics III (2003), Hege H.-C., Polthier K., (Eds.), Springer, pp. 223–244.

[16] Wang W., Pottmann H., LIU Y.: Fitting b-spline curves to point clouds by curvature-based squared distance minimization. ACM Transactions on Graphics 25, 2 (2006), pp. 214–238.

[17] Kolmogorov V., Zabih R.: What energy functions can be minimized via graph cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence 26, 2 (2004), pp. 147–159.

[18] Lempitsky V., Roth S., Carsten R.: Fusionflow: Discrete-continuous optimization for optical flow estimation. In IEEE Conference on Computer Vision and Pattern Recognition (2008), pp. 1–8.

[19] Geman S., Geman D.: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6, 6 (11 1984), pp. 721–741.

[20] Duda R., Hart P., Stork D.: Pattern Classification, 2nd Edition. Wiley, New York, NY, Nov. 2000.

[21] Yue W., Guo Q., Zhang J., Wang G.: 3d triangular mesh optimization in geometry processing for cad. In ACM symposium on Solid and physical modeling (2007), pp. 23–33.

[22] Schlesinger D.: Exact solution of permuted submodular minsum problems. In Energy Minimization Methods in Computer Vision and Pattern Recognition (2007), pp. 28–38.

[23] Boykov Y., Kolmogorov V.: An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. IEEE Trans. Pattern Anal. Mach. Intell. 26, 9 (2004), pp. 1124–1137.

[24] Cignoni P., Rocchini C., Scopigno R.: Metro: Measuring error on simplifed surfaces. Computer Graphics Forum 17, 2 (1996), pp. 167–174.