Optimizing 3D Triangulations to Recapture Sharp Edges

J. Andreas Bærentzen

June 29, 2006

Abstract

In this report, a technique for optimizing 3D triangulations is proposed. The method seeks to minimize an energy defined as a sum of energy terms for each edge in a triangle mesh. The main contribution is a novel per edge energy which strikes a balance between penalizing dihedral angle yet allowing sharp edges. The energy is minimized using edge swapping, and this can be done either in a greedy fashion or using simulated annealing. The latter is more costly, but effectively avoids local minima.

The method has been used on a number of models. Particularly good results have been obtained on digital terrain models. It is demonstrated how the method has been able to recapture sharp edges which are clearly present in the data but not reflected by the original triangulation of the elevation points. [triangulation] [simulated annealing][edge swap]

1 Introduction

When reconstructing surfaces of real-world objects, one is often starting from a set of points. This is true, for instance of laser scanned objects [LRG*00] or objects reconstructed using structure from motion [PVGV*04]. It is also true of terrain models which are often reconstructed from a set of elevation points measured by a surveyor [PK90].

This report is concerned with surfaces represented as triangle meshes and especially triangulations of elevation data. Elevation data is frequently triangulated using Delauney triangulation [Sch93] on a 2D projection of the points. This procedure maximizes the minimum angle in 2D, but it does not align the edges of the mesh with features in the data. The problem is illustrated in Figure 1. In the configuration on the left, the edge shared by the two triangles is transverse to a crease or *feature line* in the surface, and the configuration on the right is clearly preferable. Unfortunately, many triangulation methods (e.g. Delaunay triangulation) are prone to generate configurations such as the one on the left.

This work is based on the assumptions that the original surface is approximately piecewise smooth, and that the features where the piecewise smooth regions join are sampled. However, it is not assumed that the triangulation is correct, and the goal is to optimize the triangulation to better match features. Starting from the same assumptions, Dyn et al. previously proposed a solution to this problem based on a greedy minimization of a curvature-based energy functional [DHKL01]. Unfortunately, there

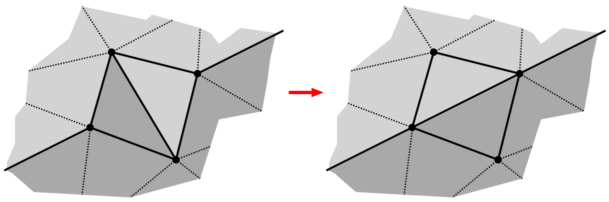

Figure 1: Edge Configurations: A triangle mesh approximates a surface with a sharp bend. On the lefte an edge is transverse to the bend. On the right an edge swap has been performed, and the new edge follows the bend.

are models for which their solution is not sufficient. In particular, I found this to be true of the terrain models.

In this report, a new energy functional is proposed, and it is demonstrated that this energy functional is able to recapture subtle features in terrain data. The method is also tested on other models with good result.

In the next section (Section 2), previous work is discussed. In Section 3 the mesh optimization scheme is discussed in detail. First, the precise energy functional is developed, and then two strategies for its minimization are discussed: A greedy strategy and simulated annealing. In Section 4 results are presented, and in Section 5 I draw conclusions and point to future work.

2 Contribution and Previous Work

The problem of finding and preserving sharp edges in geometric models built from laser scanned or otherwise reconstructed models has received a great deal of attention.

A recent example is the problem of anisotropic smoothing where several authors, e.g. [SB04] [FDCO03] have proposed schemes for the removal of noise from triangle meshes which do not remove sharp edges and corners. Hugue Hoppe's mesh optimization method which fits a triangle mesh to a set of points (which are not the vertices of the mesh) is well known, and it also aims at finding sharp edges [HDD*93].

In [AFRS03], Attene et al. describe a solution to the problem of "unchamfering" triangle meshes. If the vertices of the mesh are sampled from a piecewise smooth surface, but no or few samples lie on sharp edges or corners, the result is an unintended chamfering of these features. To fix the problem, the chamfered edges are split and the new vertices inserted on the estimated intersections of the smooth surfaces from either side. In a sense this is precisely the opposite of the problem solved here. Attene et al. change the triangulation very little but insert new vertices to approximate the corners and edges. I insert no vertices but assume that sharp edges and corners are in fact sampled. On the other hand, the triangulation is optimized in order to align with sharp edges.

This optimization is performed using the well known method of simulated annealing. Simulated annealing was introduced by Kirkpatrick et al. [KGV83] as a general technique for optimization. Simulated annealing is not fast but well suited to the case where we need to avoid local minima. Larry Schumaker initially suggested simulated annealing as a tool for computing optimal triangulations via edge swaps [Sch93]. Provided it is possible to quantify the change in total energy incurred by an edge swap, that is straightforward. One of the applications mentioned in [Sch93] is *data depen-*

dent interpolation where a triangulated surface is smoothed by swapping edges in order to minimize the dihedral angles (i.e. the angle between the face normals of adjacent triangles). In a sense this is my starting point, but the dihedral angle in itself is not a sufficient energy function when the goal is to recapture sharp edges.

Another closely related effort is the work by Dyn et al. who use edge swapping to reduce a discrete curvature measure in a greedy fashion [DHKL01]. They demonstrate that this visibly improves a number of models, and provides a better starting point for subdivision. To a large extent this improvement is, in fact, due to an alignment of the triangle edges with feature lines, but this alignment is not sufficiently powerful to recapture subtle features in terrain data.

In [SP92] Scarlatos et al. presented a very different technique for alinging edges with features. The idea of this report is to move vertices in order to ensure roughly equal curvature within triangles. While their method does perform feature alignment, the goal is to generate a reduced model which fits a more detailed model, and the detailed model is required in order to measure the error.

3 Mesh Optimization

We are given a triangulation of a point set where the triangles form a piecewise planar, manifold surface. The set of edges, $\mathcal{E}$, is simply the set of intersections of pairs of triangles. Two triangles form a quadrilateral if they share an edge and hence two vertices. An edge swap consists of replacing the shared edge by an edge connecting the two other vertices, thus dividing the quadrilateral into two different triangles (c.g. Figure 1).

It is assumed that the vertices are sampled from an underlying surface which is piecewise smooth and that the smooth regions meet along sharp edges which will be called feature lines. It is also assumed that there are samples on these feature lines, but the edges of the original triangulation do not necessarily follow those lines: For two triangles sharing an edge, the edge can either lie along the feature line (edge e in Figure 2) or it can be transverse (edge e' in Figure 2). The former case will be denoted an aligned configuration and the latter case will be denoted a transverse configuration.

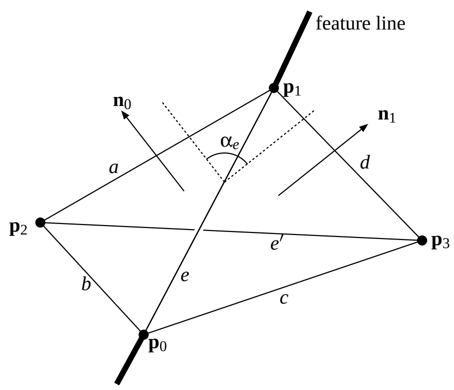

Figure 2: This illustration shows an edge e, the normals of its adjacent faces, its endpoints and the dihedral angle.

The first step is to define a per-edge energy such that going from the transverse to the aligned configuration in Figure 1 causes the total energy to decrease.

The dihedral angle, α, is the angle between the normals of two triangles (c.f. Figure 2). In either configuration the two triangles share one edge (e or e') and together they share four edges (a, b, c, and d) with other triangles. In an aligned configuration α_a, α_b, α_c, and α_d should be small since the associated edges are between triangles that approximate the same smooth region. In fact, if the surface is well sampled, these dihedral angles should be nearly 0. α_e however depends on how sharply the surfaces bends across the feature line. In the transverse case, the α_a, α_b, α_c, and α_d are likely to be larger since the vertices of the two triangles are no longer from the same smooth region: $\mathbf{p}_2$ and $\mathbf{p}_3$ lie on either side of feature line.

Thus, an energy which is simply a sum of dihedral angles would be an initial guess, since moving from the transverse to the aligned case would reduce the number of sharp edges and hence the sum. Unfortunately, it does not work since the dihedral angle of e might in some cases outweigh the sum of the other dihedral angles.

To counter this problem, I use a bias function $\text{bias}(x) = x^{1/\gamma}$ where $\gamma \geq 1$. The derivative of this function is infinite for $x = 0$ but it quickly drops. In practice this means that the energy function grows quickly initially and then the increase tapers off. The method is in fact gamma correction: A large part of the spectrum is used on small dihedral angles and big angles are compressed. The hypothesis is that this will help increase the energy values of edges in the transverse case. The results (c.f. Section 4) indicate that this hyopthesis is correct, and specifically that γ is necessary for models where all dihedral angles are fairly small.

It is also necessary to include an edge length factor. Not doing so tends to favour very long edges which in turn leads to triangles of bad aspect ratio.

These considerations lead to

$$E_e^{\alpha} = [l_e \alpha_e]^{1/\gamma} \tag{1}$$

where $l_e = \|\mathbf{p}_0 - \mathbf{p}_1\|$. E_e^{α} can also be seen as a measure of curvature if $\gamma = 1$. In fact, in Dyn et al. [DHKL01] compute the integral absolute mean curvature for a vertex v

$$\frac{1}{4} \sum_{e \in \mathcal{E}_v} l_e \alpha_e$$

where $\mathcal{E}_v$ is the set of edges incident on v.

It is slightly faster to compute $1 - \cos(\alpha_e) = 1 - \mathbf{n}_0 \cdot \mathbf{n}_1$ where $\mathbf{n}_0$ and $\mathbf{n}_1$ are the normals of the adjacent triangles as shown in Figure 2. This leads to the following energy

$$E_e^c = [l_e(1 - \cos(\alpha_e))]^{1/\gamma} \tag{2}$$

Note that if the mesh has a boundary, e may belong to the boundary. In this case, we stipulate that $E_e = 0$ corresponding to a flat edge. Boundary edges are clearly not subject to swapping.

The energy functional for a triangle mesh is simply the sum over all edges

$$F^c = \sum_{e \in \mathcal{E}} E_e^c \tag{3}$$

where $\mathcal{E}$ is the set of all edges in the triangulation. We can define a corresponding functional based on E^{α},

$$F^{\alpha} = \sum_{e \in \mathcal{E}} E_e^{\alpha} \tag{4}$$

This functional is almost the same as F_2 in [DHKL01], except that energy per edge rather than per vertex.

When an edge swap is performed, the difference in energy,

$$\Delta F = F_{\text{after}} - F_{\text{before}} \quad ,$$

is easily computed since only five edges are affected. As a convenience, let $\Delta F(e)$ denoted the energy difference associated with swapping e. The swap should be performed if $\Delta F(e) < 0$.

3.1 The Greedy Strategy

Greedy strategies are frequently used for optimization in computer graphics, for instance in [DHKL01]. The basic idea is to always make the choice that leads to the greatest reduction in energy. In the current scenario, this means that we should perform the edge swap which leads to the greatest decrease in energy. Swapping an edge only affects five edges, hence it is easy to compute, $\Delta F(e)$, the energy after minus the energy before an edge swap.

Initially, ΔF is computed for all edges, and for each edge e a pair $< \Delta F(e), e >$ is inserted into a priority queue if $\Delta F(e) < 0$.

The next step is a loop where we iteratively extract and remove the record with the ΔF corresponding to the greatest decrease in energy from the heap and swap the corresponding edge.

After an edge swap, $\Delta F(e)$ must be recomputed for any edge e which belongs to a triangle that is adjacent to one of the triangles containing the swapped edge (including the swapped edge itself). A new record is inserted in the priority queue for each of these edges if $\Delta F(e) < 0$. This entails that we sometimes have multiples records for a given edge in the priority queue. A time stamp is maintained for each edge and also stored in the records in the priority queue. If a record is extracted, and the corresponding time stamp is more recent, the record is simply discarded.

The loop continues until the priority queue is empty.

3.2 Simulated Annealing

A problem with the greedy strategy is that for many problems, including the present, all edges can be in a configuration that is locally optimal while the configuration is not globally optimal. Simulated annealing [KGV83] is a general framework for optimizaton which is well suited to avoid these local optima.

Applied to the problem at hand, the method works as follows. We iteratively, pick a random edge from $\mathcal{E}$ and compute the ΔF associated with swapping this edge. If $\Delta F <= 0$ the swap is performed since the energy decreases. If $\Delta F > 0$, however, the swap is performed with probability:

$$P_{swap}(e) = e^{\frac{-\Delta F}{T}} \tag{5}$$

where T is the temperature. A random number, r is generated, and if $r < P_{swap(e)}$ then e is swapped. For a given T, a small ΔF means a high probability, and for a given ΔF a high temperature means high probability.

Initially, the temperature is very high which means that all swaps are probable. After some time, the temperature is lowered according to an *annealing schedule*, and as the temperature approaches zero so does the probability of making swaps which cause the energy to increase. The intuition behind the method is that the initial random swaps help avoid local optima.

Simulated annealing requires some initial parameters. These have been selected experimentally based on advice given in [Sch93]. The initial temperature is

$$T_0 = -2 \min_e(\Delta F(e))$$

All edges are visited five times in random order (and either swapped or not according the scheme above) before the temperature is lowered. The new temperature is then computed according to

$$T \leftarrow T \cdot 0.9$$

When the temperature becomes very low, swaps that increase the energy cease to occur in practice, and when all edges have been visited without any swaps being made, the algorithm terminates.

3.3 Avoiding Degenerate Triangles

Not all edges should be swapped even if the swap will cause an energy reduction. Since the mehtod, essentially, tries to concentrate dihedral angle in a few edges while keeping most edges smooth, it is not surprising that very sharp edges are sometimes introduced. This should be avoided by setting an upper threshold, $\tau \in [0, \pi]$, on the dihedral angle. If an edge swap results in a dihedral angle greater than this threshold, the swap is not allowed. Usually, this threshold is set to $2.09 \approx 120°$.

In the case of terrain data, we know that the triangles should always face up, consistently. If an edge swap results in a triangle facing down, the swap is not allowed.

If we are dealing with terrain data, it seems that either of these two rules suffice. For general triangle meshes, the first rule is necessary, and the second does not apply.

However, more rules are necessary: It is assumed that the triangle mesh represents a manifold surface [Hof89], possibly with boundary. It is fairly obvious that we cannot swap boundary edges. However, some swaps will also violate the manifold property of the mesh and must be disallowed. In practical terms, manifoldness means that every edge is adjacent to two triangles, and that the triangles sharing a vertex form a single cycle around that vertex. Moreover, the intersection of two triangles should be either empty or the shared edge. To preserve manifoldness, an edge is swapped only on two

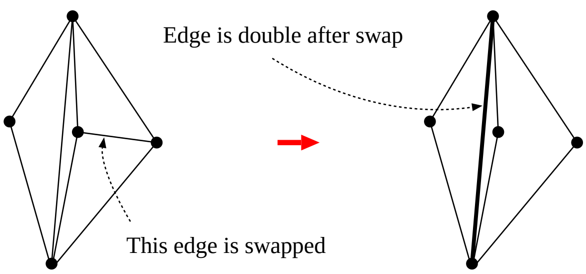

Figure 3: This figure illustrates the problem of swapping an edge which is adjacent to a vertex of valence three.

conditions:

1. Both vertices at the end-points of the edge must have valence (i.e. the number of adjacent edges) greater than three.

2. The two vertices which will be connected by the swap must not be connected by a different edge.

The first rule ensures that we do not have vertices of valence two after the swap. If an interior vertex has valence two, it is connected to two edges which in turn are both adjacent to the same two triangles. Then, these two triangles share all three vertices which means they have collapsed as shown in Figure 3.

Since all edges are straight, a violation of the second rule means that two edges after the swap are geometrically identical. Since an interior edge is shared by two faces, we would have at least three and in general four faces meeting at the same geometric edge after the swap.

4 Results

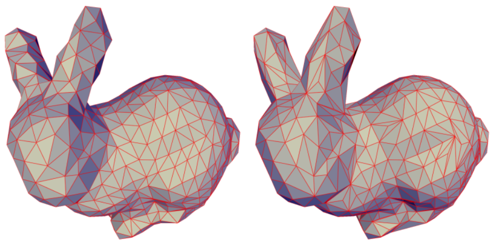

Figure 4: A greatly reduced version of the Stanford Bunny before and after minimization of F^c with $\gamma = 2$).

The method has been tested on several models. The row of images in Figure 7 show a terrain model generated using Delaunay triangulation and after various types of optimization. The most important feature is a road which runs along the left edge of the terrain. Fortunately, it is very easy to verify whether a triangulation correctly captures this feature, and that is plainly not the case for the Dealunay triangulation shown on the left. In the image on the center left, F^c has been minimized using a greedy approach. As feared, it does get stuck in a local minimum, and the edges are only partially aligned with the road. In the image on the centre right, simulated annealing has been used with $\gamma = 1$, but again the result is not satisfactory. Only the combination of $\gamma = 4$ and simulated annealing gives a good reconstruction of the road as seen in the image on the right. The images in Figure 8 show height curves from the terrain generated directly from the Delaunay triangulation and after optimization (the same as in the top right image).

It is also possible to justify numerically the need for the γ exponent. Table 1 contains the absolute values of the F^α and F^c functionals for the terrain model in Figure 7. The before values correspond to the original model, and the after values correspond to the model after minimization of $F^c, \gamma = 4$. It is obvious from the table that for $\gamma = 1$, there is no energy decrease in going from the original to the optimized model. Hence, it is unsurprising that the result is not improved.

The Venus model was used for a test which is shown in Figure 5. The initial model was corrupted by performing random edge swaps resulting in the model shown in the top left of the figure. I first tried to undo the effect of these swaps by minimizing the *integral absolute mean curvature* (i.e. F_2 from [DHKL01]) and then by minimizing

Terrain	$\gamma = 1$		$\gamma = 4$	
F^c	**0.151**	**0.153**	**303.4**	**256.2**
F^α	3.55	3.47	823.4	748.4
Venus	$\gamma = 1$		$\gamma = 4$	
F^c	23395.5	2527.88	5179.21	2668.88
F^α	**49043.6**	**10993.7**	**7026.31**	**4814.23**
Functional	before	after	before	after

Table 1: Functional values for the triangle mesh before and after optimization for two different values of γ. The boldface numbers indicate which energy was used to generate the "after" mesh.

F^α with $\gamma = 1$. In both cases simulated annealing was used. $\gamma > 1$ was also tried, but did not noticeably improve results. From Table 1 it is also apparent that the relative difference in energy before and after optimization is largest for $\gamma = 1$.

Results are shown for a different terrain with just a few hundred triangles in Figure 6. Again, the result is much improved.

The method was also tested on a greatly reduced (using a volumetric technique) version of the Stanford bunny seen in Figure 4. Notice how the biased dihedral energy has regorganized the triangles to create nearly planar regions. This is particularly visible on the right ear.

Unfortunately, simulated annealing is not efficient. The tested models all have between one and two thousand faces, and the algorithm completes in at most half a minute. However, for very large models, it is more likely to take hours. The greedy approach is, of course, much faster, but easily trapped in local minima. Table 2 shows performance measurements.

Model	energy	faces	iter	time
Bunny	$F^c, \gamma = 2$	1160	465	13
Terrain	$F^c, \gamma = 4$	2027	801	35
Terrain (Fig 6)	$F^c, \gamma = 4$	475	887	10
Venus	$F^\alpha, \gamma = 1$	1418	733	25
				seconds

Table 2: These timings show how long it takes to minimize the biased dihedral energy using simulated annealing on the models discussed in this report.

From experiments with the method, I can conclude that it is also very robust in the case of terrain data. In the case of general meshes (i.e. not height fields), there is the danger that the random swaps performed in the initial iterations of the simulated annealing will allow the mesh to degenerate to a configuration from whence it cannot recover. However, this problem is effectively countered using the max dihedral angle threshold discussed in Section 3.3. Another approach would have been to lower the initial temperature to make very bad moves less likely.

5 Discussion and Future Work

Minimization of a very simple energy functional was proposed in this report as a technique for aligning edges in a triangle mesh with features. My main concern was to recover features in terrain data, and the method works very well for that purpose thanks to the γ exponent although results are also encouraging for general meshes, but the exponent is less useful here – for a simple reason: The rôle of γ is similar to gamma correction of images where, say, low intensities are enhanced to make dark regions more visible. In the terrain data, all dihedral angles are fairly small, and the γ exponent enhances these differences.

In this report, γ was chosen experimentally, but it might be feasible to select γ automatically. Possibly, a selection based on analysis of the histogram of dihedral angles will be a promising direction of future investigation.

Arguably, simulated annealing is a very costly method, but on the other hand it is even simpler to implement than the greedy method. However, for meshes containing millions of triangles, simulated annealing may be too slow. A plausible approach in that case would be to run the annealing only on local patches and then stich these together. In that case, the greedy approach might be sufficient to fix the problems on the boundaries between regions, but this has not been attempted.

References

[AFRS03] ATTENE M., FALCIDIENO B., ROSSIGNAC J., SPAGNUOLO M.: Edge-sharpener: recovering sharp features in triangulations of non-adaptively re-meshed surfaces. In *SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing* (Aire-la-Ville, Switzerland, Switzerland, 2003), Eurographics Association, pp. 62–69.

[DHKL01] DYN N., HORMANN K., KIM S.-J., LEVIN D.: Optimizing 3d triangulations using discrete curvature analysis. In *Mathematical Methods for Curves and Surfaces: Oslo 2000* (Nashville, TN, USA, 2001), Vanderbilt University, pp. 135–146.

[FDCO03] FLEISHMAN S., DRORI I., COHEN-OR D.: Bilateral mesh denoising. *ACM Trans. Graph. 22*, 3 (2003), 950–953.

[HDD*93] HOPPE H., DEROSE T., DUCHAMP T., MCDONALD J., STUETZLE W.: Mesh optimization. *Proc ACM SIGGRAPH 93 Conf Comput Graphics and Proceedings of the ACM SIGGRAPH '93 Conference on Computer Graphics* (1993), 19–25.

[Hof89] HOFFMANN C. M.: *Geometric and Solid Modeling*. Morgan Kaufmann, 1989.

[KGV83] KIRKPATRICK S., GELATT C. D., VECCHI M. P.: Optimization by simulated annealing. *Science 3.220*, 4598 (1983), 671–680.

[LRG*00] LEVOY M., RUSINKIEWCZ S., GINZTON M., GINSBERG J., PULLI K., KOLLER D., ANDERSON S., SHADE J., CURLESS B., PEREIRA L., DAVIS J., FULK D.: The digital michelangelo project: 3d scanning of large statues. *Computer Graphics Proceedings. Annual Conference Series 2000. SIGGRAPH 2000. Conference Proceedings* (2000), 131–44.

[PK90] PETRIE G., KENNIE T. J. M. (Eds.): *Terrain Modelling in Surveying and Civil Engineering*. Whittles Publishing, 1990.

[PVGV*04] POLLEFEYS M., VAN GOOL L., VERGAUWEN M., VERBIEST F., CORNELIS K., TOPS J., KOCH R.: Visual modeling with a hand-held camera. *Int'l J. Computer Vision 59*, 3 (2004), 207–232.

[SB04] SHEN Y., BARNER K. E.: Fuzzy vector median-based surface smoothing. *IEEE Transactions on Visualization and Computer Graphics 10*, 3 (May/June 2004), 252–265.

[Sch93] SCHUMAKER L. L.: Computing optimal triangulations using simulated annealing. In *Selected papers of the international symposium on Free-form curves and free-form surfaces* (Amsterdam, The Netherlands, The Netherlands, 1993), Elsevier Science Publishers B. V., pp. 329–345.

[SP92] SCARLATOS L., PAVLIDIS T.: Optimizing triangulations by curvature equalization. *Visualization, 1992. Visualization '92, Proceedings., IEEE Conference on* (1992), 333 –339.

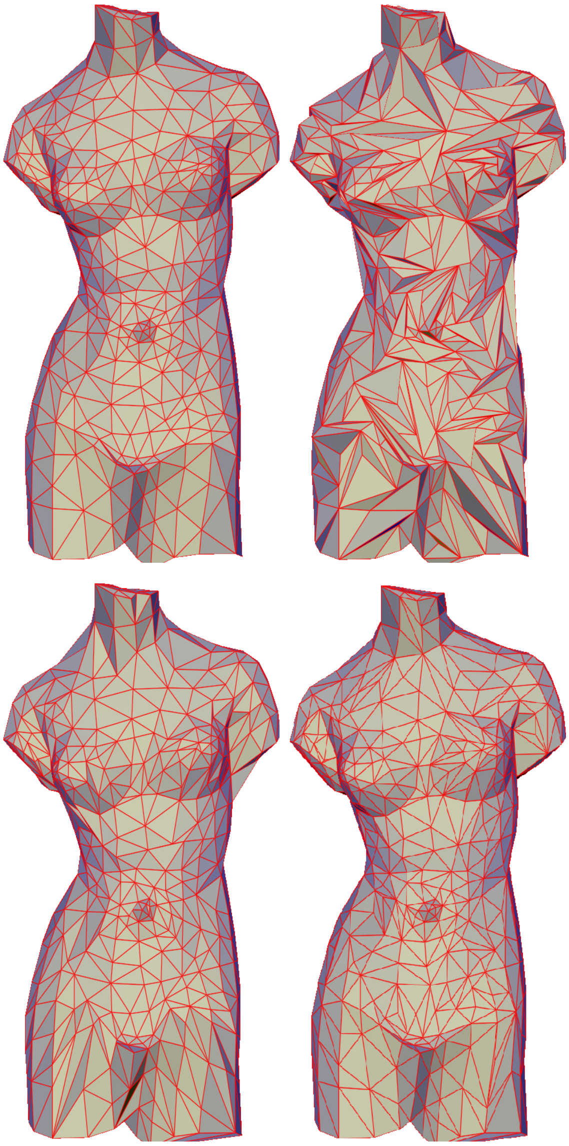

Figure 5: The original venus model (top left), the venus model corrupted by random edge swaps (top right), the venus model after minimization of a per vertex curvature measure (bottom left), and the venus model after minimization of F^{α} (bottom right). It is clear from the structure of the mesh that F^{α} performs better than the per vertex curvature measure in many places.

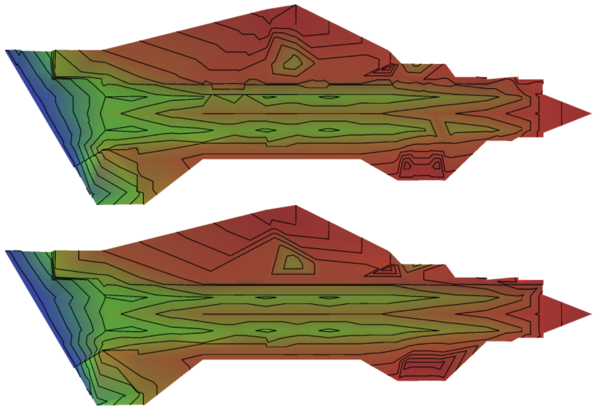

Figure 6: A simple terrain before (top) and after (bottom) minimization of F^c

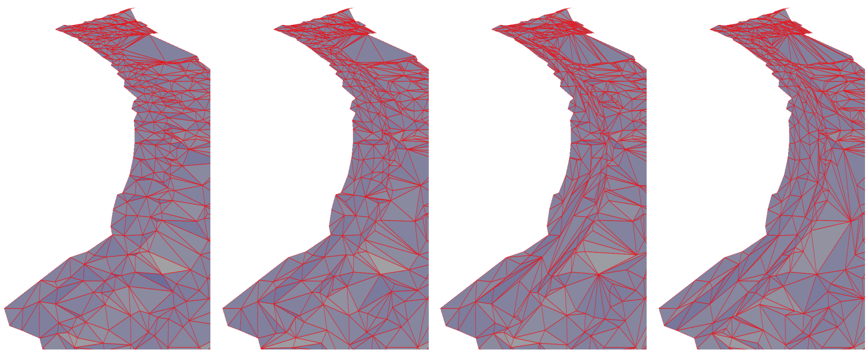

Figure 7: From left to right, this figure shows the original Delaunay triangulation of the height points, the triangulation after a greedy optimization ($\gamma = 4$), the triangulation after simulated annealing with $\gamma = 1$, and, finally, the triangulation after optimization using simulated annealing with $\gamma = 4$. Notice how the road running along the left side of the terrain is correctly reconstructed in the image on the right.

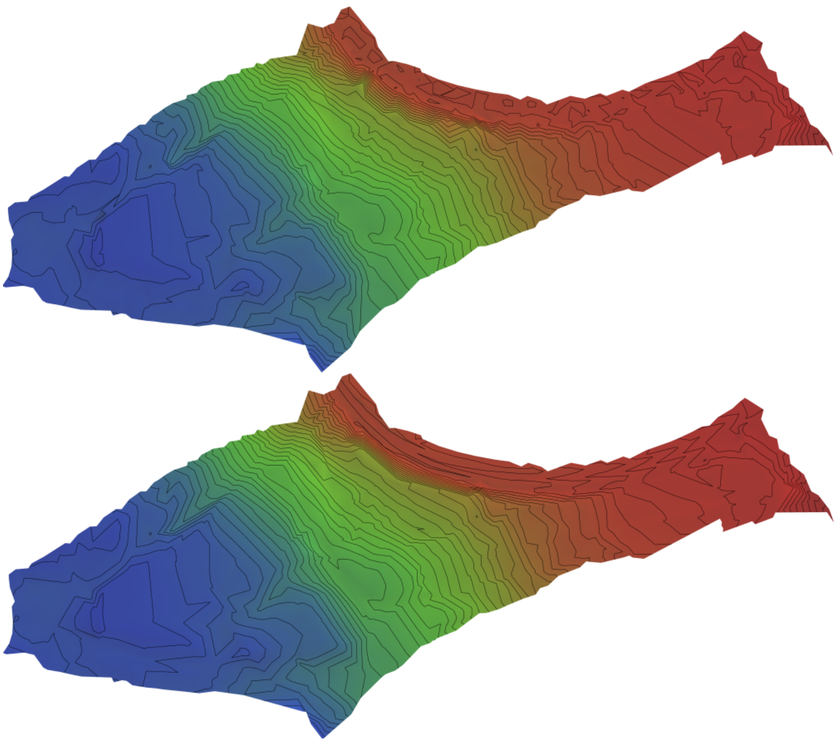

Figure 8: The two images show the height curves of the model before and after minimization of $F^c, \gamma = 4$.