MEASURING AND CONTROLLING FAIRNESS OF TRIANGULATIONS

CAIGUI JIANG, FELIX GÜNTHER, JOHANNES WALLNER, AND HELMUT POTTMANN

Abstract. The fairness of meshes which represent geometric shapes is a topic which has been studied extensively and thoroughly. However, the focus in such considerations often is not the mesh itself, but rather the smooth surface approximated by it, and fairness essentially expresses a mesh's suitability for purposes such as visualization or simulation. This paper focuses on meshes in the architectural context, where vertices, edges and faces of meshes are often highly visible, and any notion of fairness must take new aspects into account. We use concepts from discrete differential geometry (star-shaped Gauss images) to express fairness, and we also demonstrate how fairness can be incorporated into interactive geometric design of triangulated freeform skins.

Key words. polyhedral surface, smoothness, fairness, freeform skin, triangulation, optimization

1. Introduction and Motivation

Smoothness of meshes. Whenever a smooth shape is realized in a discrete manner, the smoothness resp. fairness of this approximation is of great importance. Depending on the application, different aspects of fairness play a role. For some applications like the simulation of physical processes (finite element analysis), or computer graphics rendering, the vertices and edges of the mesh are only a means to an end, and "fairness" mostly refers to the suitability of the mesh for the task at hand. Typically, it involves avoiding small angles between edges, comparable edge lengths in triangles, and avoiding vertices whose number of incident edges is not 6.

Figure 1. *Non-Smoothness from geometric constraints*: The *Cour Visconti* roof in the Louvre is a hybrid mesh consisting of both triangular and quadrilateral glass panels, for reasons of efficiency and weight optimization. The triangle mesh originally intended by the architect is achieved by placing triangular shading elements on top of each panel. Merging of two triangular faces into a quad consumes one degree of freedom, so this mesh is not as optimally smooth as would have been possible with a triangle mesh.

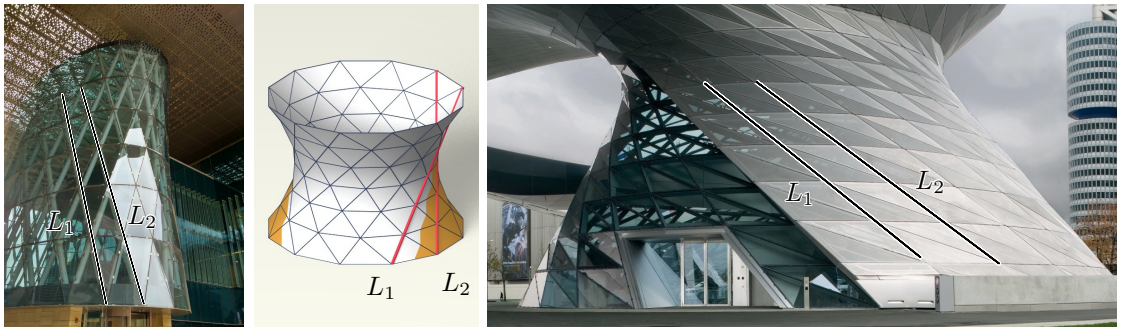

FIGURE 2. *Non-Smoothness from geometric constraints* is exhibited by Building 16 of King Abdullah University of Science and Technology (left) and by the BMW Welt building in Munich (right). These meshes contain rows of faces whose vertices alternate between two straight lines L_1, L_2, or at least approximately so. *Center:* Intersection of the mesh's surface with a plane parallel to both L_1, L_2 is a zigzag polyline whose edges are parallel to either L_1 or L_2. This shows that meshes which contain straight lines in the manner described above cannot avoid a certain degree of non-smoothness.

Smoothness of meshes in freeform architecture. In freeform architecture, the purpose of meshes typically is twofold: Firstly, to make a visual statement, and secondly, to be part of the structure. The high visibility of edges and vertices makes them a much greater part of fairness resp. smoothness than in other applications. The human eye notices minimal zigzags in edge polylines which are entirely irrelevant for physical simulations or for rendering. Similarly, reflective surfaces expose even very small kink angles between faces.

Mesh smoothness is to be distinguished from smoothness of the reference shape which the mesh is thought to approximate. A wiggly mesh can mean that a smooth reference surface is approximated in a bad manner, but it can also mean that there are wiggles in the reference shape. Unfortunately the former can sometimes not be avoided because of constraints imposed on the mesh, see examples in Figures 1 and 2.

In this paper we discuss a notion of smoothness which we believe to be consistent with expectations of users in the field of freeform architecture. We can already draw on an existing mathematical discussion by (Günther and Pottmann, 2016). We further discuss the optimization of meshes towards greater smoothness. The optimization consists of setting up hard and soft constraints, and subsequent application of standard numerical procedures.

2. Measuring smoothness

The main topic of this paper is the behaviour of meshes in the neighbourhood of vertices. This does not mean that in algorithms we neglect other contributions to visual smoothness like fairness of edge polylines (see Section 3), but these are the standard ones.

2.1. **The Gauss image.** We endow a mesh with a *Gauss image* whose vertices are the consistently oriented unit normal vectors of the faces; we think of them as pointing to the *outside* of the mesh. The Gauss image is part of the unit sphere. Each original (primal) edge separating two faces corresponds to a Gauss image edge (dual edge) connecting two unit normal vectors. Figure 3 illustrates the Gauss image $g(\mathbf{v})$ of the 1-ring neighbourhood of a vertex $\mathbf{v}$, while Figure 4 shows Gauss images of entire meshes.

Properties of Gauss images. There are certain obvious properties of Gauss images which correspond to visual smoothness: Long edges in the Gauss image correspond to large kink angles between adjacent faces (see Figure 3). Also, the shape of the Gauss image cycle of a vertex (again, see Figure 3) defines the shape of the mesh's surface in the immediate neighbourhood of a vertex. Therefore, we look for an *even pattern of dual faces* in the Gauss image.

If the dual Gauss image face $g(\mathbf{v}_i)$ of a vertex $\mathbf{v}_i$ is a proper polygon without self-intersections, then we might view any point in its interior as a candidate for a normal vector associated with the vertex $\mathbf{v}_i$. Further, we observe the sign of discrete Gauss curvature K of the mesh: We have $K > 0$ or $K < 0$ depending on whether the Gauss image $g(\mathbf{v}_i)$, and the cycle of faces incident with the vertex $\mathbf{v}_i$, have the same orientation or opposite orientations (see Figure 3).

This behaviour entirely corresponds to the behaviour of the normal vectors along a small circle in smooth surfaces (Figure 3, right). If the original circle is denoted by C and its Gauss image by $g(C)$, then the ratio of signed areas of $g(C)$ to C is the Gauss curvature K. Zero Gauss curvature implies zero signed area and thus self-intersections of the Gauss image.

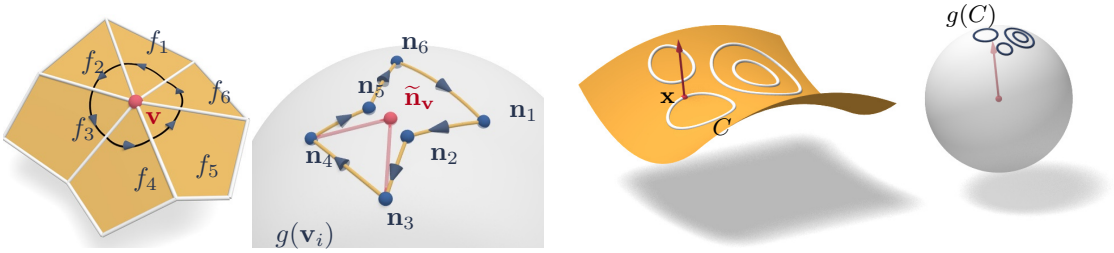

FIGURE 3. *Gauss image of a vertex* $\mathbf{v}_i$. The cycle of faces $f_1, \ldots, f_6$ incident with $\mathbf{v}_i$ defines a cycle $g(\mathbf{v}_i)$ of unit normal vectors $\mathbf{n}_1, \ldots, \mathbf{n}_6$ on the unit sphere which form the Gauss image $g(\mathbf{v})$. The kink angle between faces f_k, f_{k+1} coincides with the spherical edge length $\overline{\mathbf{n}_k, \mathbf{n}_{k+1}}$. In the case shown here the Gauss image polygon $g(\mathbf{v})$ has no self-intersections, so it is the boundary of two spherical domains — one of them contains unit vectors like $\widetilde{\mathbf{n}}_\mathbf{v}$ which point to the outside of the primal mesh; it is called the interior of $g(\mathbf{v})$. We can observe the sign of curvature (negative, from the fact that the two cycles have opposite orientations). Further, any interior point $\widetilde{\mathbf{n}}_\mathbf{v}$ of the Gauss image polygon $g(\mathbf{v})$ can be viewed as an auxiliary unit normal vector associated with the vertex $\mathbf{v}_i$. *Right:* The surface with point $\mathbf{x}$ and normal vector illustrates the smooth situation.

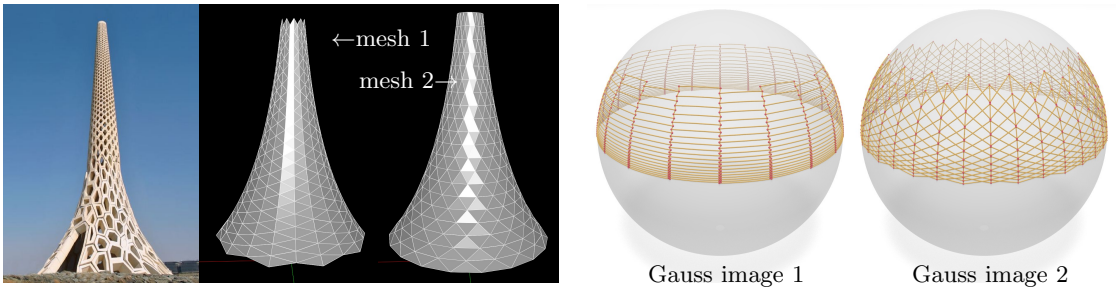

FIGURE 4. Smooth and non-smooth triangle meshes. Meshes 1 and 2 represent the same reference shape. Mesh 1 fulfills Definition 1 of "smoothness", while mesh 2 does not.

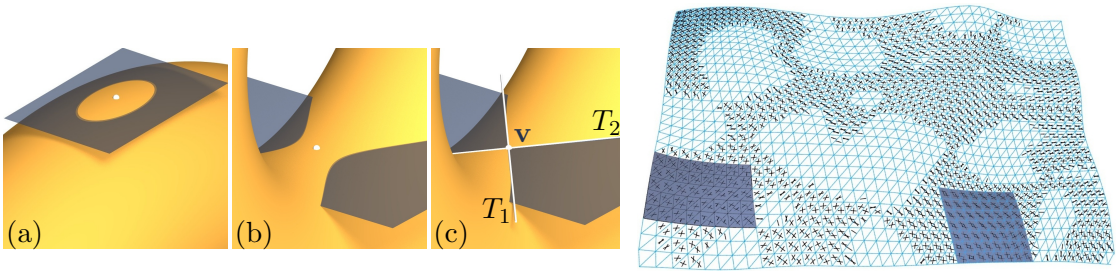

FIGURE 5. *Local shape analysis of smooth surfaces:* The intersection of a smooth surface with an almost-tangent plane generically approaches a conic called *Dupin's indicatrix* which is an ellipse in case of positive Gauss curvature (a), and a hyperbola in case of negative Gauss curvature (b). In the latter case, the intersection with a tangent plane yields two smooth curves whose tangents define the *asymptotic directions* T_1, T_2 in the point under consideration (c). *Right:* approximate asymptotic directions of the *Cour Visconti* surface (Fig. 1), computed with the jet fit method of (Cazals and Pouget, 2003).

Coming back to the discrete case, we set aside entirely the case of developable surfaces which have $K = 0$ everywhere. Apart from the rare instances where a vertex exactly marks the boundary between $K > 0$ and $K < 0$ we have non-proper Gauss images with self-intersections only if the geometry of the primal mesh is so convoluted that it is hard to even define a normal vector. We therefore formulate the main requirement for smoothness (see Figure 4):

Definition 1. *A triangle mesh is smooth, if all Gauss images of vertices are free of self-intersections.*

2.2. **Relation between Gauss image and asymptotic lines.** Closer study reveals that smoothness in the sense of Definition 1 is related to local shape properties of the surface, in particular Dupin's indicatrix and asymptotic directions, for which the reader is referred to Figure 5 or textbooks like (do Carmo, 1976). We state:

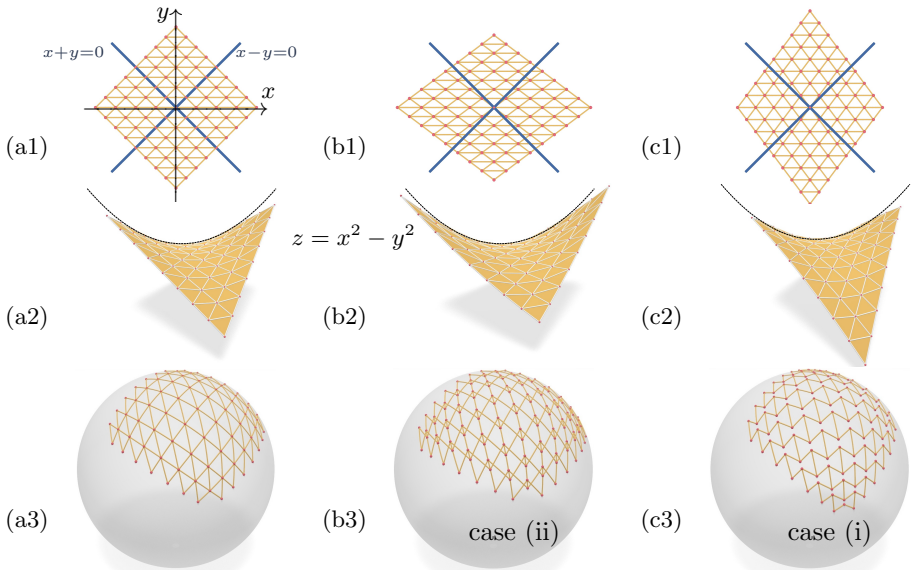

FIGURE 6. *Relevance of edge orientations for smoothness.* The graph of the function $z = x^2 - y^2$ carries two families of straight lines which correspond to $x \pm y = const$, and which are also the asymptotic directions. Images (a1)–(c1) show different tilings of the xy plane by triangles, which in (a2)–(c2) are lifted to the graph surface and generate a mesh. Their respective Gauss images have dual faces without self-intersections in (c3), and with self-intersections in (b3). Image (a3) illustrates a boundary case where self-intersections begin to occur, and where the edges coincide with the surface's asymptotic directions.

Observation 2. *Assume a mesh approximates a smooth saddle-shaped surface, and that the vertex* $\mathbf{v}$ *has face cycle* $f_1, \ldots, f_6$ *(indices modulo 6). Then, we typically have the following properties of the Gauss image hexagon* $g(\mathbf{v})$*:*

(i) $g(\mathbf{v})$ *has no self-intersections, if the quadrants bounded by the asymptotic directions do not contain faces except for a pair* f_k, f_{k+3}*, which are contained in opposite quadrants.*

(ii) $g(\mathbf{v})$ *has self-intersections, if faces* f_k, f_{k+1} *are both contained in the same quadrant between asymptotic directions, and* f_{k+3}, f_{k+4} *lie in the opposite quadrant.*

Figure 6 illustrates this phenomenon on a very simple surface. Situations (i) and (ii) correspond to Figure 6c and Figure 6b, respectively. Figure 7 demonstrates this observation by means of an actual freeform skin.

Observation 2 is not a mathematical statement, but it could be turned into one by specifying more clearly what is meant by "typically".[1]

[1]We argue as follows: the observation is true for the hyperbolic paraboloid and for lifted regular triangulations (this is an easy exercise, see Figure 6). It is true in the limit, for small faces, and limit-regular triangulations because of the hyperbolic paraboloid's capability of approximating a surface up to 2^{nd} order (thus approximating asymptotic directions and normal vectors up to 1^{st} order). The observation thus is true whenever the size of triangles is small enough and the triangulation is regular enough.

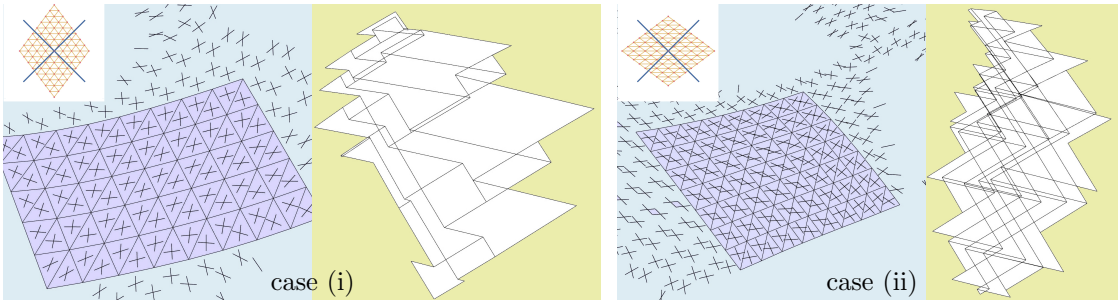

FIGURE 7. *Smooth and unsmooth meshes.* The blue subfigures show two patches of the *Cour Visconti* mesh, together with the asymptotic directions of the underlying reference surface within each face. The left and right hand patches correspond to cases (i) and (ii) of Observation 2. Consequently their respective Gauss images (green subfigures) exhibit few self-intersections in case (i) and many self-intersections in case (ii). Thus the left hand patch is revealed as smooth, the right hand patch as unsmooth. It must be admitted that these images are difficult to read, since this mesh has triangle pairs which together form a flat quadrilateral, so the Gauss image mesh has zero length edges.

Conclusions regarding mesh design. A fundamental question is to find a "smooth" triangulation of a given reference surface. The previous paragraphs give guidance for the combinatorially regular case with 6 edges per vertex: In the negatively curved regions of the reference surface, the positioning of edges must take the asymptotic directions into account. According to Observation 2, in each quadrant bounded by asymptotic directions we may place 1 or 2 edges, but not 3.

Thus the layout of a "smooth" triangulation essentially experiences the same combinatorial restrictions as the layout of quadrilateral meshes with planar faces, where in the negatively curved regions of a surface, the edges cannot deviate much from the principal curves, see (Zadravec et al., 2010).[2]

The design of "smooth" triangulations on a reference surface is therefore an instance of a well known dilemma: Choosing the surface determines much of the triangulation, and design freedom is limited. Further, it is generally not possible to optimize a triangle mesh towards smoothness by only slightly moving the vertices. Figure 7 is an instance of this, as will be shown below.

2.3. **Star-shaped Gauss images.** The constraint that Gauss image hexagons do not self-intersect is cumbersome to handle in optimization procedures. It is fortunate that another property, which is a bit stronger, is both easier to deal with and has interesting implications on the local shape of meshes. We define:

[2]Edges of smooth planar-quad meshes must follow two families of curves which constitute a conjugate network, see (Liu et al., 2006) and (Bobenko and Suris, 2008). Theoretically one family can be chosen arbitrarily and determines the other. However, in practice, the requirement of a minum angle between edges ensures that edge polylines cannot cross asymptotic curves, see (Zadravec et al., 2010).

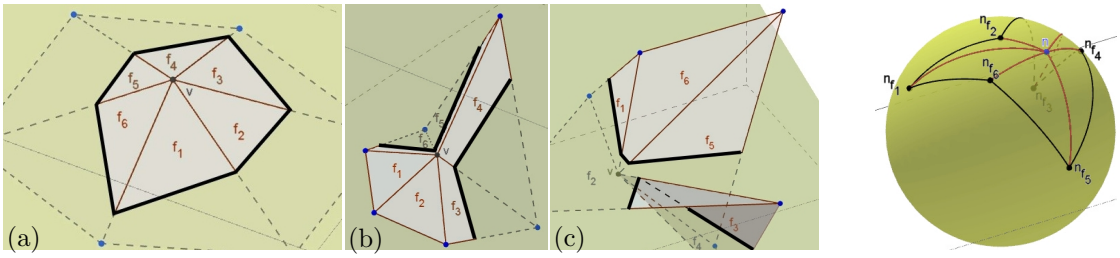

FIGURE 8. These images, taken from (Günther and Pottmann, 2016), illustrate Proposition 5. For a vertex $\mathbf{v}$ with a proper star-shaped Gauss image $g(\mathbf{v})$, the discrete indicatrix is either a discrete ellipse (i.e., a convex polygon, subfigure a) or a discrete hyperbola (i.e., it consists of two convex arcs, subfigures b,c). The Gauss image corresponding to subfigure c is shown at right.

Definition 3. *The Gauss image $g(\mathbf{v})$ is star-shaped if it has no self-intersections and there is a point $\widetilde{\mathbf{n}}_\mathbf{v}$ in its interior which can be connected to the entire circumference of the Gauss image by spherical arcs contained in that interior.*

Figures 8 and 9 show examples. In order to properly formulate the *shape properties* of meshes with star-shaped Gauss images, we recall the Dupin indicatrix of Figure 5 and define:

Definition 4. *Assume that a vertex $\mathbf{v}$ in a mesh with planar faces has a Gauss image $g(\mathbf{v})$ which is star-shaped with respect to $\widetilde{\mathbf{n}}_\mathbf{v}$. Intersecting the star of $\mathbf{v}$ with a plane close to $\mathbf{v}$ and orthogonal to $\widetilde{\mathbf{n}}_\mathbf{v}$ yields the discrete indicatrix.*

The meaning of "close to $\mathbf{v}$" is that the intersection shall not be disturbed by edges which are not incident with $\mathbf{v}$ itself. The following result, illustrated by Figure 8, has been shown by (Günther and Pottmann, 2016). It refers to the discrete Gauss curvature of triangle meshes, cf. (Banchoff, 1970).

Proposition 5. *Consider a vertex $\mathbf{v}$ in a mesh with planar faces. Its Gauss curvature is given by $K(\mathbf{v}) = 2\pi - \sum_{f\sim\mathbf{v}} \alpha_f$, where $\sum \alpha_f$ is the sum of all angles between successive edges incident with that vertex. Then, the following holds:*

(i) If $K(\mathbf{v}) > 0$ and $g(\mathbf{v})$ is free of self-intersections, then $g(\mathbf{v})$ is star-shaped and any indicatrix is a discrete ellipse, i.e., a convex polygon.

(ii) If $K(\mathbf{v}) < 0$ and $g(\mathbf{v})$ is star-shaped with respect to some point $\widetilde{\mathbf{n}}_\mathbf{v}$, then the corresponding indicatrix typically[3] is a discrete hyperbola, i.e., consists of two convex polygonal arcs. Also the reverse implication is true.

We conclude that star-shaped Gauss images imply that the local shape of a mesh in the immediate vicinity of a vertex coincides with what is expected from the local shape of a smooth surface (in particular the manner of up-down oscillations w.r.t. a fictitious tangent plane, and the convexity of intersections with near-tangent planes). This means that insisting on star-shaped Gauss images makes triangle meshes even more smooth than Definition 1 already does.

[3]The exceptions are cases where both $\widetilde{\mathbf{n}}_\mathbf{v}$ and $-\widetilde{\mathbf{n}}_\mathbf{v}$ are contained in the interior of $g(\mathbf{v})$.

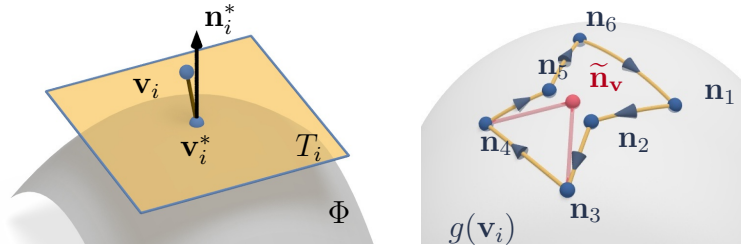

FIGURE 9. Constraints imposed on smooth meshes in our optimization procedure. *Left:* For purposes of approximation, vertices $\mathbf{v}_i$ of the mesh are kept close to the reference surface Φ, by adding the constraint that $\mathbf{v}_i$ lies in the tangent plane of Φ in the point $\mathbf{v}_i^*$ which was closest to $\mathbf{v}_i$ in the previous iteration of our optimization procedure. *Right:* The spherical polygon $\mathbf{n}_1, \ldots, \mathbf{n}_6$ is star-shaped w.r.t. the center $\widetilde{\mathbf{n}}_\mathbf{v}$ if all triangles $\mathbf{n}_k\mathbf{n}_{k+1}\widetilde{\mathbf{n}}_\mathbf{v}$ have the same orientation and it winds around $\widetilde{\mathbf{n}}_\mathbf{v}$ exactly once.

3. Optimization of meshes

We have implemented a procedure to optimize a mesh such that its Gauss images become star-shaped, which makes them "smooth" as explained in detail in the previous section. The method expresses each desired property in terms of an energy function. The variables in the optimization are positions $\mathbf{v}_i$ of vertices, normal vectors $\mathbf{n}_k$ of faces, and auxiliary normals $\widetilde{\mathbf{n}}_i$ of vertices. To express the relation between faces and normal vectors, we minimize the energy

$$E_{\text{normal}} = \sum_{\mathbf{v}_i\mathbf{v}_j \text{ is edge of face } f_k} (\mathbf{n}_k \cdot (\mathbf{v}_i - \mathbf{v}_j))^2 + \sum_{\text{faces } f_k} (\|\mathbf{n}_k\| - 1)^2.$$

We also ensure that these normal vectors are oriented consistently, i.e., cycling the face f_k in the positive sense when looking in direction $\mathbf{n}_k$, and cycling the face f_l in the positive sense when looking in direction $\mathbf{n}_l$, must assign different orientations to the common edge $f_k \cap f_l$.

Secondly, if the mesh is to approximate a reference surface Φ, we should try to minimize something like $\sum \text{dist}(\mathbf{v}_i, \Phi)^2$ which is highly nonlinear. However, we replace Φ by the tangent plane T_i in the point $\mathbf{v}_i^*$ of Φ which is closest to Φ. Thus the highly nonlinear squared distance function is substituted by its quadratic Taylor approximation without disturbing convergence of algorithms, cf. Pottmann et al. (2006). In each round of our iterative optimization procedure, we recompute the closest point $\mathbf{v}_i^*$ and the normal vector $\mathbf{n}_i^*$ there. The energy expressing closeness then reads

$$E_{\text{close}} = \sum_{\text{vertices } \mathbf{v}_i} ((\mathbf{v}_i - \mathbf{v}_i^*) \cdot \mathbf{n}_i^*)^2.$$

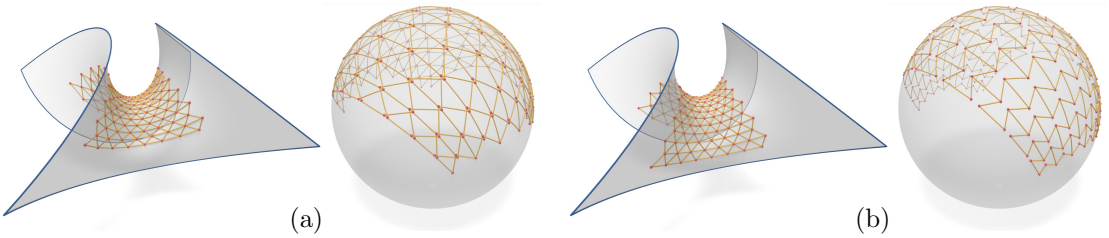

FIGURE 10. This mesh on a minimal surface together with its Gauss image (subfigure a) undergoes optimization. All Gauss image hexagons of vertices become star-shaped (subfigure b). These images illustrate the fact that the non-smoothness of certain meshes (like the one in subfigure a) may not be visible in all renderings.

Thirdly, the Gauss image polygon $\mathbf{n}_1, \mathbf{n}_2, \ldots$ of a vertex $\mathbf{v}_i$ is star-shaped with respect to the normal vector $\widetilde{\mathbf{n}}_i$ only if all triangles $\widetilde{\mathbf{n}}_i \mathbf{n}_k \mathbf{n}_{k+1}$ have the same orientation when we look at them in the direction of $\widetilde{\mathbf{n}}_i$. We therefore let

$$E_{\text{gauss}} = \sum_{\text{vertices } \mathbf{v}_i} \sum_{f_k \text{ in face cycle of } \mathbf{v}_i} \left(\det(\mathbf{n}_{k+1}, \mathbf{n}_k, \widetilde{\mathbf{n}}_i) - \omega_{ik}^2 \right)^2,$$

where ω_{ik} is a slack variable. This condition is also sufficient for star-shapedness if the polygon winds around $\widetilde{\mathbf{n}}_i$ exactly once (this is checked a posteriori, by computing angle sums). To prevent zigzag in edge polylines, we use the classical second order differences

$$E_{\text{polylines}} = \sum_{\text{successive vertices } \mathbf{v}_i, \mathbf{v}_j, \mathbf{v}_k} \|\mathbf{v}_i - 2\mathbf{v}_j + \mathbf{v}_k\|^2.$$

The total energy is a weighted linear combination of the individual energies:

$$E = w_1 E_{\text{normal}} + w_2 E_{\text{close}} + w_3 E_{\text{gauss}} + w_4 E_{\text{polylines}}.$$

Figure 10 shows the result of optimization on a simple surface.

Implementation details. Since the limit residual of the polyline fairness energy is nonzero, it is used with a low weight, in the manner of an additional regularizer. We further use units such that the typical edge lengths in the mesh are of magnitude 1. Then we may let $w_1 = w_2 = w_3 = 1$ and $w_4 = 0.01$, but some user experimenting is necessary for good results. For the actual minimization of the combined energy, we use a standard Gauss-Newton method, cf. (Kelley, 1999, pp. 22–23).

Discussion of results. Figures 11 and 12 show the behaviour of two different meshes which undergo optimization. In one case optimization is not successful, as can be seen in Figure 11c. This is not the fault of the method but rather the fault of the design itself which places mesh polylines relative to asymptotic directions such that case (ii) of Observation 2 applies. It depends on the nature of the mesh if optimization manages to move vertices such that smoothness can be achieved or not. In the case of Figure 12 this works because there is not much to do: Vertices

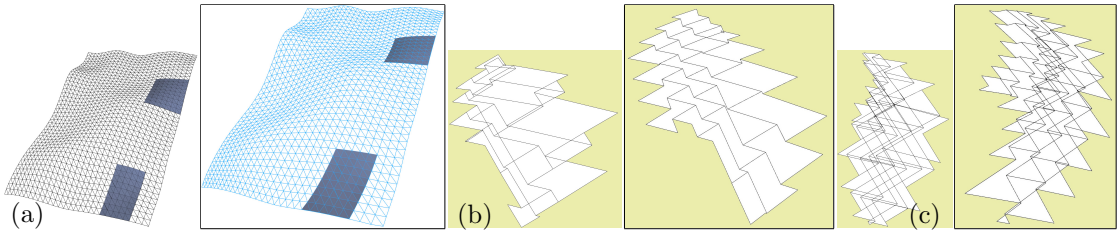

FIGURE 11. *Partly successful optimization.* The three smaller images show the *Cour Visconti* mesh of Figure 1 (a) and the Gauss images of two selected patches (b,c) before optimization. The larger figures show the situation after optimization has been performed. Since the right hand patch corresponds to case (ii) of Observation 2, optimization can hardly be successful unless we entirely rearrange the mesh layout. Our optimization procedure does not do that; rather it applies small changes which may be acceptable as an augmentation of an already existing design.

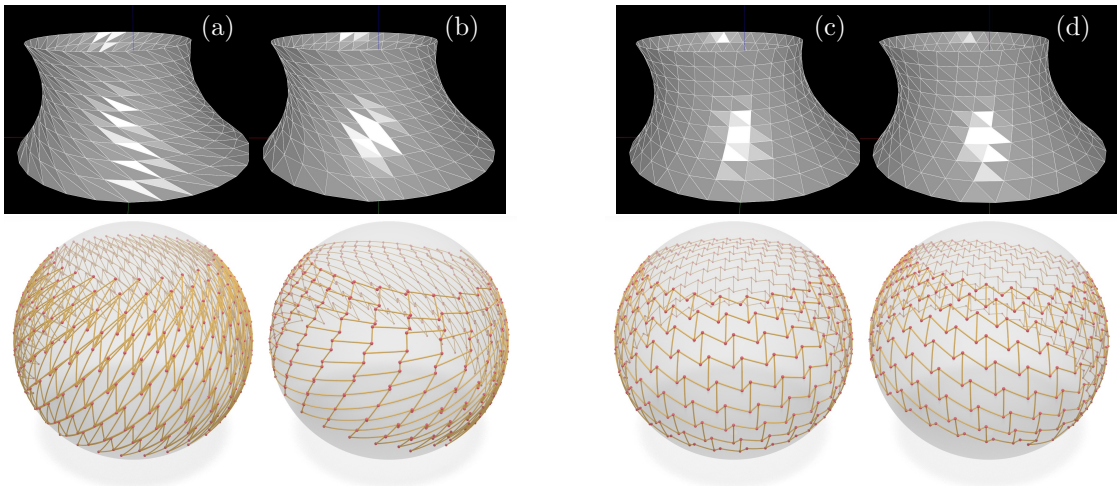

FIGURE 12. *Successful optimization.* The mesh in (a) is inspired by the skin of the *BMW Welt* building in Munich, cf. Figure 2. It is obviously unsmooth and yields mesh (b) under optimization. Below each mesh, the respective Gauss images are shown. Another triangulation of the same reference shape (c) is weakly smoooth because Gauss images of vertices are free of self-intersections, but are not star-shaped. Optimization yields mesh (d). The meshes are rendered as reflective surfaces, which allows visual inspection of smoothness.

only have to move along parallel circles a bit. In the case of Figure 11 this is not possible without completely rearranging the mesh.

A more complex example is the Eindhoven *Blob* by M. Fuksas. Figure 13 illustrates how close the optimized mesh is to the original one, and illustrates the change in Gauss images.

Comparison with other smoothing methods. There is a host of smoothing methods available in the area of geometry processing, starting with very simple methods like Laplacian smoothing a.k.a. linear diffusion (this means moving each vertex towards the average of its neighbours, see e.g. (Botsch et al., 2010)). However, most methods deal with removing noise from the shape which is described by the

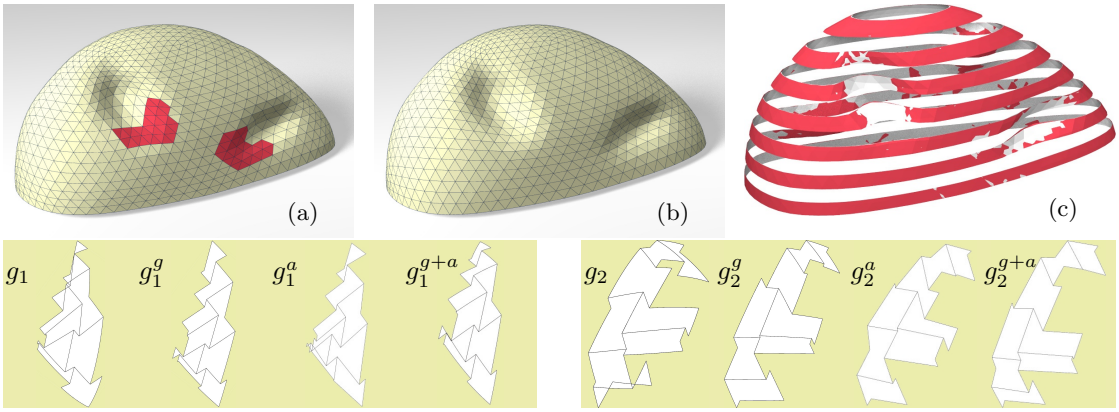

FIGURE 13. *Optimization using different energies.* In (a) we see the triangle mesh used for the Eindhoven *Blob*, together with patches No. 1 and No. 2 highlighted. The result of optimization towards star-shaped Gauss images is shown in (b). A superimposed image of slices through original mesh and optimized mesh (c) shows approximation quality. The detailed images in the bottom row show Gauss images of patch No. i ($i = 1, 2$) before optimization (labelled g_i), after optimization using E_{gauss} (labelled g_i^g), or E_{angle} (labelled g_i^a), or both energies (labelled g_i^{g+a}).

mesh, and this is not our intention. Methods which seek to represent a reference surface by a better mesh are referred to as *remeshing*, which usually means to discard the previous mesh altogether. This is also not what we are doing. Actually, from the viewpoint of geometry processing, our smoothing procedure hardly does anything at all, which is true if one forgets the important visual role which vertices and edges play in our applications. Being aware of the different aims of other smoothing methods, we really made only few comparisons, and we only observed the behaviour of meshes as they undergo Laplacian smoothing. While for some meshes like the *Blob* of Figure 13, this procedure produces almost acceptable results, it does not improve the meshes of Figure 10 and Figure 12 at all.

We might also ask a different question: What happens if we directly minimize the kink angles α_{kl} between faces f_k, f_l? With $\cos\alpha_{kl} = \mathbf{n}_k \cdot \mathbf{n}_l$ it is easy to set up an energy which directly penalizes large kink angles, namely

$$E_{\text{angle}} = \sum_{\text{edges } f_k \cap f_l} (1 - \mathbf{n}_k \cdot \mathbf{n}_l)^2.$$

The result of optimization using this energy combined with the one producing star-shaped Gauss images is illustrated by Figure 13. One can see that optimizing kink angles has an effect similar to making the Gauss image star-shaped, but weaker. Statistics show that between these two kinds of optimization (or the combined optimization of both) there is no substantial difference in kink angles. We therefore conclude that optimizing E_{gauss} can be augmented by adding E_{angle} to the total energy, but should not be replaced by it.

4. Conclusion

We have presented a two-stage definition of "smoothness" of a triangle mesh in terms of the Gauss image of vertices. A weaker version requires absence of self-intersections, a stronger one requires that Gauss images are star-shaped. We discussed the relation between smoothness and the placement of edges relative to the asymptotic directions. We conclude that in negatively curved areas, we have strong combinatorial restrictions on the placement of edges if we want the mesh to be smooth. If the stronger smoothness condition is fulfilled, we can even deduce that the piecewise-flat mesh surface has local shape properties analogous to smooth surfaces (which justifies our definition of smoothness). Finally, we show the optimization of a mesh towards smoothness and discuss in which cases this optimization can succeed.

Acknowledgments

This research has been supported by the German Research Foundation (DFG) and the Austrian Science Fund (FWF) within the framework of the SFB-Transregio Programme 109, *Discretization in Geometry and Dynamics*.

References

Banchoff, T. E. (1970). Critical points for embedded polyhedral surfaces. *Amer. Math. Monthly 77*, 475–485.

Bobenko, A. and Yu. Suris (2008). *Discrete differential geometry: Integrable Structure*. American Math. Soc.

Botsch, M., L. Kobbelt, M. Pauly, P. Alliez, and B. Levy (2010). *Polygon Mesh Processing*. A. K. Peters.

Cazals, F. and M. Pouget (2003). Estimating differential quantities using polynomial fitting of osculating jets. In *Proc. Symp. Geometry Processing*, pp. 177–187. Eurographics Association.

do Carmo, M. (1976). *Differential Geometry of Curves and Surfaces*. Prentice-Hall.

Günther, F. and H. Pottmann (2016). Smooth polyhedral surfaces. preprint.

Kelley, C. (1999). *Iterative methods for optimization*. SIAM.

Liu, Y., H. Pottmann, J. Wallner, Y.-L. Yang, and W. Wang (2006). Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graphics 25*(3), 681–689. Proc. SIGGRAPH.

Pottmann, H., Q. Huang, Y.-L. Yang, and S.-M. Hu (2006). Geometry and convergence analysis of algorithms for registration of 3D shapes. *Int. J. Computer Vision 67*(3), 277–296.

Zadravec, M., A. Schiftner, and J. Wallner (2010). Designing quad-dominant meshes with planar faces. *Computer Graphics Forum 29*(5), 1671–1679. Proc. Symp. Geometry Processing.