Variational Tangent Plane Intersection for Planar Polygonal Meshing

Henrik Zimmer and Marcel Campen

RWTH Aachen University, Computer Graphics Group

Ralf Herkrath

RWTH Aachen University, Chair of Structures and Structural Design
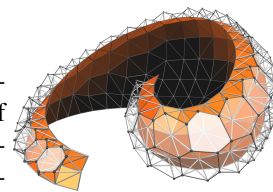
Leif Kobbelt

RWTH Aachen University, Computer Graphics Group

Abstract. *Several theoretical and practical geometry applications are based on polygon meshes with planar faces. The planar panelization of freeform surfaces is a prominent example from the field of architectural geometry. One approach to obtain a certain kind of such meshes is by intersection of suitably distributed tangent planes. Unfortunately, this simple tangent plane intersection (TPI) idea has a number of limitations. It is restricted to the generation of hexagon-dominant meshes: as vertices are in general defined by three intersecting planes, the resulting meshes are basically duals of triangle meshes. Furthermore, the explicit computation of intersection points requires dedicated handling of special cases and degenerate constellations to achieve robustness on freeform surfaces. Another limitation is the small number of degrees of freedom for incorporating design parameters.*

Using a variational re-formulation, we equip the concept of TPI with additional degrees of freedom and present a robust, unified approach for creating polygonal structures with planar faces that is readily able to integrate various objectives and constraints needed in different applications scenarios. We exemplarily demonstrate the abilities of our approach on three common problems in geometry processing.

1 Introduction

The problem of generating polygonal meshes with planar faces of given freeform surfaces has been a topic of intense research in recent years. Backed up by theoretical results, e.g., on their connection to discrete conjugate curves networks [Bobenko and Suris 2008] or to discrete conformal and minimal surfaces [Bobenko et al. 2003, Müller 2011], a wide range of well founded techniques based on these concepts have been developed leveraging the constructional and architectural benefits of such structures. Such works include [Liu et al.
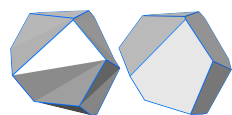
2006, Pottmann et al. 2007a, Pottmann et al. 2007b, Zadravec et al. 2010, Wang et al. 2008, Troche 2008, Eigensatz et al. 2010] discussing the economical advantages of manufacturing planar panels in general and the convenience of the (multi-layer) support structures derivable from such structures in particular.

Background. Common methods generate polygonal meshes with planar faces by means of non-linear optimization (perturbation) of the vertices of given input meshes. This *planarization* is typically a delicate procedure, the stability of which heavily depends on the input configuration. The close connection between conjugate curve networks (such as curves of principal curvature directions) and quad meshes with planar faces suggests using quad meshes with elements aligned along such curves on the surface since they are then expected to be approximately planar and facilitate the success of the optimization.
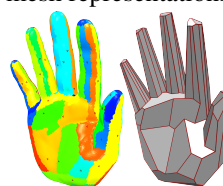
Similarly the triangulation algorithms proposed by [Troche 2008, Wang et al. 2008], used for constructing dual hexagonal meshes with planar faces, also suggest placing vertices of the triangulations along conjugate curves on the freeform surface.
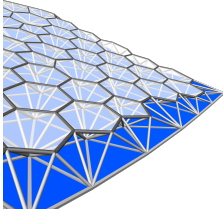
The Variational Tangent Plane Intersection (VTPI) approach of this paper uses tangent plane duality to create a polygon mesh with planar faces, similar to the TPI approach presented by [Troche 2008]. However, our variational re-formulation of the problem allows us to add new degrees of freedom, which do not only enable the introduction of additional design parameters, but also enables us to remove the typical restriction of TPI to triangle input and hexagon-dominant output meshes. The optimization formulation further eliminates the otherwise often needed handling of instabilities arising from the explicit intersection computations in developable areas of the surface. In addition, using any of the vast number of available free and commercial solver software, this formulation can easily be augmented by further constraints and energies for controlling the output meshes.

Applications. In this paper we demonstrate the applicability of VTPI on three common problems in geometry processing. For each problem scenario we propose appropriate energies and constraints for controlling the particular behavior, and show and discuss results. The first scenario is mesh planarization. We demonstrate how quad meshes can be planarized by applying VTPI to the dual of the given mesh, yielding perfectly planar faces. One notes that this necessitates intersections involving more than three tangent planes, not possible with standard TPI. The second problem is the generation of coarse mesh representations with planar faces from freeform surfaces, similar to [Cohen-Steiner et al. 2004] where, however, planarity was not enforced. When applying the technique to an architectural scenario demanding exact planarity, [Cutler and Whiting 2007] mention the problems of the intersection of several, possibly poorly aligned or even co-planar, planes while simultaneously respecting optimized cluster normals or preferred intersection points. With VTPI such problems can appropriately be taken care of.

The third application deals with tri/hex dual- and multi-layer structures, demonstrating the important case where an inner (or outer) structure is to be covered by a dual outer (or inner) structure consisting of planar faces. Such structures are of interest for different types of support structures, such as node/frame-based MERO[1]-like construction systems or the recently proposed point folding structures of [Trautz and Herkrath 2009, Zimmer et al. 2012]. The difference to the first scenarios is that such structures cannot always be considered independently from each other. We hence introduce further constraints to prevent the inner and outer layer from intersecting each other while energy functionals try to place the dual vertices at desirable positions. Also the importance of the quality of the input meshes is discussed.

Overview. After briefly reviewing the TPI idea of [Troche 2008] Section 2 presents the proposed variational formulation. In Section 3 this optimization problem is further extended by a simple set of local constraints useful for creating non-intersecting multi-layer structures. Finally in Section 4 experiments and results of the three above mentioned applications are shown and Section 5 concludes the paper.

2 Tangent Plane Intersection

For a triangulation of a freeform surface [Troche 2008] computes a dual hexagon-dominant mesh by intersection of (vertex) tangent planes. Here the tangent planes of the three incident vertices of each triangle define an intersection point. In this way also every vertex of the triangle mesh corresponds to a dual planar face, whose corners are defined by the intersection points of the (typically six) triangles incident to that vertex. The edges between dual faces lie in the intersection lines of the corresponding tangent planes as shown in Figure 1. Note that, as three planes are necessary to define a unique intersection point, the approach cannot directly be transferred to arbitrary polygons. The intersection point $\mathbf{x} = (x, y, z)^\top \in \mathbb{R}^3$ of
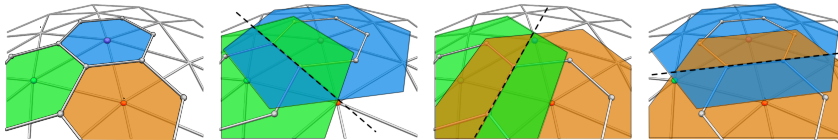

Figure 1: The plane of each dual face is defined by the tangent plane of the respective primal vertex. The dual edges lie in the pairwise intersections (dotted) of these tangent planes.

three non-co-planar tangent planes defined by normal vectors $\mathbf{n}_i \in \mathbb{R}^3$, $\|\mathbf{n}_i\| = 1$ and vertices $\mathbf{v}_i \in \mathbb{R}^3$ respectively is defined by the three equations

$$\mathbf{n}_i^\top \mathbf{x} = \mathbf{n}_i^\top \mathbf{v}_i, \quad i \in 0, 1, 2, \tag{1}$$

---

[1] www.mero.de

which can be written as a 3×3 linear system $N\mathbf{x} = \mathbf{b}$.

Note that while the explicit computation $\mathbf{x} = N^{-1}\mathbf{b}$ is not possible for rank-deficient N (co-planar tangent planes), the validity of the three equations (1) can always be checked for a given $\mathbf{x}$. This observation suggests that an intersection can be enforced by using (1) as boundary conditions for an optimization problem. An additional energy can be specified to place intersection points close to preferred positions in the unstable case of tangent plane co-planarity. Even where the intersection points are numerically well defined, they might lie quite far away from the intended surface, making it desirable to be able to prescribe preferred positions also in such stable cases.

The main idea of VTPI is to re-cast the basic TPI as a non-linear optimization problem in this way, leading to a unified approach, which allows for further quality control by specifying such additional constraints and energies (cf. Section 2.1). However, for these to have any effect, additional degrees of freedom are necessary. We propose to set the normals $\mathbf{n}_i$, which until now were considered fixed, free as additional variables to optimize. And by further allowing a parallel offset of the tangent planes along the normals, we gain even more flexibility (cf. Section 2.2).

2.1 Variational Formulation of TPI

Given a 2-manifold triangle mesh $\mathcal{M}$ with n_V vertices, n_E edges and n_F faces, we now formulate the basic optimization, which will be expanded on in the rest of the section and in the Applications:

$$minimize \quad F(\{\mathbf{x}_j\}) \quad s.t. \quad E = E_{\text{int}} := \{N_j\mathbf{x}_j - \mathbf{b}_j = 0\}_{j=1}^{n_F}, \tag{2}$$

where F is a, yet to be defined, quadratic energy functional and E_{int} a set of linear equality constraints. E_{int} enforce that the unknown intersection point $\mathbf{x}_j$ of each triangle lies in all its three incident tangent planes. This indirect intersection is the key to making tangent plane intersections stable, as it avoids the explicit intersection computations. Counting the degrees of freedom it is clear that for general configurations the minimization (2) yields the same predetermined intersections $\{\mathbf{x}_j\}_{j=1}^{n_F}$ as explicit TPI and that the energy F has no influence. However, in areas of zero Gaussian curvature, some intersection points are not well defined. These ambiguities can be handled by specifying preferred intersection points $\mathbf{f}_j$ (e.g., the barycenter of the primal face) and defining an appropriate positional energy $F(\{\mathbf{x}_j\}) := F_{\text{pos}} = \sum_{j=1}^{n_F} \|\mathbf{x}_j - \mathbf{f}_j\|^2$. While in the current simple case this functional merely controls the behaviour in degenerate configurations, it comes to its full use when we add additional degrees of freedom in the following.

2.2 Additional Degrees of Freedom

We again consider the problem formulation from above; given a set of preferred intersection points of the dual structure, we are looking for the (primal) vertex normals and corresponding offsets that minimize the deviation of the tangent plane intersections from the given points. How these extensions are added to the optimization problem (2) is described in the following.

Setting the Normals Free. Letting the normal vectors $\{\mathbf{n}_i\}_{i=1}^{n_V}$ vary, the functional has the form $F(\{\mathbf{x}_j\}, \{\mathbf{n}_j\})$. We have $3(n_F + n_V)$ unknowns and the equality constraints E_{int} are now quadratic as both the $\mathbf{n}_i$ and $\mathbf{x}_j$ are variables. To guarantee (unit) normals, E is further augmented by a set of quadratic constraints $E_{\text{norm}} := \{\|\mathbf{n}_i\|^2 - 1 = 0\}_{i=1}^{n_V}$.

Offsetting the Tangent Planes. An offset factor h is introduced into the plane equation (1) as follows $\mathbf{n}^\top(\mathbf{x} - (\mathbf{v} + h\mathbf{n})) = 0 \Rightarrow \mathbf{n}^\top\mathbf{x} = \mathbf{n}^\top\mathbf{v} - h$. The constraints E_{int} now additionally depend on the unknown offsets $\{h_i\}_{i=1}^{n_V}$ and the total number of variables is $3n_F + 4n_V$. The energy functional has the form $F(\{\mathbf{x}_j\}, \{\mathbf{n}_i\}, \{h_i\})$.

2.3 Solving the Optimization Problem

For the optimization we use $\texttt{Ipopt}$[2], an open source implementation of the primal-dual interior point method by [Wächter and Biegler 2006]. Note that such methods inherently allow for inequality constraints and can be used to optimize more general problems of the form:

$$\begin{aligned} \textit{minimize} \quad F(\mathbf{X}) \quad s.t. \quad & E(\mathbf{X}) := \{E_i(\mathbf{X}) = 0\}_{i=1}^{n_=} \\ & I(\mathbf{X}) := \{I_j(\mathbf{X}) \geq 0\}_{j=1}^{n_>} \end{aligned} \tag{3}$$

where $\mathbf{X}$ is the vector of unknowns and $E(\mathbf{X})$ and $I(\mathbf{X})$ equality and inequality constraints.

In the basic setting (above) $F(\mathbf{X})$ is the simple positional energy, $\mathbf{X} \in \mathbb{R}^{3n_F + 4n_V}$, $E(\mathbf{X}) = \{E_{\text{int}}, E_{\text{norm}}\}$ and there are no inequalities ($I(\mathbf{X}) = \emptyset$), hence we are not utilizing the full potential of (3). In the next section the optimization formulation is further extended by introducing simple but powerful inequalities for avoiding (local) intersections between a primal and a dual structure. These can be easily integrated into the minimization (3).

Until now, we only considered the generation of hexagon-dominant meshes with planar faces. Section 4.1 shows how this can be extended to the planarization of general polygons.

2.4 Remarks

We first note that after adding more degrees of freedom to the problem (cf. Section 2.2) not all variables have to occur in the energy functional. For example, in F_{pos} only the unknown intersections $\{\mathbf{x}_j\}_{j=1}^{n_F}$ appear – the other variables are linked via the constraints E_{int}.

It is also not even necessary for *any* of the variables to occur in F, the solver would then just output the first valid solution (fulfilling the constraints). One practical example would be to specify *hard* inequality constraints forcing the intersections to lie, e.g., withing the extrusion of the base triangle or at a certain height, instead of using a *soft* energy such as F_{pos}.

---

[2]https://projects.coin-or.org/Ipopt
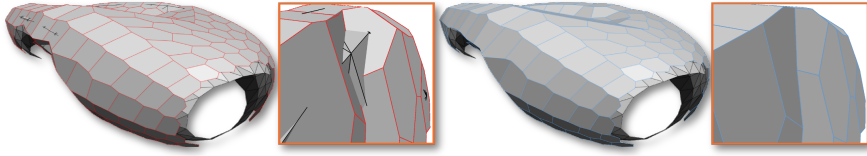
H. Zimmer, M. Campen, R. Herkrath, and L. Kobbelt

Figure 2: Red: a naïve dual structure with planar faces penetrating the primal structure in different ways. Blue: non-penetrating dual structure utilizing the constraints of Section 3.

A short note on the implementation: while [Wächter and Biegler 2006] already offers a comfortable c++ interface we additionally use the NSolver wrappers from [Bommes et al. 2010] for an almost effortless definition and prototyping of energies and constraints, with the possibility of easily exchanging solver backends.

3   Local Constraints for Multi-Layer Dual Structures

In architecture multi-layer structures are typically required to be non-intersecting. One layer carries panels (e.g., glass or inflated membrane cushions) and the other one has a structural function. Here we consider the problem of finding a hexagon (dominant) dual mesh not intersecting a given triangle mesh.

Using the method above a preferred offset distance between the structures can be prescribed by defining appropriate vertex positions $\{\mathbf{f}_j\}_{j=1}^{n_F}$, however, mesh curvature and tessellation can still lead to intersections between the structures (cf. Figure 2).

We devise three simple geometric constraints for avoiding (local) intersections which can be easily integrated into the above optimization framework. The three constraints are based on three identified intersection scenarios: *vertex, face and*

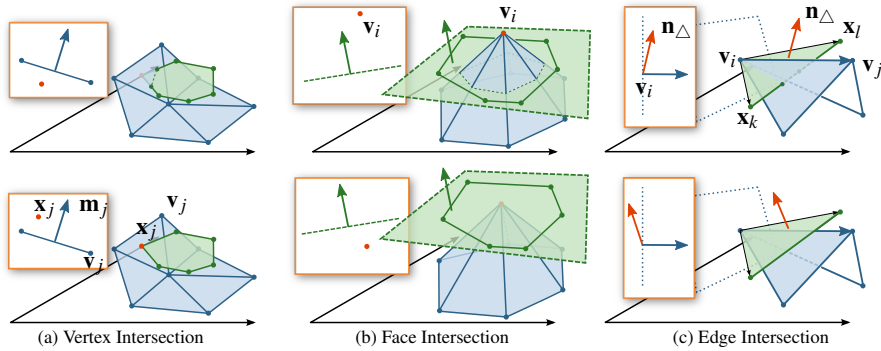(a) Vertex Intersection   (b) Face Intersection   (c) Edge Intersection

Figure 3: Local intersection scenarios between a primal (blue) and dual (green) structure. (a) A simple projection suffices to check whether a dual vertex lies "in front of" a primal face. (b) Likewise, one needs to check if a dual face lies "above" the primal vertex. (c) A dual edge can penetrate two neighboring primal faces, lying "underneath" the primal edge. A constraint can be derived by comparing the normal (red) of the plane spanned by the dual edge and a primal vertex with the primal edge.

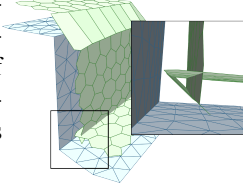*edge intersections* detailed below (cf. Figure 3).

Vertex Intersections   occur when a dual vertex lies behind a primal face (cf. Figure 3 (a)) and can easily be avoided by forcing the intersection point to lie in front of the face: inequalities $\{\mathbf{m}_j^\top (\mathbf{x}_j - \mathbf{v}_j) \geq 0\}_{j=1}^{n_F}$, where $\mathbf{v}_j$ is any (fixed) incident vertex of face j and $\mathbf{m}_j$ the (fixed) face normal, are added to $I(\mathbf{X})$.

Face Intersections   occur when a primal vertex pokes through a dual face (cf. Figure 3 (b)) we avoid this by demanding the tangent planes to lie above the primal vertices $\mathbf{v}_i$. The inequalities $\{h_i \geq 0\}_{i=1}^{n_V}$ are added to $I(\mathbf{X})$.

Edge Intersections   occur when a dual edge lies "below" the corresponding primal edge (cf. Figure 3 (c)). As a primal edge alone does not define a unique above or below relation, it is unclear how to express this as direct positional constraints on the dual vertices $\mathbf{x}_k$ and $\mathbf{x}_l$ at the primal edge e_{ij} between $\mathbf{v}_i$ and $\mathbf{v}_j$, instead we utilize the orientation inherent in the 2-manifold base mesh and formulate this as an orientation problem: If the normal of the triangle $\triangle_{ikl}$ points away from the edge e_{ij} the dual edge lies "above" the primal, otherwise "below". The inequalities $\{((\mathbf{x}_k - \mathbf{v}_i) \times (\mathbf{x}_l - \mathbf{v}_i)) \cdot (\mathbf{v}_j - \mathbf{v}_i) \geq 0\}_{e_{ij} \in \mathcal{M}}$ are added to $I(\mathbf{X})$.

Note that the edge-intersection constraints depend on the relative position of $\mathbf{x}_k$ and $\mathbf{x}_l$. While this simple formulation was sufficient for our experiments, further constraints could be implemented to guarantee the correct orientation (cf. Section 2.4). Figure 2 demonstrates the effect of adding the above constraints to (3) and further results are shown in Section 4.3.

Discussion.   Note, that these constraints prevent local intersections between the primal and the dual structure. Unsuitable offset-distances may still lead to intersections of the dual structure with itself as depicted in the inset. Aesthetic architectural meshes and practical offset-distances are, however, typically not very prone to such problems.

4   Applications

4.1   Planarization

To planarize a given polygon mesh $\mathcal{M}$ with non-triangle faces a dual (virtual) mesh $\mathcal{M}^*$ is created and submitted to the VTPI of Section 2 which then produces a planarized version of $\mathcal{M}$. The only difference to the previous setting is that the input mesh $\mathcal{M}^*$ is generally no longer a triangle mesh and one needs to make sure that the n tangent planes of an n-polygonal face all intersect at the same point $\mathbf{x}$. This is achieved by simply adding more equations of the type (1) to E_{int} in (2), yielding a $n \times 3$ dimensional matrix N for each face. Note that this overdetermined system would in general have no solution for fixed tangent planes (as in the standard TPI approach). The additional degrees of freedom provided by free normals and offsets

in VTPI facilitate such intersections. The vertices of $\mathcal{M}$ can be used as preferred points of intersection.

Here we briefly demonstrate the results of this approach and compare them to the recently proposed planarizing flow of [Alexa and Wardetzky 2011]. Figure 4 shows the results of both methods on the FERTILITY and FELINE quad mesh. The results are comparable but the methods strongly differ.

The planarizing flow is practically parameter-free, simple to use and yields high-quality planarizations. However, extending the method by prescribing additional constraints and integrating it into an optimization framework such as [Wächter and Biegler 2006] requires computing Jacobians (and possible Hessians), a delicate task for the SVD involved in the computation of the Laplacian. The VTPI presented here only depends on simple-to-derive polynomials, but does necessitate specifying weighting factors for the linear combination of energy functionals.

Energies. Additional to the simple positional energy in (1) our experiments yielded that energies preserving lengths of edges and face diagonals, as well as the directions of edges are useful for controlling the quality of the planarized version. Figure 5 briefly shows a few exemplary combinations of these energies. The "HHH" energy weighting is identical to the setting used for both examples in Figure 4.
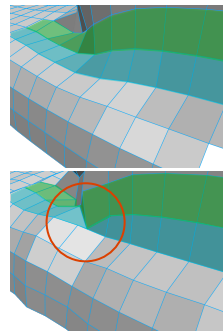
As the different functionals have different value ranges (and partly also different polynomial degrees) choosing good weighting factors depends on the scale of the input object, the type of functionals and their respective relations. For this reason we just state approximative relational sizes (Low/High) between the weights.

Discussion. One notices an interesting effect when observing the planarization of quad-strips that undergo $\geq 90°$ rotations along their trajectory in the non-planar state, e.g., a strip that goes from the flat pedestal on the FERTILITY around her feet or the strip coming from behind her back going onto the pedestal. The top figure on the right shows a close up of two strips. The planarization of such strips is handled quite differently by the two methods.

VTPI tends to concentrate these rotations to single points (see bottom figure on the right) yielding "triangles" (degenerate quads), whereas the method of [Alexa and Wardetzky 2011] tends to preserve the quad structure better at the expense of introducing ripples and plateaus (cf. the marked regions in Figure 4).

An interesting direction for future work would be to investigate if the identification of such turning points can help guide a manual or an automatic (re-)placement of nearby singularities or the introduction of new singularities needed for a better planarization.

The running time of our method is highly problem dependent. A torus can be tessellated with $10k$ planar hexagons in less than 15 sec, while the topologically and geometrically more involved FELINE model took about 2.5 min. to planarize.
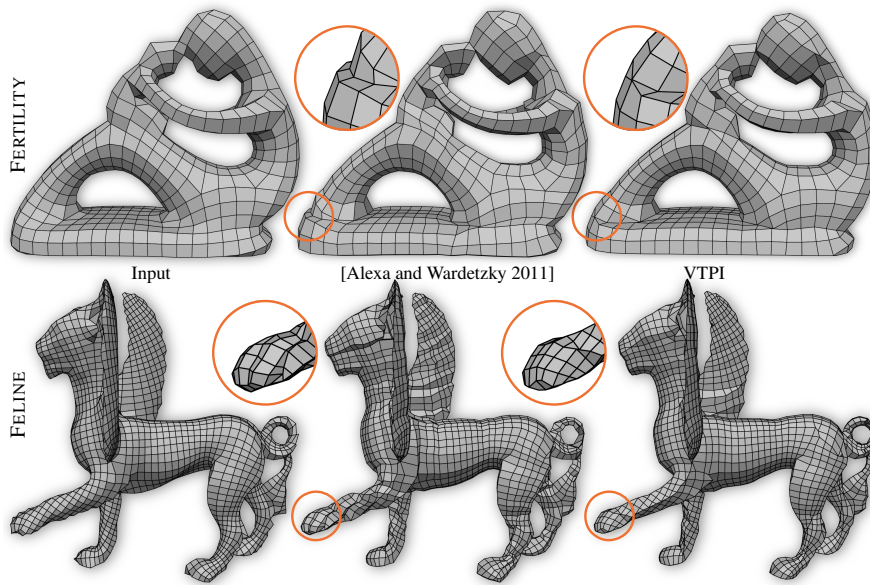
Figure 4: The two compared methods yield very similar results in good-natured areas. However, in areas where the input quads are not aligned to the principal curvatures (front leg of FELINE) the results strongly differ. Also see the Discussion in Section 4.1.
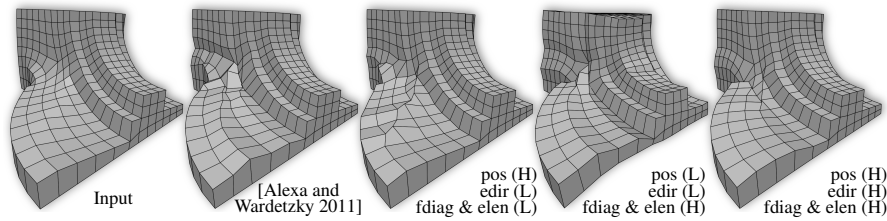

Figure 5: Results for different parameters for planarization. When primarily preserving the shape of the quads ("LLH") ripples similar to [Alexa and Wardetzky 2011] occur. Additionally enforcing high positional and directional constraints ("HHH") yields the mesh on the right.

4.2 Variational Shape Approximation

In the second test scenario we applied the VTPI to *Constrained Planar Remeshing for Architecture* [Cutler and Whiting 2007], a method based on *Variational Shape Approximation* [Cohen-Steiner et al. 2004] to approximate complex models by planar, efficiently manufacturable panels.

[Cutler and Whiting 2007] use a combination of different metrics for the Llyod relaxation algorithm to cluster the faces of an input triangle mesh. Subsequently the intersections of the planar proxies of neighboring clusters are sought to yield a coarse approximation with planar faces. Here typical intersection problems arising
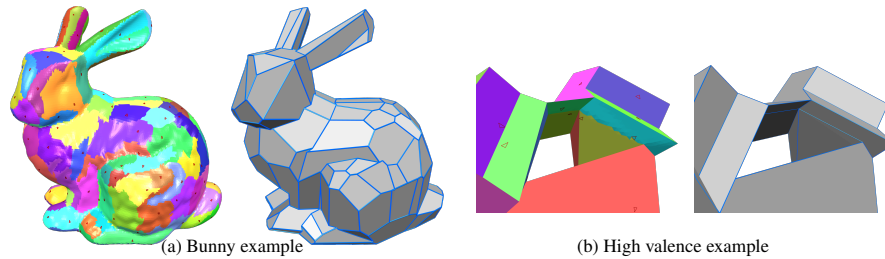
(a) Bunny example   (b) High valence example

Figure 6: Variational Shape Approximation with planar panels.

from parallel planes or poorly oriented planes defining intersection points far away from the expected cluster corner can occur as discussed by the authors.

Energies.   The above mentioned intersection scenario and associated problems map perfectly to VTPI. Additional to a positional energy keeping intersections close to the computed cluster corners we implemented a simple energy penalizing deviation from the optimized proxy normals. Figure 6 shows a coarse BUNNY and an example with high-valence intersection points.

Discussion.   Note that this section just demonstrates the applicability of the VTPI. No post-processing (e.g., fixing acute angles) was implemented. For the VTPI concave faces and acute angles do not pose a problem. Naturally, energies or constraints limiting the interior angles can be added. However, simply forbidding acute angles in the optimization will generally not lead to good results or even a valid solution, such issues need to be properly addressed already in the clustering step.

Regarding the intersection problems, re-locating clusters and rerunning the clustering or locally *tilting* the planes for better intersections as suggested in [Cutler and Whiting 2007] might lead to acceptable solutions. Nevertheless, it is valuable to have a reliable tool such as VTPI for intersection computations, which is additionally able to handle high-valence vertices.

4.3   Dual and Multi-Layer Structures

For the third application scenario we consider the creation of dual hexagon-dominant structures with planar faces at a preferred offset-distance from a given triangle meshes.

Energies.   For the planarization of given meshes (Section 4.1) energies preserving edge lengths and diagonals proved useful. Here we compute dual structures whose geometry is yet unknown and strongly depends on other constraints and parameters such as the wanted offset of the dual structure. Hence, to compute aesthetic outer/inner shells of given designs we use a combination of element fairing and normal smoothness energies instead. Our element fairing energy simply penalizes positional deviation from the c.o.g. of a vertex' neighbors, and normal smoothness the deviation of the normals of neighboring faces. Figure 7 briefly demonstrates the effect of these energies. The bottom row shows dual structures guaranteed to not
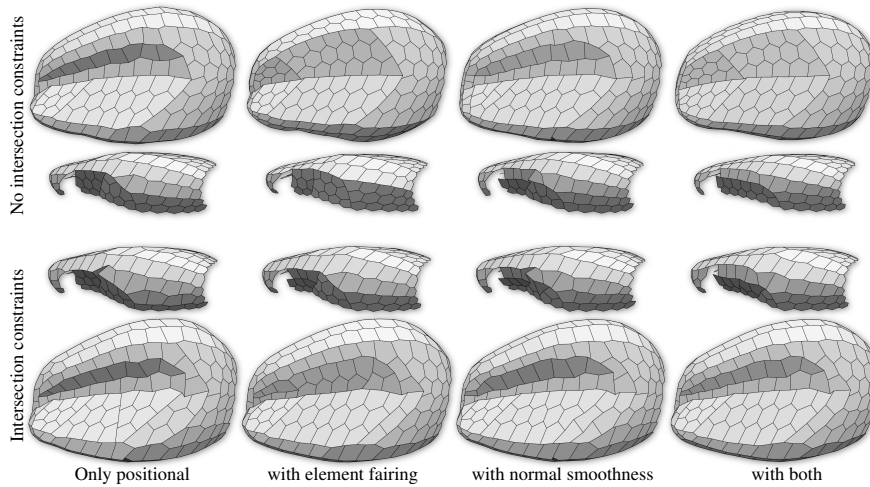
Figure 7: Shows the effect of the element fairing and normal smoothness energies when computing dual structures of the ALPINEHUT model, also shown in Figure 2. The second row uses the intersection constraints from Section 3. All four examples used the positional energy. Additionally using element fairing tends to concentrate curvature to sharp creases, segmenting the mesh into flat or low-curvature patches consisting mainly of regular convex faces, while the normal smoothness leads to a rounder shape. These effects are most visible in the top two rows, in the absence of constraints.

intersect the primal structure locally (using the constraints from Section 3) and the upper row shows free (unconstrained) dual structures.

Element quality. In practical applications such as architecture, aesthetic qualities (regularity, convex angles, etc.) of the dual elements are important. While VTPI is a practical tool for computing dual structures with planar faces, it cannot circumvent the topological and geometrical problems inherent in the task, e.g., concavities, degeneracies etc. Although specifying additional energies and constraints can to some extent hide or relocate some effects (such as the curvature concentrated along the sharp creases in Figure 7), it always comes at the cost of other qualities such as approximation accuracy.

The importance of good input meshes. One of the most effective ways to control the quality of the dual mesh is by specification of an appropriate primal mesh. One important quality factor in this regard is symmetry. Figure 8 demonstrates how symmetric dual structures can be achieved from symmetric input, whereas the typical smooth, highly regular triangle meshes resulting from standard remeshing approaches yield an unstructured less pleasing output. Hence, automatic triangulation methods, such as the ones employed in the pre-processing stage by [Troche 2008] and [Wang et al. 2008], can lead to good dual structures of abstract organic objects, but are not suited for working with symmetric or technical input shapes.

While we assume qualitative input to our algorithm, we also added a couple of basic heuristics to support manual improvement. Local self-intersection or dual

vertices lying too close to each other are identified and highlighted, suggesting to the designer to, e.g., locally introduce higher valence faces or flip the corresponding edge in the triangle mesh.
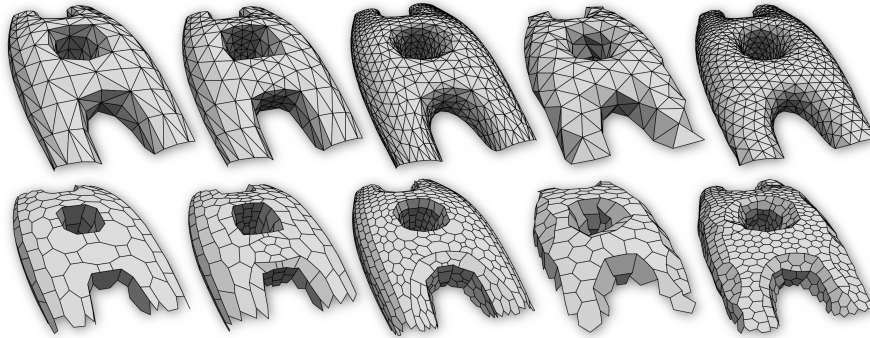

Figure 8: Dual structures of symmetric and non-symmetric input triangulations for the TRAINSTATION design, all computed using the same parameters.

Discussion. The here proposed creation of planar dual panels at a desired offset distance is in a sense related to the recent rationalization of point folding structures [Zimmer et al. 2012]. There a dual (hexagon-dominant) structure arose implicitly from the generation of the point folding elements, but, since the dual faces were in general not planar, the supporting construction framework could not be covered by an outer simple surface. Unifying the two concepts is an interesting direction for future work and would require interleaving the typical discrete optimization used in rationalization with the here presented continuous optimization possibly along the lines of [Eigensatz et al. 2010].

5 Conclusion

We presented a generic optimization framework for planar polygonal meshing that is based on a variational formulation of tangent plane intersections. The formulation allows for the introduction of additional degrees of freedom which we utilize to enrich the otherwise rigid concept of tangent plane intersection with various energies and constraints to make it applicable to different geometry processing scenarios, even computing intersection points of more than three tangent planes.

One advantage of VTPI is the fact that the dual faces are always planar and their normals can be used during the optimization, e.g., to define further energies or constraints (such as the face constraints of Section 3). Our approach utilizes the strengths of modern solvers and is slightly more brute-force than some state-of-the-art methods dealing with constrained solutions spaces (e.g., [Yang et al. 2011]). On the other hand it requires only a minimal implementation effort to prototype different energies and constraints.

Controlling surfaces properties (such as planarity of panels) is only one step towards more profitable manufacturing. Reducing the number of different panels or molds (cf. [Eigensatz et al. 2010, Fu et al. 2010, Singh and Schaefer 2010, Zimmer et al. 2012]) is another important aspect. While both problems are well understood on their own, the combined problem of simultaneously optimizing both, the number of panels and the properties of the surface, is inherently difficult. Not only is the one typically a discrete and the other a continuous optimization problem, but the reduction of the number of panels and the quality optimization of the same are practically two contradicting goals for which no common, acceptable solution must exist in general. Nevertheless, such a method would be invaluable in modern architecture.

Acknowledgments

References

ALEXA, M., AND WARDETZKY, M. 2011. Discrete Laplacians on general polygonal meshes. *ACM Trans. Graph. 30*, 4, 102:1–102:10.

BOBENKO, A., AND SURIS, Y. 2008. *Discrete Differential Geometry: Integrable Structure*. Graduate Studies in Mathematics. American Mathematical Society.

BOBENKO, A. I., HOFFMANN, T., AND SPRINGBORN, B. A. 2003. Minimal surfaces from circle patterns: Geometry from combinatorics. *Annals of Mathematics 164*, 231–264.

BOMMES, D., ZIMMER, H., AND KOBBELT, L. 2010. Practical Mixed-Integer Optimization for Geometry Processing. In *Curves and Surfaces*, Springer, J.-D. Boissonnat, P. Chenin, A. Cohen, C. Gout, T. Lyche, M.-L. Mazure, and L. L. Schumaker, Eds., vol. 6920 of *Lecture Notes in Computer Science*, 193–206.

COHEN-STEINER, D., ALLIEZ, P., AND DESBRUN, M. 2004. Variational shape approximation. *ACM Trans. Graph. 23*, 3, 905–914.

CUTLER, B., AND WHITING, E. 2007. Constrained planar remeshing for architecture. In *Proceedings of Graphics Interface 2007*, ACM, GI '07, 11–18.

EIGENSATZ, M., KILIAN, M., SCHIFTNER, A., MITRA, N. J., POTTMANN, H., AND PAULY, M. 2010. Paneling architectural freeform surfaces. *ACM Trans. Graph. 29*, 4, 45:1–45:10.

FU, C.-W., LAI, C.-F., HE, Y., AND COHEN-OR, D. 2010. K-set tilable surfaces. *ACM Trans. Graph. 29*, 44:1–44:6.

LIU, Y., POTTMANN, H., WALLNER, J., YANG, Y.-L., AND WANG, W. 2006. Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graph. 25*, 3, 681–689.

MÖBIUS, J., AND KOBBELT, L. 2010. OpenFlipper: An Open Source Geometry Processing and Rendering Framework. In *Curves and Surfaces*, Springer, J.-D. Boissonnat, P. Chenin, A. Cohen, C. Gout, T. Lyche, M.-L. Mazure, and L. L. Schumaker, Eds., vol. 6920 of *Lecture Notes in Computer Science*, 488–500.

MÜLLER, C. 2011. Conformal hexagonal meshes. *Geometriae Dedicata 154*, 27–46.

POTTMANN, H., BRELL-COKCAN, S., AND WALLNER, J. 2007. Discrete surfaces for architectural design. In *Curves and Surface Design: Avignon 2006*, P. Chenin, T. Lyche, and L. L. Schumaker, Eds. Nashboro Press, 213–234.

POTTMANN, H., LIU, Y., WALLNER, J., BOBENKO, A., AND WANG, W. 2007. Geometry of multi-layer freeform structures for architecture. *ACM Trans. Graph. 26*, 3, 65:1–65:11.

SINGH, M., AND SCHAEFER, S. 2010. Triangle surfaces with discrete equivalence classes. *ACM Trans. Graph. 29*, 46:1–46:7.

TRAUTZ, M., AND HERKRATH, R. 2009. The application of folded plate principles on spatial structures with regular, irregular and free-form geometries. In *Proc. IASS*, 1019–1031.

TROCHE, C. 2008. Planar hexagonal meshes by tangent plane intersection. In *Advances in Architectural Geometry*.

WÄCHTER, A., AND BIEGLER, L. T. 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming 106*, 25–57.

WANG, W., LIU, Y., YAN, D., CHAN, B., LING, R., AND SUN, F. 2008. Hexagonal meshes with planar faces. Tech. Rep. TR-2008-13, Department of Computer Science, The University of Hong Kong.

YANG, Y.-L., YANG, Y.-J., POTTMANN, H., AND MITRA, N. J. 2011. Shape space exploration of constrained meshes. *ACM Trans. Graph. 30*, 6, 124:1–124:12.

ZADRAVEC, M., SCHIFTNER, A., AND WALLNER, J. 2010. Designing quad-dominant meshes with planar faces. *Comput. Graph. Forum 29*, 5, 1671–1679.

ZIMMER, H., CAMPEN, M., BOMMES, D., AND KOBBELT, L. 2012. Rationalization of Triangle-Based Point-Folding Structures. *Computer Graphics Forum 31*, 2, 611–620.