

Planar Quads in Architecture Surfaces

Christian Dimitri, UPC BarcelonaTech

September 2018

Abstract

Planar quads are found everywhere in architectural free-form surfaces. The first chapter will cover the problems behind planar quads for constructability, there benefits, there metrics and there constraints goals for optimization. The second chapter will embed several pre-processing algorithms that generates a candidate mesh for being a planar quads mesh. Therefore, in the third chapter, those candidate meshes will be optimized according to some conditions that qualify them to be PQ meshes. Those conditions are based on scientific papers. And finally by combining the chapter two and chapter three, iteratively a subdivision method algorithm and a quad planarization will be done in order to have a PQ mesh.

1 Construction

PQ meshes must show different results from the mere geometry since the planarity of faces should obey the goals in order to fulfil the basis properties of the *planar quad meshes* (Zadravec, Schiftner, and Wallner 2010).

1.1 PQ Geometric Properties

Since a triangular face in space is always planar, then they are easier to deal with when the curvature is very high. The latter leads the quad panels to bend and the four vertices will want to move from the plane. Such constraint is a disadvantage for *PQ meshes* over triangular ones. Thus, if the warping height exceeds a certain limit while measuring it, the four vertices of each of the faces should be independent from its neighbouring face's vertex see fig. 1.

In theory two parallel vectors in space enclosed at each point by two other vectors not necessarily parallel form a planar face, for more information refer to the author (Glymph et al. 2004). Consider each row of faces $f_{i,j}$ is a PQ strip. PQ meshes are composed by vertices $\mathbf{v}_{i,j}$ with a valence $\pm k/4 (k \in \mathbb{Z})$ where along each vertex a curve of family A and a curve of family B intersect see (Liu et al. 2006). N-geons can appear with a valence $k \neq 4$ so called singularities.

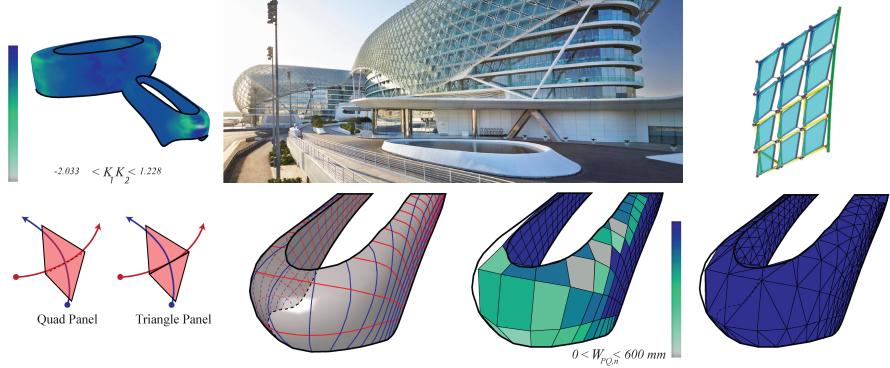


Figure 1: The hinge is affected by the high Gaussian curvature on the surface of the *Yas Island Hotel By Zaha Hadid* (Leon and Shelden 2011). The difference between *PQ meshes* and *triangule meshes*.

1.2 Benefits

Planar quad meshes are preferred over *triangule meshes* for aesthetical and constructional reasons. The advantages of *planar quads* meshes for construction over other meshes is that:

- *PQ meshes* are less heavy.
- *PQ meshes* are a less material use.
- *PQ meshes* consumes less energy during fabrication.

1.3 Metrics-Measures(Quality)

To have *planar quads*, several measures are mentioned below. For a better quality, the mathematical measures and the conditions are classified by face and by mesh[fig. 2 and fig. 3]. In addition to that, some conditions will be translated to *custom goals* that will improve the quality of the mesh.

The measurements and conditions applied to the mesh itself are:

The measurements and conditions applied to the elements of the mesh are:

2 Algorithmic Strategies

2.1 Several Pre-Processing Techniques

Several pre-processing techniques will be adapted in order to generate a *PQ* mesh with *planar faces*. The technique will be used depending on the surface

Measures	Technical Drawings	Mathematical Description
Element Area: The area on each quad is divided by two.		$PQ_{area} = \frac{\max(d) \times \min(d)}{2}$
Diagonals Aspect Ratio: Maximum distance between diagonals of the quad face divided by the minimum distance of diagonals.		$\eta_{PQ} = \frac{\max(d)}{\min(d)}$

Figure 2: Table showing the main measures of PQ meshes see (Gokhale (2008), guide (2018) and Geometry and Generator (2016))

Measures	Technical Drawings	Mathematical Description
Planarity and Convexity: A panel is flat and convex if and only the difference between the sum of the internal angles and 2π should be zero.		$\delta_{PQ} = \sum \alpha_i - 2 = 0$
Warping Height: The measure of a quadrilateral element from being planar. Max element corner normal angular deviation from normal of mean plane.		The warping W_{PQ} at the vertex v is measured by projecting the point on the plane and measuring its distance

Figure 3: Table showing the main measures of PQ meshes see (Gokhale (2008), guide (2018) and Geometry and Generator (2016))

type. Translational surfaces is an easy and fast algorithm. However architecture free-form surfaces with high curvature, requires more complex algorithm to generate *PQ meshes*.

2.1.1 Translational Surfaces

Translational surfaces are limited and easy to generate. The quads generated are 100% the proof that it is generated thru a set of parallel vectors which results a planar face. In addition to that they are homogeneous because adding the same length vector as a constraint leads to have evenly spaces faces and reduce the variance. If the sectional curves are plane and the vectors are parallel having the same length the result will respond to the design principle of a translation surface. Assuming that one direction of the quad mesh net to be the sectional curve, two design principle can appear:

- The row of longitudinal sectional curves form parallel vectors.
- The row of lateral sectional curves form parallel vectors.

2.1.1.1 Row of sectional curves translated over a set of parallel vectors

The family of sectional curves $p(\mathbf{u})$ translated over a set of parallel vectors is generated as follow: A random spatial curve $p(\mathbf{v})$ called generatrix is translated against another random spatial curve $p(\mathbf{u})$ called directrix as seen in fig. 4. Thus considering that translation by equal length gives homogeneous results of the planar quads.

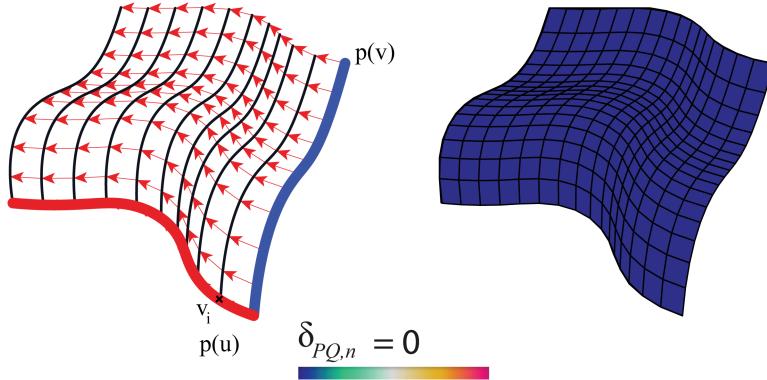


Figure 4: Geometric principle for translation surfaces and planarity measure fulfilled.

Several geometrical shapes have been developed in architecture during the history using translation surfaces. The elliptical paraboloid is the most familiar shape found. It is generated using the same principle, translating one parabolic curve against another.

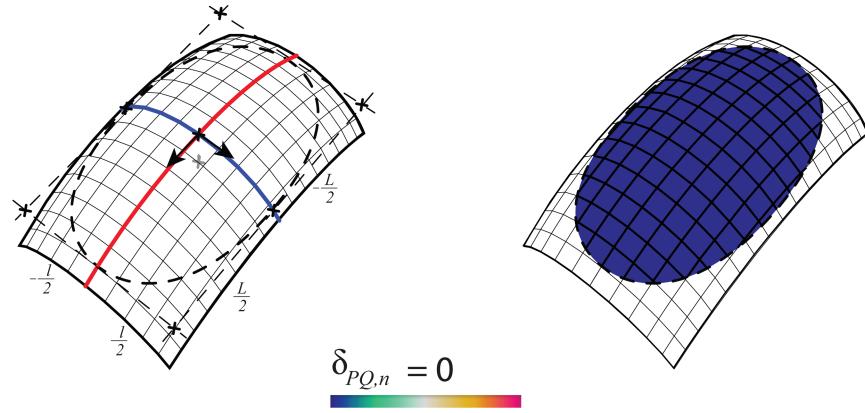


Figure 5: Elliptical paraboloid

In translational surfaces, some geometrical shapes admit boolean and joining operations, for example, the hyperbolic paraboloid is a type of translational surfaces that acknowledge such operations. By translating a parabolic curve over a hyperbolic the result is as seen in fig. 6

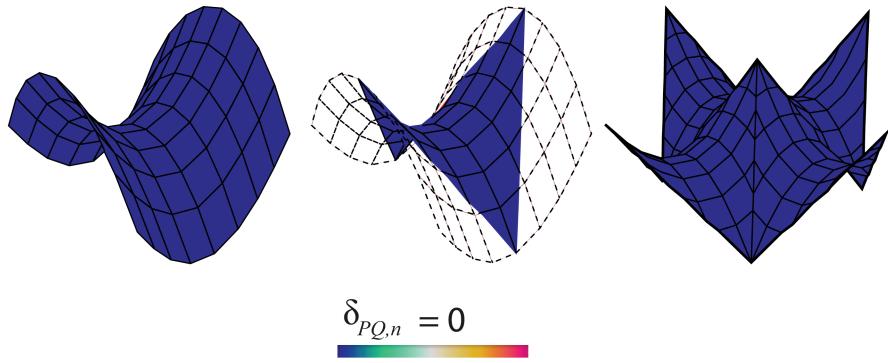


Figure 6: Translated hyperbolic paraboloid and joining possibilities

2.1.1.2 Scale-Translation Surfaces

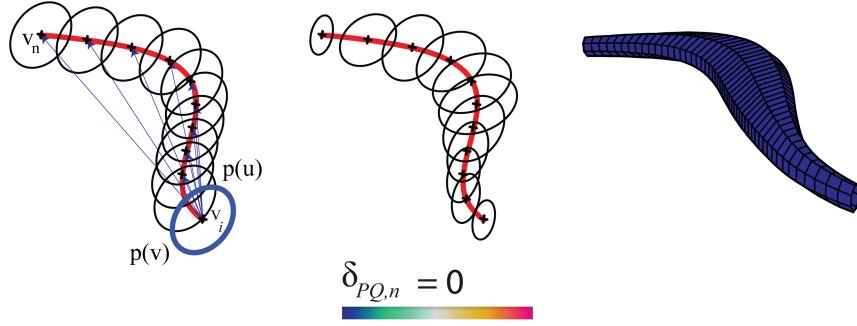


Figure 7: Centric scale-translation expansion

Scale Translational surfaces are generating by adding a scale parameter to the output curves C_n . After translating the sectional curve $p(u)$ on each point v_i equally distant at the directrix curve $p(v)$, the output curves can be scaled uniformly or non-uniformly controlled by the user. The centrical expansion of any curve gives a new curve having parallel edges. The center of expansion can be chosen randomly (Glymph et al. 2004). In this technique the centric expansion has been chosen. The resulting algorithm gives planar quad meshes see fig. 7.

2.1.2 Conjugate networks

Some curve networks are robust and efficient method to extract *PQ meshes* (Liu et al. 2006). Such method admit a huge variety of free-form surfaces. The advantage of designing a *conjugate direction field* is that the user possess freedom while controlling the *PQ mesh* layout (Zadravec, Schiftner, and Wallner 2010). Thus, the panels are flat and *discretize the principle curvature lines* see (Liu et al. 2006).

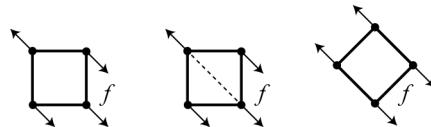


Figure 8: Left: High twisting moment. Middle: Stiffening by triangulation. Right: Torsion free alignment.(Zadravec, Schiftner, and Wallner 2010)

In addition to that, it can admit free torsion node while aligning the curve

networks with the stress and curvature field see fig. 8 for more information on statics sensitive layout refer to (Shiftner and Balzer 2010).

2.1.2.1 What is a CDF on a *triangular mesh*.

On a smooth surface $S \subset \mathbb{R}^3$, the tangent vectors $\mathbf{v}_p, \mathbf{w}_p$ are conjugate if and only if they are treated as two vectors in \mathbb{R}^3 (Liu et al. 2011). The CDF is a tool for non-photorealistic rendering in order to visualize the surface topology. Therefore it is useful for surface remeshing and alignment control. On a triangular face f_i as seen in fig. 9 of a triangular mesh $\mathbb{R}^3 = (V, E, F)$ a CDF is:

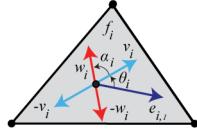


Figure 9: A CDF on a Triangular face based on (Liu et al. 2011).

- Four vectors $\{\mathbf{v}_i, \mathbf{w}_i, -\mathbf{v}_i, -\mathbf{w}_i\}$
- Two scalar parameters $\{\theta_i, \alpha_i\}$:
 - θ_i oriented angle between $e_{i,1}$ and \mathbf{v}_i
 - α_i oriented angle between \mathbf{v}_i and \mathbf{w}_i
 - They define the following: $\mathbf{v}_i = (\cos\theta_i, \sin\theta_i)^T$ and $\mathbf{w}_i = (\cos(\theta_i + \alpha_i), \sin(\theta_i + \alpha_i))^T$

2.1.2.2 The relation between PQ meshes and *conjugate networks*.

Conjugate curve networks are families of curves $A, B \subset \Phi$: For each $p \in \Phi$ unique curves of both family A, B should appear. Since T_1, T_2 are conjugate then they predefine A and get B by *integrating the vector field directions conjugate of family A.*(Liu et al. 2006)

Examples of Conjugate Curve Networks on Surfaces:

- Suited for PQ meshes: (Liu et al. 2006)
 - *The network of principle curvature lines* see (fig. 10 left).
 - In a translational surface of the form $p(u, v) \mathbf{p}(u)$ a sectional curve is translated along another curve generatrix $\mathbf{p}(v)$ and vice versa see fig. 4.
- Not suited for PQ meshes:
 - *Epipolar curves:* The translation of a point p along a line l and the intersection of the planes threw the points $p(i)$ with that surface Φ generate asymptotic curves that are not suited for such meshing see (fig. 10 center).

- Isophotic curves are conjugate to the system of the steepest descent curves with respect to the z -axis see (fig. 10 right).

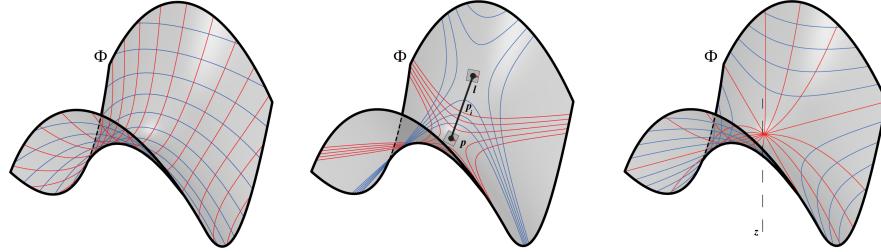


Figure 10: Various conjugate networks

2.1.2.3 Generating quad-dominant meshes via conjugate direction field

The input mesh should not be random, it is preferable to use a remeshing tool such as mesh machine to control the edges length and the fixed boundaries or others, for more info see (Piker, n.d.).

2.1.2.3.1 Alignment with the curvature (Mueller and Adriaenssens 2018).

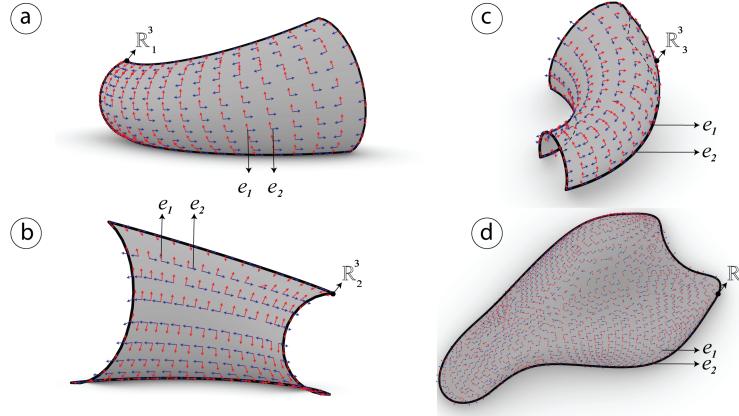


Figure 11: Surface tangency extracted on each of the meshes \mathbb{R}_i^3 by computing the minimum e_1 and maximum e_2 principle directions in red and blue.

The quality of the mesh is always better when the panels are aligned with the curvature or the stress lines. Given four different meshes \mathbb{R}^3 , the orthogonality

will be introduced for each of the meshes \mathbb{R}_i^3 by computing the *principle directions* e_1 and e_2 and storing them in $[e_1, e_2]$ see fig. 11. This method has been used by (Liu et al. 2011).

2.1.2.3.2 Interpolating vector field with *N-PolyVector Field* (Mueller and Adriaenssens 2018).

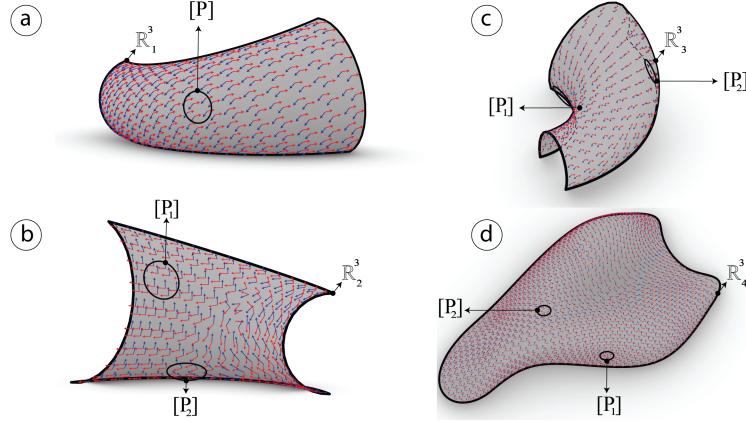


Figure 12: Smoothed vector field using n-polyVector field algorithm

In order to find a smoothed and well aligned vector field $[e_1, e_2]$ on each of the four meshes \mathbb{R}^3 , in this section a plugin called (Greco 2018) developed by (Mueller and Adriaenssens 2018) has been used. The algorithm is based on finding the trade-off between neighboring faces f_i so that the parallel transport succeed. It uses the novel method proposed by (Diamanti et al. 2014) called *N-Poly Vector Field*. While selecting a subset of points $[P]$, the vector field $[e_1, e_2]$ is able to be generated smoothly and continuously. It finds the smoothest field by interpolating the two vectors parallelly. This method is different from the one used in (Liu et al. 2011) where it uses a signed permutation technique in order to find the correct vector's relation between neighbouring vertices. In fig. 12, it is well clear how the smoothed vector field and the parallel transport have been well generated.

2.1.2.3.3 *Conjugate direction field*

After smoothing the vector field in the previous step, a quad mesh can be generated after defining the conjugate networks, (Liu et al. 2011) took further this topic for more information refer to the reading. From the previous step a conjugate vector field $[\mathbf{v}_i, \mathbf{w}_i]$ with (Greco 2018) is computed using an algorithm provided in (LibDirectional 2018) see [#fig:Conjugatefield].

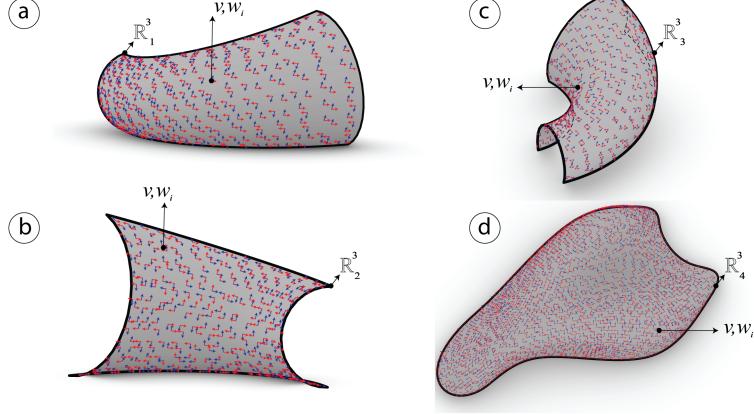


Figure 13: Conjugate field $[v_i, w_i]$ after smoothing previously the vector field $[e_{i,1}, e_{i,2}]$.

2.1.2.3.4 Global parametrization using frame fields

If the mesh possess negative curvature and is varying highly, the parametrization has to be done by patches, see fig. 14 otherwise the parametrization can be done on a single patch see fig. 15. The latter depend on the umbilics and the rigids. the algorithm succeeded with all the meshes except for \mathbb{R}_4^3 which is in continuous research.

This method is based on the *global parametrization with frame fields* fig. 14 it is generated using a custom component developed by the author of (Greco 2018). It is used to shape the new mesh in a different typology, the latter has to be aligned with some given vetors v_i, w_i by interest of the user. The Mixed-integer quadrangulation by (Bommes, Zimmer, and Kobbelt 2009) is one way to do that and another way is using *Anisotropic remeshing to concentrate the elements in the regions with more details*(Mueller and Adriaenssens 2018) published by (Diamanti et al. 2014) an open source library can do that see (Libgil 2017) and (Ebke et al. 2013b).

2.1.2.3.5 Tracing streamlines

The streamlines are traced on the 2D maps after integrating the Vector field then they are remapped on the 3D meshes. They are generated using a custom component in (Greco 2018) that is developed using the 4th order Runge-Kutta for more information refer to (Mueller and Adriaenssens 2018) see fig. 14.

2.1.2.3.6 Extracting the candidate PQ Meshes

The meshes are generated after tracing the streamlines in the previous step.

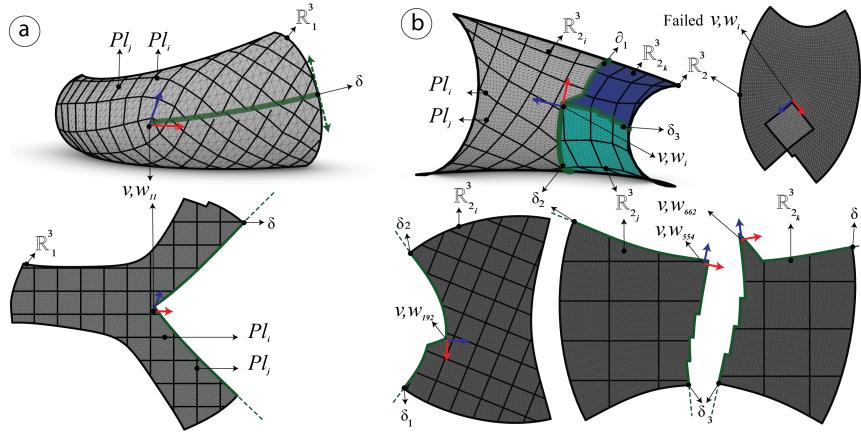


Figure 14: a) The first mesh \mathbb{R}_1^3 , b) The second mesh \mathbb{R}_2^3 . In green the boundary of the cutting path δ , Pl_i the isolines, and $\mathbf{v}_i, \mathbf{w}_i$ the frame fields chosen at index i .

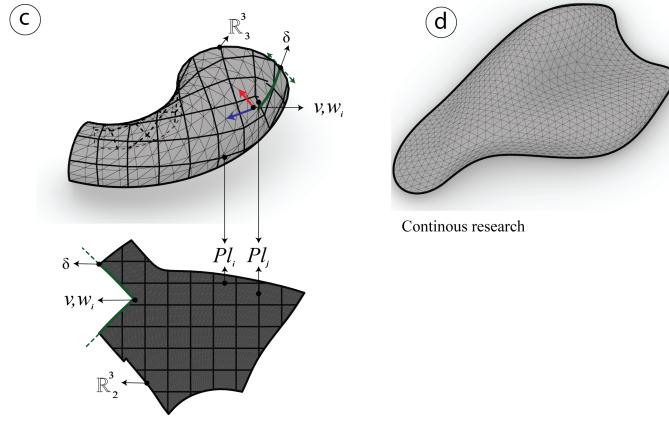


Figure 15: c) The third mesh \mathbb{R}_1^3 , d) The fourth mesh \mathbb{R}_1^3 which is still in continuous research.

Therefore the results are seen in fig. 16. The resulting meshes are a candidate PQ mesh ready for optimization.

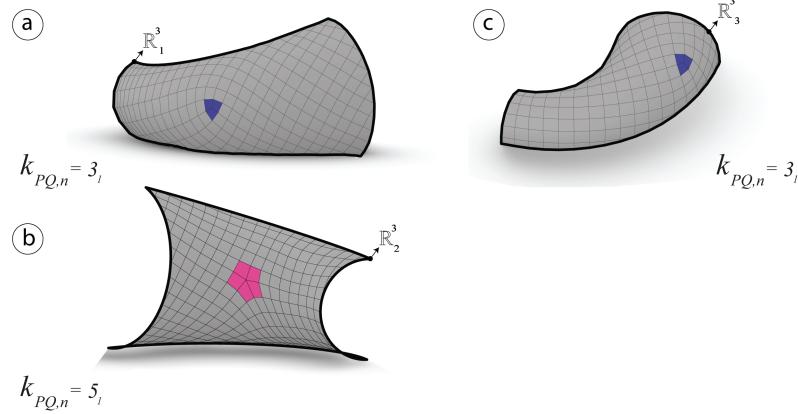


Figure 16: The resulting candidate PQ meshes have a semi regular valence. a) \mathbb{R}_1^3 with one singularitie $k_{PQ} = 3_1$. b) \mathbb{R}_2^3 with one singularitie $k_{PQ} = 5_1$. c) \mathbb{R}_3^3 with one singularitie $k_{PQ} = 3_1$.

2.1.2.4 Generating quad-dominant meshes via principle curvature networks

This method is different from the previous one. The network of curves [Pl] will be generated on each of the four meshes \mathbb{R}_i^3 using(Michalatos 2017), however the output is not sorted. Although without a special library like (Jacobson, Panozzo, and others 2017) and (Ebke et al. 2013b) to extract automatically a robust-quad mesh (Ebke et al. 2013a) is very hard to achieve. This method is based on the on the mixed integer quadrangulation by (Bommes, Zimmer, and Kobbelt 2009). Therefore, an algorithm had to be developed in order to extract that candidate PQ mesh using conformal mapping.

2.1.2.4.1 Computing curvature networks

The *principle curvature networks*[Pl] will be generated automatically using the reparameterized component developed in (Michalatos 2017) see fig. 17.

2.1.2.4.2 Global parametrization using conformal mapping

The *curve networks* [Pl] previously extracted, are reparametrized using conformal mapping. Then they are analyzed and rebuilt in order to close naked nodes and form meshes with a semi regular valence.

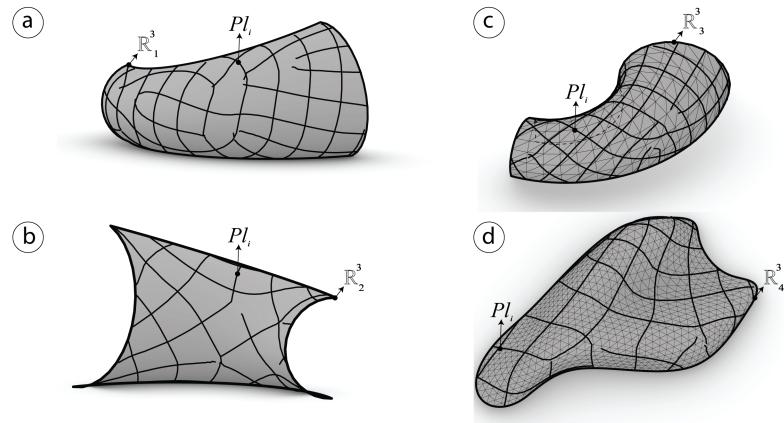


Figure 17: Curve network Pl_i computed using (Michalatos 2017) on each of the input meshes \mathbb{R}_i^3 .

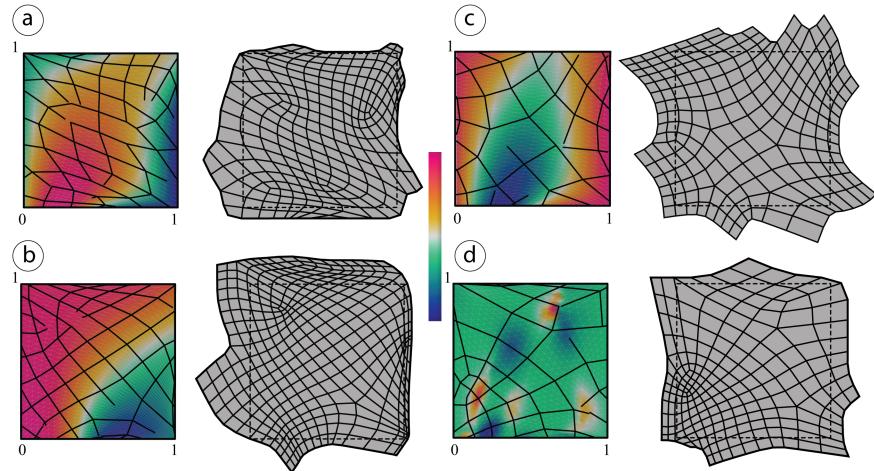


Figure 18: Conformal mapping parametrization on a unit plane and curvature color gradient. Pl_i remapped and rebuilt on the 2D map.

2.1.2.4.3 Extracting the candidate PQ meshes

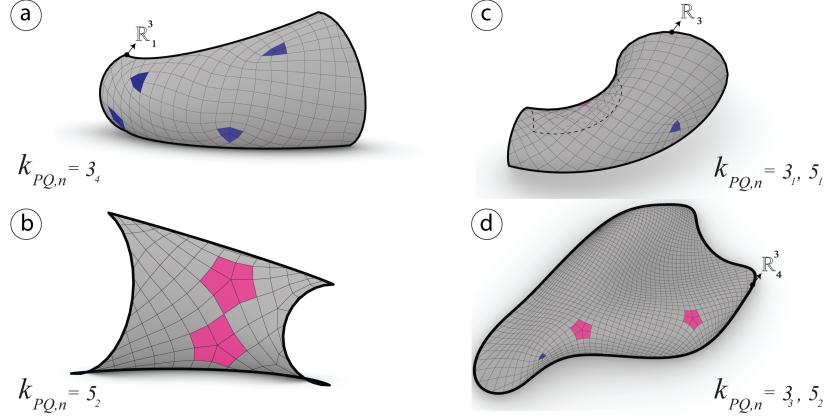


Figure 19: The resulting candidate PQ meshes have a semi regular valence. a) \mathbb{R}_1^3 with four singularities $k_{PQ} = 3_4$. b) \mathbb{R}_2^3 with two singularities $k_{PQ} = 5_2$. c) \mathbb{R}_3^3 with one singularity $k_{PQ} = 3_1$ and one singularity $k_{PQ} = 5_1$.

After mapping the *curve networks* and rebuilding the quad mesh on the unit plane, it is now possible remap the meshes on the original surfaces see fig. 18 and fig. 19}.

2.1.3 Conical meshes

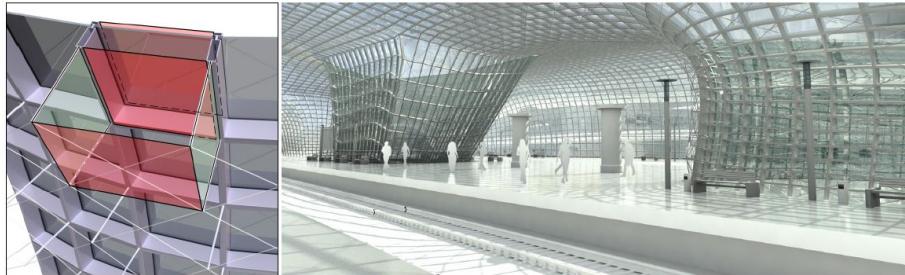


Figure 20: Left: Offset property of a conical mesh. Right: *Railway station by B.Schneider* (Liu et al. 2006) a conical mesh as glass structure that *discretizes the principle curvature*.

Conical meshes are planar quad meshes which *discretize principle curvature lines*, posses and offset at a constant distance as well as planar connecting elements (Liu et al. 2006) see fig. 22. A conical mesh is conical if and only if all of its

vertices \mathbf{v}_i are conical which means that the four face planes meeting at \mathbf{v} are tangent to a common sphere (Liu et al. 2006) see fig. 21.

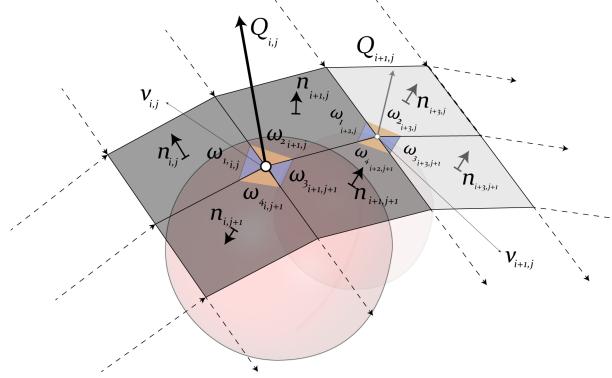


Figure 21: Faces configuration of a conical mesh (Liu et al. 2006).

2.1.3.1 The angle criterion of a conical mesh

The sum of the opposite angles on a vertex \mathbf{v} should always be equals to zero so see fig. 21. \mathbf{v} is a conical vertex if and only if the characterization of a conical mesh interior angles should respond to this function:

$$\omega_1 + \omega_3 = \omega_2 + \omega_4$$

2.1.3.2 The Offset Properties

Triangular meshes are missing the offset property at a constant distance. However conical meshes have this property while generating conical meshes at the offset (Liu et al. 2006).

The fact that the faces of a conical mesh are incident to a common vertex $\mathbf{v}_{i,j}$ and tangent to a cone with an axis $Q_{i,j}$. After offsetting the axis remains the same and the faces are still tangent to the cone (Liu et al. 2006).

The Languerre transformation (Liu et al. 2006) contains one of the instances for offsetting planes by a fixed distance along their normal vector. The Languerre transformation preserves the conical meshes at the offset.

2.1.3.3 The Normals

The spherical image is a fact where the vertice \mathbf{v}_{ij} of a PQ mesh built on a unit sphere are converted to the normal vectors of $Q_{i,j}$. As the four faces incident to a common vertex \mathbf{v}_{ij} tangent to the same cone, the normal vectors $n_{i,j}$ on each of the four faces share the same angle with the cone's axis $Q_{i,j}$ see fig. 21.

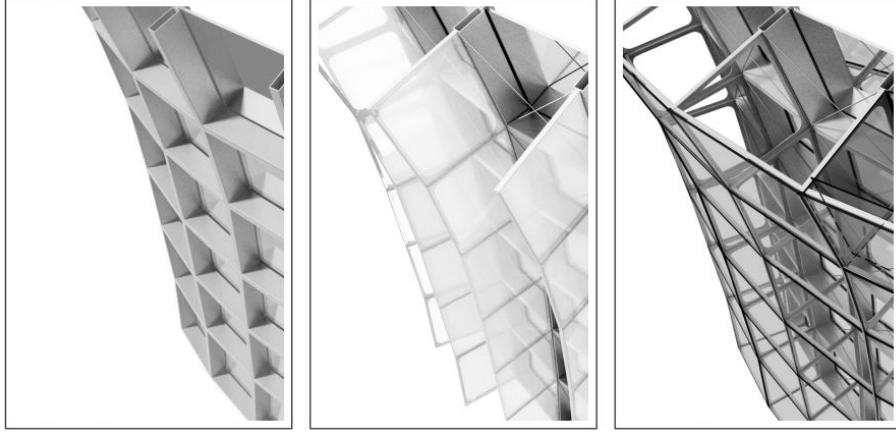


Figure 22: Constant offset of a Conical Mesh see (Pottmann and Wallner 2008).

Consequently the spherical image of the principle curvature network returns an orthogonal curve network on a sphere (Liu et al. 2006).

2.1.3.4 Conical optimization

PQ meshes generated after computing the principle curve networks are well suited to be optimized using conical meshes conditions. In order to do that, the angles and normals are measured and visualized with a gradiant color that varies in a range between the common meshes see fig. 23, fig. 24, fig. 25 and fig. 26.

2.1.3.4.1 Angles analysis

2.1.3.4.2 Normals analysis

The first mesh $\mathbb{R}^{\mathbb{M}_1}$

2.1.3.4.3 Angles optimization

For each vertex $\mathbf{v}_{i,j}$ on the mesh \mathbb{R}^3 minimize the sum of the opposite angles equals to zero $\omega_1 + \omega_3 - \omega_2 - \omega_4 = 0$ using (Piker 2010) solver.

2.1.3.4.4 Normals optimization

For each vertex $\mathbf{v}_{i,j}$ on the mesh \mathbb{R}^3 minimize the angles difference between the four adjacents faces normals and the cones normal should be equal to zero $n_{i,j}\angle Q_{i,j} - n_{i+1,j}\angle Q_{i,j} - n_{i,j+1}\angle Q_{i,j} - n_{i+1,j+1}\angle Q_{i,j} = 0$.

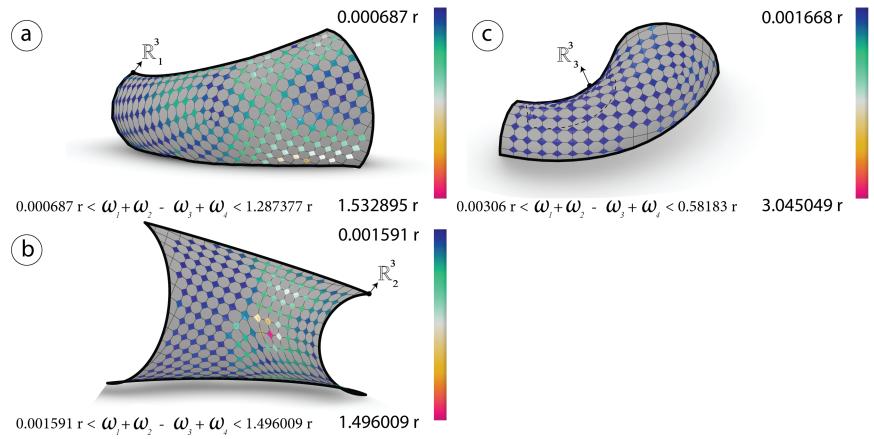


Figure 23: Before optimization: The sum of the opposite angles ω_i measured in radians for each vertex $\mathbf{v}_{i,j}$ of the meshes \mathbb{R}_i^3 generated via CDF.

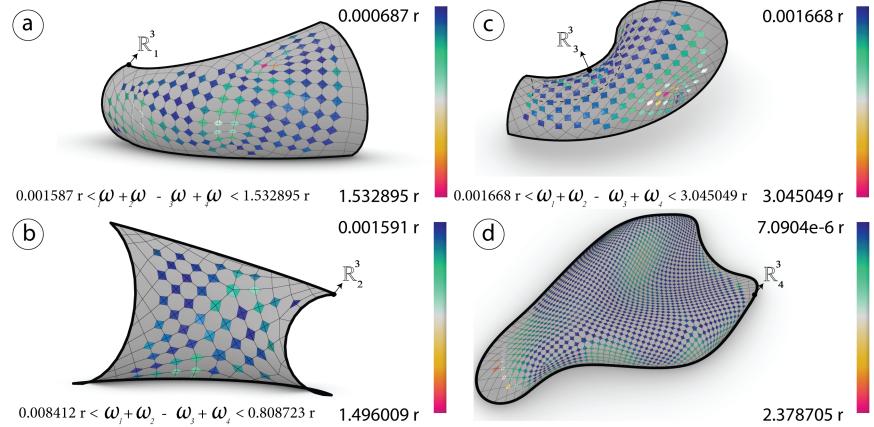


Figure 24: Before optimization: The sum of the opposite angles ω_i measured in radians for each vertex $\mathbf{v}_{i,j}$ of the meshes \mathbb{R}_j^3 generated via principle curvature networks.

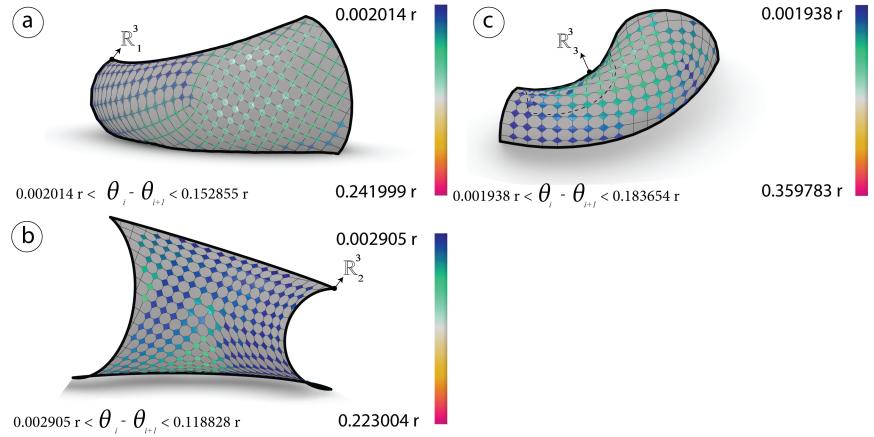


Figure 25: Before optimization: The angles difference θ_i between the normals $n_{i,j}$ and the cones normal $Q_{i,j}$ are measured in radians for each vertex $\mathbf{v}_{i,j}$ of the meshes \mathbb{R}_i^3 generated via *conformal mapping*.

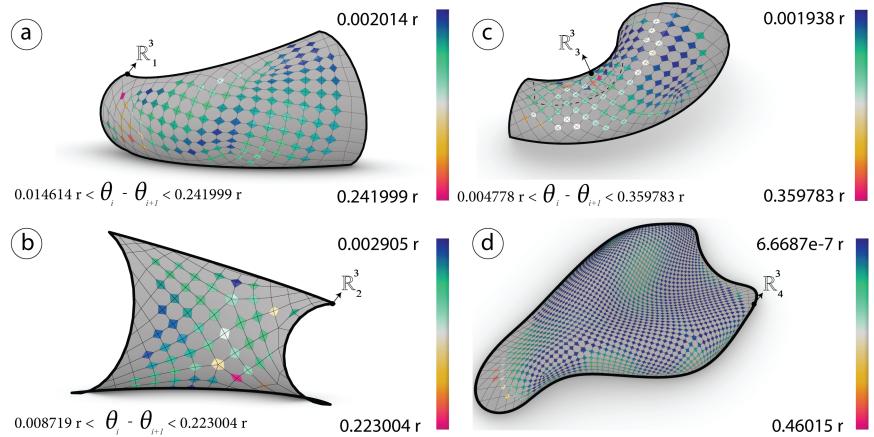


Figure 26: Before optimization: The angles difference θ_i between the normals $n_{i,j}$ and the cones normal $Q_{i,j}$ are measured in radians for each vertex $\mathbf{v}_{i,j}$ of the meshes \mathbb{R}_j^3 generated via *frame field*.

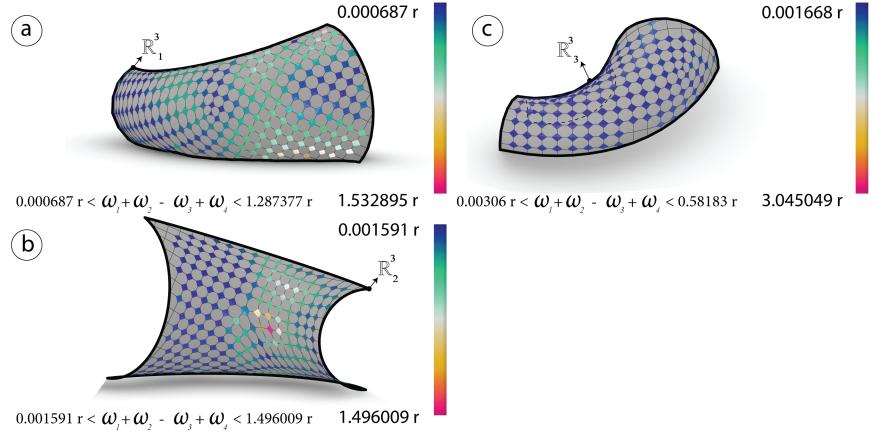


Figure 27: After optimization: The sum of the opposite angles ω_i in radians of the meshes \mathbb{R}_i^3 generated via *conformal mapping*.

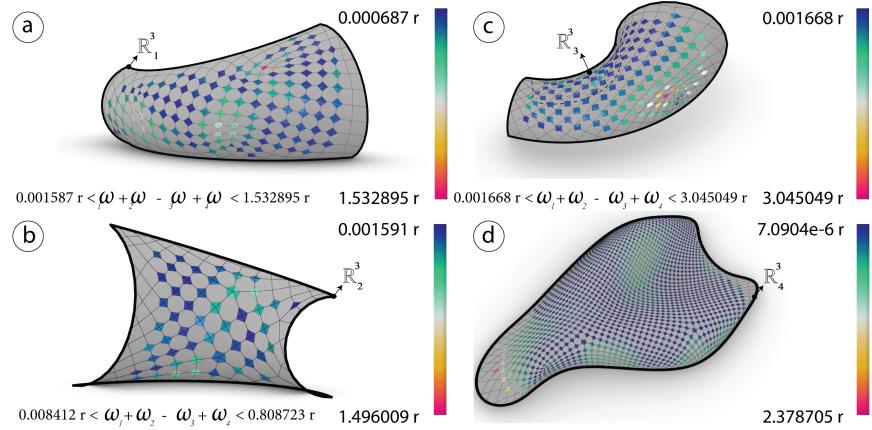


Figure 28: After optimization: The sum of the opposite angles ω_i in radians of the meshes \mathbb{R}_j^3 generated via *conformal mapping*.

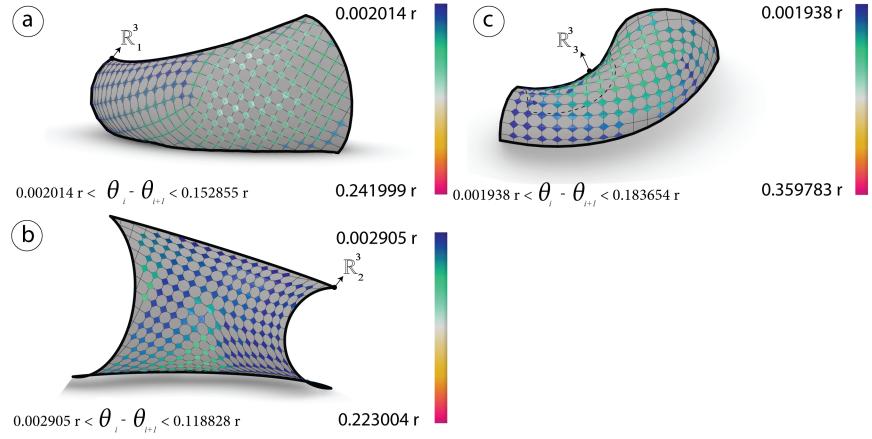


Figure 29: After optimization: The angles difference θ_i in radians of the meshes \mathbb{R}_i^3 generated via frame field

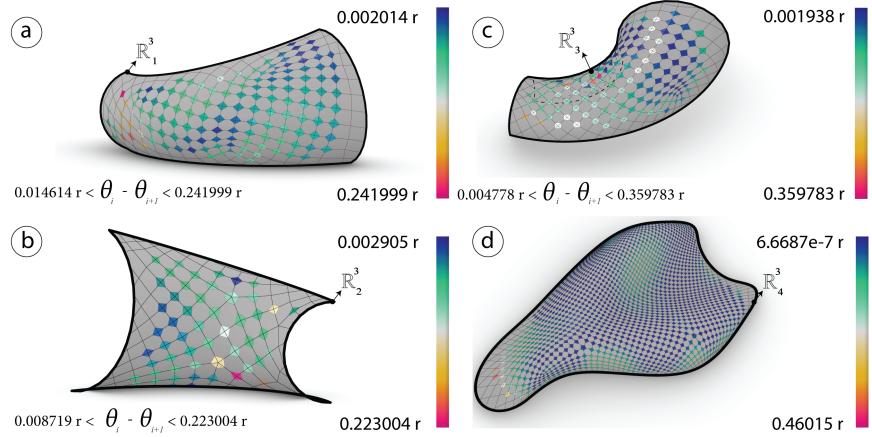


Figure 30: After optimization: The angles difference θ_i of the meshes \mathbb{R}_j^3 generated via frame field.

2.2 Optimization (K2)goals

2.2.1 Planarity

2.2.1.1 Analysis

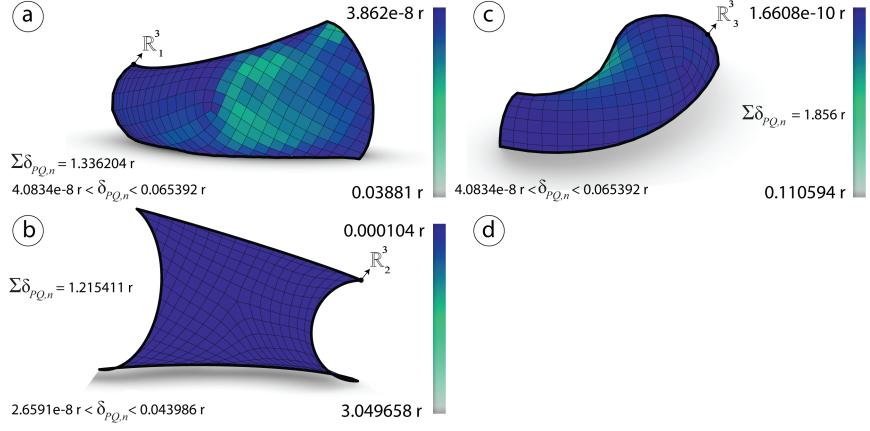


Figure 31: Before optimization: Planarity δ_{PQ} measured in radians of the meshes \mathbb{R}_i^3 generated via *frame field*.

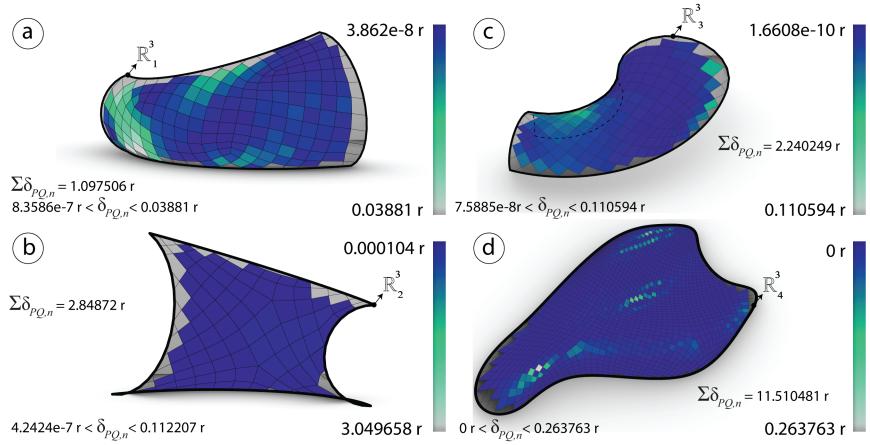


Figure 32: Before optimization: Planarity δ_{PQ} measured in radians of the meshes \mathbb{R}_i^3 generated via *conformal mapping*.

2.2.1.2 Optimization

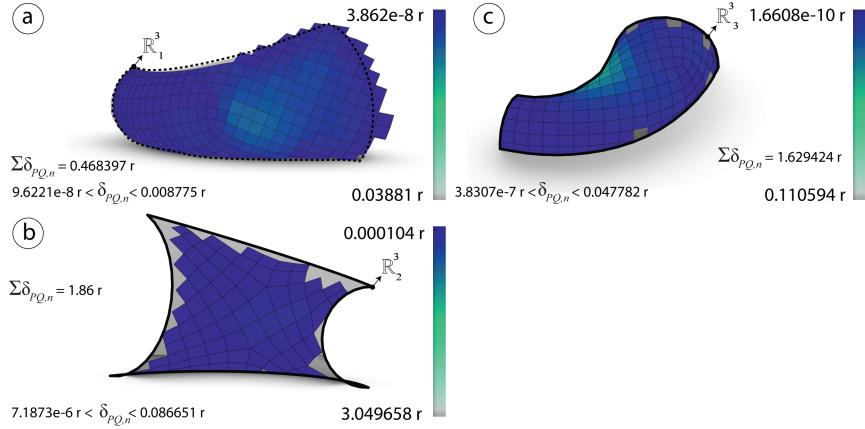


Figure 33: After optimization: Planarity δ_{PQ} measured in radians of the meshes \mathbb{R}_i^3 generated via *frame field*.

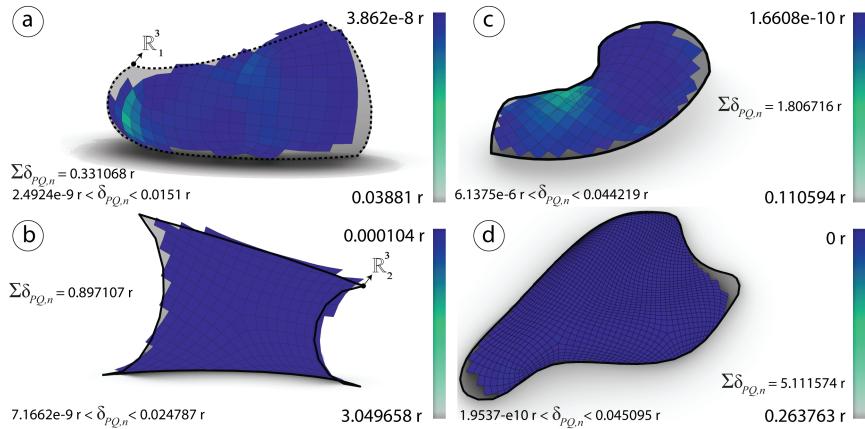


Figure 34: After optimization: Planarity δ_{PQ} measured in radians of the meshes \mathbb{R}_i^3 generated via *frame field*.

2.2.2 Diagonals Aspect Ratio

2.2.2.1 Analysis

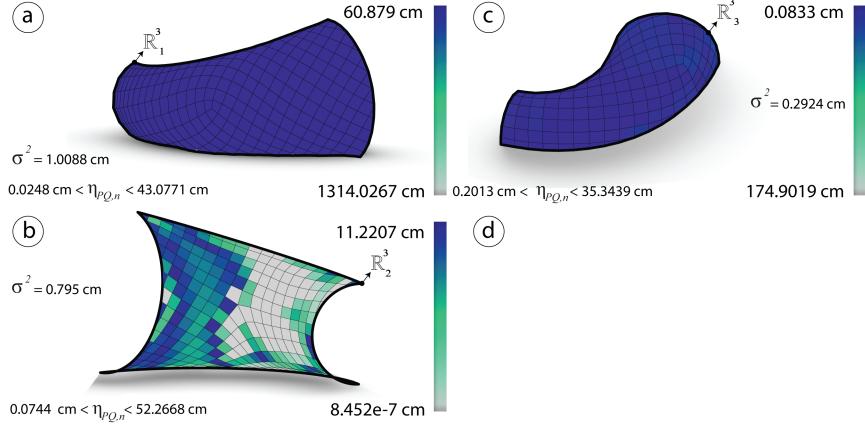


Figure 35: Before optimization: Diagonals aspect ratio η_{PQ} measured in cm of the meshes \mathbb{R}_i^3 generated via *frame field*.

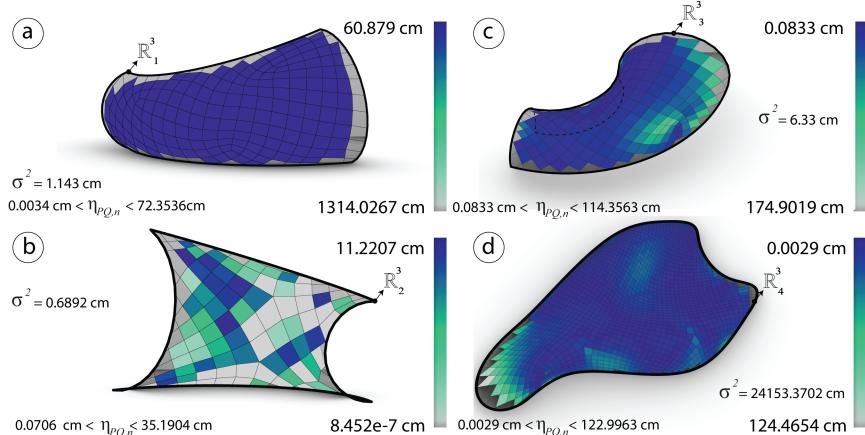


Figure 36: Before optimization: Diagonals aspect ratio η_{PQ} measured in cm of the meshes \mathbb{R}_i^3 generated via *conformal mapping*.

2.2.2.2 Optimization

2.2.3 Element areas

2.2.3.1 Analysis

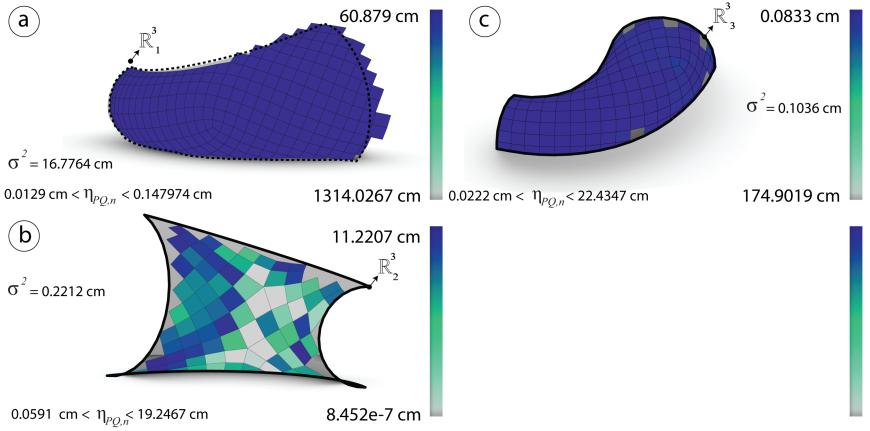


Figure 37: After optimization: Diagonals aspect ratio η_{PQ} measured in cm of the meshes \mathbb{R}_i^3 generated via *frame field*.

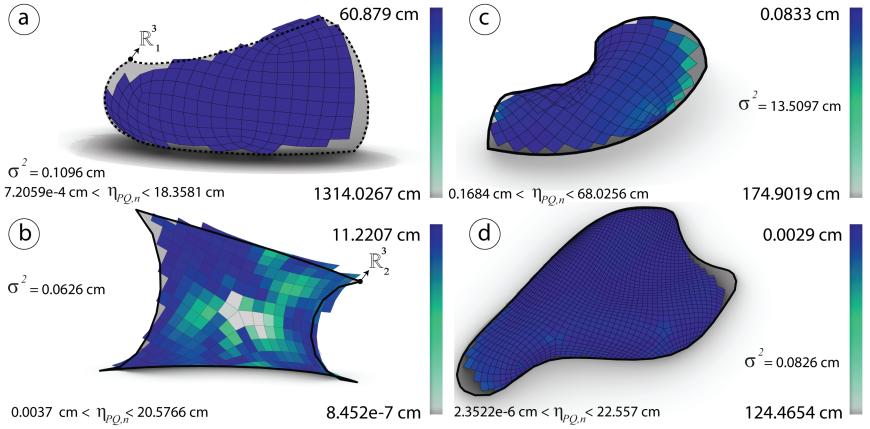


Figure 38: After optimization: Diagonals aspect ratio η_{PQ} measured in cm of the meshes \mathbb{R}_i^3 generated via *conformal mapping*.

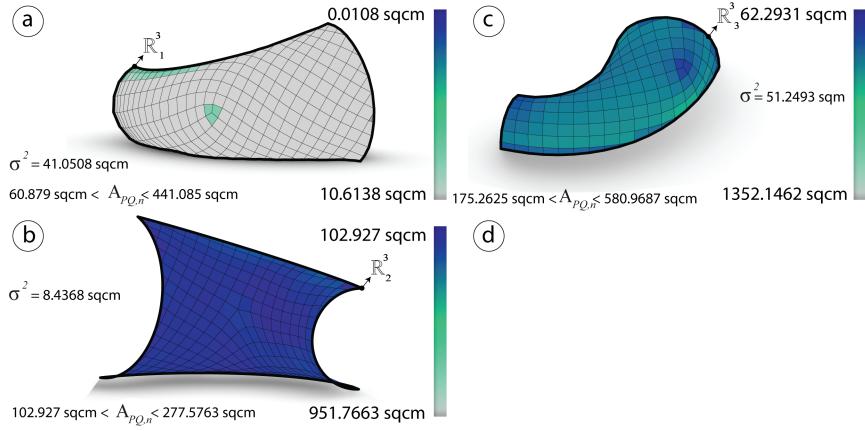


Figure 39: Before optimization: Area η_{PQ} and the variance measured in sqcm of the meshes \mathbb{R}_i^3 generated via *frame field*.

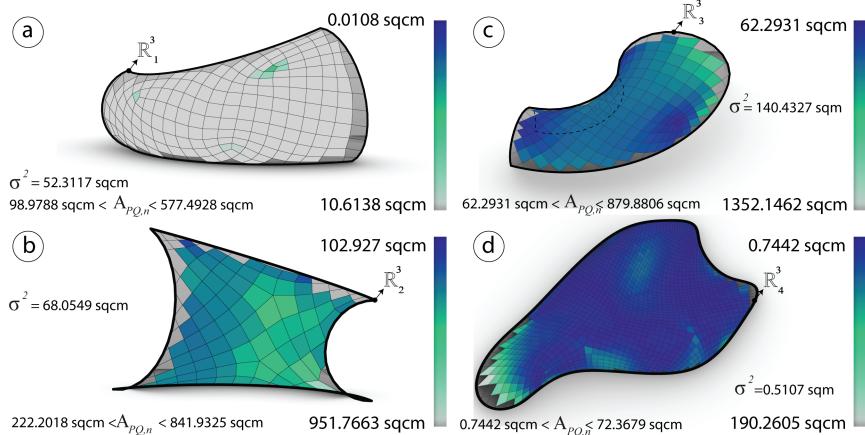


Figure 40: Before optimization: Area η_{PQ} and the variance measured in sqcm of the meshes \mathbb{R}_i^3 generated via *conformal mapping*.

2.2.3.2 Optimization

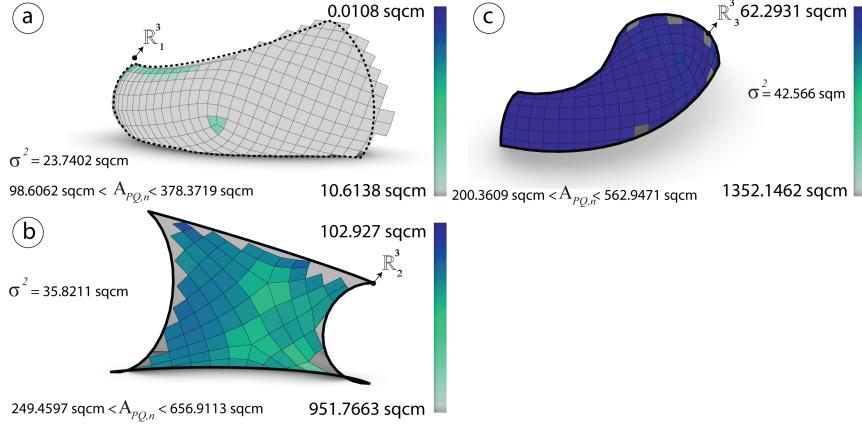


Figure 41: Before optimization: Area η_{PQ} and the variance measured in sqcm of the meshes \mathbb{R}_i^3 generated via frame field*.

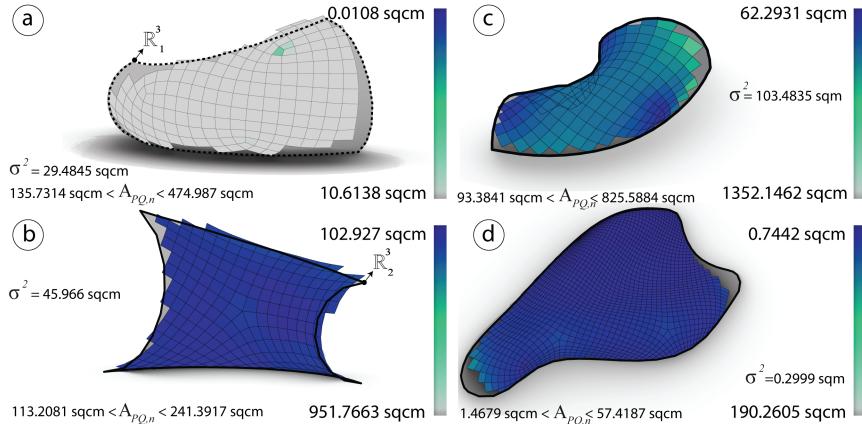


Figure 42: Before optimization: Area η_{PQ} and the variance measured in sqcm of the meshes \mathbb{R}_i^3 generated via conformal mapping.

2.2.4 Warping height

2.2.4.1 Analysis

2.2.4.2 Optimization

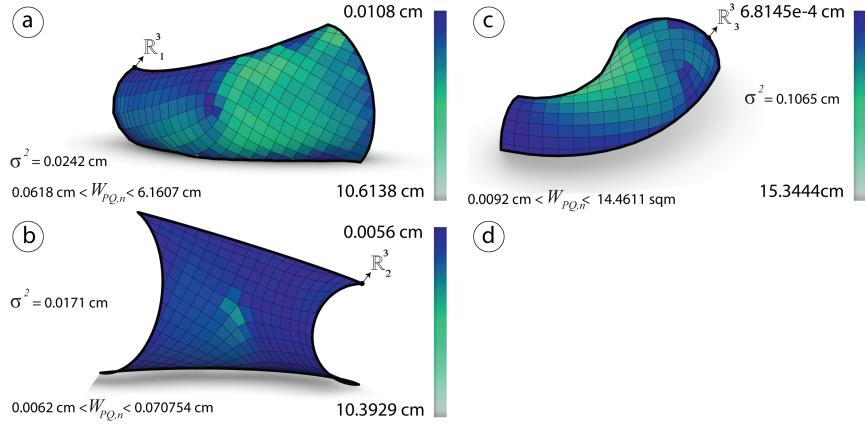


Figure 43: Before optimization: The height h and the variance measured in cm of the meshes \mathbb{R}_i^3 generated via *frame field*.

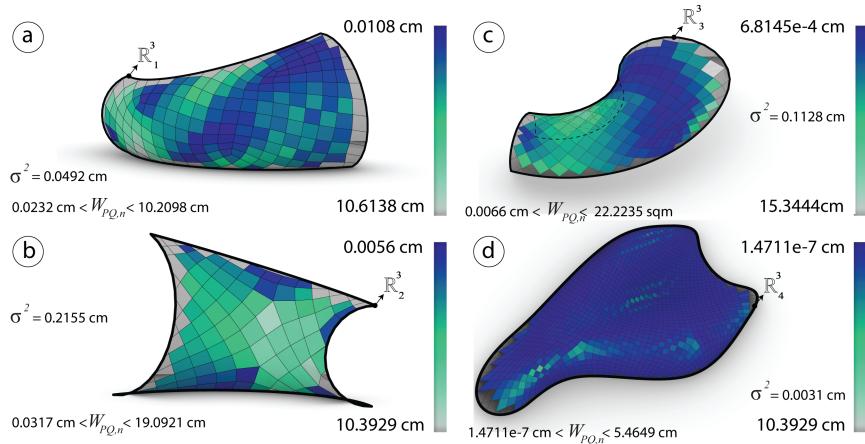


Figure 44: Before optimization: The height h and the variance measured in cm of the meshes \mathbb{R}_i^3 generated via *conformal mapping*.

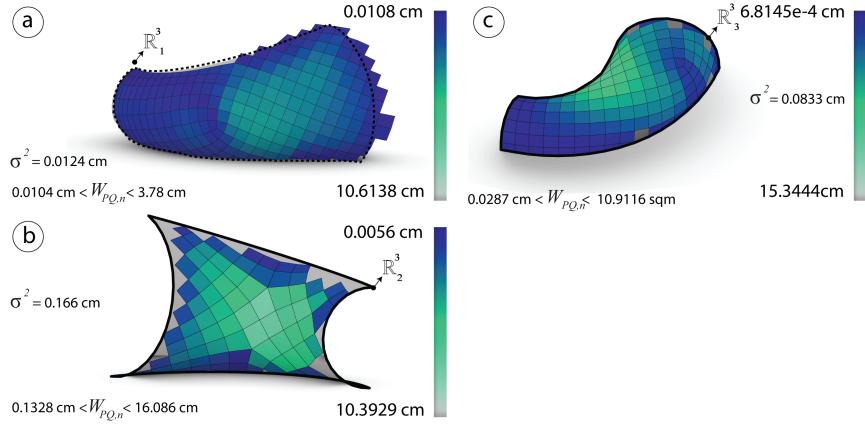


Figure 45: After optimization: The height h and the variance measured in cm of the meshes \mathbb{R}_i^3 generated via *frame field*.

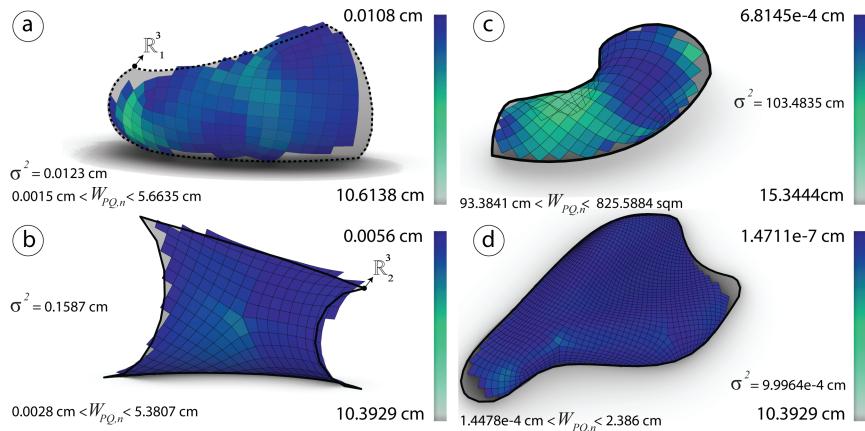


Figure 46: After optimization: The height h and the variance measured in cm of the meshes \mathbb{R}_i^3 generated via *conformal mapping*.

2.3 Subdivision Strategy (Starting with a Coarse Quad-Dominant mesh)

2.3.1 Subdivision Strategy principles

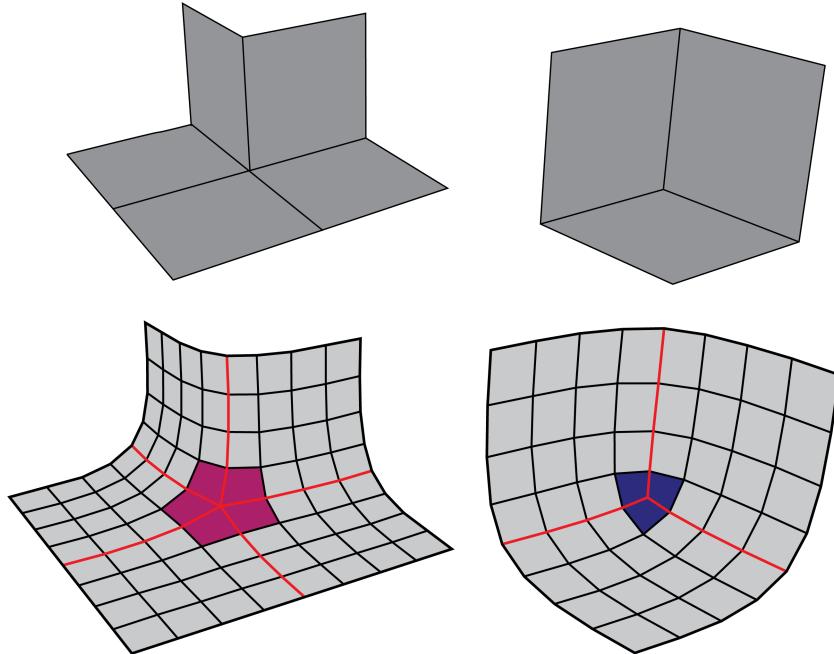


Figure 47: Left: Singularities with negative indices. Right: Singularities with positive indices.

A coarse mesh that approximate the topology of a input surface can be subdivided using catmull clark algorithm (Piacentino 2010). For PQ meshes, the valence of the each vertex should be four, vertices with a valence more then four are considered as singularities. After applying the subdivision on the coarse mesh, singularities with negative indices take a negative curvature and singularities with a positive indices take a positive curvature see fig. 47.

2.3.2 Curvature and singularities analysis

Givin four different input surfaces, first the curvature is analyzed and the singularities are placed by index see fig. 48.

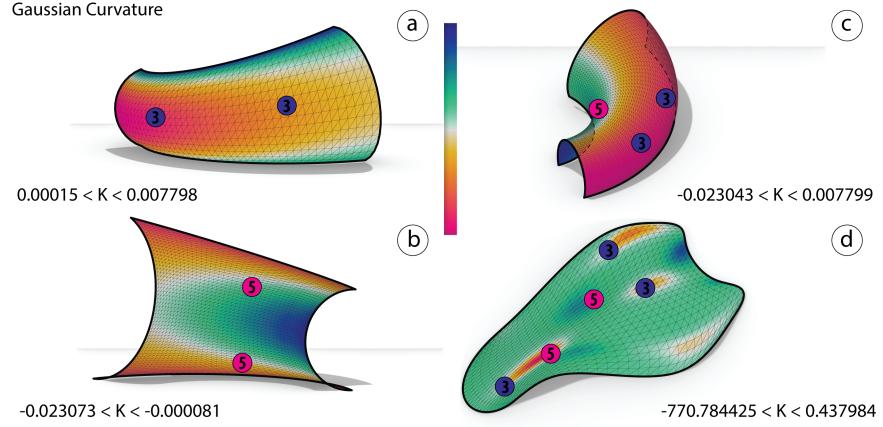


Figure 48: Singularities with indices $-\frac{1}{4}$ and $\frac{1}{4}$ are placed according to the curvature.

2.3.3 Generating the coarse mesh

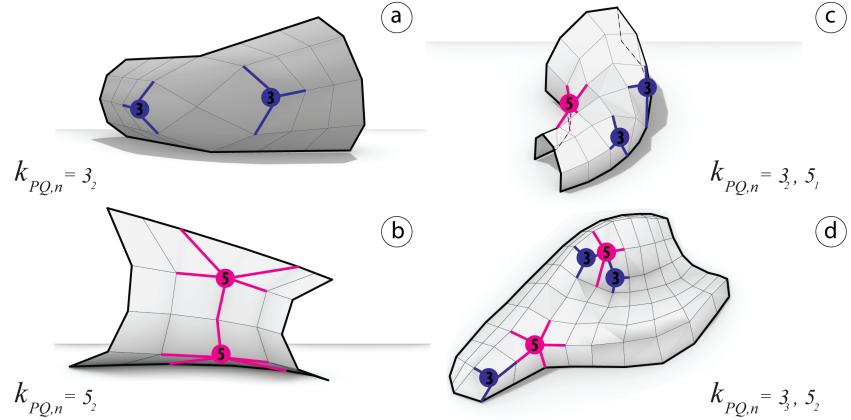


Figure 49: The coarse meshes are generated thru isocurves

Subsequently to the previous step, a 2D map by patches have been generated. Such method can help out to predict the pre-networking between singularities and to avoid unexpected ones. Therefore it now possible to generate the coarse mesh following the 2D map see fig. 49.

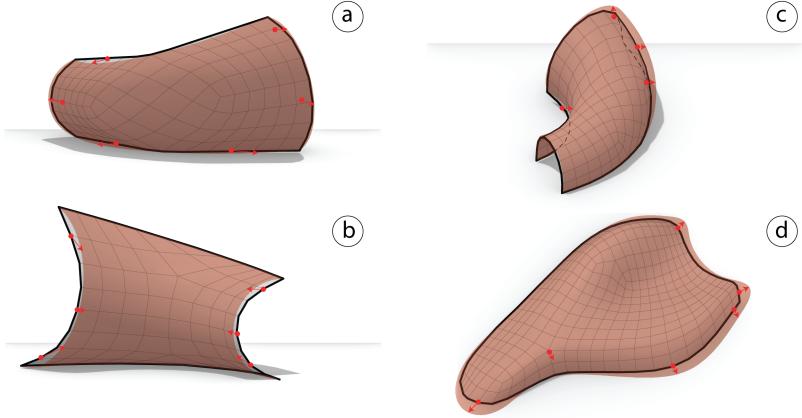


Figure 50: Left: Subdivided mesh using catmull-clark algorithm and singularities in color. Right: Pulling the subdivided mesh to the input surface.

2.3.4 Catmull-clark subdivision and pull to mesh

Thereafter, the catmull-clark algorithm has been added to the coarse mesh. Therefore, using kangaroo2 (Piker 2010) the coarse mesh has been pulled by constraining the latter's points on the input mesh. Finally the coarse mesh takes the input shape see fig. 50.

2.3.5 Optimization

The panels are ready to be fully optimized by constraining the faces under the planarity goal using (Piker 2010) solver, surface fairness and planarity and continuously modified in order to achieve the objectives.

2.3.5.1 Planarity

2.3.5.1.1 Analysis

2.3.5.1.2 Optimization

2.3.5.2 Diagonals aspect ratio

2.3.5.3 Analysis

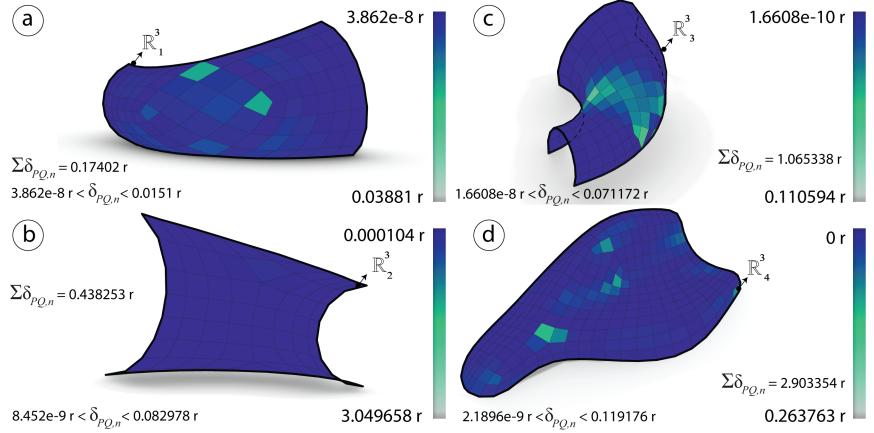


Figure 51: Before optimization: Planarity δ_{PQ} measured in radians of the meshes \mathbb{R}_i^3 generated via subdivision technique.

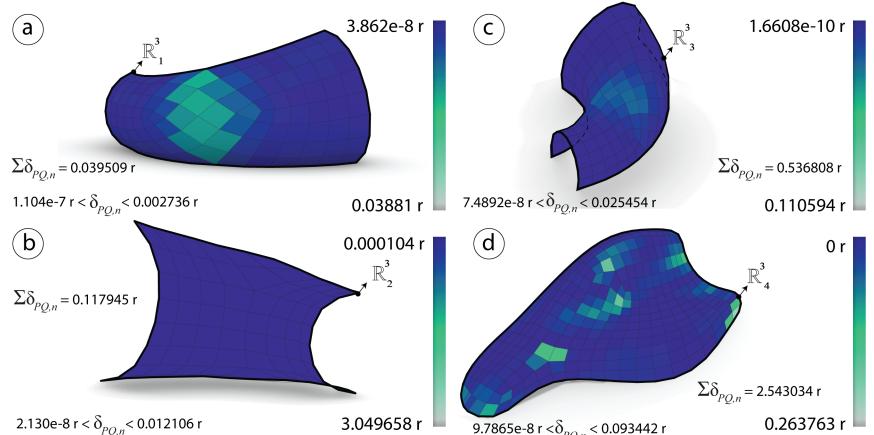


Figure 52: After optimization: Planarity δ_{PQ} measured in radians of the meshes \mathbb{R}_i^3 generated via subdivision technique.

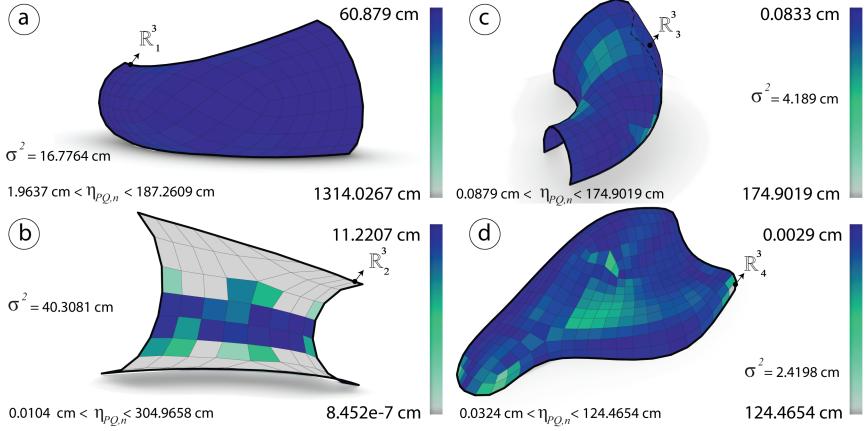


Figure 53: Before optimization: Diagonals aspect ratio η_{PQ} measured in cm of the meshes \mathbb{R}_i^3 generated via subdivision technique.

2.3.5.4 Optimization

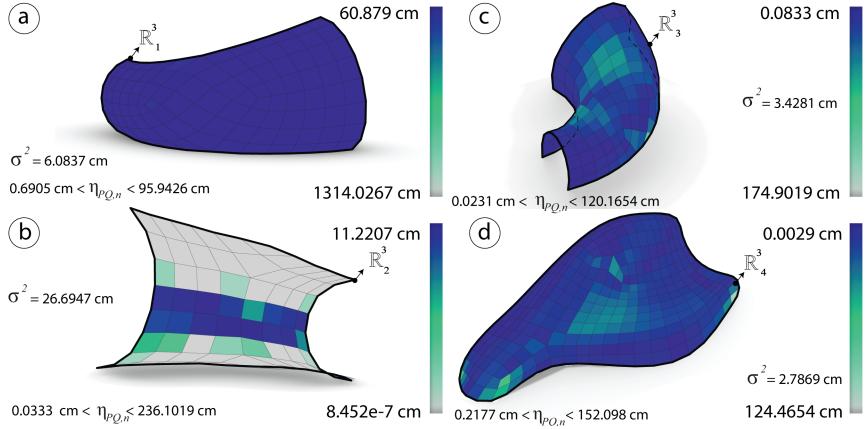


Figure 54: After optimization: Diagonals aspect ratio η_{PQ} measured in cm of the meshes \mathbb{R}_i^3 generated via subdivision technique.

2.3.5.5 Element areas

2.3.5.6 Analysis

2.3.5.7 Optimization

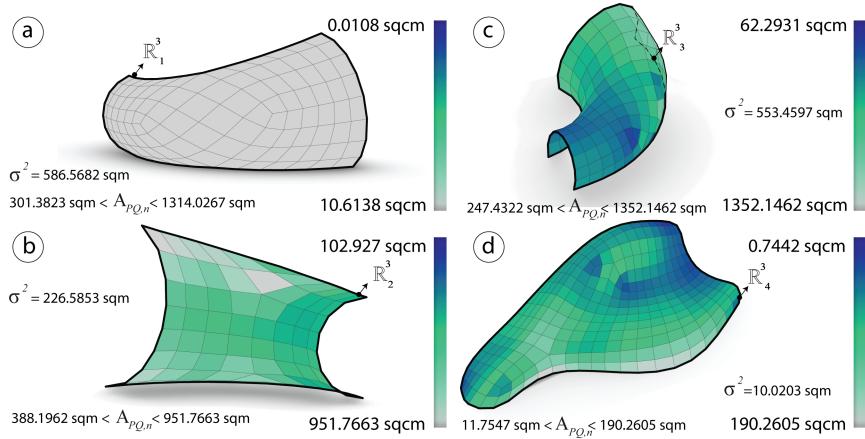


Figure 55: Before optimization: Area η_{PQ} and the variance measured in sqcm of the meshes \mathbb{R}_i^3 generated via subdivision technique.

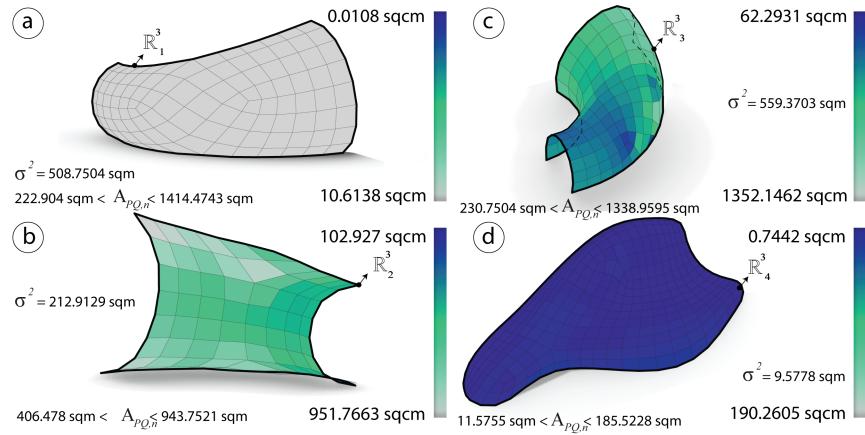


Figure 56: After optimization: Area η_{PQ} and the variance measured in sqcm of the meshes \mathbb{R}_i^3 generated via subdivision technique.

2.3.5.8 Warping height

2.3.5.9 Analysis

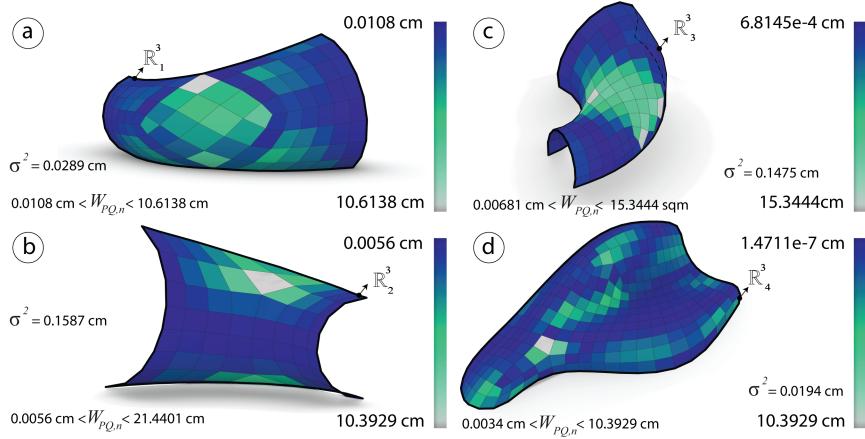


Figure 57: Before optimization: The height h and the variance measured in cm of the meshes \mathbb{R}_i^3 generated via subdivision technique.

2.3.5.10 Optimization

2.4 Comparison & Synthesis

3 Conclusion

PQ meshes are very hard to deal with when the input surface is a free-form. However some algorithms have shown the differences between them and there results. Having a conjugate direction field as a tool to control the mesh layout is very useful. Thus generating PQ meshes from curve network is robust as well. The two different methods are almost planar after generation since they are extracted from the principal directions. The conical optimization has proved its robustness over planar quad meshes. By optimizing and combining the methods the last one was to generate planar quads by subdividing a coarse mesh and then optimize it. The boundary condition has been neglected, for further research the latter will be taken in consideration while genearting the PQ meshes.

Bommes, David, Henrik Zimmer, and Leif Kobbelt. 2009. “Mixed-Integer Quadrangulation.” In *ACM Transactions on Graphics (Tog)*, 28:77. 3. ACM.

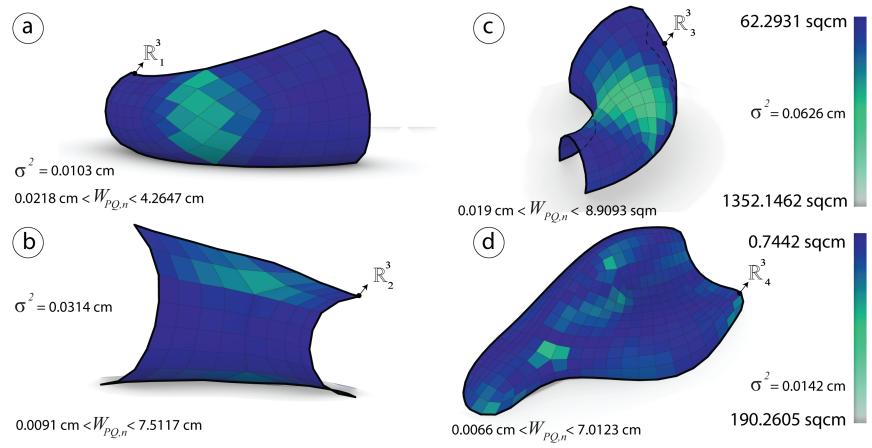


Figure 58: After optimization: The height h and the variance measured in cm of the meshes \mathbb{R}_i^3 generated via subdivision technique.

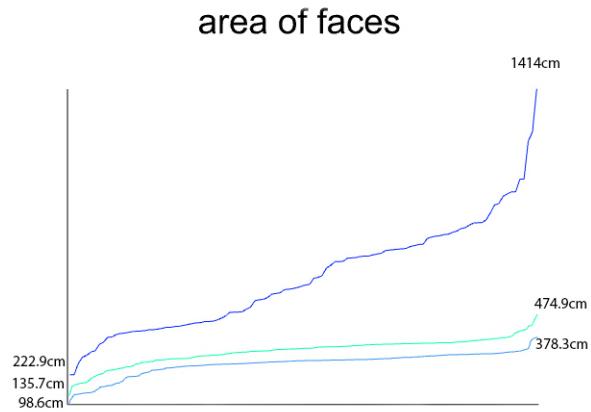


Figure 59: Graph showing the difference between the elements areas and the number of panels for each of the first mesh \mathbb{R}_1^3 generated by the three different techniques

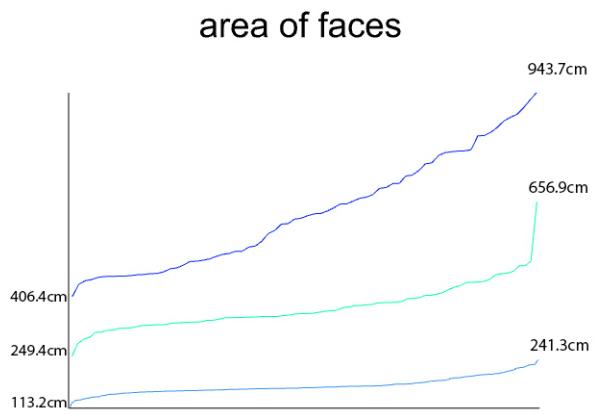


Figure 60: Graph showing the difference between the elements areas and the number of panels for each of the first mesh \mathbb{R}_2^3 generated by the three different techniques

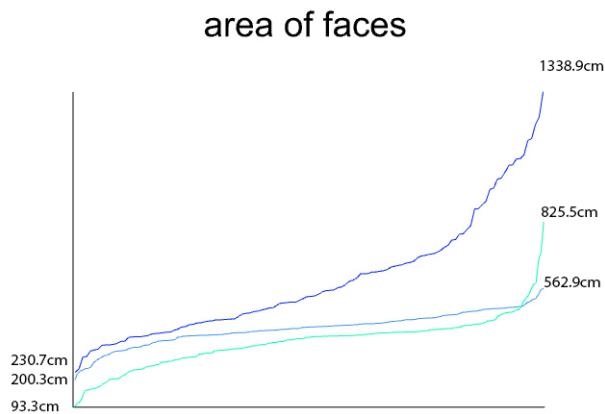


Figure 61: Graph showing the difference between the elements areas and the number of panels for each of the first mesh \mathbb{R}_3^3 generated by the three different techniques

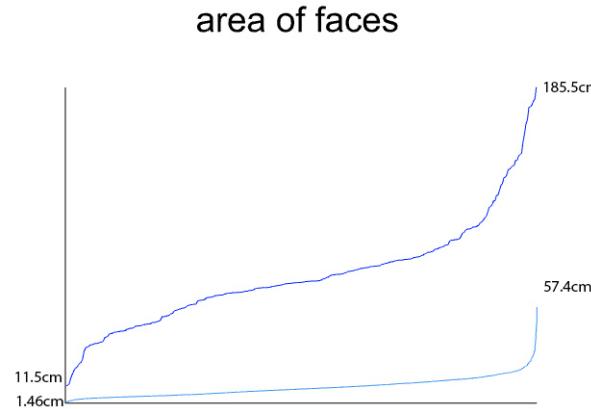


Figure 62: Graph showing the difference between the elements areas and the number of panels for each of the first mesh \mathbb{R}_4^3 generated by the three different techniques

Diamanti, Olga, Amir Vaxman, Daniele Panozzo, and Olga Sorkine-Hornung. 2014. “Designing N-Polyvector Fields with Complex Polynomials.” In *Computer Graphics Forum*, 33:1–11. 5. Wiley Online Library.

Ebke, Hans-Christian, David Bommes, Marcel Campen, and Leif Kobbelt. 2013a. “QEx: Robust Quad Mesh Extraction.” *ACM Transactions on Graphics (TOG)* 32 (6). ACM:168.

———. 2013b. “QEx: Robust Quad Mesh Extraction.” *ACM Trans. Graph.* 32 (6). New York, NY, USA: ACM:168:1–168:10. <https://doi.org/10.1145/2508363.2508372>.

Geometry, CUBIT Sandia, and Mesh Generator. 2016. “Metrics for Quadrilateral Elements.” 2016. https://cubit.sandia.gov/public/15.2/help_manual/WebHelp/mesh_generation/mesh_quality_assessment/quadrilateral_metrics.htm.

Glymph, James, Dennis Shelden, Cristiano Ceccato, Judith Mussel, and Hans Schober. 2004. “A Parametric Strategy for Free-Form Glass Structures Using Quadrilateral Planar Facets.” *Automation in Construction* 13 (2). Elsevier:187–202.

Gokhale, Nitin S. 2008. *Practical Finite Element Analysis*. Finite to infinite.

Greco, Lorenzo. 2018. “Plugin for Advanced Mesh Editing, Remeshing, Quads from Fields, Mesh Analysis.” 2018. <https://www.food4rhino.com/app/capybara-guide>.

Autodesk Nastran Reference. 2018. “Geometry Check Options.” 2018. <https://knowledge.autodesk.com/support/nastran/learn-explore/caas/CloudHelp/cloudhelp/2019/ENU/NSTRN-Reference/files/GUID-69125711-AB89-4A46-8FA9-B7DB2C856A13.htm.html>.

- Jacobson, Alec, Daniele Panozzo, and others. 2017. “libigl: A Simple C++ Geometry Processing Library.”
- Leon, Alexander Peña de, and Dennis Shelden. 2011. “A Technique for the Conditional Detailing of Grid-Shell Structures: Using Cellular Automata’s as Decision Making Engines in Large Parametric Model Assemblies.” In *Computational Design Modelling*, 267–73. Springer.
- LibDirectional. 2018. “Matlab Library for Directional Statistics and Directional Estimation.” 2018. <https://github.com/libDirectional>.
- Libgil. 2017. “Simple C++ Geometry Processing Library.” 2017. <https://github.com/libgil/libgil>.
- Liu, Yang, Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, and Wenping Wang. 2006. “Geometric Modeling with Conical Meshes and Developable Surfaces.” In *ACM Transactions on Graphics (ToG)*, 25:681–89. 3. ACM.
- Liu, Yang, Weiwei Xu, Jun Wang, Lifeng Zhu, Baining Guo, Falai Chen, and Guoping Wang. 2011. “General Planar Quadrilateral Mesh Design Using Conjugate Direction Field.” In *ACM Transactions on Graphics (ToG)*, 30:140. 6. ACM.
- Michalatos, Panagiotis. 2017. “Millipede for Structural Analysis and Optimization.” 2017. <http://www.sawapan.eu/>.
- Mueller, Caitlin, and Sigrid Adriaenssens. 2018. “Optimized Quad Gridshell from Stress Field and Curvature Field.”
- Piacentino, Giulio. 2010. “Weaverbird – Topological Mesh Editor.” 2010. <http://www.giuliopiacentino.com/weaverbird/>.
- Piker, Daniel. 2010. “Kangaroo Is an Interactive Physics/Constraint Solver and Grasshopper Plugin for Designers.” 2010. <http://kangaroo3d.com/>.
- . n.d. “MeshMachine for Remeshing.” <https://www.grasshopper3d.com/profiles/blogs/meshmachine-update?id=2985220%3ABlogPost%3A1085830&page=2>.
- Pottmann, Helmut, and Johannes Wallner. 2008. “The Focal Geometry of Circular and Conical Meshes.” *Adv. Comp. Math* 29:249–68.
- Schiftner, Alexander, and Jonathan Balzer. 2010. “Statics-Sensitive Layout of Planar Quadrilateral Meshes.” *Advances in Architectural Geometry 2010*. Springer, 221–36.
- Zadravec, Mirko, Alexander Schiftner, and Johannes Wallner. 2010. “Designing Quad-Dominant Meshes with Planar Faces.” In *Computer Graphics Forum*, 29:1671–9. 5. Wiley Online Library.