Localized remeshing for polyhedral splines

Kęstutis Karčiauskas[a], Jörg Peters[b,*]

[a] *Vilnius University*
[b] *University of Florida*

Abstract

Smooth spline surfaces can now be built with polyhedral control nets rather than just grid-like tensor-product control nets. However, irregularities such as T-junctions, multi-sided facets, and n-valent vertices need to be sufficiently separated. Automatically generated quad-dominant meshes, and meshes created by designers unaware of the requirements for spline surfaces often pack irregularities too tightly.

Global refinement, e.g. via two steps of subdivision, can sufficiently separate irregularities. However, each refinement quadruples the number of polynomial pieces. Moreover, first-step artifacts can lead to oscillating and pinched highlight line distributions. We therefore investigate minimal, single edge insertion, re-connection and localized refinement of quad-dominant meshes to make them suitable for polyhedral splines.

Keywords: polyhedral control net, quad-dominant mesh, free-form surface, polyhedral spline, T-junction, localized refinement

1. Introduction

Tensor-product splines [6, 9] can convert faceted meshes into smooth surfaces provided the meshes have the connectivity of a grid. Polyhedral splines [16] can convert more general control nets that can include T-junctions, multi-sided facets, and n-valent vertices – provided they are sufficiently separated. For example, non-4-sided facets must be surrounded by quads. However designers or algorithms that are unaware of this requirement often generate almost, but not quite admissible control nets, see Fig. 1 and Fig. 17. Typically such meshes cannot be sent back or regenerated for improvement, but can only be post-processed. Other than redesign by hand, currently the only option available to practitioners is global refinement, e.g. applying two subdivision steps globally. This separates the irregular configurations by quadrilateral strips, but comes at the cost of many more polynomial pieces. Moreover, shape can suffer, see Fig. 2d: First step artifacts [1], can create unwanted oscillations and pinched highlight lines [2] and can destroy alignment with feature lines. All these flaws are unacceptable for higher-end design.

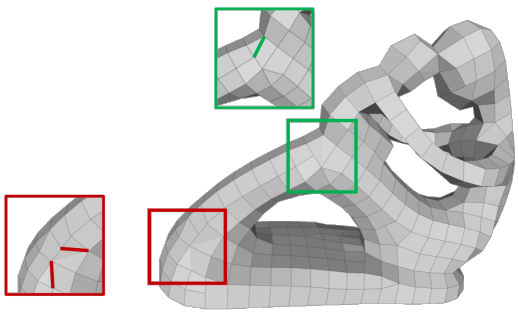
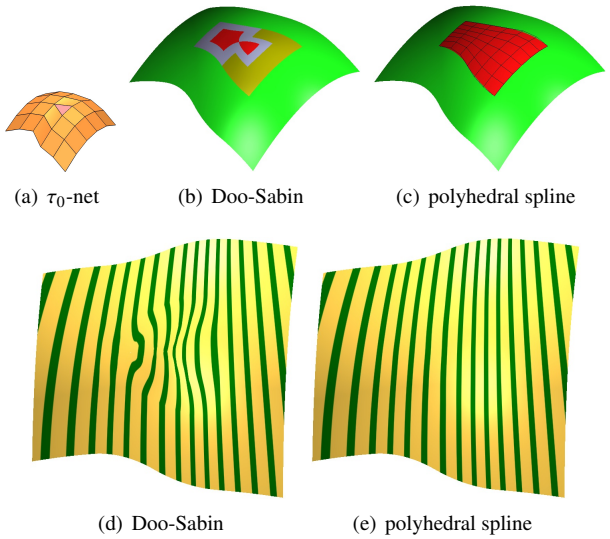


Figure 1: Local re-connection to τ_0-nets to avoid τ_1 τ_1 adjacency, or τ_1 τ_0 adjacency.

This paper therefore explores minimal *local* re-connection and refinement recipes for making the mesh

*Corresponding author
Email address: jorg.peters@gmail.com (Jörg Peters)



(a) τ_0-net (b) Doo-Sabin (c) polyhedral spline

(d) Doo-Sabin (e) polyhedral spline

Figure 2: (a) Polyhedral input net with τ_0 configuration. Applying (b,d) Doo-Sabin subdivision, (c,e) polyhedral splines. (b,c) For context, the configuration is surrounded by a strip of C^1 bi-2 splines (green). i.e. compare the normal variation. Uniform highlight line distribution [2] is a popular criterion for good shape. The highlight lines (e) for polyhedral spline are uniform, whereas (d) Doo-Sabin's oscillate.

suitable as a polyhedral spline control net. Since neither designers nor algorithms create random quad-dominant meshes, we do not consider all possible scenarios. That is, we do not seek algorithms that guarantee suitability for arbitrary input meshes. Such general remeshing could be as hard as creating sparse polyhedral spline-aware layout ab initio. In particular, since automatic quad-dominant algorithms such as [10, 19] generate ever more tightly packed irregularities when the resolution is reduced, we assume that the quad-count has been prescribed so that only isolated small submeshes require local adjustment.

This analysis of minimal *local* re-connection and refinement recipes focuses neither on (quad-)remeshing nor on surface construction. Rather the goal is to better understand how to distribute algorithmic complexity between generalized spline constructions and (quad-)meshing. That is, we observe how much structure the meshing algorithm must provide vs how complex the spline surface must be: requiring pure quad meshes challenges the meshing algorithm whereas unstructured quad-dominant meshes forces many specialized spline constructions. We therefore analyze strategies for *localized post-processing* of meshes with tightly-packed irregular points, facets and T-junctions to convert the mesh into a control net suitable for polyhedral splines and so improve the interplay between meshing and high-end surfacing. Turning the resulting insights into a complete algorithmic framework require future collaboration towards 'spline-aware meshing' between meshing specialists and spline specialists. Here our contributions are to show options for simple, localized post-processing of insufficiently-structured meshes:

- analyzing shape considerations for single edge insertion,
- introducing local Augmented Refinement for isotropic n-gon refinement,
- local T-gon refinement to honor preferred directions and
- unified refinement for abutting T- and n-gons.

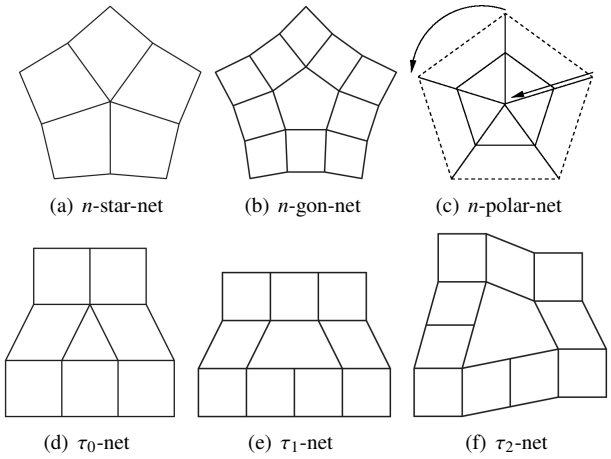
Outline Section 1.1 reviews constructions that generalize bi-2 splines, including polyhedral splines. Section 2 summarizes the mesh configurations that can serve as control nets for polyhedral splines. Section 3 considers the shape implications of adding single edges to a mesh. Section 4 introduces localized Augmented Refinement for multi-sided facets. Section 5 introduces localized refinement for T-gon facets. Section 6 organizing overlappiong refinements for abutting T-gon and n-gon facets. Section 7 compares the refined meshes and their effect on surface shape.

1.1. Generalized bi-2 splines, and fast quad-dominant meshing

Polyhedral splines [16] generalize tensor-product bi-quadratic (bi-2) splines by combining algorithms from [13, 14, 15]. The polynomial pieces of a polyhedral spline have degree at most bi-cubic (bi-3). There is also a purely degree bi-2 construction [18] but the shape is reportedly not satisfactory. Bi-2 splines have minimal polynomial bi-degree for smoothing out a quadrilateral mesh. When the input is a quad-dominant mesh, Doo-Sabin subdivision [8] comes to mind. However, this classic generalization of bi-2 splines consists of an infinite sequence of nested (contracting) polynomial surface rings and fails to yield good shape due to artifacts generated already in the

first steps – as the highlight line distribution [2] of Fig. 2d demonstrates. (Similar, but lesser artifacts afflict Catmull-Clark subdivision [5], see [11]). The highlight line distortions of classical subdivision algorithms also disqualify them as a post-processing step to turn quad-dominant meshes into pure quad meshes for splines (see for example in [10]). Augmented Subdivision [12] yields much better shape by adding a carefully chosen central guide point.

We focus here on the quad-dominant meshing algorithms in [10, 19], primarily because code is available and due to the complexity and higher quad-count of strict quad-meshing algorithms, see e.g. [3, 4, 17, 20]. There appears to be little use trying to compile statistics on how many meshes generated by automatic quad-dominant algorithms fail to be polyhedral spline-suitable: the outcome depends too much on the geometry of the object and the required resolution. High resolution meshes are almost always suitable but low quad-count is desirable when the final shape can be captured by a few splines rather than highly-refined quad-facet approximations.



(a) n-star-net (b) n-gon-net (c) n-polar-net

(d) τ_0-net (e) τ_1-net (f) τ_2-net

Figure 3: Polyhedral control net patterns of [16]. The valence in (a,b,c) is $n = 5$. The triangle in (d) is called a T_0-gon. The pentagon in (e) is called a T_1-gon. The hexagon in (f) is called a T_2-gon.

2. Meshes and control nets

Regular 3×3 grids of quadrilateral facets (quads) can be interpreted as the control net of a bi-2 spline in uniform B-spline form [7]. A natural generalization are polyhedral meshes that can be interpreted as control nets of polyhedral splines. Suitable polyhedral control nets include nodes where $n \neq 4$ quads join (Fig. 3a), nets with n-sided facets surrounded by quads (Fig. 3b), polar nets (Fig. 3c) and special multi-sided facets, called T-gons with T-junctions, surrounded by quads (Fig. 3d,e,f). Many hand-crafted models naturally include these irregularities to merge n features (a,b,c) or adjust mesh density (d,e,f). The triangles in configuration (c) can be viewed as quadrilaterals with one edge collapsed to close a cylinder. This can model umbilics such as at a finger tip.

Formally, for $m > 4$, a T_{m-4}-gon is an m-gon surrounded by quads such that all its vertices are either T-junctions or have valence 4. An τ_0-net (Fig. 3d) is a special configuration with a triangle (a T_0-gon) at the center surrounded by 7 quads. The τ_0-net has two vertices of valence 4 and one of valence 5. A T_1-gon (Fig. 3e) has one T-junction and is formally a pentagon, The T_0 configuration frequently arises in quad-dominant meshing algorithms, for example [10, 19], see Fig. 4. A T_2-gon (Fig. 3f) has two T-junctions and is formally a hexagon.

A mesh is suitable for polyhedral splines if it consists of the polyhedral patterns shown in Fig. 3. In the available implementations, the valence n of the n-star-net, n-gon-net or n-polar-net is expected to be less than 9, although for polar nets the valence can certainly be higher without harming shape.

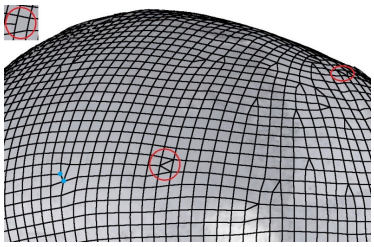


Figure 4: Mesh generated by [19]. There are many τ_0 configurations in the mesh but only some pack irregularities too close for polyhedral splines. (see red circles: the upper left configuration is allowable, but not the packed τ_0 configurations.) Pairs of non-separated irregular nodes, e.g. those marked as • present no problem for polyhedral splines.

3. Edge insertion and shape

Fig. 1 and Fig. 4 illustrate how adding or removal of a single edge can turn the output of [10, 19] into a polyhedral spline control net.

As with tensor-product patches, some basic rules should be followed to attain good shape. To avoid 'camel-back' oscillations as for tensor-product splines with control points elevated along a diagonal. For best outcomes, T-gons should be aligned with creases as in Fig. 5g (see the polyhedral spline surface in Fig. 5h). Placing a T_1-gon across a sharp crease (Fig. 5a) causes an unexpected (unwanted) dip marked by ↓ in the polyhedral spline surface figreffig:creased. Splitting the T_1-gon into a quad facet and a T_0-gon to its right, i.e. a right bias in the split to avoid that a T-gon straddles the sharp crease, results in a surface with a barely noticeable dip, see Fig. 5e. The dip in Fig. 5f stems from the twisted quad in the left-biased split (Fig. 5c) that generates a T_0-gon and a straddling quad facet. Even the right-biased option has a dip since

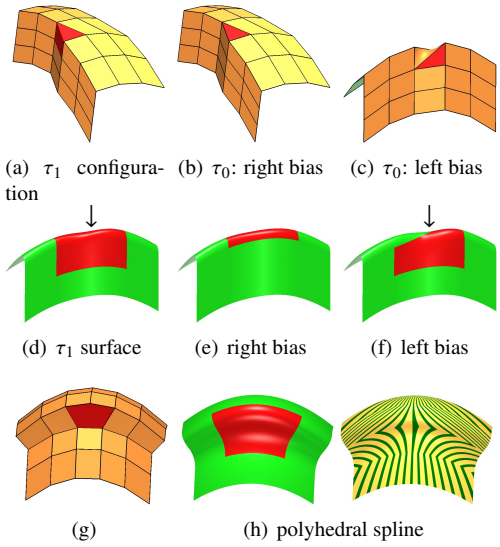


Figure 5: *top* rows: Base of the T_1-gon (non-contracting direction) is not aligned with the ridge. The ↓ indicates a dip. *bottom* row: better shape due to aligned ridge.

the base direction along which the the grid lines are not merged, i.e. the non-contracting direction, w,s not aligned with the crease direction. This waviness can be avoided by proper alignment of the base with the crease as in Fig. 5g. (The horizontal sharp crease in the highlight lines in Fig. 5h originates with the regular bi-2 surroundings.) Similarly the base of a T_0-gon with the valence 4 nodes should be placed along the crease direction.

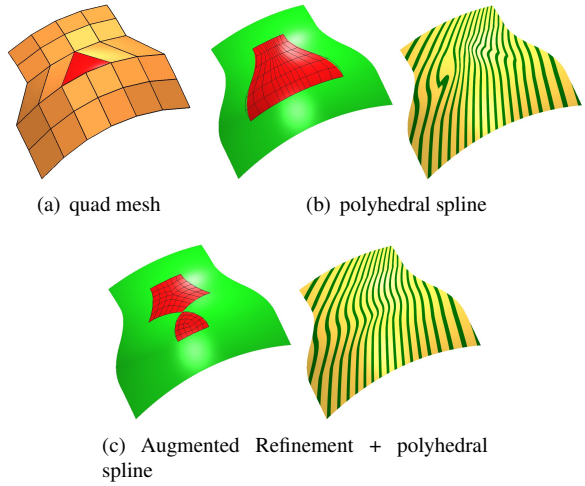


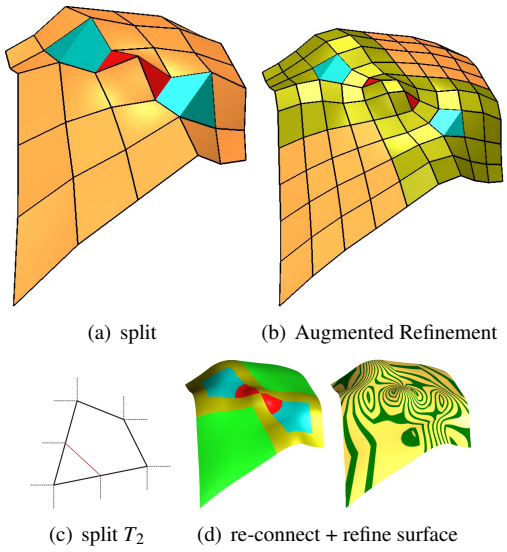
Figure 6: Quad meshes with high contraction ratios challenge polyhedral splines. Here $r = 1:3$, i.e. 3 control nodes merge to 1 via the red, nominally 4-sided facet. Note that, to emphasize the shape problem, the control net is placed sideways (not symmetric) to the highlight line direction.

Edge insertion (or removal) should not create T-gons that have more than one T-junction more on one side than on the other. Violating this rule means poorly graded transitions. Let r denote the ratio of the vertex count at the sparse side divided by the dense side of the T-gon, e.g. $r = 1{:}2$ for τ_0 since there is one node at the sparse and two at the dense end. For τ_1, $r = 2{:}3$. A polyhedral spline for a T-net with lower r, e.g. $r = 1{:}3$ in Fig. 6a, is typically not well-shaped, see Fig. 6b, even after Augmented Refinement Fig. 6c, explained in the next section.

For tight configurations, as in Fig. 7a, edge-insertion that splits a T_2-gon into a triangle and a pentagon (Fig. 7c), needs to be followed by local Augmented Refinement, (Fig. 7b), as defined next.

4. Local "Augmented Refinement"

Subdivision can be used to separate the irregularities in an input mesh. However, already a single Doo-Sabin subdivision step can cause oscillation in the highlight line



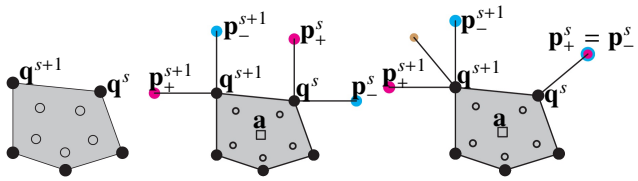
(a) split (b) Augmented Refinement

(c) split T_2 (d) re-connect + refine surface

Figure 7: Two T_2-gons sharing a vertex. (a) each of the two T_2-gons (nominally a hexagon with two T-junctions, see black in (c)) is split into a triangle and a pentagon, see red line in (c). (b) Augmented Refinement of the input mesh (d) regular bi-2 patches obtained after re-connection and refinement, 3-sided and 5-sided cap.

distribution. By adding a carefully-chosen auxiliary central point $\mathbf{a}$ (see Fig. 8b), Augmented Subdivision [12] is known to yield better shape. Here $\mathbf{a}$ is an affine combination of the nodes $\mathbf{q}^s$, $s = 0, \ldots, n-1$, of the n-sided facet and direct neighbor nodes $\mathbf{p}^s_+$ and $\mathbf{p}^s_-$. At global boundaries, we can choose to add an outer sacrificial layer to either mirror the immediate interior layer across the boundary sequence of edges, or replicate the boundary edges to form boundary facets of zero area. We can thus restrict the discussion to interior faces. As for Doo-Sabin subdivision, after one step all new nodes of the multi-sided facet are of valence 4.

Augmented Refinement is a generalization of one step of Augmented Subdivision. Augmented Refinement allows the nodes $\mathbf{q}^s$ to have valence that is not 4. Consider the general situation of Fig. 8c: $\mathbf{p}^s_-$ and $\mathbf{p}^s_+$ denote the neighbor nodes of $\mathbf{q}^s$ on edges directly preceding or succeeding the polygon edges when walking around $\mathbf{q}^s$. Augmented Refinement counts a neighbor $\mathbf{p}^s_+ = \mathbf{p}^s_-$ twice for $n = 3$, and leaves out other neighbors for $n > 4$ when



(a) Doo-Sabin (b) Augmented Subdivision (c) Augmented Refinement

Figure 8: Support (stencils) of three alternative refinements. Old facet points • and new facet points represented as ◦. (a) Doo-Sabin. (b,c) Augmented Subdivision and Augmented Refinement: •, • and •, are used to initialize $\mathbf{a}$ and generate new nodes ◦.

applying the rule

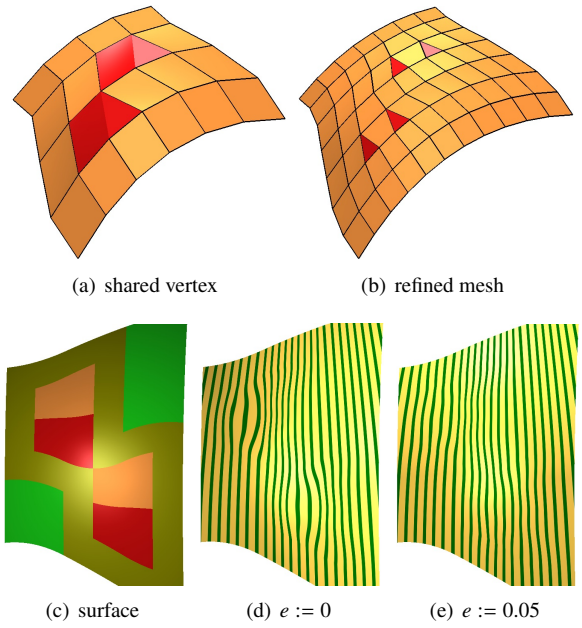
$$\mathbf{a} := := \mu_n \mathbf{q} + (1 - \mu_n)\frac{\mathbf{p}_- + \mathbf{p}_+}{2}, \quad \mu_n := \begin{cases} 3/5, & n = 3, \\ 2\frac{n-2}{n}, & n \geq 4. \end{cases} \tag{1}$$

$$\mathbf{q} := \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{q}^i, \quad \mathbf{p}_- := \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{p}^i_-, \quad \mathbf{p}_+ := \frac{1}{n}\sum_{i=0}^{n-1} \mathbf{p}^i_+. \tag{2}$$

This, ultimately simple, sub-selection of only some neighbors and double counting for $n = 3$ is based on extensive tests of alternative affine combinations of all neighbors that show that additional neighbors are not needed to obtain a much improved shape. To prevent the undue flatness observed for Doo-Sabin subdivision for a convex control net, the affine combination of averages places $\mathbf{a}$ outside the convex hull. Since $\mathbf{a}$ is an auxiliary point and not even part of the control net, this placement does not imply that the resulting surface extends outside the convex hull.

5. Local T-gon refinement

T-gon refinement addresses configurations where T-gons share vertices (see Fig. 9) or even edges. While subdivision can separate features, and Augmented Refinement improves the resulting shape considerably over Doo-Sabin, T-gon refinement is the best choice for shape near T_1-gons since it preserves the inherent preferred two directions that distinguish, say a T_1-gon from a pentagonal facet.



(a) shared vertex (b) refined mesh

(c) surface (d) $e := 0$ (e) $e := 0.05$

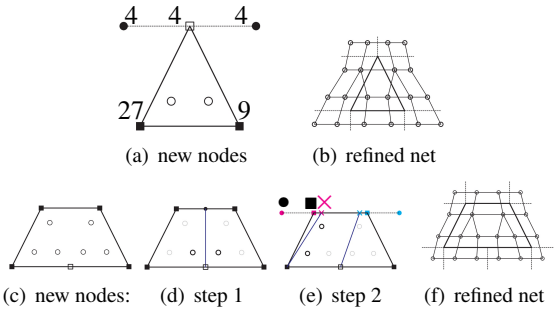
Figure 9: Two T_1-gons sharing a common vertex.

T_0-gon refinement inserts two new nodes (marked as $\circ$ in Fig. 10a) into the T_0-gon (augmented subdivision would generate three nodes). The weights, multiplied by 48, for the left $\circ$ are shown in Fig. 10a. The weights for the right are obtained by symmetry. Connecting the new nodes forms two overlapping τ_0-nets, see Fig. 10b, a permissible configuration for polyhedral splines. We note that the uniform bi-2 spline refinement stencil weights are $27, 9, 3, 9$ divided by 48. The weights $4, 4, 4$ of Fig. 10a, *top*, can therefore be viewed as a uniform re-distribution of a (collapsed) edge of a bi-2 spline with weights $3, 9$.

For *T_1-gon refinement* insert six new nodes into the T_1-gon, see Fig. 10c, (augmented subdivision would generate five nodes) as follows. As step 1, (Fig. 10d), split the T_1-gon into two quads by connecting the midpoint $\bullet$ of the top edge with the 3-valent T-vertex. Apply regular bi-2 subdivision to define the two new nodes marked as $\circ$. As step 2, (Fig. 10e), replace the initial $\blacksquare$ by $\times$ where

$$\times := (1 + e)\blacksquare - e\bullet, \quad e^{\text{default}} := 0.05, \tag{3}$$

and do alike for the right. Apply regular bi-2 subdivision to the resulting quad to define two new nodes marked as thick circles (the other two are obtained by symmetry).

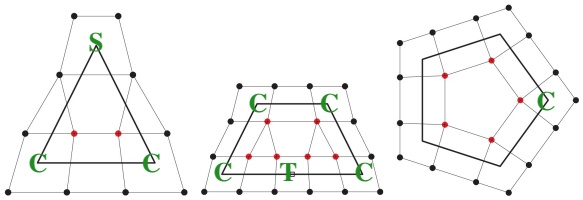


(a) new nodes (b) refined net

(c) new nodes: (d) step 1 (e) step 2 (f) refined net

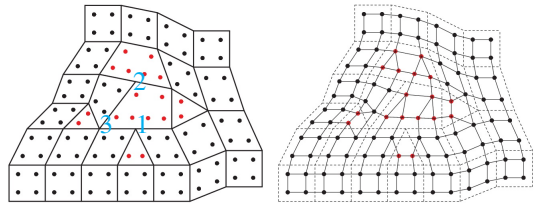
Figure 10: *Top* row: T_0-gon refinement with (a) stencil weights for the left $\circ$ to be divided by 48; *bottom* row: T_1-gon refinement.

Connect the new nodes to form two overlapping τ_0-nets as shown in Fig. 10f. The rule would be simpler if we had chosen $e = 0$. Then the refinement rules would be restricted to the vertices of the T-gon. However, as illustrated in Fig. 9, this setting of e in place of the default $e = e^{\text{default}} := 0.05$ results in visible artifacts.



(a) T_0-gon (b) T_1-gon (c) n-gon

Figure 11: Unified refinement.



(a) input mesh. new nodes $\bullet$, $\bullet$ (b) refined mesh

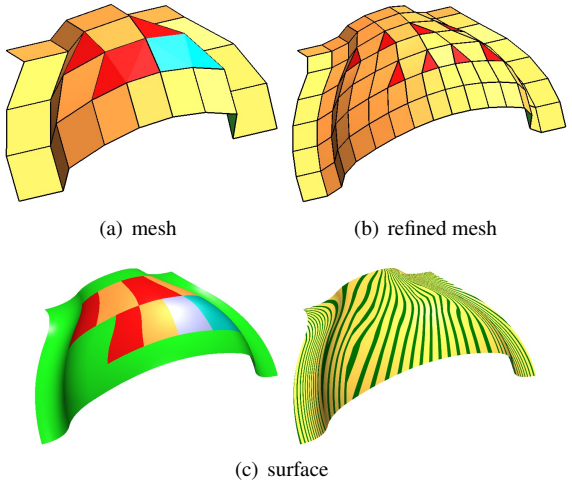
Figure 12: Unified refinement of tight meshes with non-4-valent vertices and T-junctions in close proximity. (a) $\bullet$ are defined by T-gon refinement. (b) Connecting the new nodes.

6. Unified refinement for tight configurations of T-gons and *n*-gons

We can unify the refinement where T-gons and n-gons abut and their control nets overlap. In Fig. 11, we mark as• new nodes from augmented or T-gon refinement that only depend on horizontal neighbor facets of the T-gon (• in Fig. 10a for a T_0-gon, and • or • in Fig. 10e for a T_1-gon). The other refined nodes are marked •. Now denote as T-vertex the valence 3 vertex of a T_1-gon, as S-vertex the valence 5 vertex of a T_0-gon, and as C-vertex the valence 4 (corner) vertex of a T-gon. Then refinement surrounds a

- C-vertex with one • and three • neighbors,
- S-vertex with four • neighbors,
- T-vertex with two • and two • neighbors.

Fig. 12 illustrates unified refinement. The input node 1 of valence 4 is a T-vertex in a T_1-gon but an S-vertex in a T_0-gon. Input node 2 of valence 3 is a C-vertex in the lower T_1-gon but a T-vertex in the upper T_1-gon. Input node 3 of valence 5 is a C-vertex in the T_1-gon but an S-vertex in the T_0-gon. Where both are regular, i.e. knot



(a) mesh (b) refined mesh

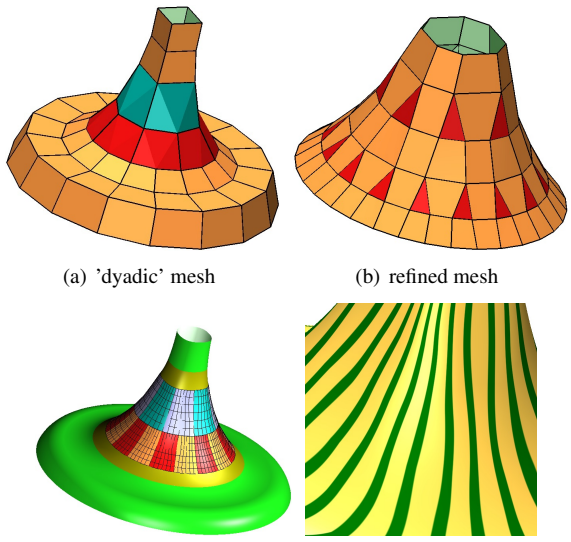
(c) surface

Figure 13: Tight configuration with two T_1-gons and two T_0-gons.

insertion splits patches, the refined patches smoothly join their coarser neighbors.

7. Examples of re-connection and refinement

The input mesh of Fig. 13a contains two T_1-gons and two T_0-gons. T-gon refinement into six overlapping τ_0-nets (Fig. 13b) makes the mesh suitable as control net for a polyhedral spline of reasonable shape given the poor initial layout, see Fig. 13c.



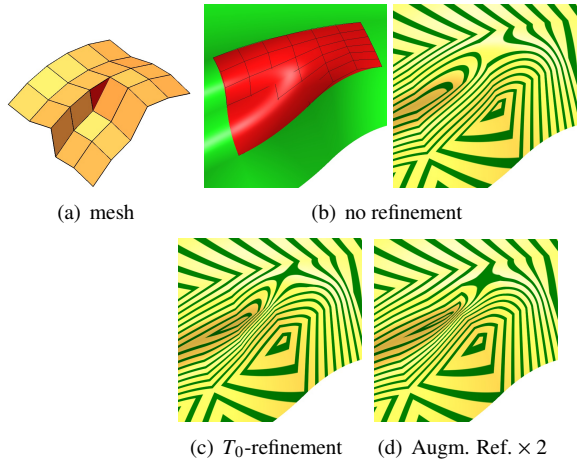
(a) 'dyadic' mesh (b) refined mesh

(c) surface (d) zoom

Figure 14: A rapid reduction in the number of quad strips by adjacent and cascading T_1-gons. (a) Input mesh. (b) Refined mesh for τ_0-surfaces. (c) BB-net overlaid on the τ_0-surface. Regular bi-2 surfaces, additional regular bi-2 surfaces from refinement. (d) zoom in into area of τ_0-surfaces.

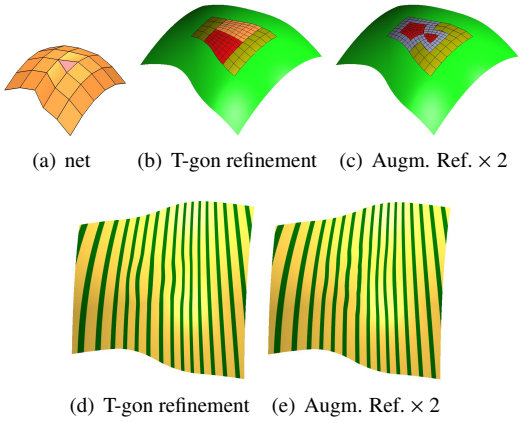
Fig. 14 features a rapid reduction in number of quad strips using tightly packed T_1-gons. T-gon refinement makes the mesh suitable as a polyhedral spline control net. While a skilled designer can model the reduction directly by placing τ_0-nets, this requires careful placement of twice as many facets. Compared to a tensor-product spline with its parameter lines aligned to the circular and vertical directions of the cylindrical shape, the highlight lines oscillate slightly due to the alternating diagonal. The simple configuration seems to directly admit (periodic) LR splines or T-splines. However, these require a globally consistent interval length assignment that is not usually available in polyhedral modelling environments, and in some cases can not exist at all. Moreover, the LR and

T-spline implementations that we are aware of do not accept a mesh with T-junctions as an input, but require the modeling of T-junctions within their framework.



(a) mesh (b) no refinement (c) T_0-refinement (d) Augm. Ref. $\times$ 2

Figure 15: T_0-gon for merging two quad strips. (c) T_0 refinement; (d) two steps of Augmented Refinement.



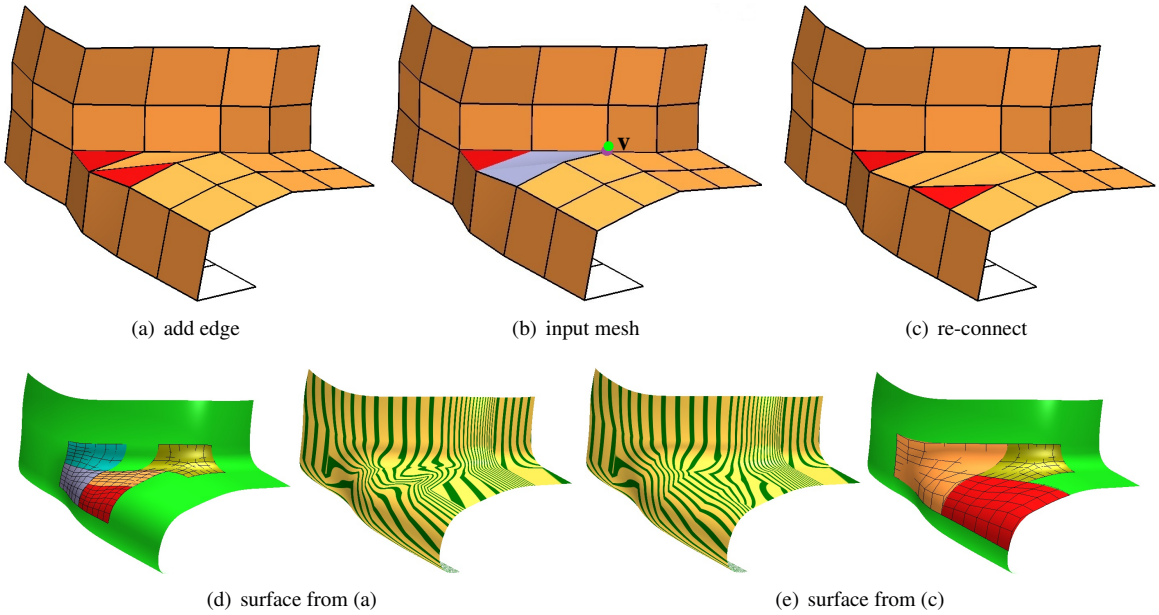
(a) net (b) T-gon refinement (c) Augm. Ref. $\times$ 2 (d) T-gon refinement (e) Augm. Ref. $\times$ 2

Figure 16: Polyhedral net of Fig. 2. Surface (b,d) T_0-refinement, (c,e) two steps of Augmented Refinement.

The T_0-configuration Fig. 15a is common in automobile outer surface design when blending two primary surfaces. To better understand the cost of localized refinement, Fig. 15b,c,d, compare direct modeling using (b) polyhedral spline with (c) T_0-refinement and (d) two Augmented Refinement steps followed by polyhedral splines. Although the overall shape remains visually alike, the highlight line distribution in the area of interest is more uniform in (b) than in (c) or (d). Fig. 16 repeats the experiment for the convex net of Fig. 2. Again, Doo-Sabin refinement (Fig. 2d) is the worst option and application of polyhedral spline without refinement (Fig. 2e) is the best option, compared to Fig. 16d,e where we observe slight oscillations when the mesh is first refined. If possible, a designer should therefore avoid refinement since refinement can introduce unwanted variation in the surface.

The mesh in Fig. 17b (a submesh of the initial Fig. 1) includes a T_0-gon adjacent to a T_1-gon. Moreover, one of the T_1-gon vertices (marked •) has valence 5. One T_0-, T_1-gon or bi-2 refinement as applicable would yield (not shown) a pentagon in place of $\mathbf{v}$ and a T_0-gon that shares one vertex with the pentagon. This would trigger a rare second round: a T_0-gon refinement and Augmented Refinement of the pentagon. Since the resulting shape is worse compared to two alternatives, it is not shown here. Inserting an edge in Fig. 17a enables T_0-refinement and a permissible n-valent star configuration at $\mathbf{v}$. Even better, re-connecting as in Fig. 17c yields sufficiently separated τ_0-nets suitable for polyhedral splines. Note that sharp changes in the highlight lines originate in the surrounding bi-2 splines. The T_0 configurations still have their 5-valent vertex poorly placed with respect to the crease, as discussed for Fig. 5. In both cases the highlight line distribution is not truly satisfactory. This is due to the placement of the T-gons with respect to the crease: the τ_0-nets have their 5-valent vertex on the crease which places the opposite line of the triangle across the crease rather than aligned, see the discussion of Fig. 5. Fig. 18 displays the results for some well-known larger-scale meshes. Note that the large-scale oscillations on the bunny surface occur in the regular bi-quadratic surface, i.e. are due to the geometry generated by the regular regions of the quadrangulation.

Putting it all together. At most two localized (Augmented) refinements guarantee a valid polyhedral spline control net: after one refinement, all new vertices have valence 4 but there can be adjacent m-sided faces that are separated in a second step. While two steps of localized refinement represent a complete algorithm, examples Fig. 15 and Fig. 16 indicate that opportunistically minimizing the number of refinement steps is preferable. Fig. 17 shows another scenario where simple re-



(a) add edge (b) input mesh (c) re-connect

(d) surface from (a) (e) surface from (c)

Figure 17: Detail from Fig. 1: Tight configuration with a T_0-gon (red) and a T_1-gon (teal) sharing an edge. One vertex $\mathbf{v}$ (•) of the T_1-gon is of valence $n = 5$. Separation by re-connection (single edge flip) yields better shape.

connection results in better shape. Ideally, future quad-meshing algorithms will therefore arise from a collaboration of meshing and spline expertise to take advantage of the flexibility of polyhedral splines, but also respect their layout limitations. As a practical step, until such algorithms are available, and given the negligible cost of generating and evaluating the polyhedral splines for a local mesh, a palette of alternative local surfaces should be presented to the designer to choose from. That is, for each isolated cluster of tight configurations, the plug-in or designer should test, in order until resolved,

- removal and insertion of edges to separate adjacent T-gons,
- unified refinement of T-gons and n-gons,
- Augmented Refinement

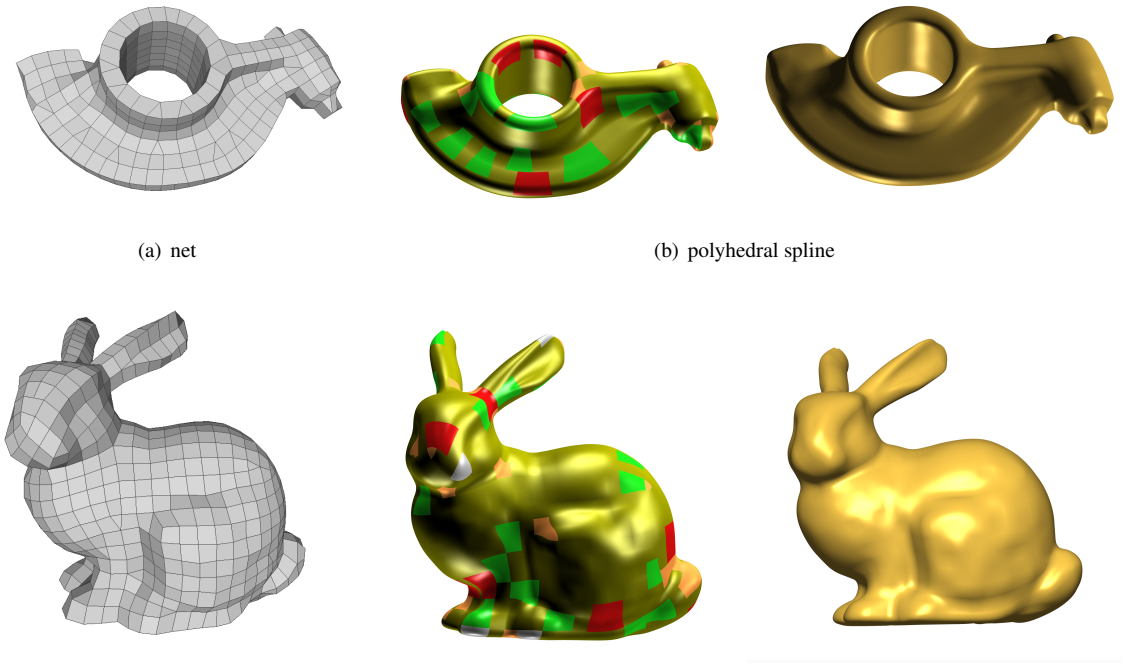
to generate nets suitable for polyhedral splines.

8. Conclusion

We explored recipes for *localized post-processing* of meshes with tightly-packed irregular points, facets and T-junctions to convert the mesh into a control net suitable for polyhedral splines and improve the interplay between meshing and high-end surfacing. We gained the following insights:

- if possible, re-connect so that T-gons appear only isolated;
- align the non-contracting direction of T_0-gons and T_1-gons with creases;
- in tight configurations, Augmented Refinement can yield good shape near T_0-gons, but not near T_1-gons;
- T-refinements are preferred over Augmented Refinement to preserve the two dominant directions of a regular surface layout.

While each of the rules is easy to apply and present to a designer, to choose from, turning the insights into an automatic algorithmic framework means that meshing specialists and spline specialists must collaborate to develop spline-aware meshing algorithms. Increasing the number of polyhedral spline configurations, to address yet more complex meshes, appears to ask of splines what may be better addressed by meshing. That is, automatic spline-aware meshing must optimize the trade-off between the



(a) net (b) polyhedral spline

(c) net (d) polyhedral spline

Figure 18: Two polyhedral splines (see [16] for the interpretation of colors) whose control nets are obtained by localized remeshing of meshes generated by [10].

complexity of meshing and the complexity of spline constructions.

Acknowledgements. Kyle Lo implemented the code to render Fig. 18.

References

[1] Ursula H. Augsdoerfer, Neil A. Dodgson, and Malcolm A. Sabin. 2009. Removing Polar Rendering Artifacts in Subdivision Surfaces. *J. Graphics, GPU, & Game Tools* 14, 2 (2009), 61–76.

[2] Klaus-Peter Beier and Yifan Chen. 1994. Highlight-line algorithm for realtime surface-quality assessment. *Computer-Aided Design* 26, 4 (1994), 268–277.

[3] David Bommes, Bruno Lévy, Nico Pietroni, Enrico Puppo, Claudio Silva, Marco Tarini, and Denis Zorin. 2013. Quad-mesh generation and processing: A survey. In *Computer Graphics Forum*, Vol. 32. Wiley Online Library, 51–76.

[4] Marcel Campen. 2017. Partitioning Surfaces Into Quadrilateral Patches: A Survey. *Comput. Graph. Forum* 36, 8 (2017), 567–588.

[5] E. Catmull and J. Clark. 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. *Computer-Aided Design* 10 (Sept. 1978), 350–355.

[6] C. de Boor. 1978. *A Practical Guide to Splines*. Springer.

[7] Carl de Boor. 1987. B-Form Basics, Gerald Farin (Ed.). SIAM, Philadelphia, 131–148.

[8] D. Doo and M. Sabin. 1978. Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design* 10 (1978), 356–360.

[9] Gerald Farin. 1988. *Curves and Surfaces for Computer Aided Geometric Design: A Practical Guide*. Academic Press.

[10] Wenzel Jakob, Marco Tarini, Daniele Panozzo, and Olga Sorkine-Hornung. 2015. Instant field-aligned meshes. *ACM Trans. Graph* 34, 6 (2015), 189:1–189:15.

[11] Kęstutis Karčiauskas, Daniele Panozzo, and Jörg Peters. 2017. T-junctions in spline surfaces. *ACM Tr on Graphics, ACM Siggraph* 36, 5 (2017), 170:1–9.

[12] Kęstutis Karčiauskas and Jörg Peters. 2015. Point-augmented biquadratic C^1 subdivision surfaces. *Graphical Models* 77 (2015), 18–26.

[13] Kęstutis Karčiauskas and Jörg Peters. 2015. Smooth multi-sided blending of biquadratic splines. *Computers & Graphics* 46 (2015), 172–185.

[14] Kęstutis Karčiauskas and Jörg Peters. 2020. Low degree splines for locally quad-dominant meshes. *Computer Aided Geometric Design* 83 (2020), 1–12.

[15] Kęstutis Karčiauskas and Jörg Peters. 2020. Smooth polar caps for locally quad-dominant meshes. *Computer Aided Geometric Design* 81 (06 2020), 1–12. PMC7343232.

[16] Kyle Lo and Jörg Peters. [n.d.]. Polyhedral Splines. https://uf-cise-surflab.github.io/blender-polyhedral-splines-web/.

[17] Max Lyon, Marcel Campen, and Leif Kobbelt. 2021. Simpler quad layouts using relaxed singularities. In *Computer Graphics Forum*, Vol. 40. 169–180.

[18] Ulrich Reif. 1995. Biquadratic G-spline surfaces. *Computer Aided Geometric Design* 12, 2 (1995), 193–205.

[19] Nico Schertler, Marco Tarini, Wenzel Jakob, Misha Kazhdan, Stefan Gumhold, and Daniele Panozzo. 2017. Field-aligned online surface reconstruction. *ACM Trans. Graph* 36, 4 (2017), 77:1–77:13.

[20] Ryan Viertel, Braxton Osting, and Matthew Staten. 2019. Coarse quad layouts through robust simplification of cross field separatrix partitions. *arXiv preprint arXiv:1905.09097* (2019).