Phong Tessellation and PN Polygons for Polygonal Models

G.J. Hettinga[1], J. Kosinka[1]

[1]Johann Bernoulli Institute, University of Groningen, Nijenborgh 9, 9747 AG Groningen, The Netherlands

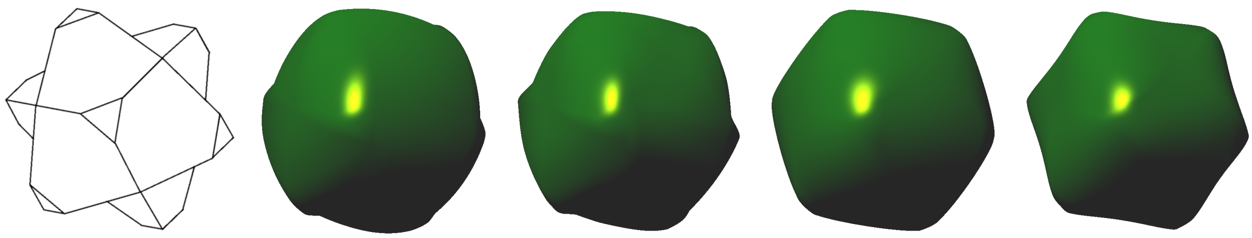

Figure 1: *Rendering a model with 8 triangles and 12 non-planar hexagons. Left to right: input model, original Phong tessellation [BA08], original PN triangles [VPBM01] (hexagons have been triangulated), extended Phong tessellation, and PN triangles extended to PN polygons.*

Abstract

We extend Phong tessellation and point normal (PN) triangles from the original triangular setting to arbitrary polygons by use of generalised barycentric coordinates and S-patches. In addition, a generalisation of the associated quadratic normal field is given as well as a simple algorithm for evaluating the polygonal extensions for a polygon with vertex normals on the GPU.

Categories and Subject Descriptors (according to ACM CCS):
I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Surface representation; splines I.3.5 [Computer Graphics]: Methodology and Techniques—Graphics data structures

1. Introduction

The creation of visually smooth geometry has long been a problem in computer graphics. The use of Phong shading gives a good enough illusion of a smooth surface by interpreting interpolated normals as the normal field of some fictional underlying surface. Although Phong shading produces nicely varying shading on flat polygons, it lacks in this regard at the edges and contours of the input geometry, where artifacts are clearly visible. This problem has been solved by generating additional geometry from the existing geometry, typically by employing Bézier triangles or tensor product Bézier patches on quadrilaterals. Bézier surfaces have been extended to higher order polygons by use of generalised barycentric coordinates, resulting into S-patches in the case of regular polygons [LD89], and mean value Bézier maps or surfaces defined on arbitrary polygons [LBS08].

A persisting problem with this approach is the automatic setting of the control points for the patches. The control points have to be placed such that patches on a mesh are continuous (C^0) and preferably tangent-plane continuous (G^1). Phong tessellation [BA08] and (curved) PN triangles [VPBM01] give constructions for quadratic and cubic Bézier triangles, respectively, such that their control points are determined by vertex positions and vertex normals. Both techniques are local and suitable for hardware tessellation, and assure at least C^0 continuity across edges, and additionally G^1 continuity at vertices in the case of PN triangles.

This paper describes an extension of Phong tessellation and PN triangles to arbitrary polygons such that the resulting surfaces are S-patches with control points determined from vertex positions and normals. This allows for a uniform technique to process meshes with several face types. This technique can be implemented using hardware tessellation and is, unlike Phong tessellation and PN triangles, independent of the triangulation of the input polygons.

2. Preliminaries

Generalised barycentric coordinates provide a coordinate system in which any point on a planar polygon can be expressed as a linear combination of the polygon's vertices. Given cyclically ordered vertices v_i, $i = 1 \ldots n$ of a polygon P, and a point $p \in P$, generalised barycentric coordinate functions ϕ_i, $i = 1 \ldots n$ on P are defined by the following properties:

- Partition of unity: $\sum_{i=1}^{n} \phi_i(p) = 1$;
- Non-negativity $\phi_i(p) \geq 0, \forall i$;
- Linear reproduction: $\sum_{i=1}^{n} \phi_i(p) v_i = p$.

The coordinate functions linearly interpolate on the boundary of P and satisfy the Lagrange property: $\phi_i(v_j) = \delta_{ij}$, where δ_{ij} is the Kronecker delta. There exist various forms of generalised barycentric coordinates such as Wachspress coordinates [Wac75] and many other types [Flo15].

Phong tessellation [BA08] uses the unique barycentric coordinates on triangles and orthogonal projections to create a quadratic Bézier triangle. Consider a triangle in 3D with vertices $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, and corresponding normals $\mathbf{n}_1, \mathbf{n}_2, \mathbf{n}_3$. A point on the triangle can be expressed as

$$\mathbf{p}(u,v,w) = (u,v,w)(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3)^\intercal, \tag{1}$$

where u,v,w denote barycentric coordinates. Let $\pi_i(\mathbf{p})$ be the orthogonal projection of a point $\mathbf{p}$ onto the tangent plane at $\mathbf{v}_i$ given by $\mathbf{n}_i$. Phong tessellation [BA08] is defined as the barycentric combination of the orthogonally projected barycentric combinations:

$$\mathbf{p}^2(u,v,w) = (u,v,w)\begin{pmatrix}\pi_1(\mathbf{p}(u,v,w))\\ \pi_2(\mathbf{p}(u,v,w))\\ \pi_3(\mathbf{p}(u,v,w))\end{pmatrix}. \tag{2}$$

As proposed in [BA08], linearly interpolated normals are used.

A (curved) PN triangle [VPBM01] is a cubic Bézier triangle also constructed using barycentric coordinates and orthogonal projections, and is given in Bernstein-Bézier form

$$\mathbf{p}^3(u,v,w) = \sum_{\substack{i+j+k=3\\ i,j,k\geq 0}} \frac{3!}{i!j!k!}\mathbf{b}_{ijk} u^i v^j w^k, \tag{3}$$

where

$$\begin{aligned}
&\mathbf{b}_{300} = \mathbf{v}_1,\ \mathbf{b}_{030} = \mathbf{v}_2,\ \mathbf{b}_{003} = \mathbf{v}_3,\\
&\mathbf{b}_{210} = \frac{2}{3}\mathbf{v}_1 + \frac{1}{3}\pi_1(\mathbf{v}_2),\ \mathbf{b}_{120} = \frac{2}{3}\mathbf{v}_2 + \frac{1}{3}\pi_2(\mathbf{v}_1),\\
&\mathbf{b}_{201} = \frac{2}{3}\mathbf{v}_1 + \frac{1}{3}\pi_1(\mathbf{v}_3),\ \mathbf{b}_{102} = \frac{2}{3}\mathbf{v}_3 + \frac{1}{3}\pi_3(\mathbf{v}_1),\\
&\mathbf{b}_{021} = \frac{2}{3}\mathbf{v}_2 + \frac{1}{3}\pi_2(\mathbf{v}_3),\ \mathbf{b}_{012} = \frac{2}{3}\mathbf{v}_3 + \frac{1}{3}\pi_3(\mathbf{v}_2),\\
&\quad\mathbf{e} = (\mathbf{b}_{210} + \mathbf{b}_{120} + \mathbf{b}_{201} + \mathbf{b}_{102} + \mathbf{b}_{021} + \mathbf{b}_{012})/6,\\
&\quad\mathbf{c} = (\mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3)/3, \text{ and } \mathbf{b}_{111} = \mathbf{e} + (\mathbf{e} - \mathbf{c})/2.
\end{aligned}$$

Since cubics are capable of representing inflection points, it is a good choice to use a normal field which supports this. To this end, a quadratically varying normal field was proposed in [VPBM01]:

$$\mathbf{n}^2(u,v,w) = u^2\mathbf{n}_{200} + v^2\mathbf{n}_{020} + w^2\mathbf{n}_{002} + uv\mathbf{n}_{110} + uw\mathbf{n}_{101} + wv\mathbf{n}_{011}$$

with

$$\begin{aligned}
&\mathbf{n}_{200} = \mathbf{n}_1, \mathbf{n}_{020} = \mathbf{n}_2, \mathbf{n}_{002} = \mathbf{n}_3,\\
&\mathbf{n}_{110} = \frac{\mathbf{h}_{110}}{||\mathbf{h}_{110}||}, \mathbf{h}_{110} = \mathbf{n}_1 + \mathbf{n}_2 - t_{12}(\mathbf{v}_2 - \mathbf{v}_1),\\
&\mathbf{n}_{101} = \frac{\mathbf{h}_{101}}{||\mathbf{h}_{101}||}, \mathbf{h}_{101} = \mathbf{n}_1 + \mathbf{n}_3 - t_{13}(\mathbf{v}_3 - \mathbf{v}_1),\\
&\mathbf{n}_{011} = \frac{\mathbf{h}_{011}}{||\mathbf{h}_{011}||}, \mathbf{h}_{011} = \mathbf{n}_2 + \mathbf{n}_3 - t_{23}(\mathbf{v}_3 - \mathbf{v}_2),
\end{aligned} \tag{4}$$

where

$$t_{ij} = 2\frac{(\mathbf{v}_j - \mathbf{v}_i)\cdot(\mathbf{n}_i + \mathbf{n}_j)}{(\mathbf{v}_j - \mathbf{v}_i)\cdot(\mathbf{v}_j - \mathbf{v}_i)}.$$

We now extend these constructions to arbitrary polygons.

3. Extending Phong tessellation and PN triangles

To parametrise an arbitrary and possibly non-planar polygon $\mathbf{P}$ of valency n with vertices $\mathbf{v}_1 \dots, \mathbf{v}_n$, we consider a regular polygon P of the same valency. Other parametrisation options are discussed in Section 6. This planar domain allows for any point $p \in P$ to be expressed using generalized barycentric coordinates $\boldsymbol{\phi}$ on P. Having obtained these coordinates, they are used to parametrise $\mathbf{P}$.

3.1. Extended Phong tessellation

Given some generalised barycentric coordinates $\boldsymbol{\phi} = (\phi_1 \dots \phi_n)$ with respect to P corresponding to $\mathbf{P}$, we have that

$$\mathbf{p}(\boldsymbol{\phi}) = (\phi_1, \phi_2, \dots, \phi_n)(\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n)^\intercal. \tag{5}$$

Then, extended Phong tessellation, cf. (2), is the barycentric combination of the projections of this point onto the tangent planes defined by the normals at the vertices of $\mathbf{P}$:

$$\mathbf{p}^2(\boldsymbol{\phi}) = (\phi_1, \phi_2, \dots, \phi_n)\left(\pi_1(\mathbf{p}(\boldsymbol{\phi})), \dots, \pi_n(\mathbf{p}(\boldsymbol{\phi}))\right)^\intercal. \tag{6}$$

3.2. PN polygons

In the case of PN triangles, observe that any Bézier cubic can be written as a linear combination of Bézier quadratics. It then follows that we can rewrite the triangular construction in (3) as

$$\mathbf{p}^3(u,v,w) = (u,v,w)\begin{pmatrix}(u,v,w)\begin{pmatrix}\pi_1(\mathbf{p}(u,v,w))\\ \mathbf{p}(u,v,w)\\ \mathbf{p}(u,v,w)\end{pmatrix}\\ (u,v,w)\begin{pmatrix}\mathbf{p}(u,v,w)\\ \pi_2(\mathbf{p}(u,v,w))\\ \mathbf{p}(u,v,w)\end{pmatrix}\\ (u,v,w)\begin{pmatrix}\mathbf{p}(u,v,w)\\ \mathbf{p}(u,v,w)\\ \pi_3(\mathbf{p}(u,v,w))\end{pmatrix}\end{pmatrix} + \boldsymbol{\theta}_3$$

with

$$\boldsymbol{\theta}_3 = 6uvw\left(\frac{-\mathbf{c}_{123} + \sum_{q\in\{1,2,3\}}\pi_q(\mathbf{c}_{123})}{4}\right),$$

where $\mathbf{c}_{ijk} = (\mathbf{v}_i + \mathbf{v}_j + \mathbf{v}_k)/3$. This extra term $\boldsymbol{\theta}_3$ is needed to achieve reproduction of quadratics [Far86] and is derived from the central coefficient in the original PN triangle definition. Based on this, we extend this construction to arbitrary polygons by

$$\mathbf{p}^3(\boldsymbol{\phi}) = \boldsymbol{\phi}\left(\boldsymbol{\phi}\begin{pmatrix}\pi_1(\mathbf{p}(\boldsymbol{\phi}))\\ \cdots\\ \mathbf{p}(\boldsymbol{\phi})\end{pmatrix}, \dots, \boldsymbol{\phi}\begin{pmatrix}\mathbf{p}(\boldsymbol{\phi})\\ \cdots\\ \pi_n(\mathbf{p}(\boldsymbol{\phi}))\end{pmatrix}\right)^\intercal + \boldsymbol{\theta}_n \tag{7}$$

with

$$\boldsymbol{\theta}_n = 6\sum_{i=1}^{n-2}\sum_{j=i+1}^{n-1}\sum_{k=j+1}^{n}\phi_i\phi_j\phi_k\left(\frac{-\mathbf{c}_{ijk} + \sum_{q\in\{i,j,k\}}\pi_q(\mathbf{c}_{ijk})}{4}\right). \tag{8}$$

In (6) and (7) we construct an S-patch of degree two and three, respectively. For triangles ($n = 3$), both extended Phong tessellation and PN polygons yield the exact same result as the known triangular forms. If bilinear coordinates (or equivalently Wachspress coordinates on rectangles) are used in the case of quads ($n = 4$), Phong tessellation for quads is also reproduced [Bou10]. The correction term $\boldsymbol{\theta}_n$ is derived similarly to the triangular case ($\boldsymbol{\theta}_3$) by degree elevating a quadratic S-patch [SS15] so as to reproduce quadratics in $\boldsymbol{\phi}$ when possible.

Since our extended constructions yield S-patches, they can also be given in Bernstein-Bézier form using the multi-index notation $\alpha = (\alpha_1, \ldots, \alpha_n)$ and degree d defined in [LD89]:

$$\mathbf{p}^d(\boldsymbol{\phi}) = \sum_{|\alpha|=d} \mathbf{b}_\alpha B^d_\alpha(\boldsymbol{\phi}) \quad \text{with} \quad B^d_\alpha(\boldsymbol{\phi}) = \frac{d!}{\alpha!}\phi_\alpha, \tag{9}$$

where the control points in the quadratic case (Phong tessellation) are defined as

$$\mathbf{b}^2_{\alpha_i,\alpha_j} = \frac{\pi_j(\mathbf{v}_i) + \pi_i(\mathbf{v}_j)}{2}, \ \mathbf{b}^2_{\alpha_i} = \mathbf{v}_i,$$

and in the cubic case (PN polygons) as

$$\mathbf{b}^3_{\alpha_i,\alpha_j,\alpha_k} = \frac{3\mathbf{c}_{ijk} + \sum_{q\in i,j,k} 3\pi_q(\mathbf{c}_{ijk})}{12},$$

$$\mathbf{b}^3_{\alpha_i,\alpha_j} = \frac{2\mathbf{v}_i + \pi_i(\mathbf{v}_j)}{3} \quad \text{when} \quad \alpha_i > \alpha_j, \quad \text{and} \quad \mathbf{b}^3_{\alpha_i} = \mathbf{v}_i.$$

3.3. Normal interpolation

As in the triangular setting, the extended cubic form should be accompanied by a quadratic normal field. The linear normal field is simply $\mathbf{n}(\boldsymbol{\phi}) = (\phi_1, \ldots, \phi_n)(\mathbf{n}_1, \ldots, \mathbf{n}_n)^\intercal$. Defining the reflection of a normal as

$$\boldsymbol{\tau}_i(\mathbf{n}, \mathbf{v}) = \begin{cases} \mathbf{n}, & \text{if } \mathbf{v} = \mathbf{v}_i \\ \mathbf{n} - 2\frac{\mathbf{n}\cdot(\mathbf{v}-\mathbf{v}_i)}{(\mathbf{v}-\mathbf{v}_i)\cdot(\mathbf{v}-\mathbf{v}_i)}(\mathbf{v}-\mathbf{v}_i), & \text{otherwise,} \end{cases}$$

the quadratic normal field is then given by

$$\mathbf{n}^2(\boldsymbol{\phi}) = \boldsymbol{\phi}\left(\boldsymbol{\phi}\begin{pmatrix}\boldsymbol{\tau}_1(\mathbf{n}_1,\mathbf{v}_1)\\ \cdots \\ \boldsymbol{\tau}_1(\mathbf{n}_n,\mathbf{v}_n)\end{pmatrix}, \ldots, \boldsymbol{\phi}\begin{pmatrix}\boldsymbol{\tau}_n(\mathbf{n}_1,\mathbf{v}_1)\\ \cdots \\ \boldsymbol{\tau}_n(\mathbf{n}_n,\mathbf{v}_n)\end{pmatrix}\right)^\intercal. \tag{10}$$

This does not exactly reproduce the triangular version of [VPBM01] in the case $n = 3$ since the intermediate normalisation of the edge coefficients, see (4), is lost in this form. Generally, the quadratic normal fields look the same and differ only slightly as can be seen in Figure 2.

4. Implementation

The implementation of the extended definitions was done using the tessellation stage in the OpenGL 4.x pipeline. To render an arbitrary (curved) polygon of valency n, we divide the polygon in $n-2$ triangular patches. These patches correspond to a triangulation of a regular n-gon. For simplicity, a fanning triangulation was chosen; see Figure 3 and Section 6.

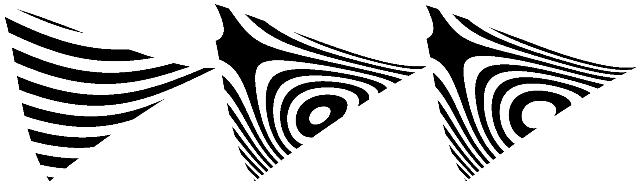

Figure 2: *Reflection lines on a cubic triangle for various normal interpolation methods. Left to right: linear interpolation, quadratic interpolation for PN triangles, and the new quadratic interpolation for arbitrary PN polygons.*

By use of vertex indexing, barycentric coordinates u, v, w on triangles, and indexing of sub-triangles, a position $p_\triangle$ on a sub-triangle $\triangle$ with vertices v_i, v_j and v_k of P is found. Then generalized barycentric coordinates $\boldsymbol{\phi}$ of $p_\triangle$ on P are calculated corresponding to u, v, w on $\triangle$ produced in the tessellation stage. The coordinates can then be used to obtain the associated position on $\mathbf{P}$ or to evaluate (6) or (7). This gives a point on the surface of the Phong tessellation or PN polygon on $\mathbf{P}$, respectively.

Care must be taken on the boundary of P and at vertices. A simple criterion can be evaluated on the generated barycentric coordinates u, v, w corresponding to vertex indices i, j, k of some sub-triangle of P in order to determine whether these describe a point on an edge or vertex. Alternatively, control points of the S-patch can be pre-computed by following the form described in (9).

5. Results and performance

Immediate improvements over the original triangular methods can be seen in the case of polygons parametrised using a regular parametrisation domain. In Figure 3, two different triangulations are imposed on an octagon. This shows that the extended techniques are indifferent to the underlying triangulation as both resulting surfaces are identical. The same can be seen in the context of a model containing different types of polygons, as shown in Figure 1.

Regarding the performance of the extended algorithms, they are slower than the original triangular methods. Explicitly calculating control points drastically increases performance as can be seen in Figure 4. The measured FPS rates were obtained on a PC with an Intel Core i3-2310M, 4GB RAM and NVIDIA GeForce GT 520MX 1GB graphics card running OpenGL 4.1 with NVIDIA drivers for Linux.

6. Discussion, limitations and future work

As mentioned above, the resulting surfaces are independent of the chosen triangulation of the polygons as the triangulation step is only used to facilitate the calculation of generalised barycentric coordinates in the parametrisation domain. This triangulation step is currently unavoidable due to GPU limitations. Its purpose is to convert regular barycentric coordinates on a sub-triangle to generalised barycentric coordinates with respect to the whole polygon. Although this slows down the extended methods compared to the original versions (cf. Figure 4), it provides much smoother results on polygonal models (see Figure 1).

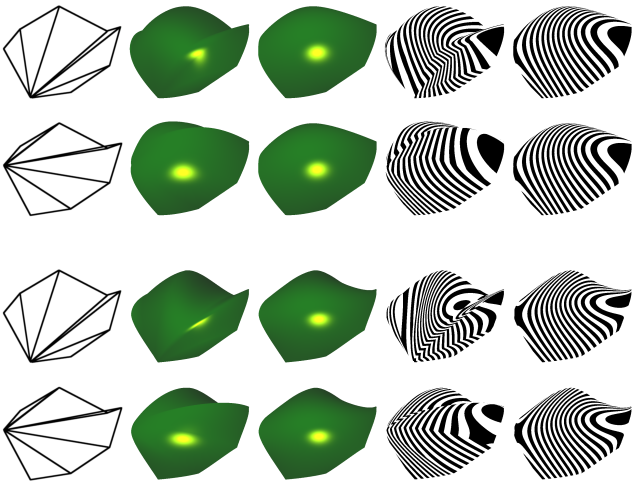

Figure 3: *Comparison: Two different triangulations of a non-planar octagon. Top two rows: Phong tesselation and extended Phong tessellation, reflection lines of linear normal fields on Phong tessellation and extended Phong tessellation. Bottom two rows: PN triangles and PN polygons, reflection lines of the new quadratic normal fields, see* (10)*, on PN triangles and PN polygons. Note that extended Phong tessellation and PN polygons are independent of the triangulation of the input octagon.*

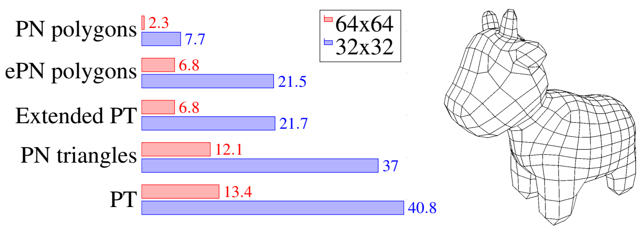

Figure 4: *Performance comparison in FPS per method: rendering a model (right) containing 56 triangles, 634 quads, 40 pentagons, and 4 hexagons at two tessellation depths. ePN polygons refers to the explicit calculation of control points, see* (9)*, and PT refers to Phong tessellation.*

Regarding our choice of parametrisation, a regular polygon can be viewed as the simplest and most efficient choice. Additionally, the use of regular polygons allows us to utilise Wachspress coordinates, which are only defined on convex polygons, but are arguably the most efficient (a closed-form rational formula is available) generalisation of barycentric coordinates to polygons.

Moreover, our tests suggest that this choice allows one to create very complex surfaces from polygonal boundaries which can self intersect and twist without introducing other artifacts when compared to other coordinate types.

The use of other types of parametrisation domains has been investigated, namely a projection of the polygon to the best fitting plane and a chord length parametrisation variant. These methods pose restrictions on the type of barycentric coordinates since the domain may become non-convex. In addition to this, the use of these techniques requires a uniquely constructed parametrisation domain for each polygon in a mesh, thus slowing down the method.

In theory, there is no maximum polygonal valency, but increasing the number of vertices adds computational complexity to the calculation of generalised barycentric coordinates and S-patches. In the current implementation an individual shader is implemented for valencies 3 to 8 because of constraints on input vertex numbers. However, future developments in graphics drivers could lift this restriction. Still a more efficient means of (pre-)computing generalised barycentric coordinates is sought, e.g., by means of textures, since it could save valuable computation time.

7. Conclusion

The generalisations of Phong tessellation and curved PN triangles to arbitrary polygons provide simple ways of creating quadratic and cubic S-patches in the context of polygonal models. The generalised quadratic normal fields do not stray far from the original triangular version when applied to triangles and are equally effective on arbitrary polygons. The increased complexity of the computation of the control points or evaluation of the patches makes it slower than the original triangular methods but this does not undo the power of the techniques in that they are independent of the triangulation of the polygons and can create very complex surfaces. The continuity properties of the original methods are transferred to the polygonal extensions and the new methods can also be implemented into the rendering pipeline, requiring only vertices and normals to be sent to the GPU.

Acknowledgements This paper is in part based on the first author's MSc thesis at the University of Groningen, and more details can be found therein. We would like to thank Pieter Barendrecht for his help with our OpenGL implementation.

References

[BA08] BOUBEKEUR T., ALEXA M.: Phong tessellation. *ACM Trans. Graph. 27*, 5 (Dec. 2008), 141:1–141:5.

[Bou10] BOUBEKEUR T.: *GPU Pro - Advanced Rendering Techniques*. ShaderX Book Series. A.K. Peters, 2010, ch. As-Simple-As Possible Tessellation for Interactive Applications.

[Far86] FARIN G.: Triangular Bernstein-Bézier patches. *Computer Aided Geometric Design 3*, 2 (1986), 83–127.

[Flo15] FLOATER M. S.: Generalized barycentric coordinates and applications. *Acta Numerica 24* (2015), 161–214.

[LBS08] LANGER T., BELYAEV A., SEIDEL H.-P.: Mean value Bézier maps. In *Advances in Geometric Modeling and Processing*. Springer, 2008, pp. 231–243.

[LD89] LOOP C. T., DEROSE T. D.: A multisided generalization of Bézier surfaces. *ACM Transactions on Graphics (TOG) 8*, 3 (1989), 204–234.

[SS15] SMITH J., SCHAEFER S.: Selective degree elevation for multi-sided Bézier patches. *Computer Graphics Forum 34*, 2 (2015), 609–615.

[VPBM01] VLACHOS A., PETERS J., BOYD C., MITCHELL J. L.: Curved PN triangles. In *Proceedings of the 2001 symposium on Interactive 3D graphics* (2001), ACM, pp. 159–166.

[Wac75] WACHSPRESS E.: *A Rational Finite Element Basis*. Academic Press rapid manuscript reproductions. Academic Press, 1975.