Universidade Federal da Bahia

IME – PGCOMP

MATE06 - TÓPICOS EM COMPUTAÇÃO VISUAL I

Seminário

Professor: Antônio L. Apolinário Jr.

Aluno: Fernando Ferraz Ribeiro

Conceitos:

Tessellation

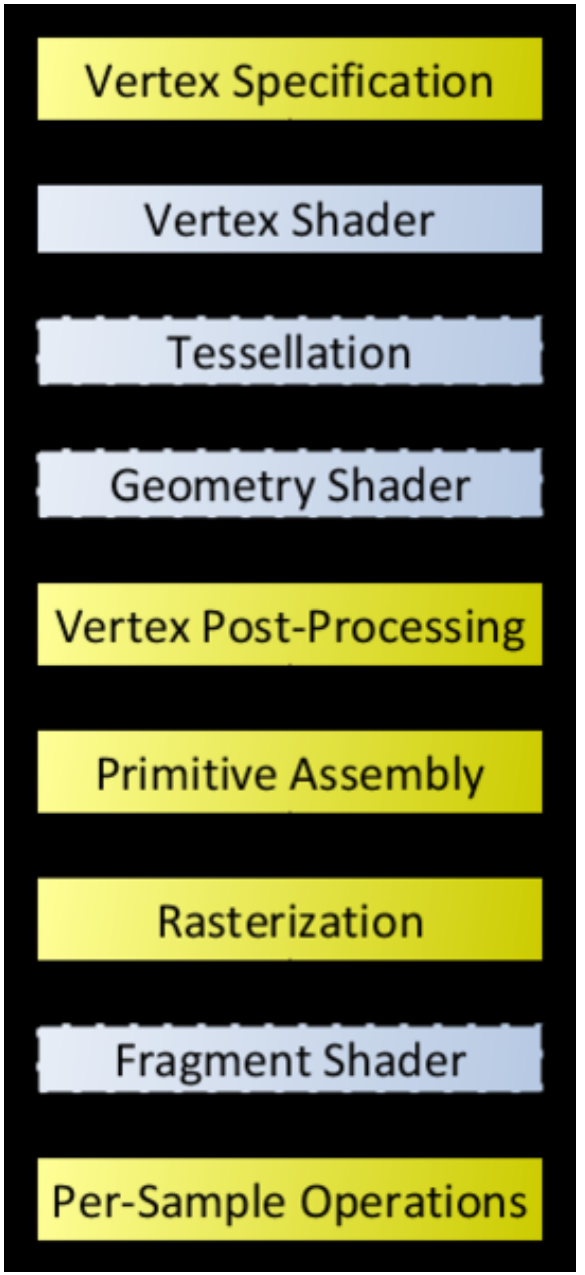
Bezier

Spline

Nurbs

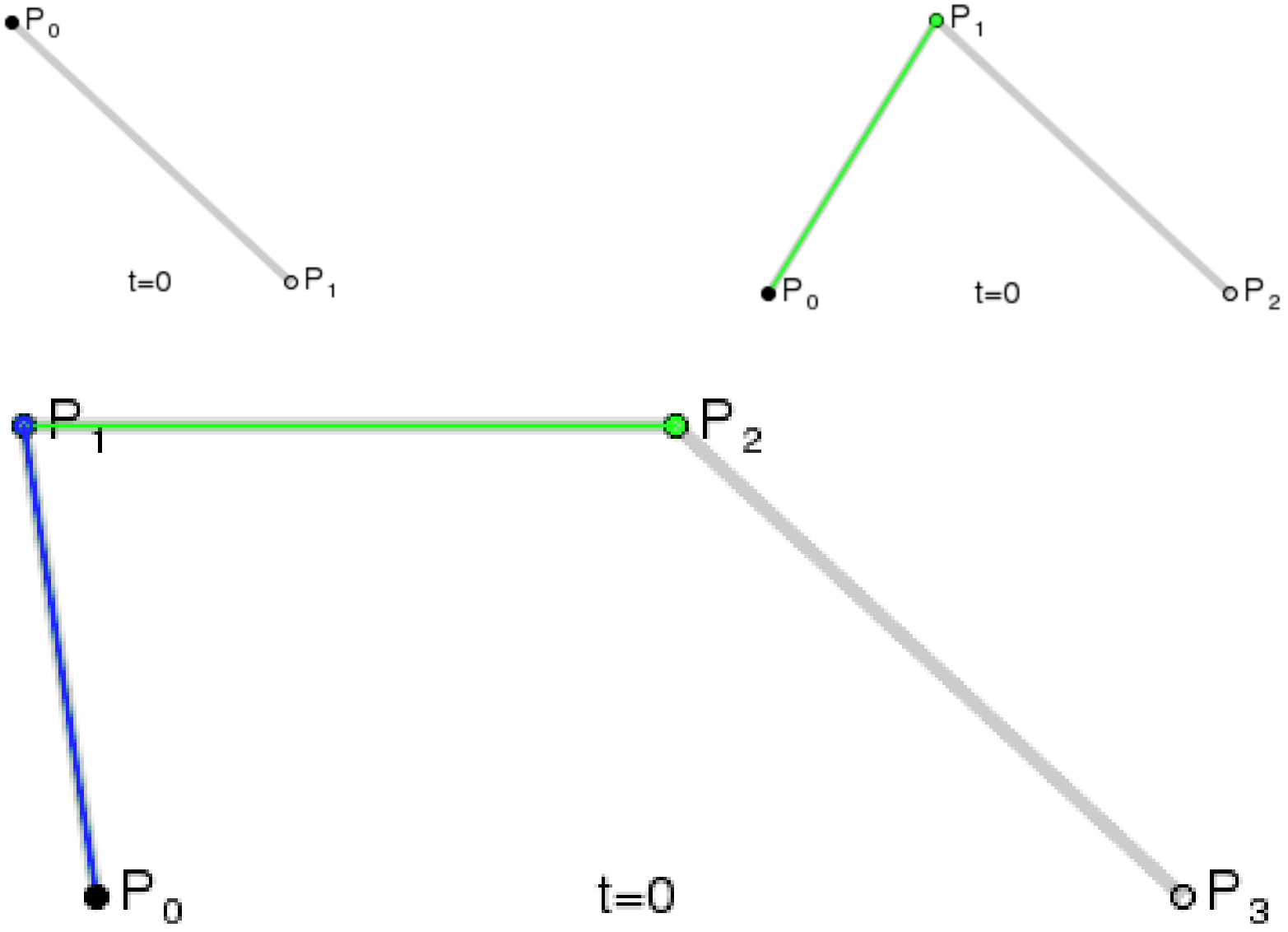
Tessellation

Tessellation é a etapa do processamento dos vértices do pipeline gráfico Opengl que transforma primitivas do modelo em primitivas menores.



https://www.khronos.org/opengl/wiki/File:RenderingPipeline.png

Bezier



https://pt.wikipedia.org/w/index.php?title=Especial:Entrar&returnto=Curva+de+B%C3%A9zier

ARTIGO 1:

Phong Tessellation

Tamy Boubekeur Marc Alexa
TU Berlin

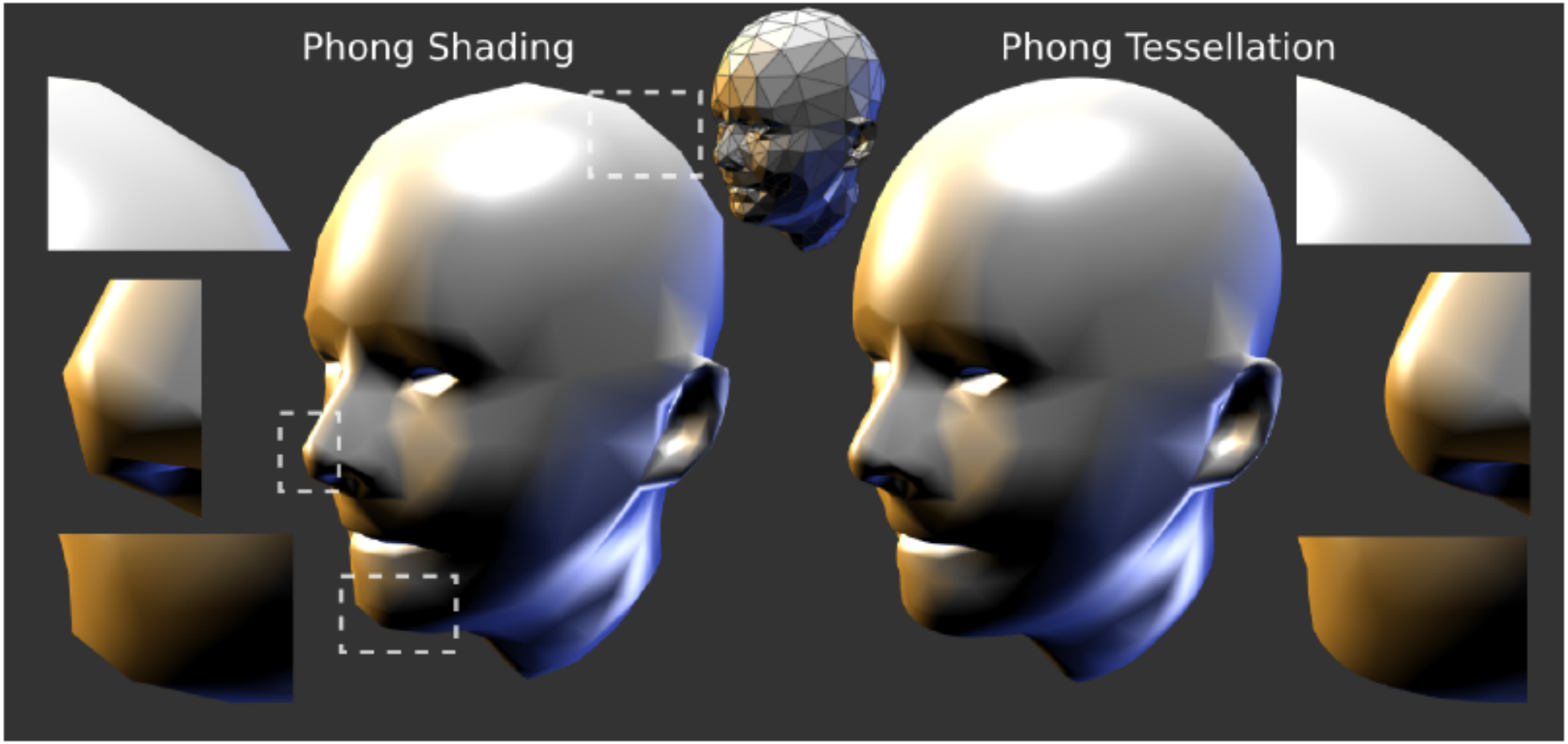


Figure 1: *Phong Tessellation completes Phong Shading.*

Gouraud - Phong

Gouraud $$\mathbf{p}(u,v) = (u,v,w)(\mathbf{p}_i, \mathbf{p}_j, \mathbf{p}_k)^{\mathsf{T}} \qquad (1)$$

Phong $$\mathbf{n}'(u,v) = (u,v,w)(\mathbf{n}_i, \mathbf{n}_j, \mathbf{n}_k)^{\mathsf{T}}, \quad \mathbf{n}(u,v) = \mathbf{n}'/\|\mathbf{n}'\| \qquad (2)$$

Phong Tessellation

1. Calcular a interpolação linear.

2. Projetar o ponto interpolado nos 3 planos tangentes a cada vértice que contenham as normais dos vértices.

3. Calcular a interpolação linear em relação ao triângulo gerado pelos pontos projetados.

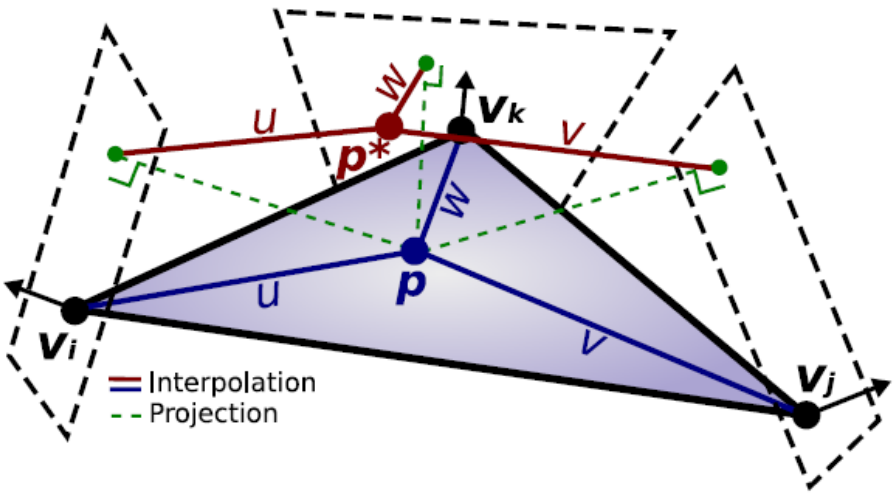


Figure 3: *Phong Tessellation principle. Instead of interpolating normals as in Phong Shading, we interpolate projection onto vertices tangent plane to define a curve geometry for each triangle.*

Phong Tessellation

$$\pi_i(\mathbf{q}) = \mathbf{q} - \left((\mathbf{q} - \mathbf{p}_i)^{\mathsf{T}} \mathbf{n}_i\right) \mathbf{n}_i$$

$$\mathbf{p}^*(u,v) = (u,v,w) \begin{pmatrix} \pi_i(\mathbf{p}(u,v)) \\ \pi_j(\mathbf{p}(u,v)) \\ \pi_k(\mathbf{p}(u,v)) \end{pmatrix}$$

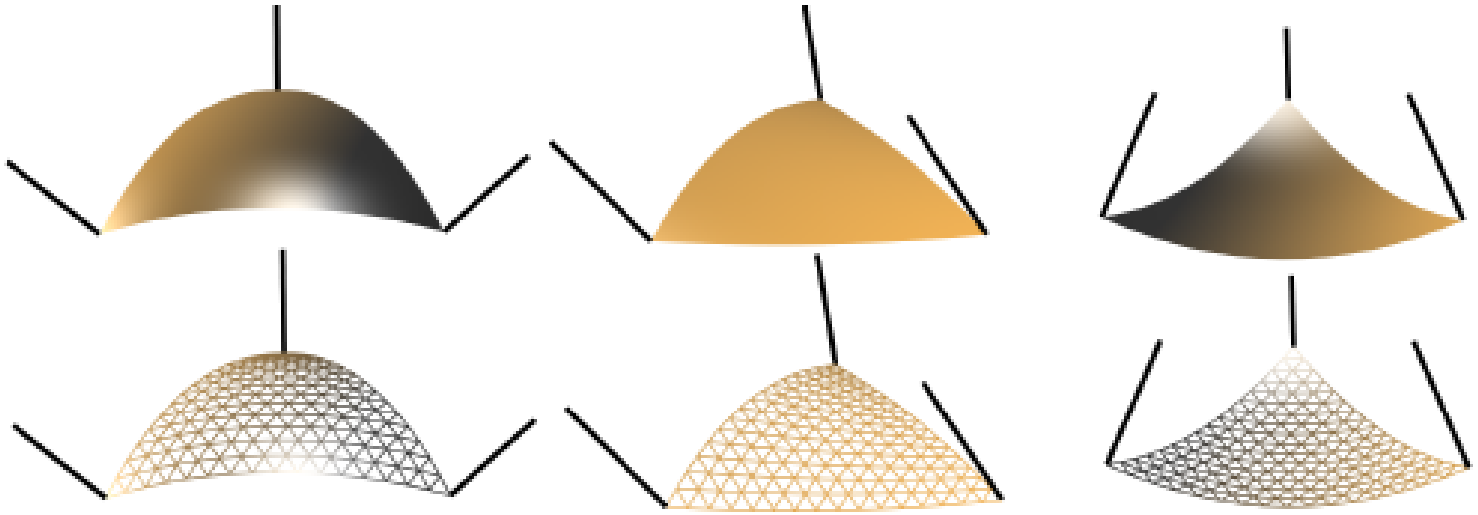


Figure 5: *Convex, saddle and concave normals configuration.*

Fator Alfa

$$\mathbf{p}^*{}_{\alpha}(u,v) = (1-\alpha)\mathbf{p}(u,v) + \alpha(u,v,w)\begin{pmatrix}\pi_i(\mathbf{p}(u,v))\\ \pi_j(\mathbf{p}(u,v))\\ \pi_k(\mathbf{p}(u,v))\end{pmatrix} \qquad (4)$$

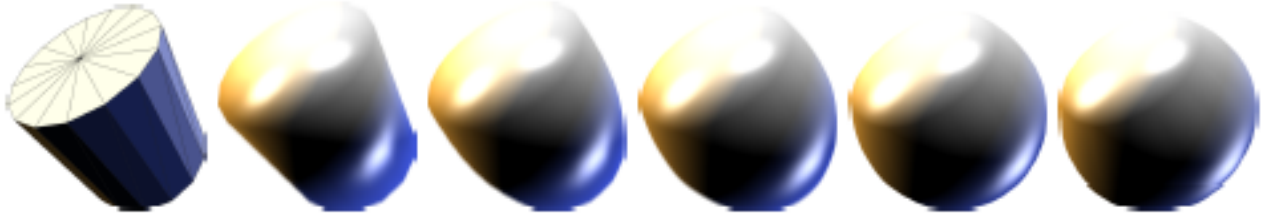


Figure 4: *Coarse mesh, followed by various Phong Tessellations with* α *equal to* 0, 1/4, 1/2, 3/4 *and* 1.

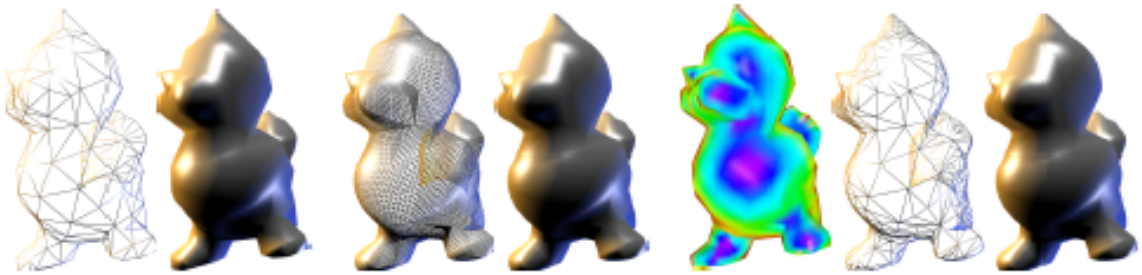
Propriedades

$$\mathbf{p}^*(u,v) = u^2\mathbf{p}_i + v^2\mathbf{p}_j + w^2\mathbf{p}_k + uv\left(\pi_i(\mathbf{p}_j) + \pi_j(\mathbf{p}_i)\right) + vw\left(\pi_j(\mathbf{p}_k) + \pi_k(\mathbf{p}_j)\right) + wu\left(\pi_k(\mathbf{p}_i) + \pi_i(\mathbf{p}_k)\right)$$

Refinamento Adaptativo

$$d_i = \left(1 - \left\| \mathbf{n}_i^T \frac{\mathbf{c} - \mathbf{p}_i}{\|\mathbf{c} - \mathbf{p}_i\|} \right\|\right) m$$

Comparações



(a) Input (b) Uniform (c) Adaptive View-Dependent

Figure 7: *Adaptive view-dependent Phong Tessellation for real-time geometric upsampling on silhouettes and contours only.*

Figure 9: *Comparison: Modified Butterfly subdivision (middle-left), PN Triangles (middle-right) and Phong tessellation (right).*

ARTIGO 2:

Efficient GPU Rendering of Subdivision Surfaces using Adaptive Quadtrees

Wade Brainerd* Activision

Tim Foley* NVIDIA

Manuel Kraemer NVIDIA

Henry Moreton NVIDIA

Matthias Nießner Stanford University

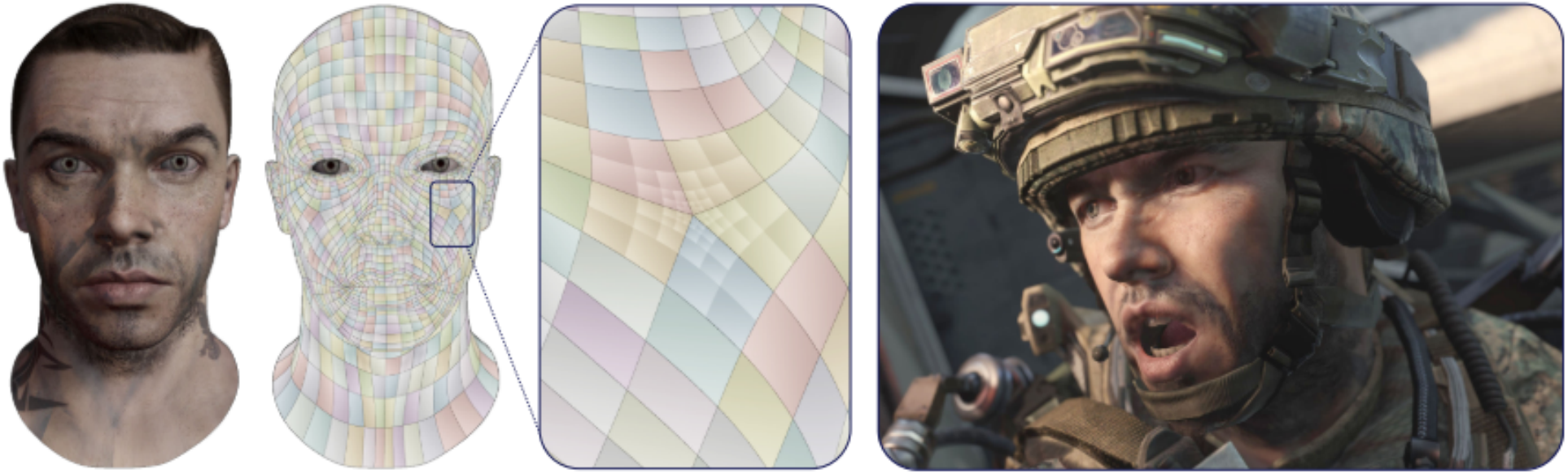
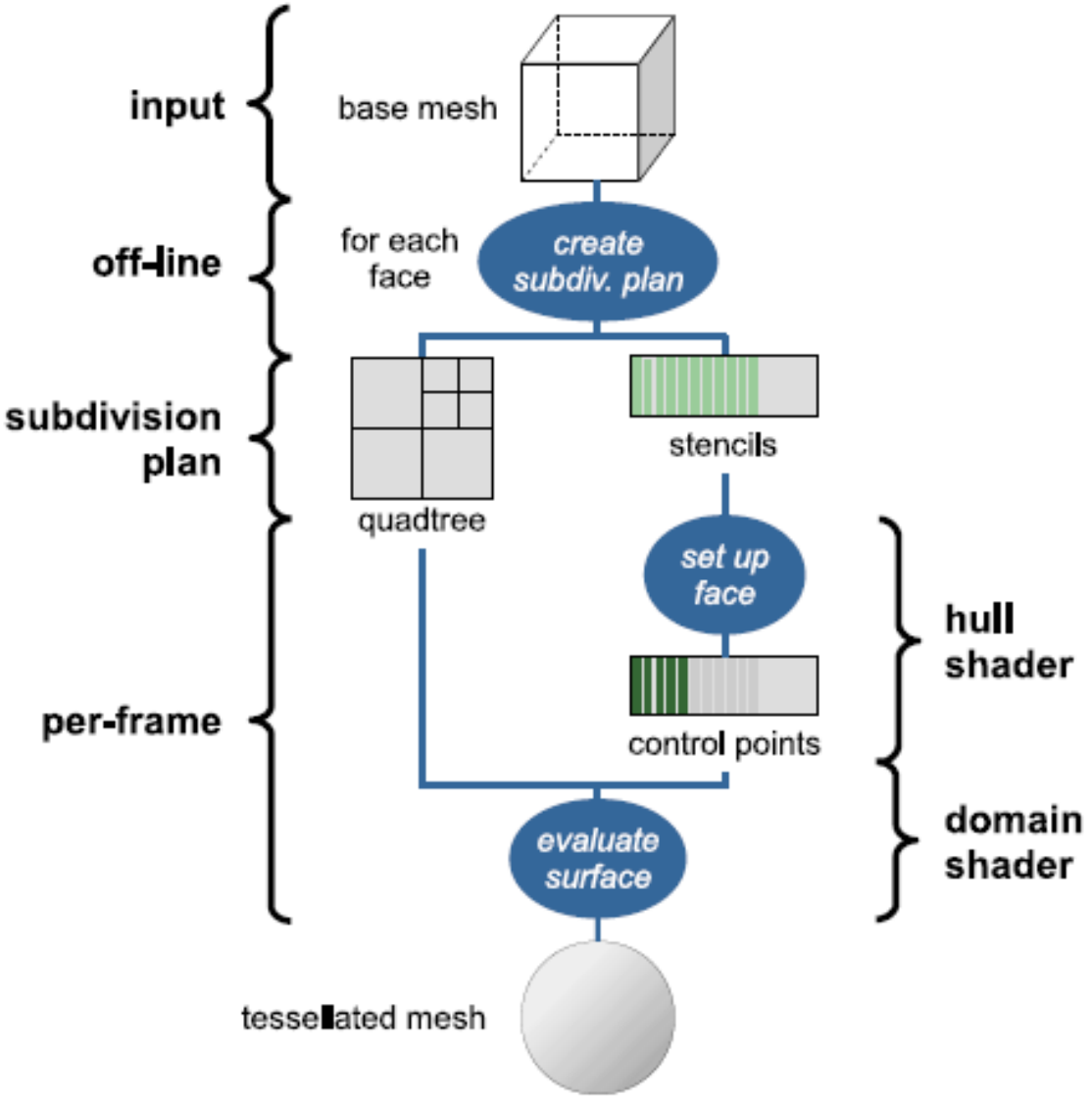


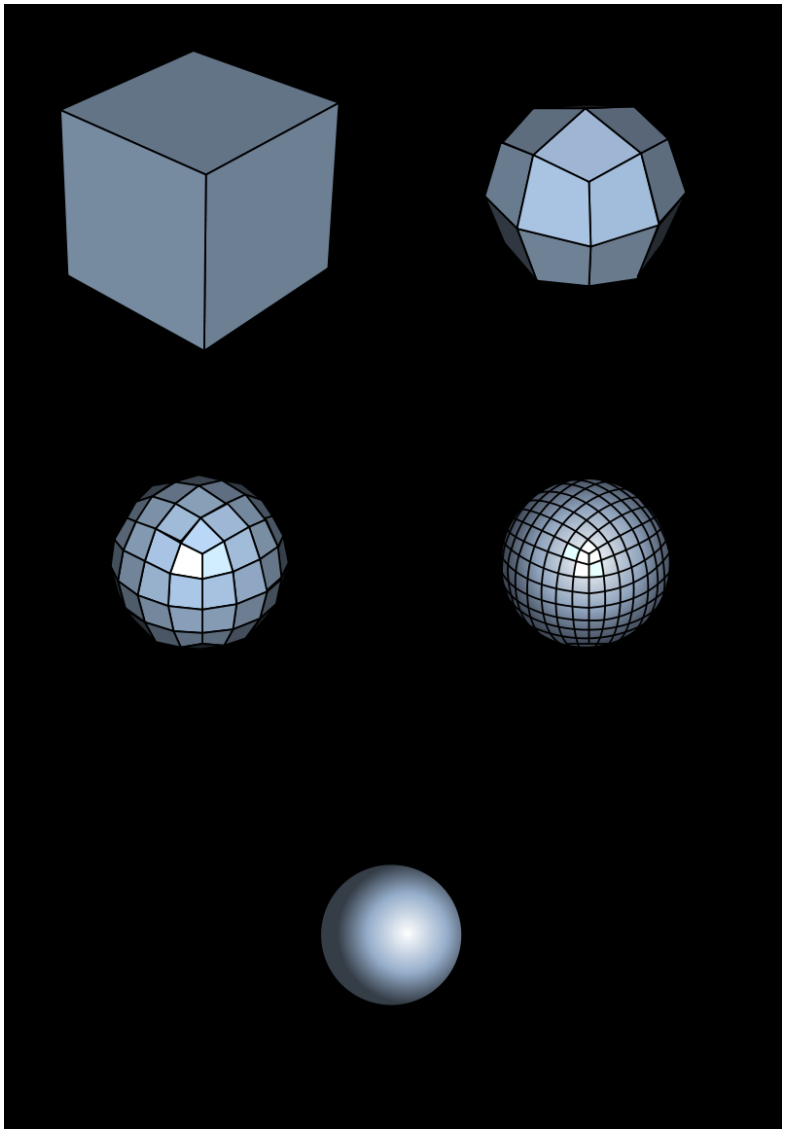
Figure 1: *In our method, a subdivision surface model (left) is rendered in a single pass, without a separate subdivision step. Each quad face is submitted as a single tessellated primitive; a per-face adaptive quadtree is used to map tessellated vertices to the appropriate subdivided face (middle). Our approach makes tessellated subdivision surfaces easy to integrate into modern video game rendering (right).*

Visão Geral


input
off-line
subdivision plan
per-frame
base mesh
for each face
create subdiv. plan
quadtree
stencils
set up face
control points
evaluate surface
tessellated mesh
hull shader
domain shader

Entrada de dados

- Catmull-Clark base mesh
  - Topologia
  - Tags de características
  - Vértices
- Atualizações dinâmicas de topologia



https://en.wikipedia.org/wiki/Catmull%E2%80%93Clark_subdivision_surface#/media/File:Catmull-Clark_subdivision_of_a_cube.svg

Plano de Subdivisão

- Uma estrutura de dados que representa uma hierarquia de subdivisões para uma face até um nível de detalhe máximo definido.
  - Uma *quadtree* das subdivisões hierárquicas de cada face
  - Uma lista ordenada de *Stencils*.
    - Cada *Stencil* representa uma soma balanceada dos vértices das faces na vizinhança “1-ring” do vértice

Plano de Subdivisão

- Catmull-Clark base mesh
  - Topologia
  - Tags de características
  - Vértices
- Atualizações dinâmicas de topologia

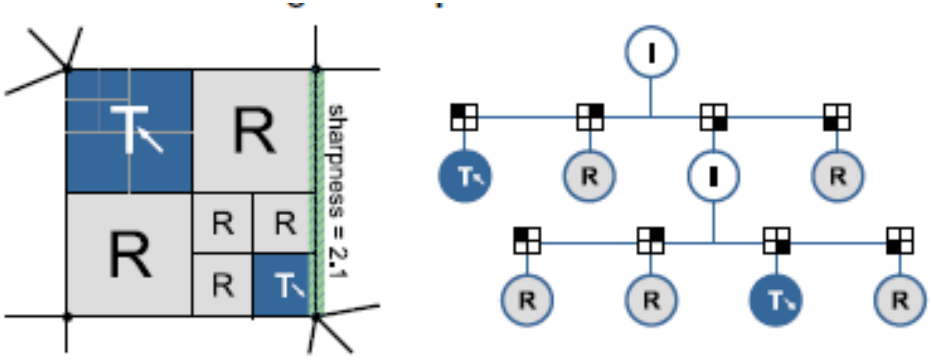


Figure 5: *Quadtree for a face with two extraordinary vertices, and one semi-sharp edge. The quadtree comprises internal (I), regular (R), and terminal nodes (T). Our terminal nodes do not support direct semi-sharp crease evaluation, so the bottom-right quadrant is recursively subdivided until the sharp feature is eliminated.*

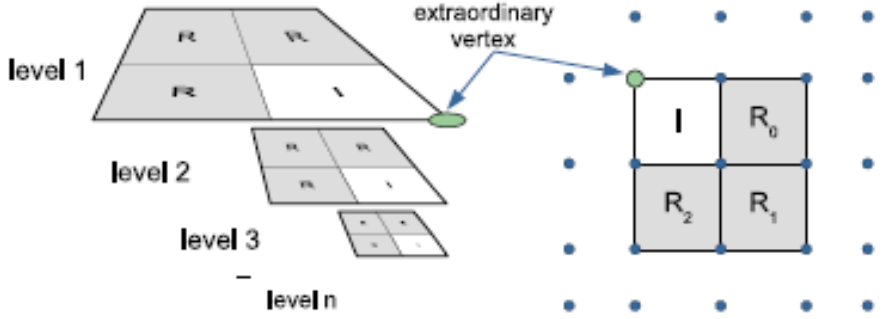
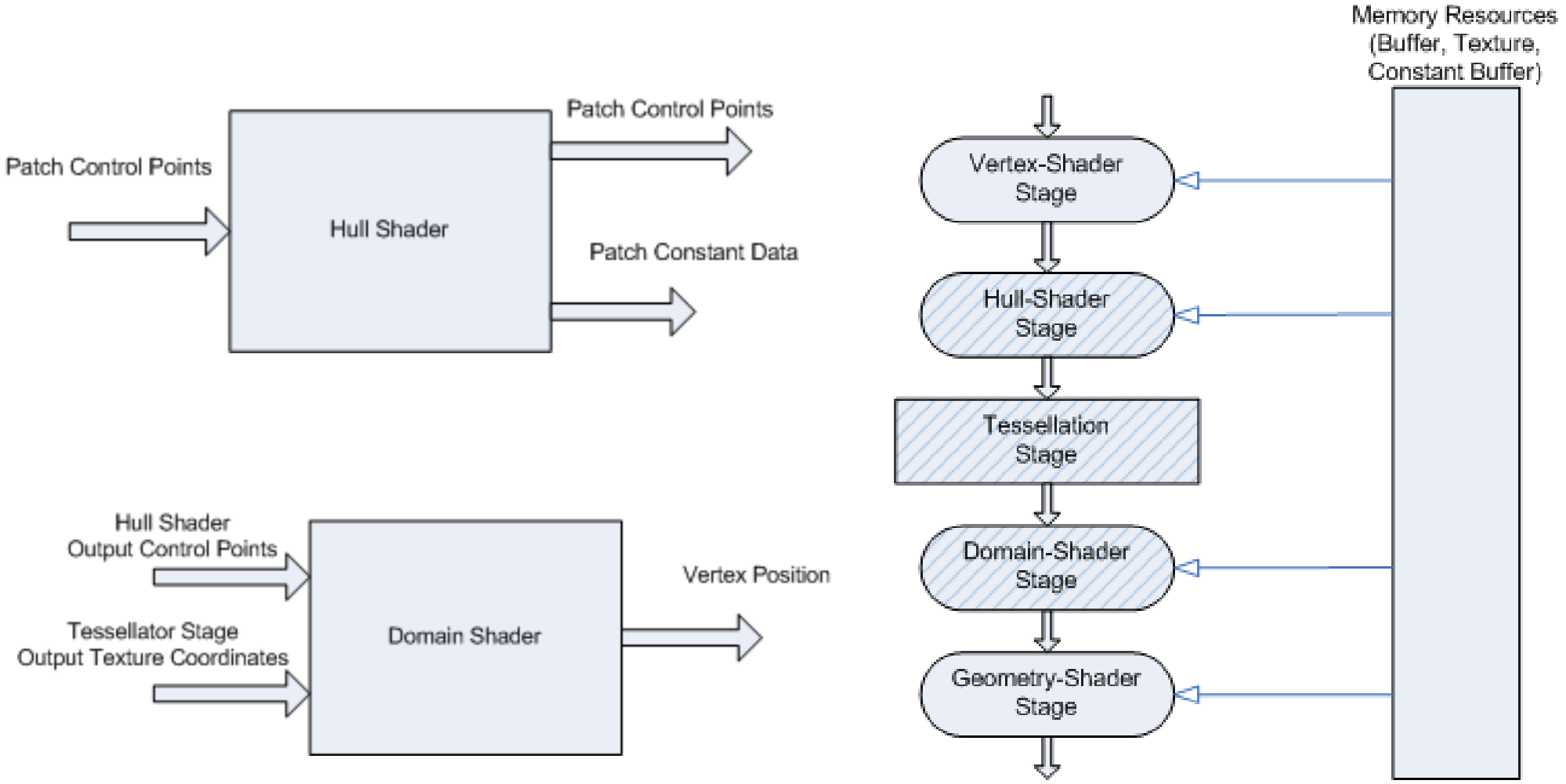


Figure 6: *A terminal node captures the repeating pattern of adaptive subdivision at an extraordinary vertex (left). For each subdivision level, the node references 24 control points shared by three regular sub-domains (right).*

Face Setup - Subdivision



https://docs.microsoft.com/en-us/windows/desktop/direct3d11/direct3d-11-advanced-stages-tessellation

Vértice extraordinário

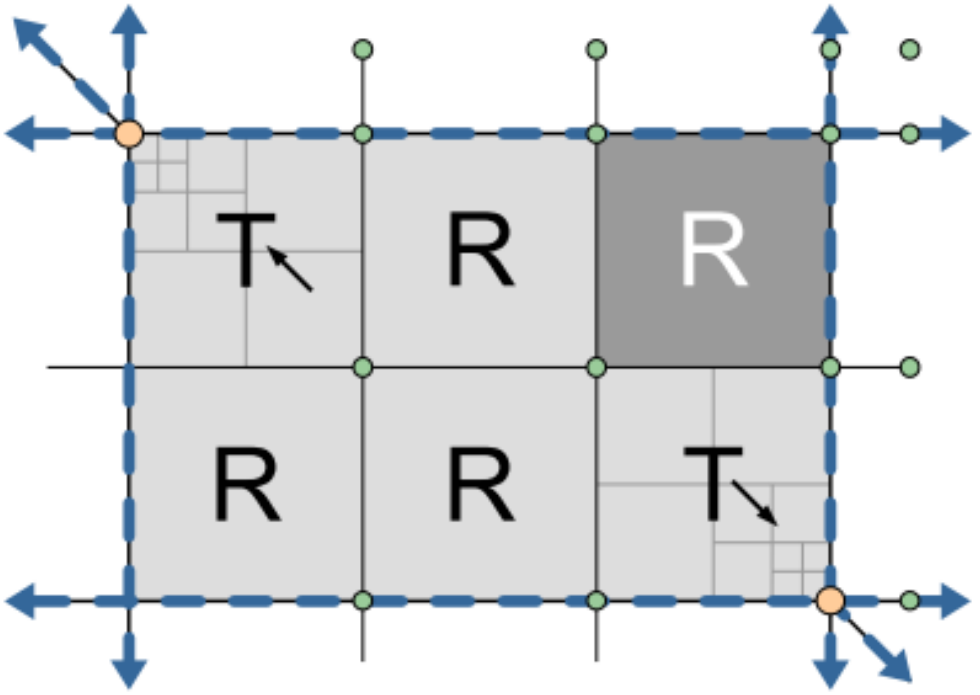


Figure 7: *A 3 × 2 rectangular macro face formed by the intersections of separatrices from two extraordinary vertices (orange). Aggregated irregular faces (T) reference quadtree root nodes, while regular faces (R) share a grid of B-spline control points; the control points (green) used by the shaded regular face are shown.*

Quadriláteros regulares e Irregulares

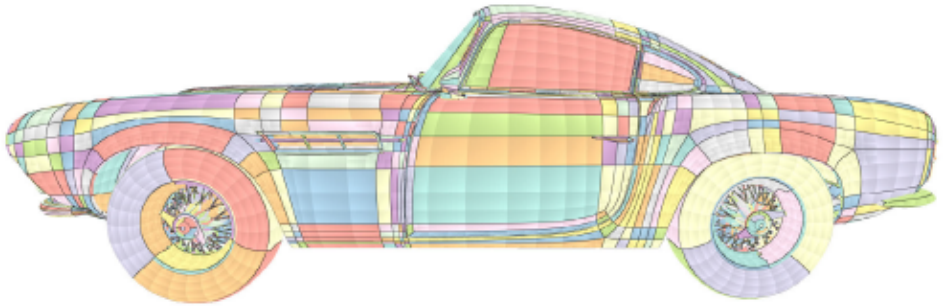


Figure 8: *The Stirling model (©Disney/Pixar) rendered with rectangular macro faces. Colors indicate different macro faces; shading shows the parameter (sub-)domains of aggregated face quadtrees. The wheels show that macro face dimensions are limited to 16, allowing a maximum per-face tessellation factor of 4.*

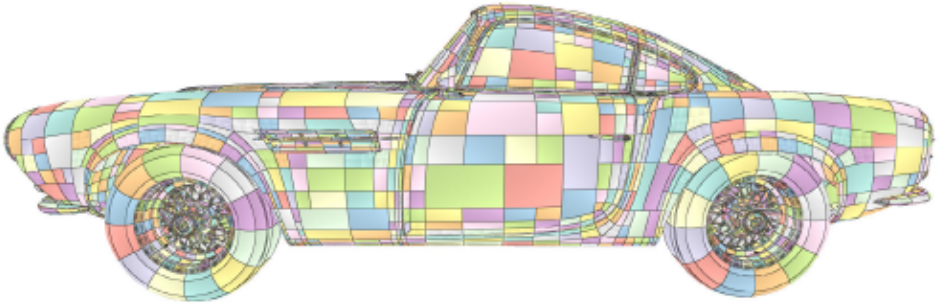
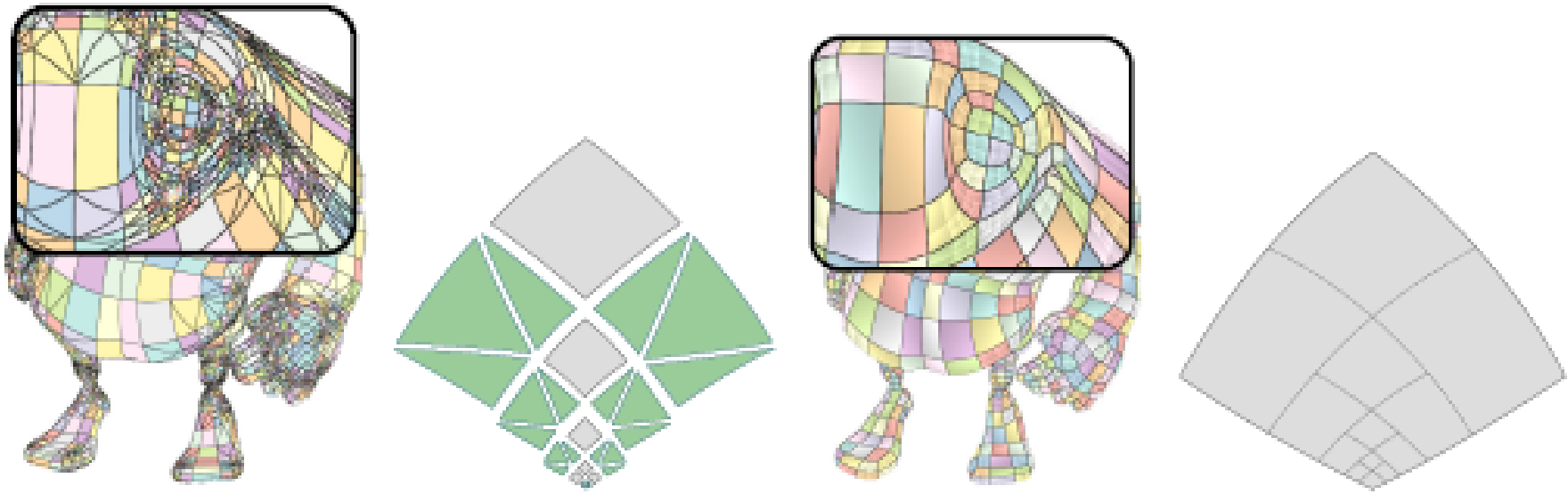


Figure 10: *The Stirling model (©Disney/Pixar) with regular, quartet, and irregular faces. As the tessellation factor is uniform, quartets are allowed to be offset relative to one another.*

8.1 Subdivision Ring Buffer

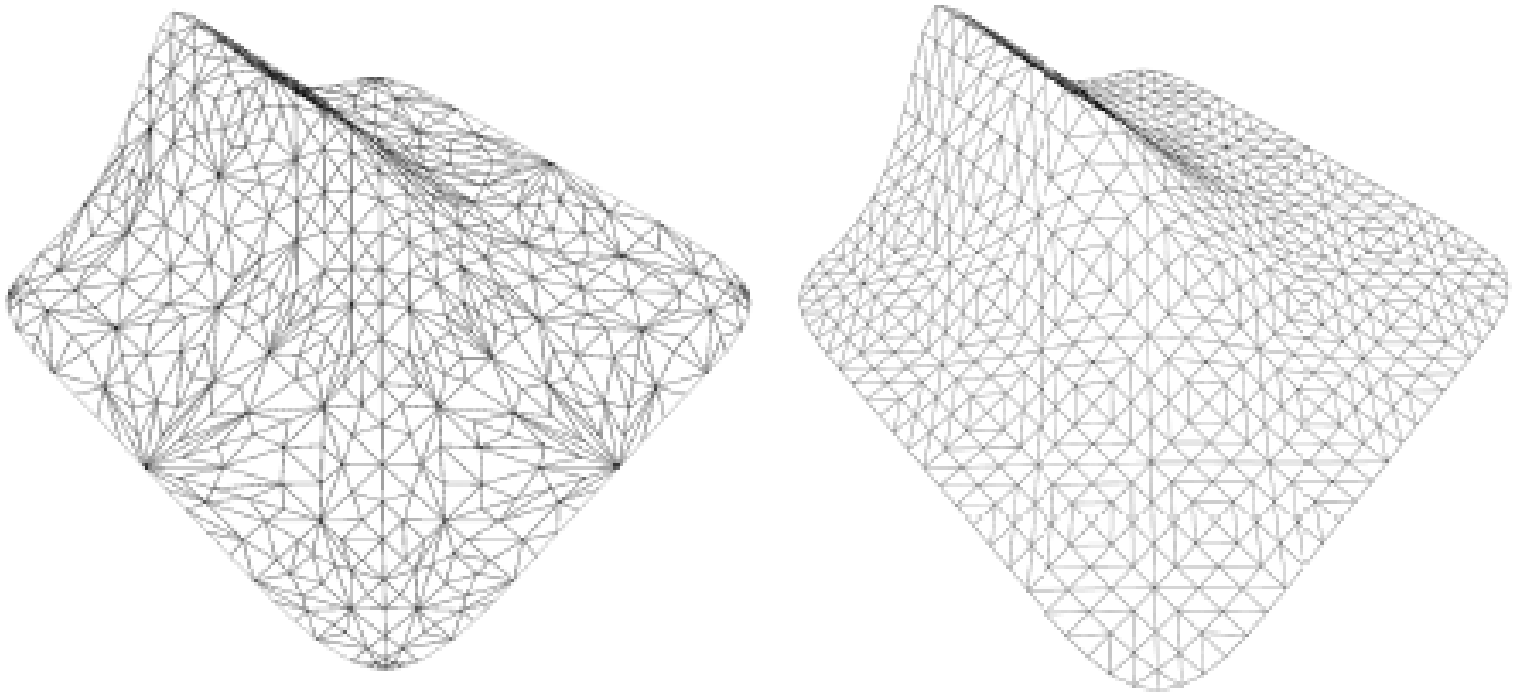
Comparações



(a) *FAS* (b) *Ours*

Figure 2: *In order to isolate an extraordinary vertex, FAS subdivides a face into multiple primitives, including transition patches (green) to stitch T-junctions. Our algorithm uses a single primitive per quad face, with a precomputed internal hierarchy.*

Comparações



(a) *FAS* (b) *Ours*

Figure 3: *Comparison of tessellation pattern quality between FAS and our approach. Our tessellation density is more uniform due to our one-to-one mapping between submitted and rendered faces.*

Comparações

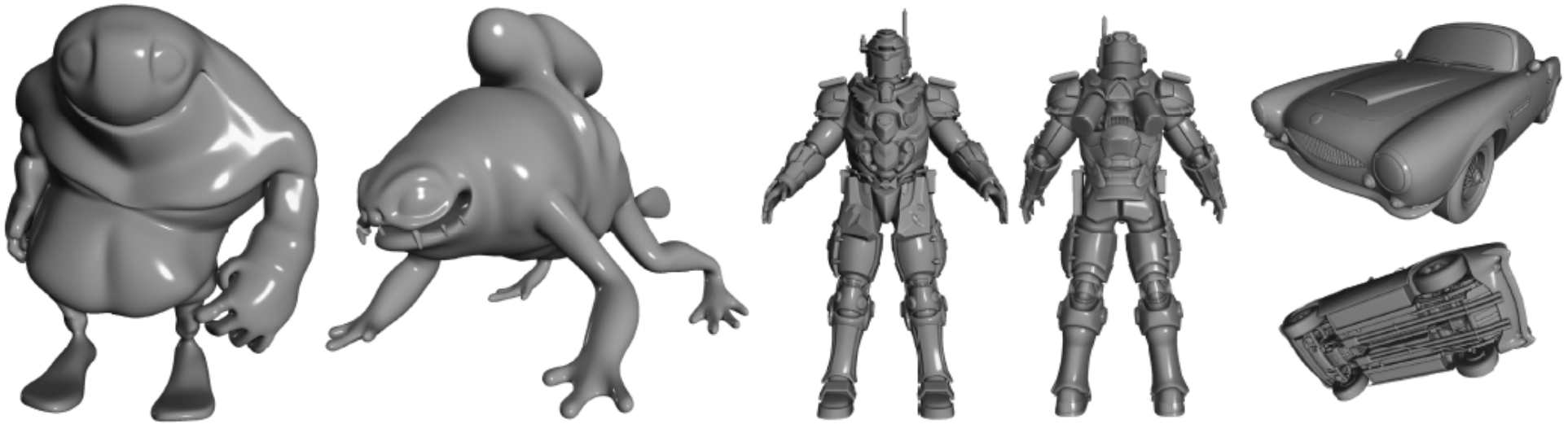
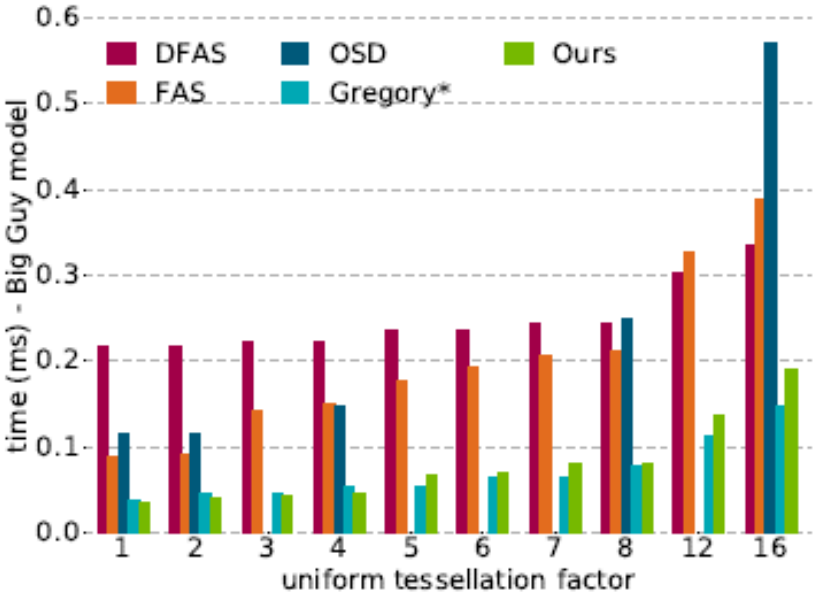
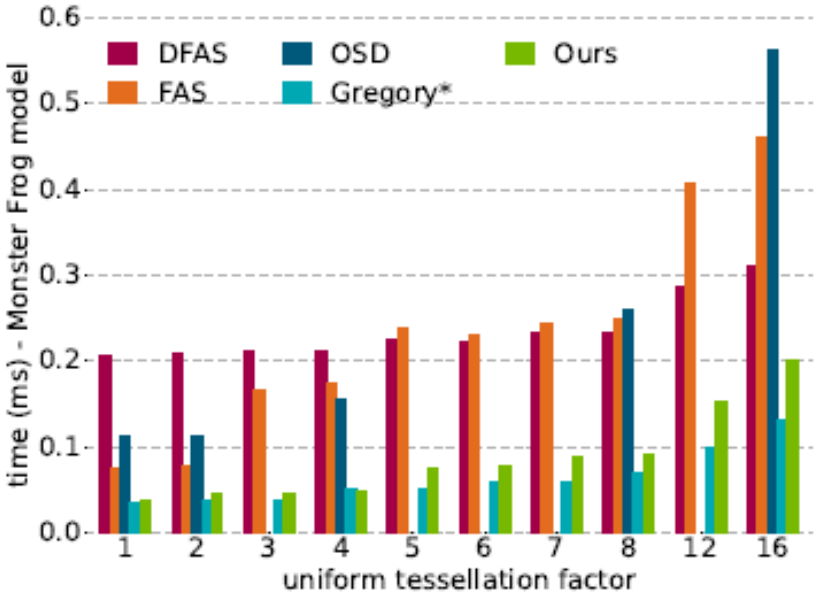
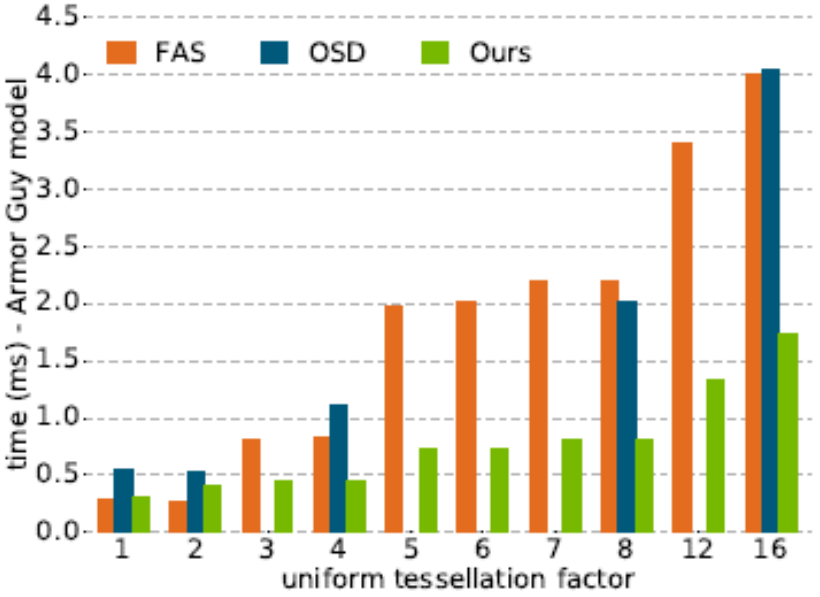


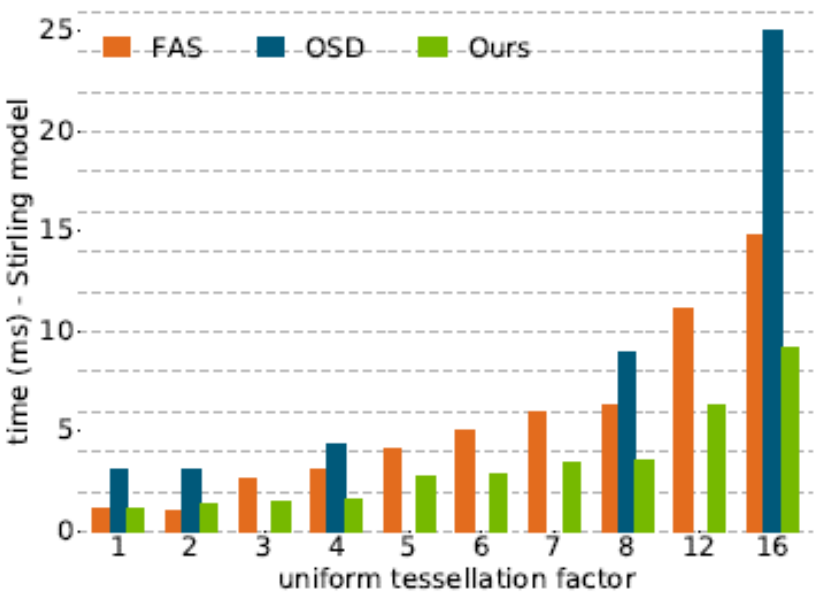
Figure 12: *Models used in our evaluation, from left to right: Big Guy, Monster Frog, Armor Guy (©2014 DigitalFish, Inc. All rights reserved), and Stirling (©Disney/Pixar).*

Comparações









Limitações

- Número fixo de profundidade máxima
- Número de vértices que o vertex shader pode passar para o hull shader (32 na implementação estudada).
- Funciona apenas com o modelo de subdivisão de Catmull-Clark

Artigo 3:

EUROGRAPHICS 2017/ A. Peytavie and C. Bosch

Short Paper

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

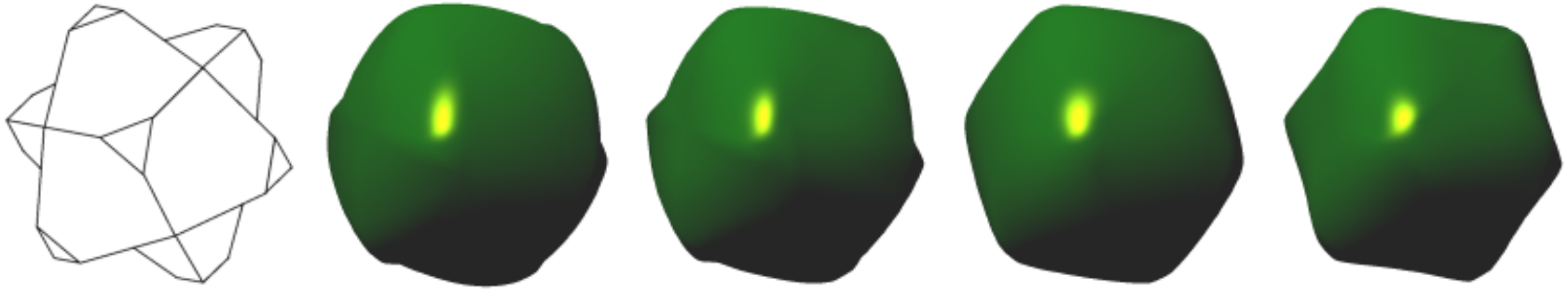


Figure 1: *Rendering a model with 8 triangles and 12 non-planar hexagons. Left to right: input model, original Phong tessellation [BA08], original PN triangles [VPBM01] (hexagons have been triangulated), extended Phong tessellation, and PN triangles extended to PN polygons.*

Generalização das coordenadas baricêntricas

Para um polígono qualquer:

- Partição da unidade
- Não negativos
- Reprodução linear

Existem várias formulações para se generalizar coordenadas baricêntricas.

Tesselagem linear e quadrática

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

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

Tesselagem cúbica

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

where

$$\mathbf{b}_{300} = \mathbf{v}_1,\ \mathbf{b}_{030} = \mathbf{v}_2,\ \mathbf{b}_{003} = \mathbf{v}_3,$$

$$\mathbf{b}_{210} = \frac{2}{3}\mathbf{v}_1 + \frac{1}{3}\pi_1(\mathbf{v}_2),\ \mathbf{b}_{120} = \frac{2}{3}\mathbf{v}_2 + \frac{1}{3}\pi_2(\mathbf{v}_1),$$

$$\mathbf{b}_{201} = \frac{2}{3}\mathbf{v}_1 + \frac{1}{3}\pi_1(\mathbf{v}_3),\ \mathbf{b}_{102} = \frac{2}{3}\mathbf{v}_3 + \frac{1}{3}\pi_3(\mathbf{v}_1),$$

$$\mathbf{b}_{021} = \frac{2}{3}\mathbf{v}_2 + \frac{1}{3}\pi_2(\mathbf{v}_3),\ \mathbf{b}_{012} = \frac{2}{3}\mathbf{v}_3 + \frac{1}{3}\pi_3(\mathbf{v}_2),$$

$$\mathbf{e} = (\mathbf{b}_{210} + \mathbf{b}_{120} + \mathbf{b}_{201} + \mathbf{b}_{102} + \mathbf{b}_{021} + \mathbf{b}_{012})/6,$$

$$\mathbf{c} = (\mathbf{v}_1 + \mathbf{v}_2 + \mathbf{v}_3)/3, \text{ and } \mathbf{b}_{111} = \mathbf{e} + (\mathbf{e} - \mathbf{c})/2.$$

Extended Phong Tesselation

$$\mathbf{p}^2(\boldsymbol{\phi}) = (\phi_1, \phi_2, \ldots, \phi_n)\,(\pi_1(\mathbf{p}(\boldsymbol{\phi})), \ldots, \pi_n(\mathbf{p}(\boldsymbol{\phi}))^{\mathsf{T}}. \qquad (6)$$

Subdivisão cúbica representada em combinação de subdivisões quadráticas

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

with

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

Interpolação de normais

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

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

Resultados

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.*

Resultados

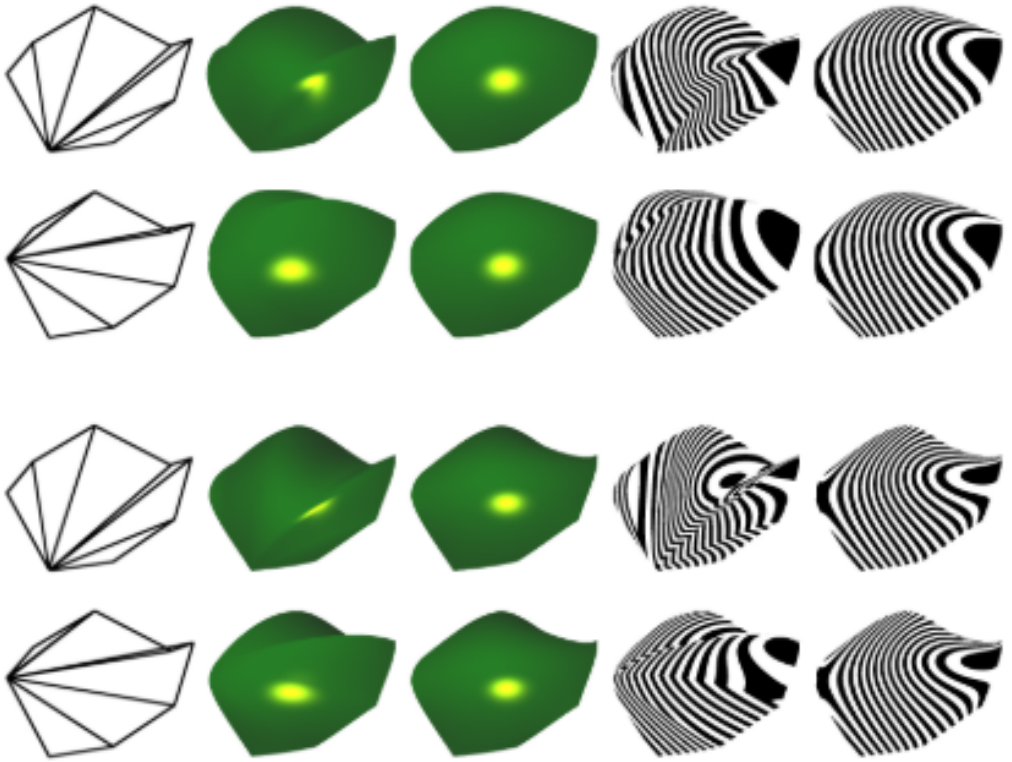


Figure 3: *Comparison: Two different triangulations of a non-planar octagon. Top two rows: Phong tesselation and extended Phong tessellation, reflection lines of linear normal fields on Phong tessellation and extended Phong tessellation. Bottom two rows: PN triangles and PN polygons, reflection lines 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.*

Desempenho

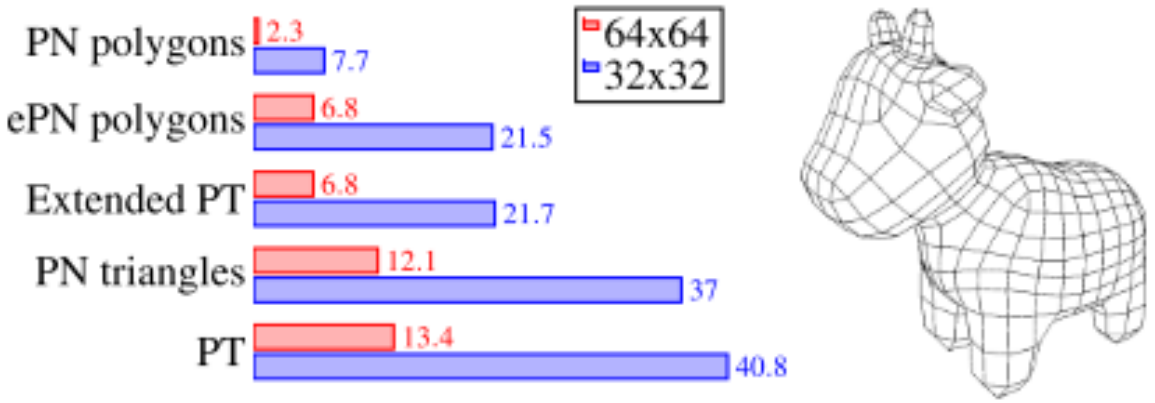


Figure 4: *Performance comparison in FPS per method: rendering a model (right) containing 56 triangles, 634 quads, 40 pentagons, and 4 hexagons at two tessellation depths. ePN polygons refers to the explicit calculation of control points, see* (9)*, and PT refers to Phong tessellation.*

Conclusão

Apesar do desempenho mais lento em relação as soluções comparadas, as extensões propostas no artigo conseguem criar superfícies complexas independente da triangulação do polígono