

POINT NORMAL TRIANGLES

Rick van Veen Laura Baakman

December 14, 2015

Advanced Computer Graphics

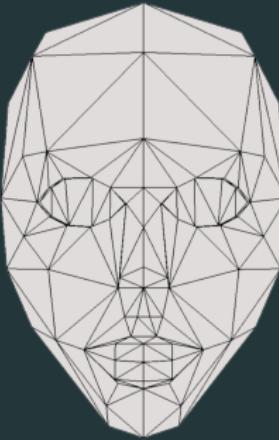
Point Normal triangles

2015-12-12

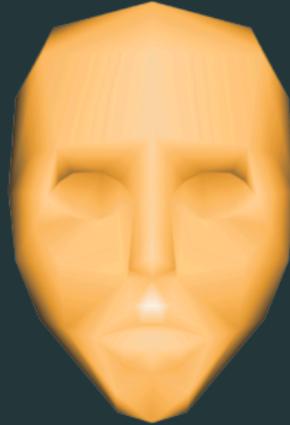
POINT NORMAL TRIANGLES

Rick van Veen Laura Baakman
December 14, 2015
Advanced Computer Graphics

[Rick] Welcome everybody. Tell people that PN means Point Normal triangles.



INPUT MESH



GOURAUD



PN GEOMETRY



PN TRIANGLES

Point Normal triangles

2015-12-12

[Rick] Why PN triangles? Look at the nice result it gives :-) and we will see that it's easy to extend it to the 'existing' pipeline.



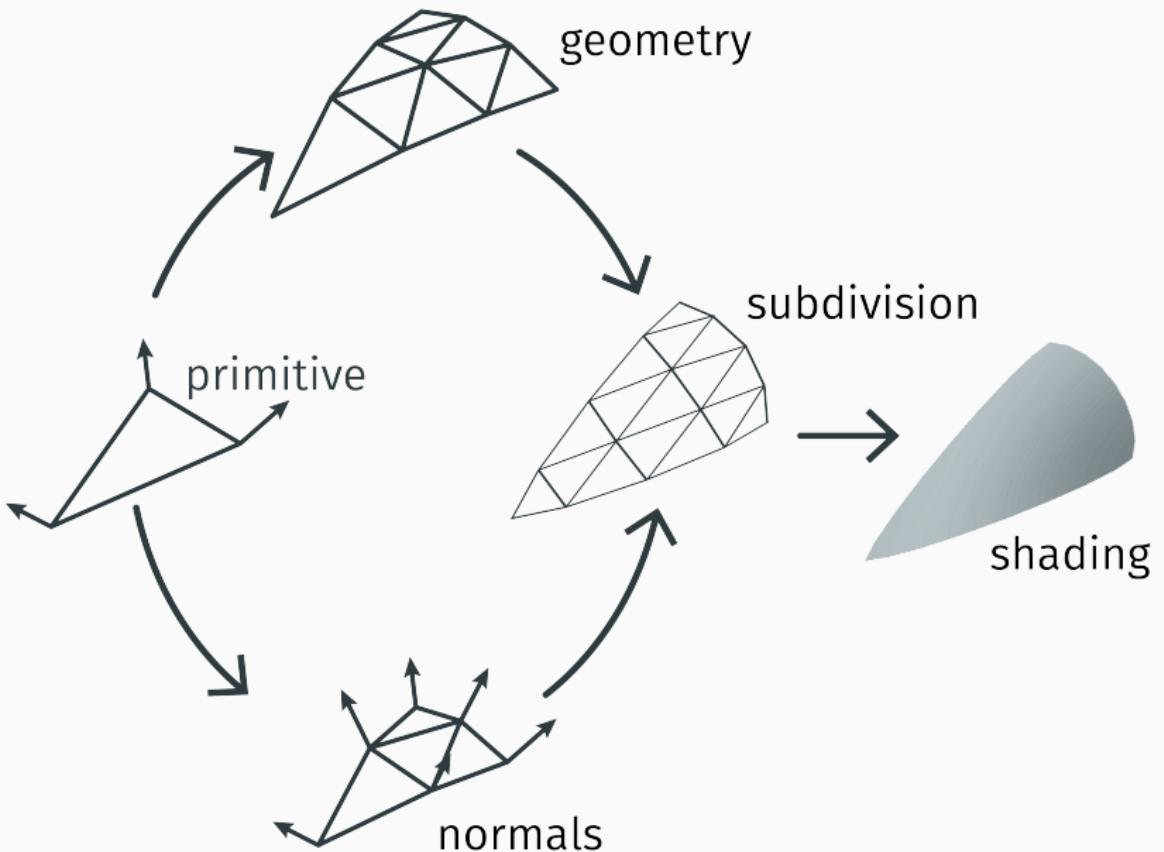
SINGLE PN TRIANGLE

Point Normal triangles
└ Single PN Triangle

2015-12-12

SINGLE PN TRIANGLE

[Rick] How does one construct a single PN triangle?
Overview on the next slide

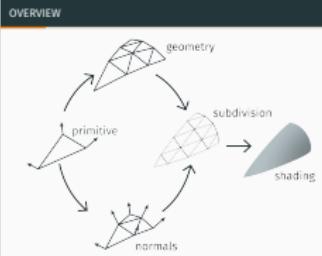


Point Normal triangles

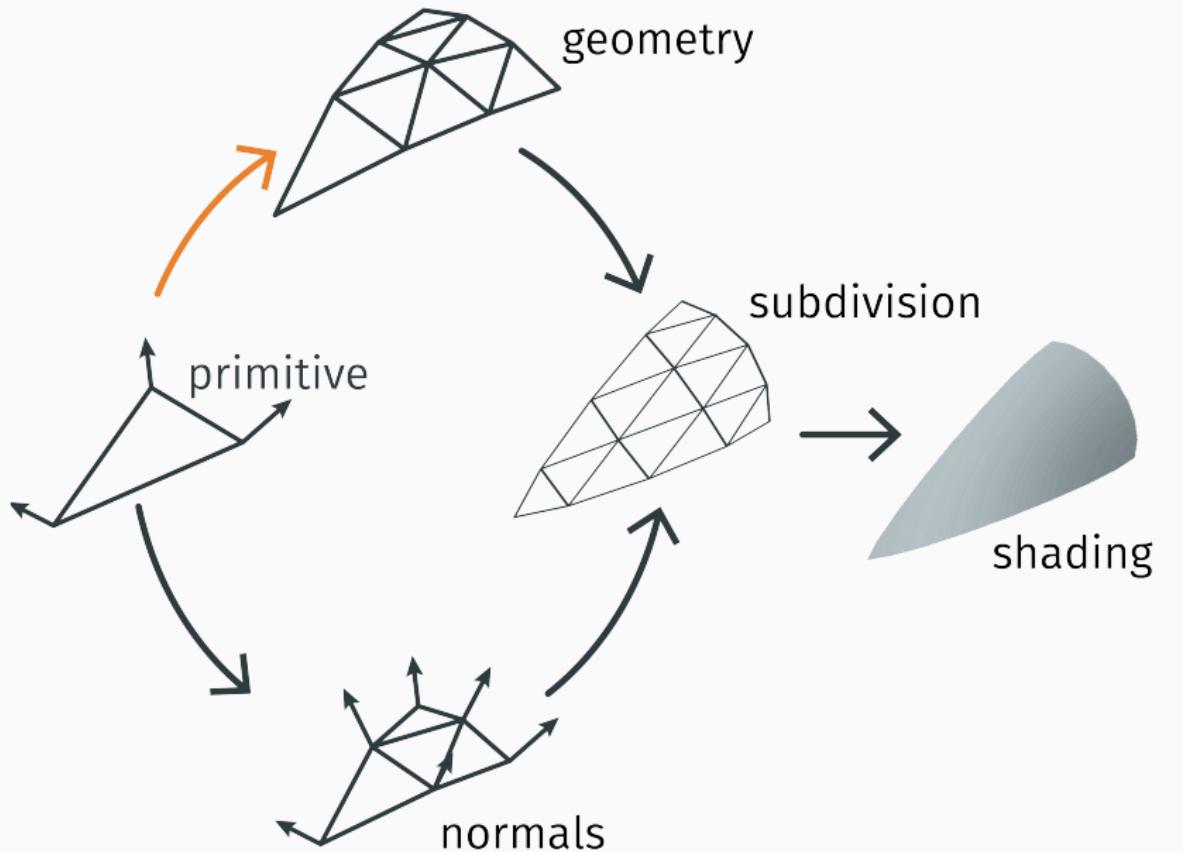
Single PN Triangle

Overview

2015-12-12



[Laura] PN triangle is defined by geometry and normal component.

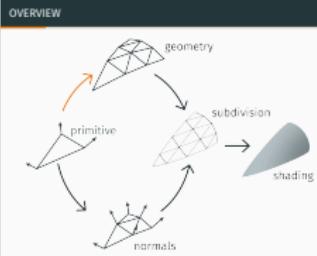


Point Normal triangles

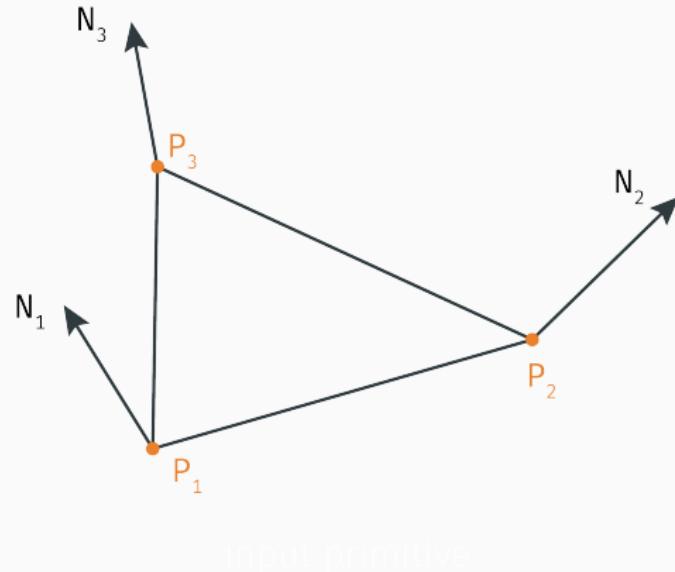
Single PN Triangle

Overview

2015-12-12



Laura From geometric component of input primitive to geometric component of PN triangle

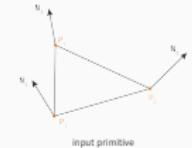


Point Normal triangles

└ Single PN Triangle

└ Geometry

2015-12-12

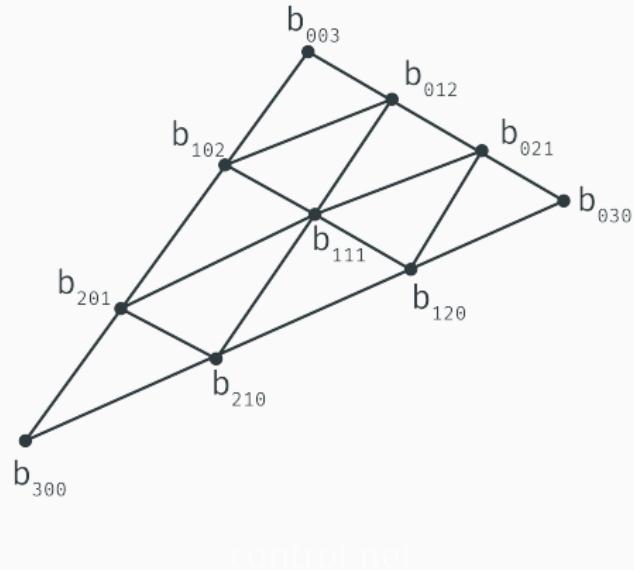


[Laura] This is a standard triangle primitive, defined by its vertices and normals.

Focus on getting the different control primitives.

Note that we only have this input primitive, without information about its neighbours.

GEOMETRY - VERTEX COEFFICIENTS



$$b_{ijk} = (iP_1 + jP_2 + kP_3)/3$$

$$b_{300} = P_1,$$

$$b_{030} = P_2,$$

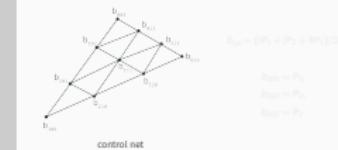
$$b_{003} = P_3$$

Point Normal triangles

└ Single PN Triangle

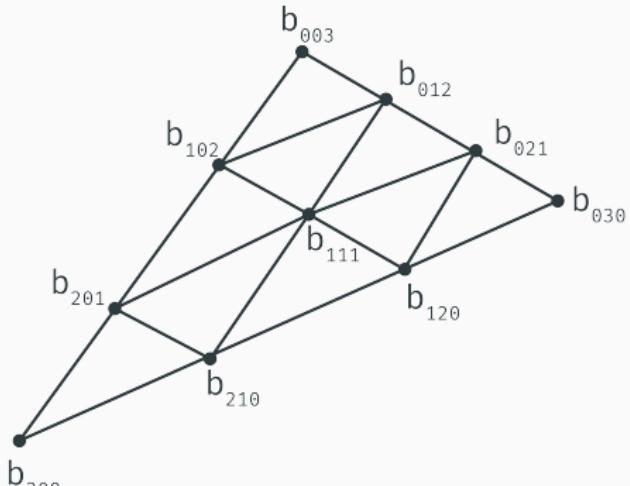
└ Geometry - Vertex Coefficients

2015-12-12



[Laura] These are all the initial control point. Evenly divided on the triangle. -> formula **Laura** Mention which vertices are vertex, tangent and center coefficient.

GEOMETRY - VERTEX COEFFICIENTS



control net

$$b_{ijk} = (iP_1 + jP_2 + iP_3)/3$$

$$b_{300} = P_1,$$

$$b_{030} = P_2,$$

$$b_{003} = P_3$$

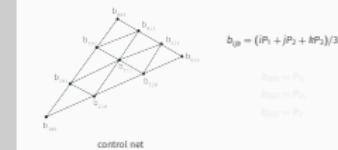
Point Normal triangles

└ Single PN Triangle

└ Geometry - Vertex Coefficients

2015-12-12

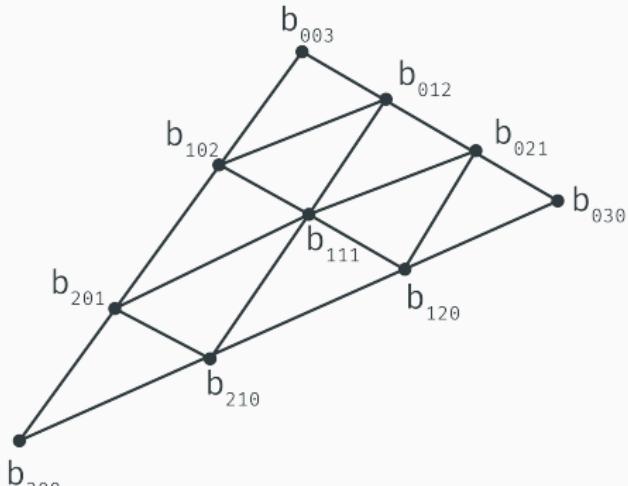
[Laura] Nice formula



$$b_{ijk} = (iP_1 + jP_2 + iP_3)/3$$

$b_{ijk} = P_1$
 $b_{i+1,j,k} = P_2$
 $b_{i,j+1,k} = P_3$

GEOMETRY - VERTEX COEFFICIENTS



control net

$$\begin{aligned}
 b_{ij0} &= P_1 \\
 b_{ij1} &= P_2 \\
 b_{ij2} &= P_3
 \end{aligned}$$

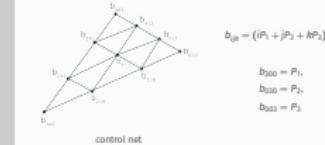
Point Normal triangles

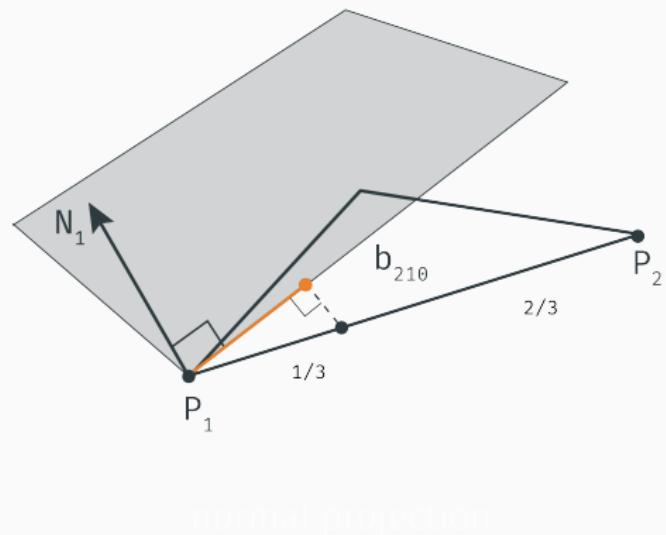
└ Single PN Triangle

└ Geometry - Vertex Coefficients

2015-12-12

[Laura] Stress that the vertex coefficients/control points are the one on the original vertices and that they do not move.





$$w_{ij} = (P_j - P_i) \cdot N_i \in \mathbb{R}$$

$$b_{210} = \frac{2P_1 + P_2 - w_{12}N_1}{3},$$

⋮

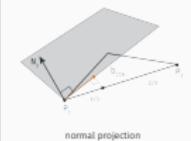
$$b_{201} = \frac{2P_1 + P_3 - w_{13}N_1}{3}$$

Point Normal triangles

└ Single PN Triangle

└ Geometry - Tangent Coefficients

2015-12-12

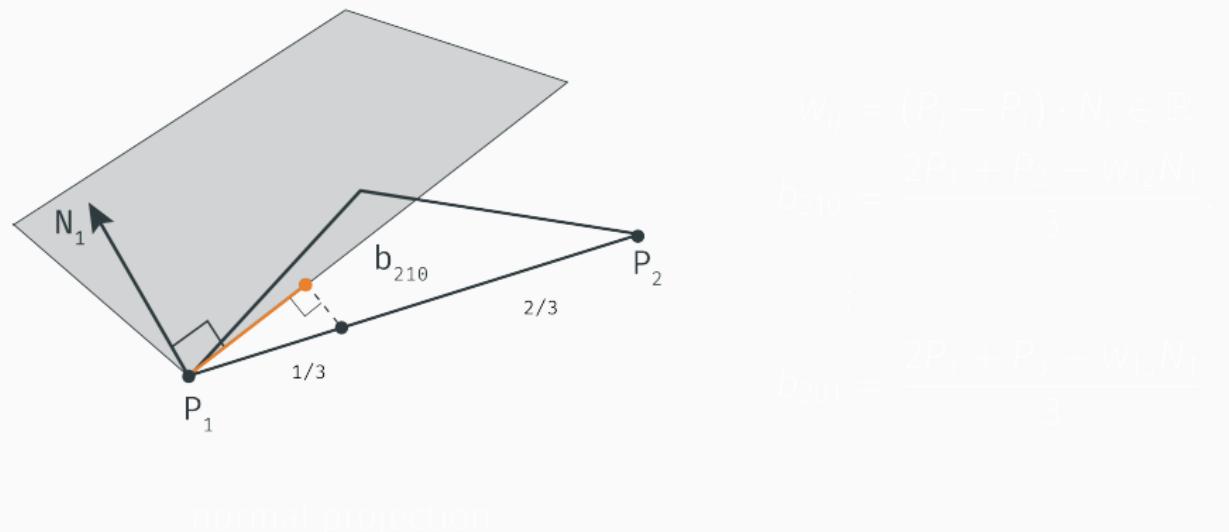


$$w_{ij} = (P_j - P_i) \cdot N_i \in \mathbb{R}$$

$$b_{ijk} = \frac{2P_i + P_j + P_k - w_{ij}N_i - w_{ik}N_i - w_{jk}N_i}{3}$$

[Laura] Define a plane using the closest vertex and its normal. Find the point on this plane that is closest to the uniformly distributed point.

GEOMETRY - TANGENT COEFFICIENTS

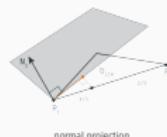


Point Normal triangles

└ Single PN Triangle

└ Geometry - Tangent Coefficients

2015-12-12



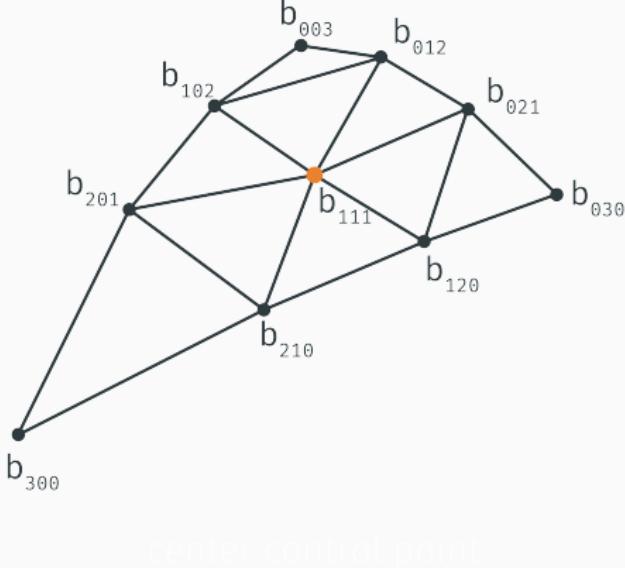
$$w_0 = (P_j - P_i) \cdot N_j \in \mathbb{R}$$

$$b_{210} = \frac{2P_1 + P_2 - w_{10}N_1}{3}$$

$$\vdots$$

$$b_{201} = \frac{2P_1 + P_3 - w_{10}N_1}{3}$$

GEOMETRY - CENTER COEFFICIENT

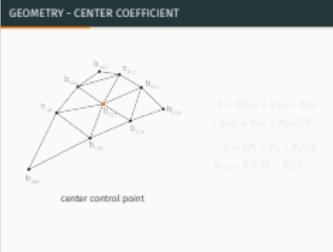


$$E = (b_{210} + b_{120} + b_{021} + b_{012} + b_{102} + b_{201})/6,$$
$$V = (P_1 + P_2 + P_3)/3,$$
$$b_{111} = E + (E - V)/2$$

Point Normal triangles
└ Single PN Triangle

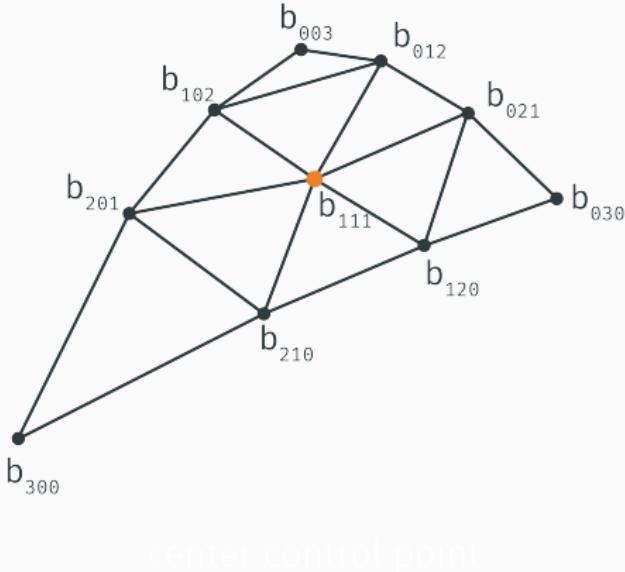
└ Geometry - Center Coefficient

2015-12-12



[Laura] Note that this is the result of the previous step -> now only center coefficient is left.

GEOMETRY - CENTER COEFFICIENT



$$E = (b_{003} + b_{012} + b_{021} + b_{030} + b_{102} + b_{111} + b_{120} + b_{201}) / 8$$

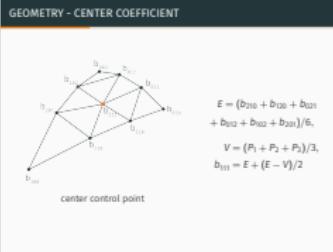
8

Point Normal triangles
└ Single PN Triangle

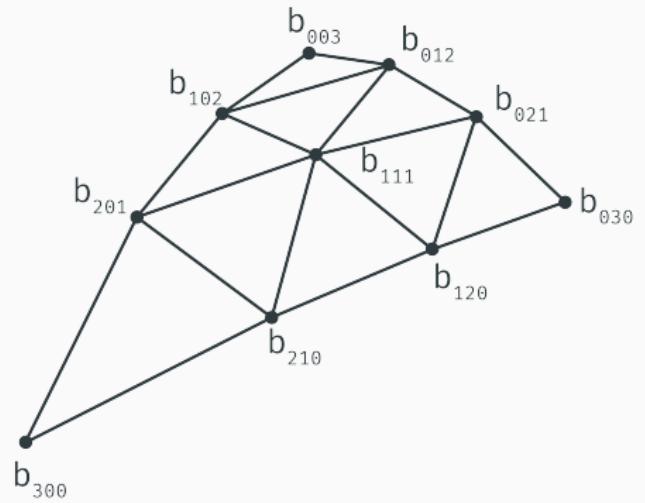
└ Geometry - Center Coefficient

2015-12-12

[Laura] Average of the tangent coefficients plus half the difference between the tangent and vertex coefficients.



GEOMETRY - RESULT



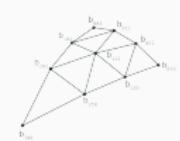
Point Normal triangles

└ Single PN Triangle

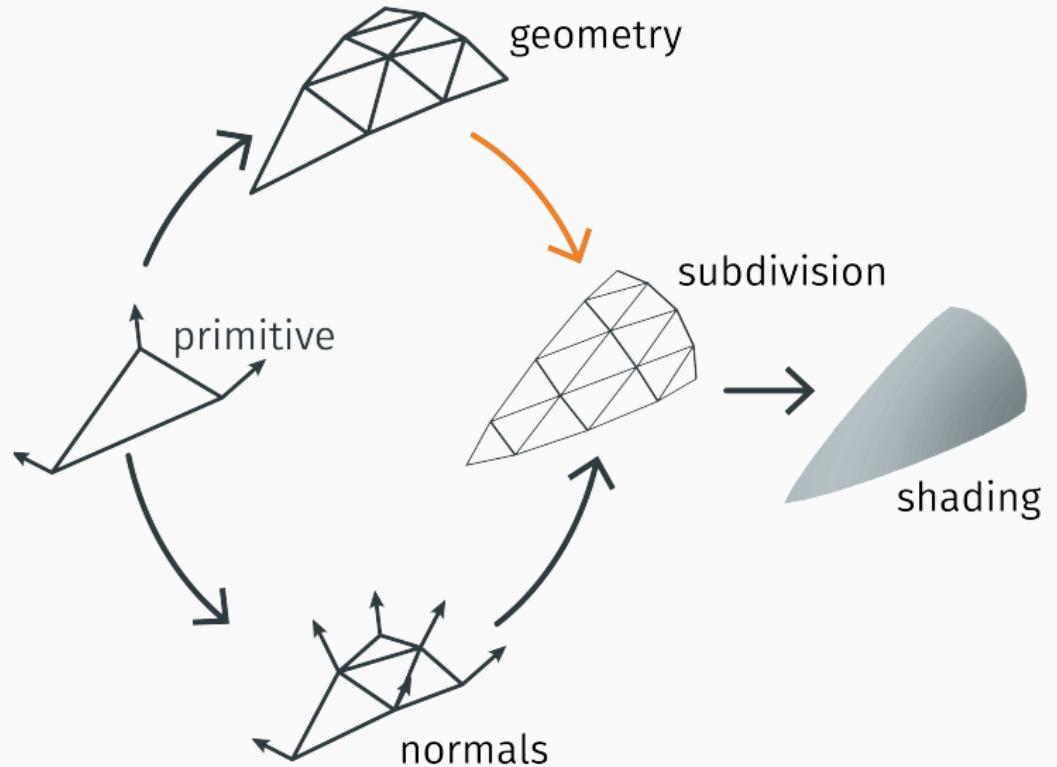
└ Geometry - Result

2015-12-12

[Laura] Results



OVERVIEW

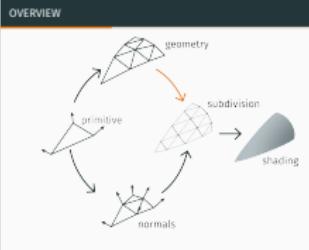


Point Normal triangles

└ Single PN Triangle

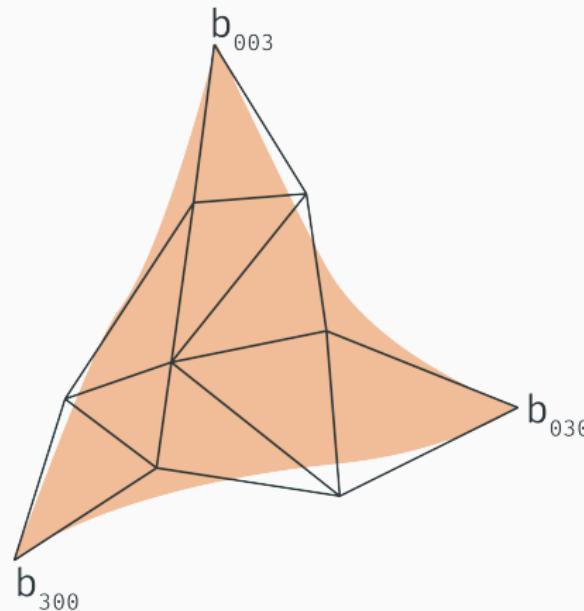
└ Overview

2015-12-12



[Rick] Overview -> how to get from this to shading.
Sample/subdivide with formula on following slide.

CUBIC BÉZIER PATCH



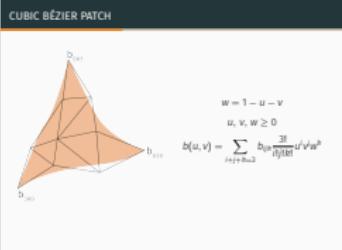
$$b(u, v) = \sum_{i+j+k=3} b_{ijk} \frac{u^i v^j w^k}{i! j! k!}$$

Point Normal triangles

└ Single PN Triangle

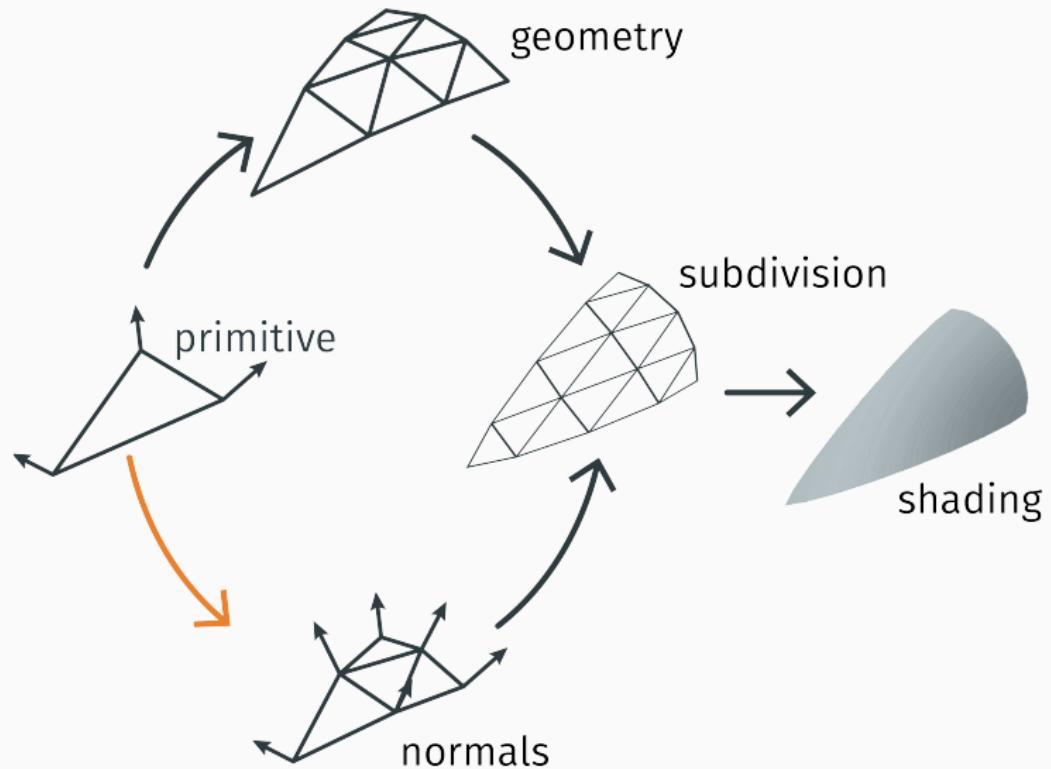
└ Cubic Bézier patch

2015-12-12



[Rick] Very nice formula with a nice picture. Mention **Barycentric coordinates**, u, v, w are a convex combination

OVERVIEW



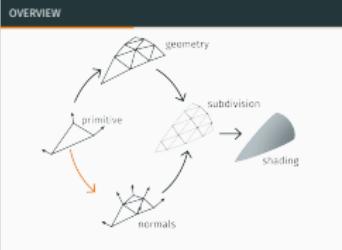
Point Normal triangles

└ Single PN Triangle

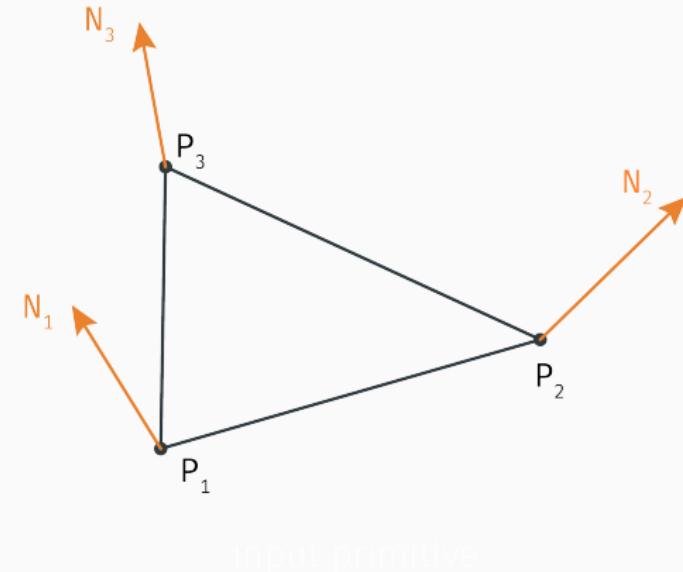
└ Overview

[Rick] From the primitive normals the the PN triangle normals

2015-12-12



NORMALS



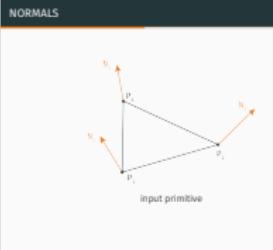
Point Normal triangles

└ Single PN Triangle

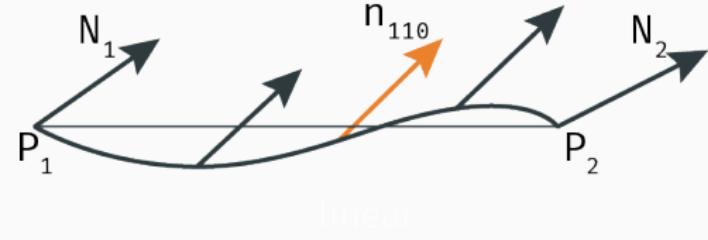
└ Normals

2015-12-12

[Rick] Recap input primitive and with emphasis on the normals.



NORMALS - THEORY



quadratic

Point Normal triangles
└ Single PN Triangle

└ Normals - theory

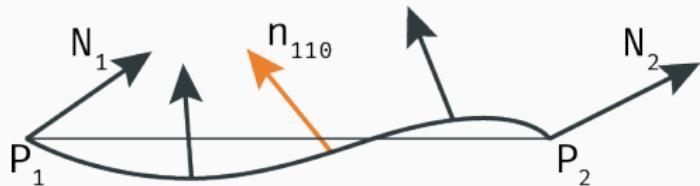
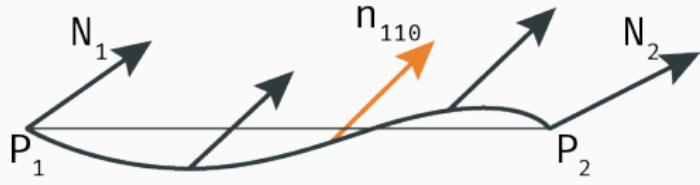
2015-12-12

[Rick] Stress that there is a need to capture the cubic bezier curve (inflection points) and that this cannot be

NORMALS - THEORY



NORMALS - THEORY



Point Normal triangles
└ Single PN Triangle

└ Normals - theory

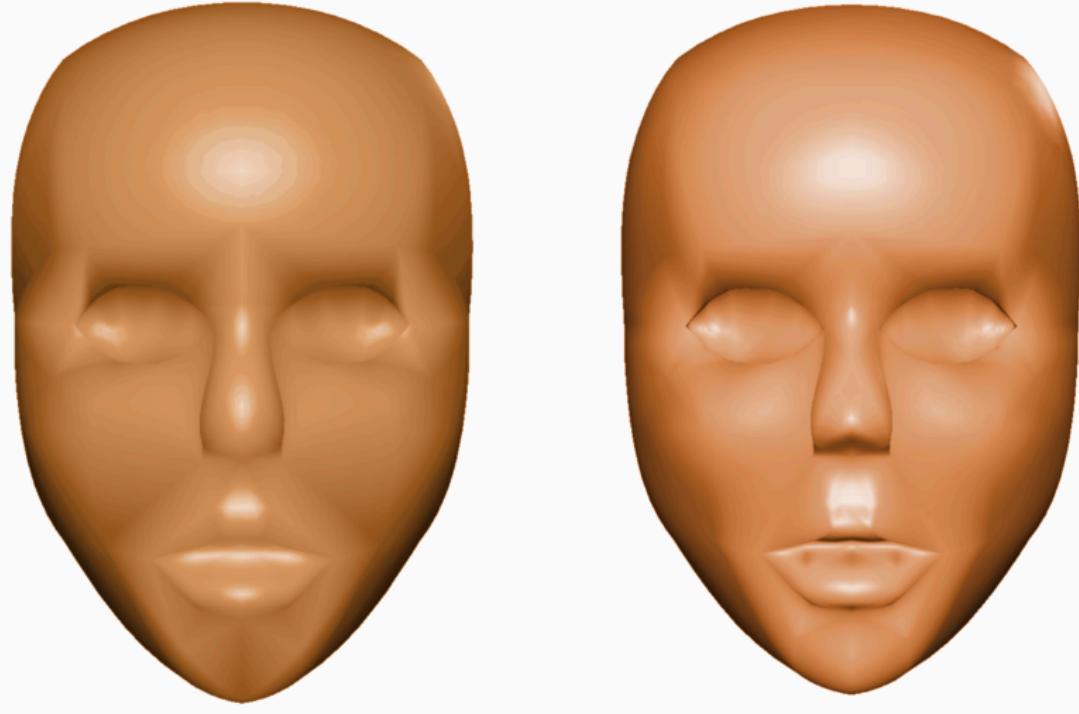
2015-12-12

NORMALS - THEORY



[Rick] Quadratic does capture inflection points. Trade off between performance and result (maybe?)

NORMALS - EXAMPLE



Point Normal triangles

└ Single PN Triangle

└ Normals - example

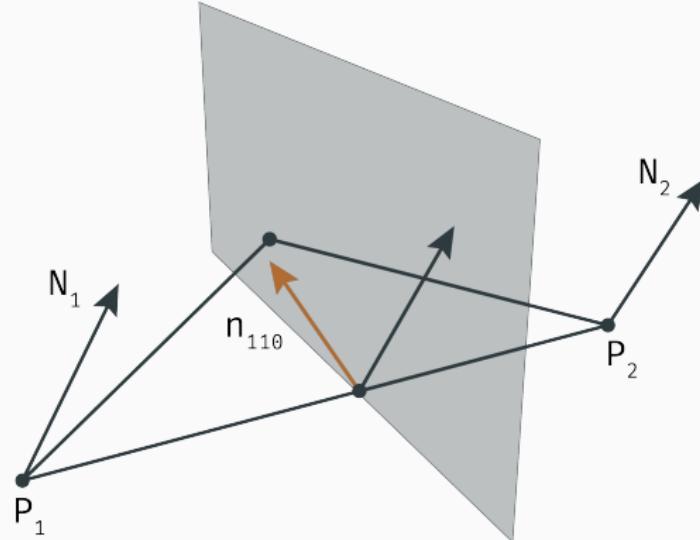
[Rick] Look how pretty.

2015-12-12

NORMALS - EXAMPLE



NORMALS - THEORY



$$v_{ij} = 2 \frac{(P_j - P_i) \cdot (N_i + N_j)}{(P_j - P_i) \cdot (P_j - P_i)} \in \mathbb{R}$$

$$h_{110} = N_1 + N_2 - v_{12}(P_2 - P_1)$$

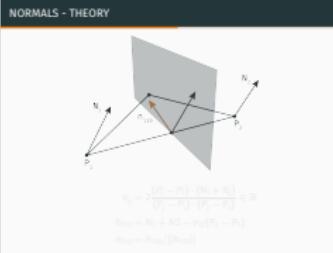
$$n_{110} = h_{110} / \|h_{110}\|$$

Point Normal triangles

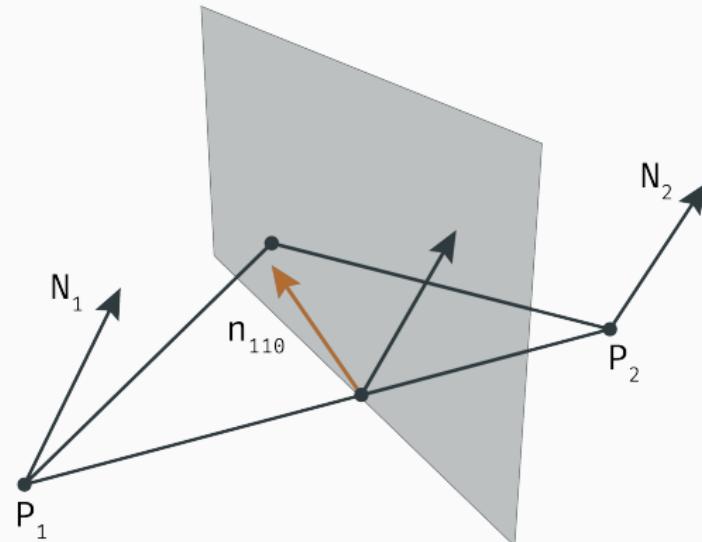
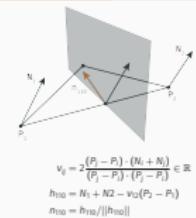
└ Single PN Triangle

└ Normals - theory

2015-12-12



[Rick] Formula in words: reflect the averaged normal (average of N1 and N2) on the plane orthogonal/perpendicular the the edge at the mid point.



$$\begin{aligned} n_{110} &= \frac{(P_1 - P) \cdot (N_1 + N_2)}{(P_1 - P_2) \cdot (P_2 - P)} \\ h_{110} &= N_1 + N_2 - v_{110}(P_2 - P_1) \\ n_{110} &= h_{110}/\|h_{110}\| \end{aligned}$$

Point Normal triangles

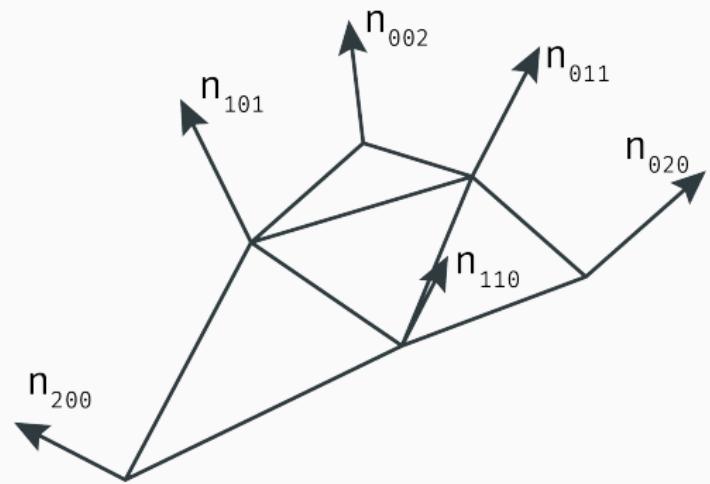
└ Single PN Triangle

└ Normals - theory

2015-12-12

[Rick] Formula in words: reflect the averaged normal (average of N_1 and N_2) on the plane orthogonal/perpendicular the the edge at the mid point.

NORMALS - RESULT



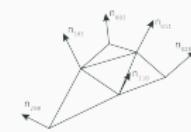
Point Normal triangles

└ Single PN Triangle

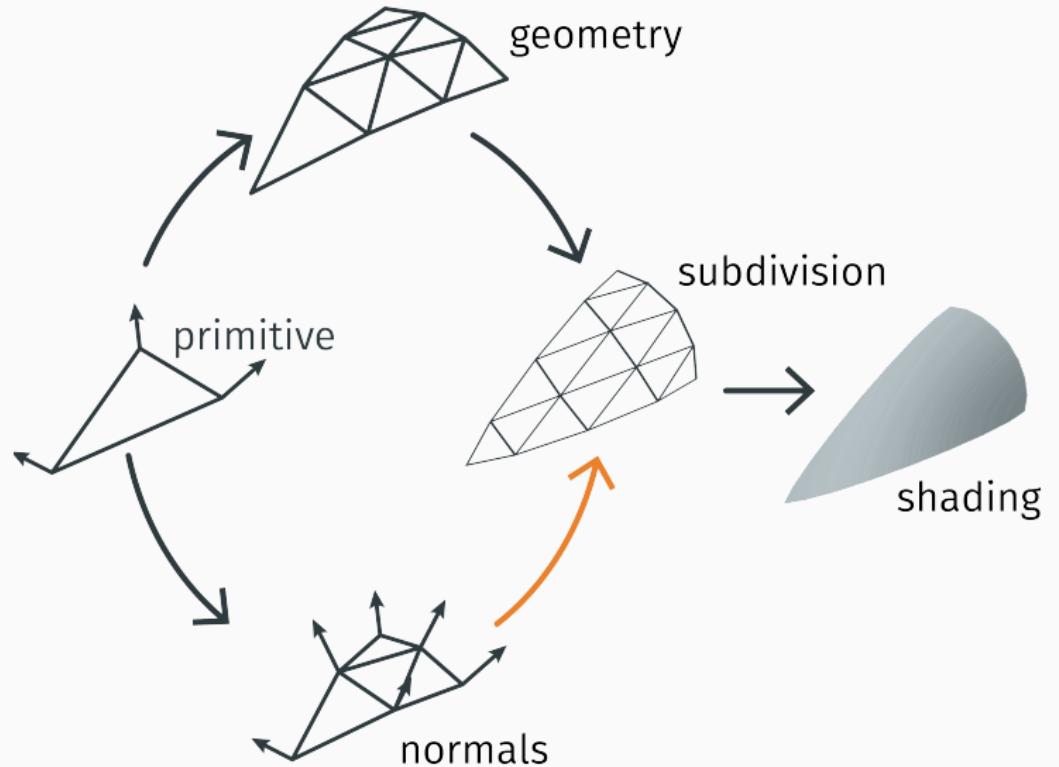
└ Normals - result

[Rick] Result

2015-12-12



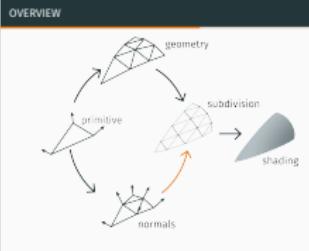
OVERVIEW



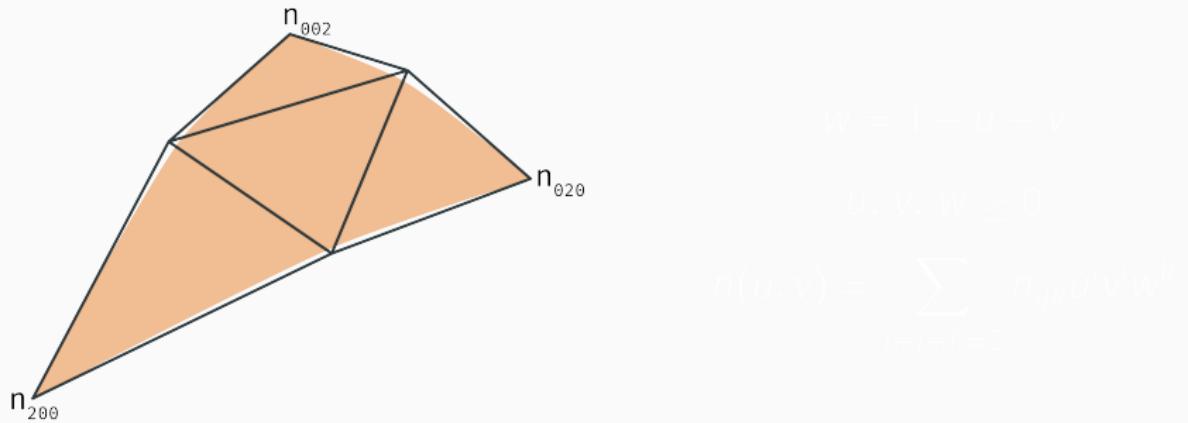
Point Normal triangles
└ Single PN Triangle
 └ Overview

2015-12-12

laura



QUADRATIC PATCH

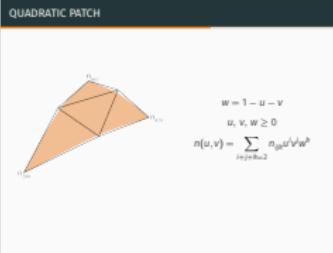


Point Normal triangles

└ Single PN Triangle

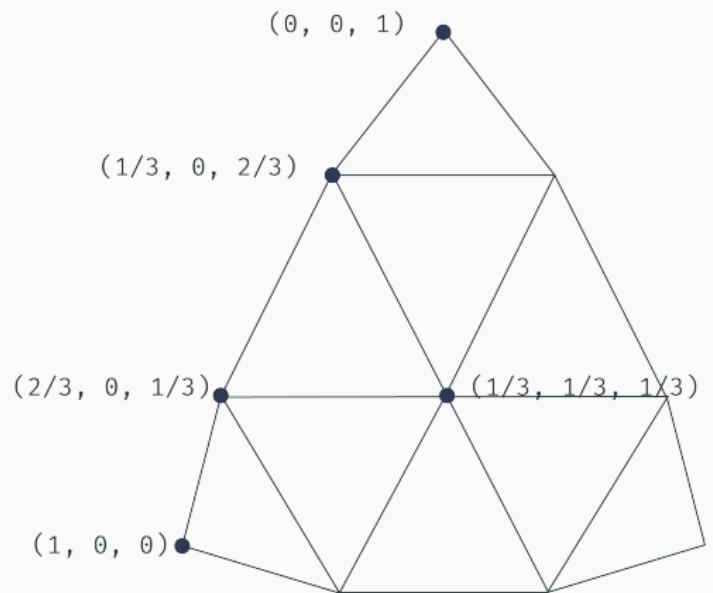
└ Quadratic Patch

2015-12-12



[Laura] u, v and w are convex combinations

LEVEL OF DETAIL

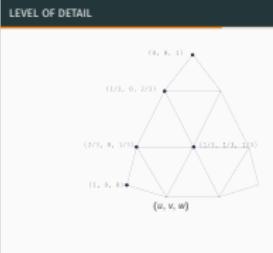


Point Normal triangles
└ Single PN Triangle

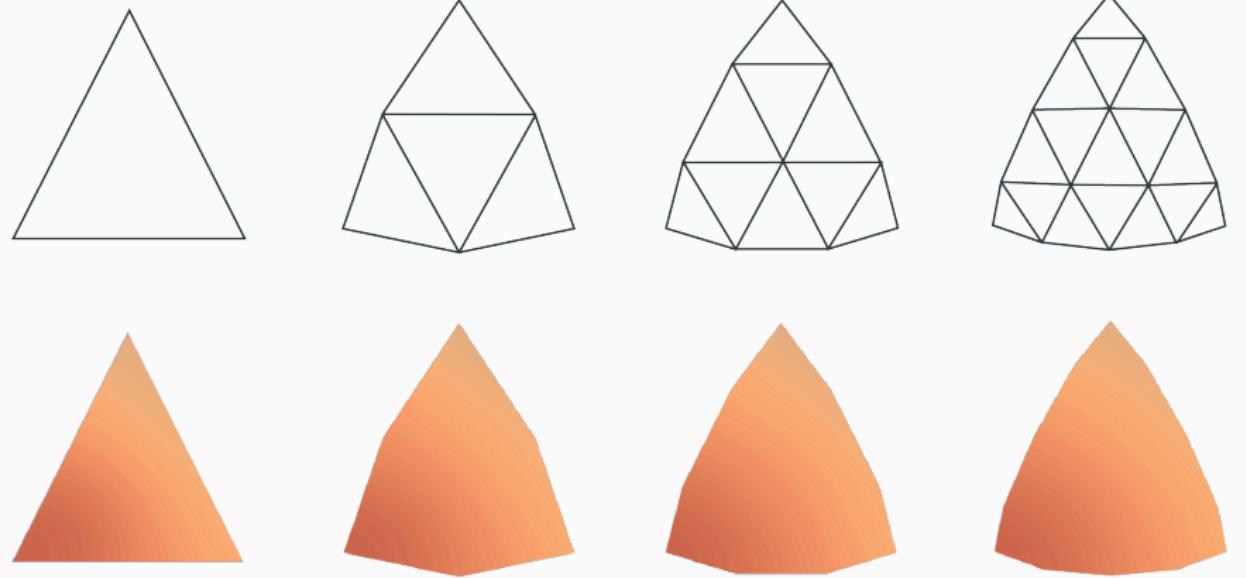
2015-12-12

└ Level Of Detail

[Laura] LOD = 2



LEVEL OF DETAIL

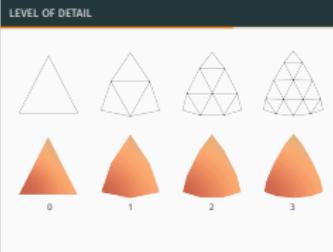


Point Normal triangles

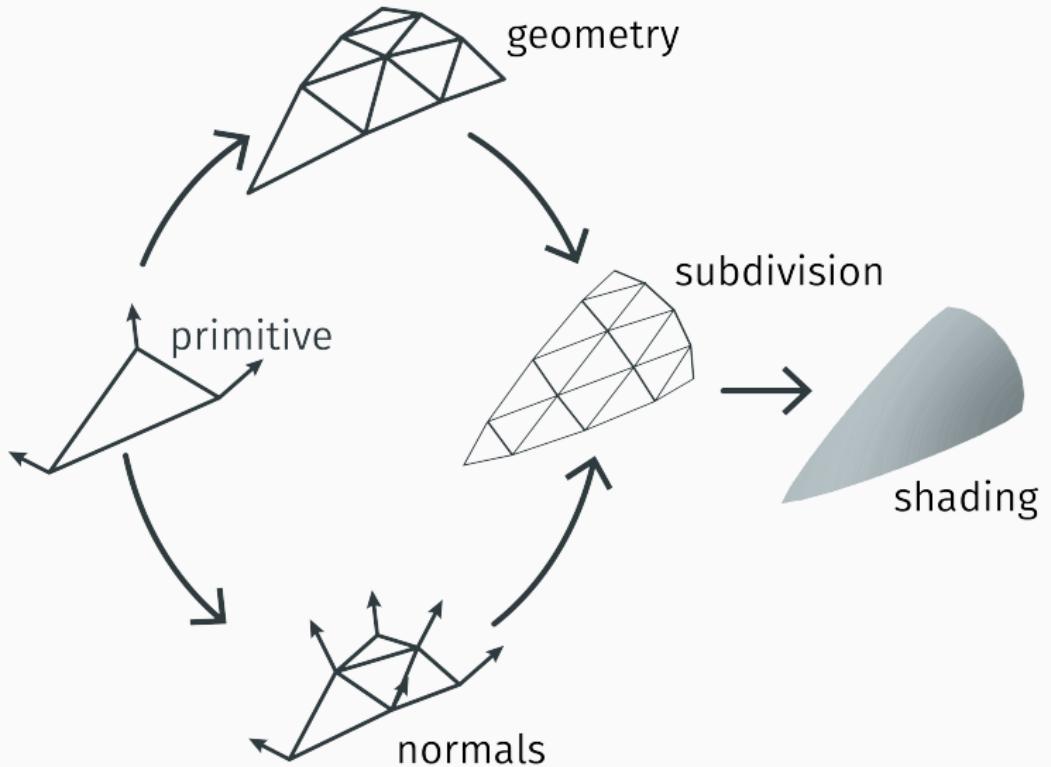
└ Single PN Triangle

└ Level Of Detail

[Laura] Level of detail -> subdivision -> how many triangles go through to the next shaders.



2015-12-12

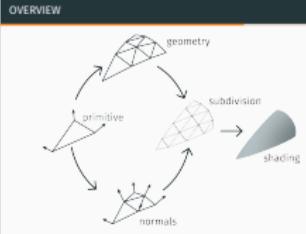


Point Normal triangles

└ Single PN Triangle

└ Overview

2015-12-12



1. [Laura] Recap of the whole process
2. [Laura] Shading out of the scope of this presentation
3. [Laura] Why quadratic patch for normals, why cubic patch for geometry: We need at least cubic geometry and quadratic normals to capture inflections. There are no additional data to suggest higher degree patches.
4. [Laura] Why PN triangles? Look at the nice result it gives :-) and we will see that it easy to extend it to the 'existing' pipeline.

A TRIANGLE MESH

Point Normal triangles
└ A Triangle Mesh

2015-12-12

[rick] blaat

A TRIANGLE MESH

“PN triangles should not deviate too much from the original triangle to preserve the shape and avoid interference with other curved triangles.”¹

¹Vlachos et al.

Point Normal triangles

└ A Triangle Mesh

└ Properties

2015-12-12

[Rick] Problem when combining multiple triangles, so this is an important property

“PN triangles should not deviate too much from the original triangle to preserve the shape and avoid interference with other curved triangles.”¹

¹Marchis et al.

CONTINUITY

PN triangles have²

- C^1 continuity in the vertex points
- C^0 continuity along the edges
- C^∞ everywhere else

²Jiao and Alexander

Point Normal triangles

└ A Triangle Mesh

└ Continuity

2015-12-12

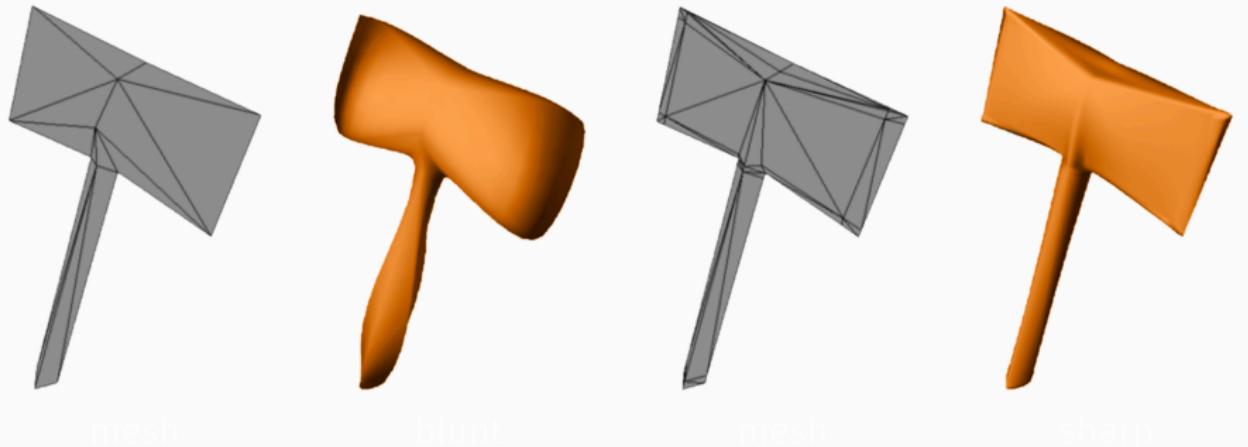
[Rick] Continuity C^0 is important -> no gaps. Higher is better because this gives a more smooth result.

PN triangles have²

- C^1 continuity in the vertex points
- C^0 continuity along the edges
- C^∞ everywhere else

²Jiao and Alexander

SHARP EDGES



Point Normal triangles

└ A Triangle Mesh

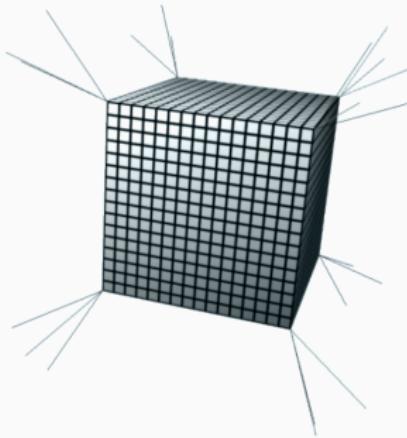
└ Sharp Edges

2015-12-12



[Rick] Curved triangles do not always give the preferred results -> sharp edges. Solution is to insert more triangles at the sharp edges -> model needs to be changed :(

SEPARATE NORMALS



normals



smooth

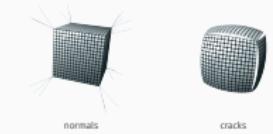
Point Normal triangles

└ A Triangle Mesh

└ Separate Normals

2015-12-12

SEPARATE NORMALS



GRAPHICS PIPELINE

Point Normal triangles
└ Graphics Pipeline

2015-12-12

GRAPHICS PIPELINE

[Laura] How does one construct a single PN triangle?

HARDWARE - PIPELINES



2015

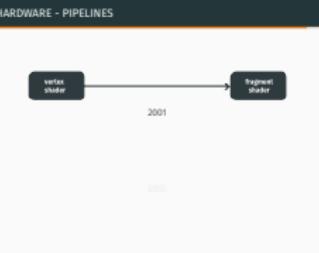
29

Point Normal triangles

└ Graphics Pipeline

└ Hardware - Pipelines

2015-12-12



[Laura] Great part of the paper stresses the point that it can easily be implemented as a preprocessing step (CPU).
2001 pipeline (OpenGL 1.3)

HARDWARE - PIPELINES

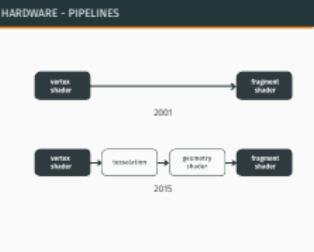


Point Normal triangles

└ Graphics Pipeline

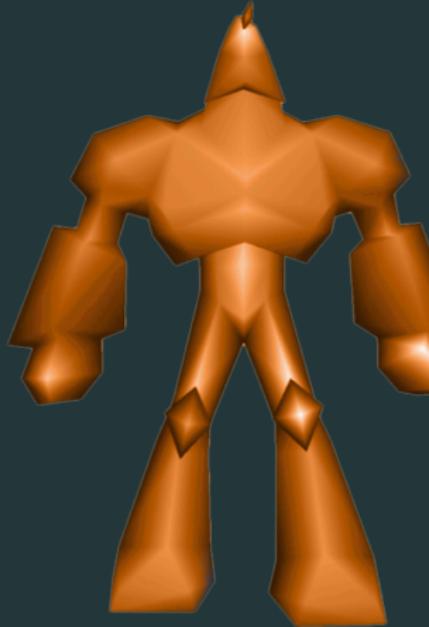
└ Hardware - Pipelines

2015-12-12

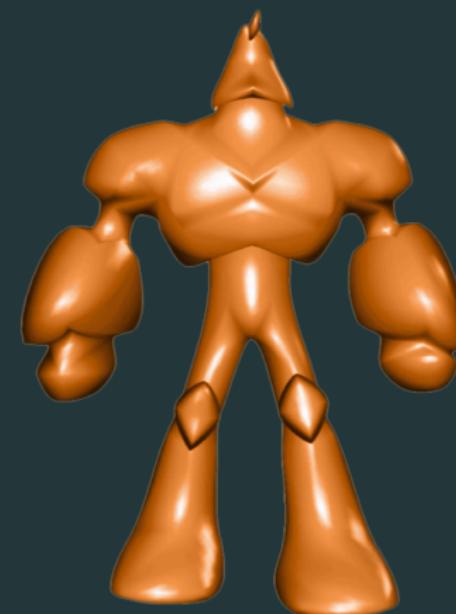


[Laura] 2015 we have OpenGL 4.5 with more programmable shaders and the whole process can be done on the GPU. Since PN triangles only uses the primitive, no neighboring primitives, easy in shaders.

QUESTIONS?



TRIANGLES



PN TRIANGLES

Point Normal triangles
└ Graphics Pipeline

2015-12-12

Laura Why PN triangles are not suited to rendering for CAD: On real objects, the normal field is determined by the geometry and thus fixed. If you invent fake normals the rendering looks better but the user gets an unpleasant surprise when the product is actually manufactured.



Point Normal triangles

└ Graphics Pipeline

└ References

2015-12-12

- Xiangmin Jiao and Phillip J Alexander. "Parallel feature-preserving mesh smoothing". In: *Computational Science and Its Applications–ICCSA 2005*. Springer, 2005, pp. 1180–1189.
- J McDonald and M Kilgard. *Crack-free point-normal triangles using adjacent edge normals*. 2010.
- Alex Vlachos et al. "Curved PN triangles". In: *Proceedings of the 2001 symposium on Interactive 3D graphics*. ACM, 2001, pp. 159–166.

- Xiangmin Jiao and Phillip J Alexander. "Parallel feature-preserving mesh smoothing". In: *Computational Science and Its Applications–ICCSA 2005*. Springer, 2005, pp. 1180–1189.
- J McDonald and M Kilgard. *Crack-free point-normal triangles using adjacent edge normals*. 2010.
- Alex Vlachos et al. "Curved PN triangles". In: *Proceedings of the 2001 symposium on Interactive 3D graphics*. ACM, 2001, pp. 159–166.