

# POINT NORMAL TRIANGLES

---

Rick van Veen   Laura Baakman

December 14, 2015

Advanced Computer Graphics

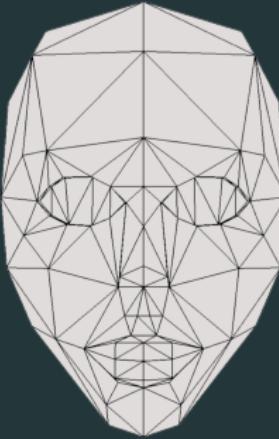
## Point Normal triangles

2015-12-11

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

2015-12-11

[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

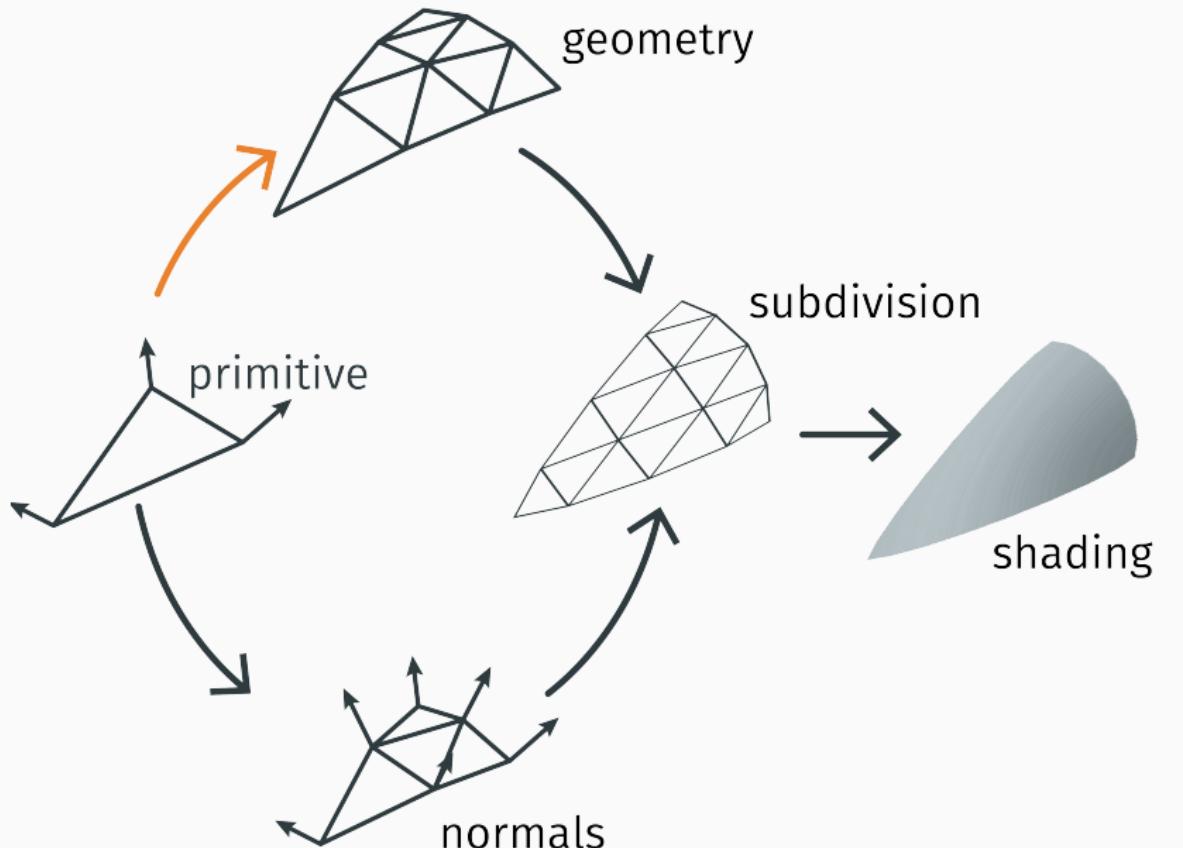
---

Point Normal triangles  
└ Single PN Triangle

2015-12-11

SINGLE PN TRIANGLE

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

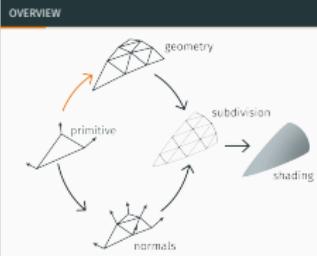


## Point Normal triangles

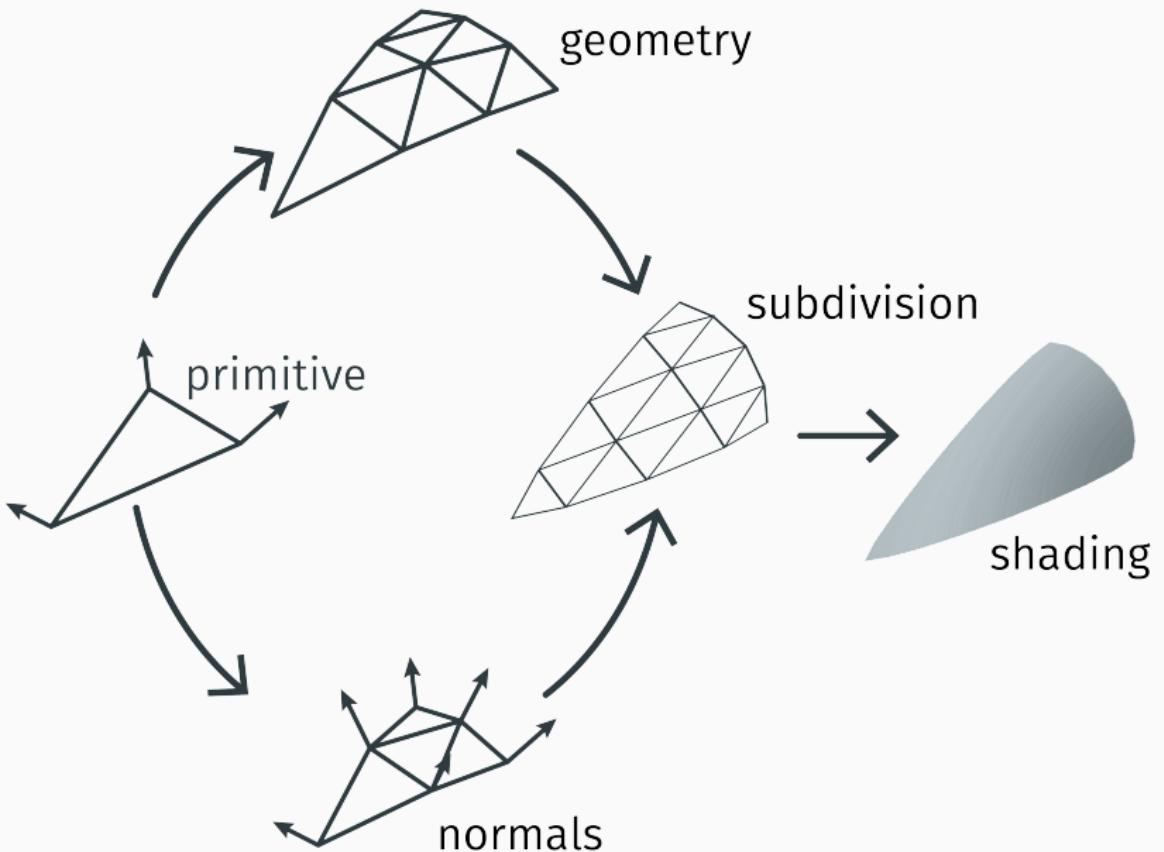
### Single PN Triangle

#### Overview

2015-12-11



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



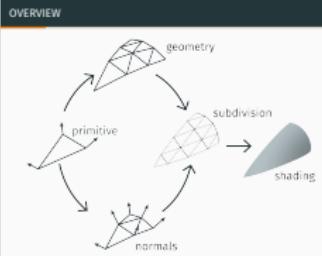
## Point Normal triangles

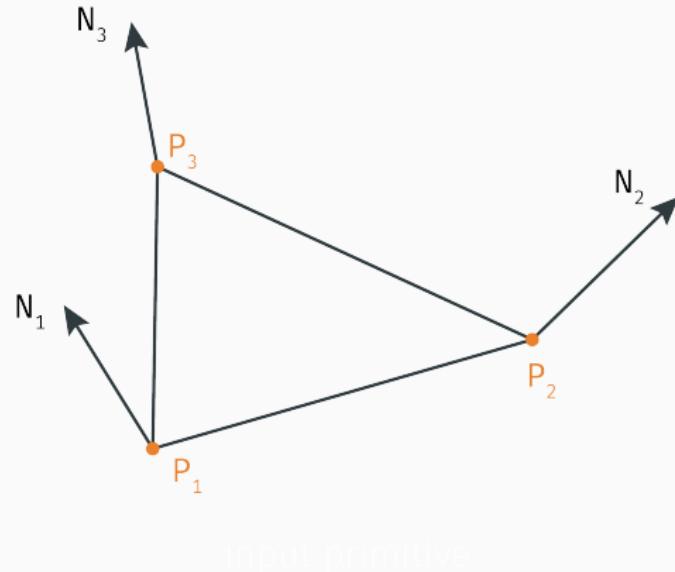
### └ Single PN Triangle

#### └ Overview

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

2015-12-11





Point Normal triangles

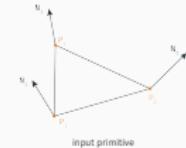
└ Single PN Triangle

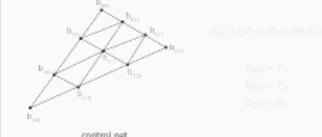
└ Geometry

2015-12-11

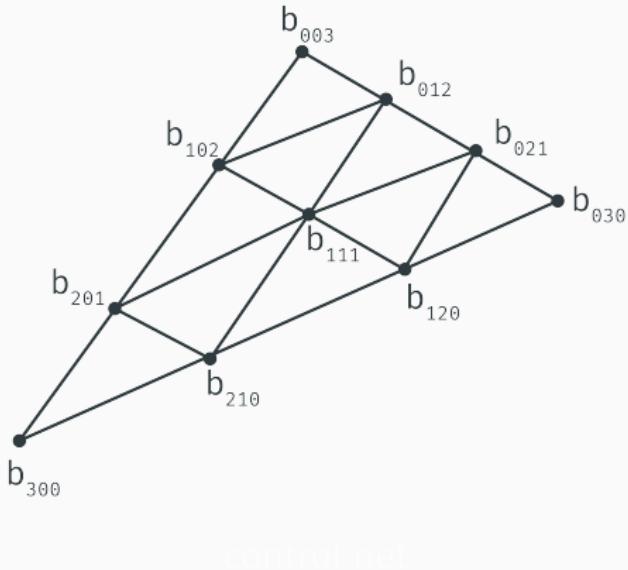
**[Laura]** This a standard triangle primitive, defined by its vertices and normals.

Focus on getting the different control primitives.





# GEOMETRY - VERTEX COEFFICIENTS



$$b_{ijk} = (P_1 + P_2 + P_3)/3$$

$$b_{300} = P_1,$$

$$b_{030} = P_2,$$

$$b_{003} = P_3$$

Point Normal triangles

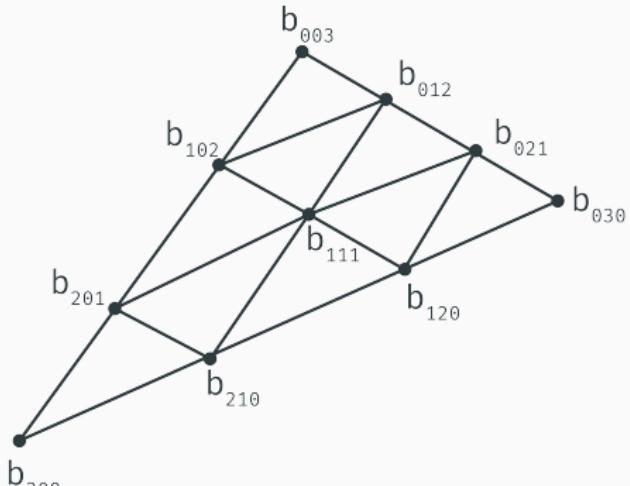
└ Single PN Triangle

└ Geometry - Vertex Coefficients

2015-12-11

**[Laura]** These are all the initial control point. Evenly divided on the triangle. -> formula

# GEOMETRY - VERTEX COEFFICIENTS



control net

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

$$b_{300} = P_1,$$

$$b_{030} = P_2,$$

$$b_{003} = P_3$$

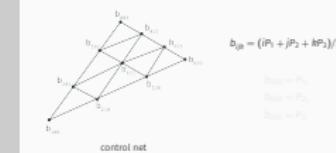
Point Normal triangles

└ Single PN Triangle

└ Geometry - Vertex Coefficients

2015-12-11

[Laura] Nice formula



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

$b_{ijk}$

$b_{i+1,j,k}$

$b_{i,j+1,k}$

$b_{i,j,k+1}$

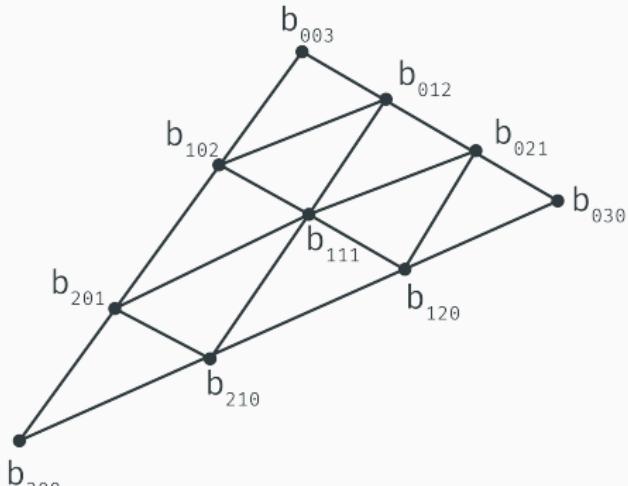
$b_{ijk}$

$b_{i+1,j,k}$

$b_{i,j+1,k}$

$b_{i,j,k+1}$

# GEOMETRY - VERTEX COEFFICIENTS



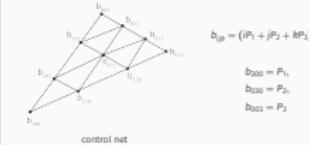
$$\begin{aligned}
 b_{ij0} &= P_1 \\
 b_{ij1} &= P_2 \\
 b_{ij2} &= P_3
 \end{aligned}$$

Point Normal triangles

└ Single PN Triangle

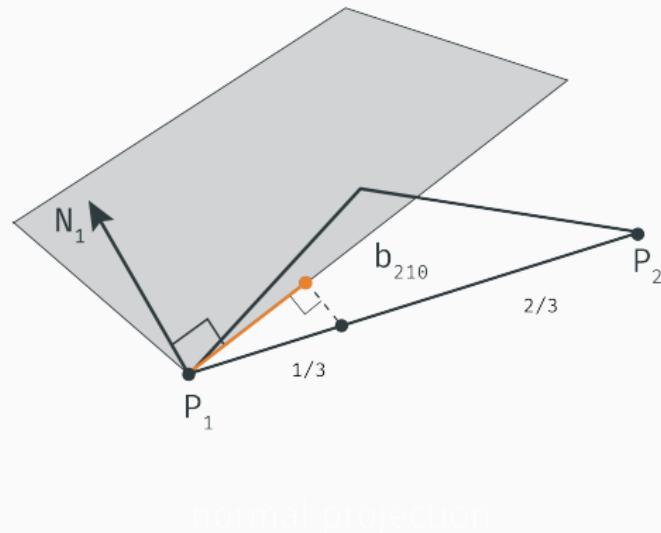
└ Geometry - Vertex Coefficients

2015-12-11



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

# GEOMETRY - TANGENT COEFFICIENTS



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

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

⋮

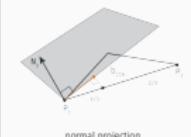
$$b_{201} = \frac{2P_1 + P_3 - w_{13}N_1}{3}$$

Point Normal triangles

└ Single PN Triangle

└ Geometry - Tangent Coefficients

2015-12-11



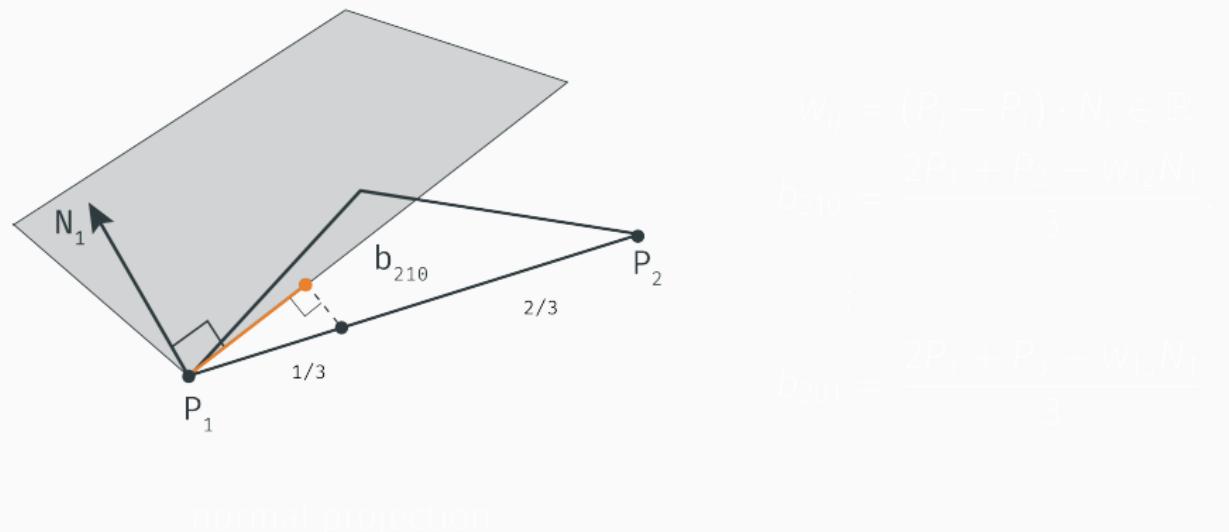
$$w_{ij} = (P_j - P_i) \cdot N_i \in \mathbb{R}$$

$$b_{ij0} = \frac{2P_i + P_j - w_{ij}N_i}{3}$$

$$b_{0ij} = \frac{2P_i + P_j - w_{ji}N_j}{3}$$

**[Laura]** How to get the tangent coefficient (the ones on the edge but now curvy)

# GEOMETRY - TANGENT COEFFICIENTS

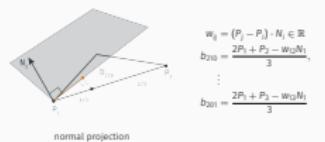


Point Normal triangles

└ Single PN Triangle

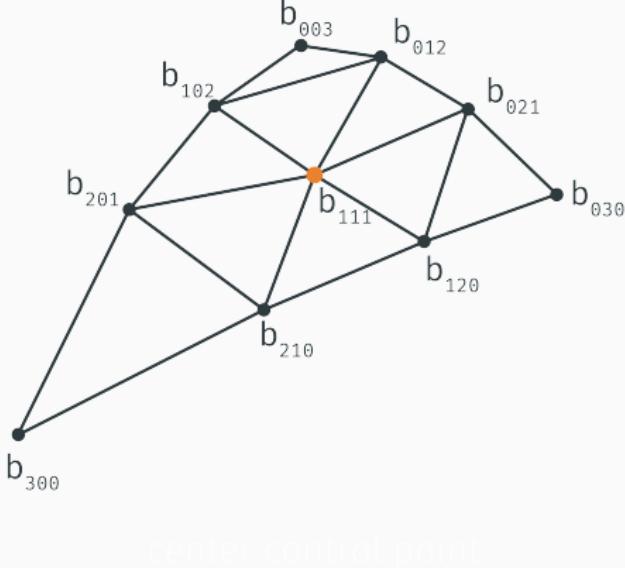
└ Geometry - Tangent Coefficients

2015-12-11



**[Laura]** Projection of the initial control points on the normal plane of a vertex.

# GEOMETRY - CENTER COEFFICIENT



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

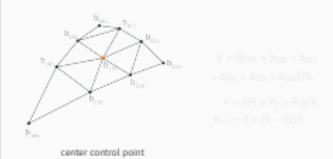
Point Normal triangles

└ Single PN Triangle

└ Geometry - Center Coefficient

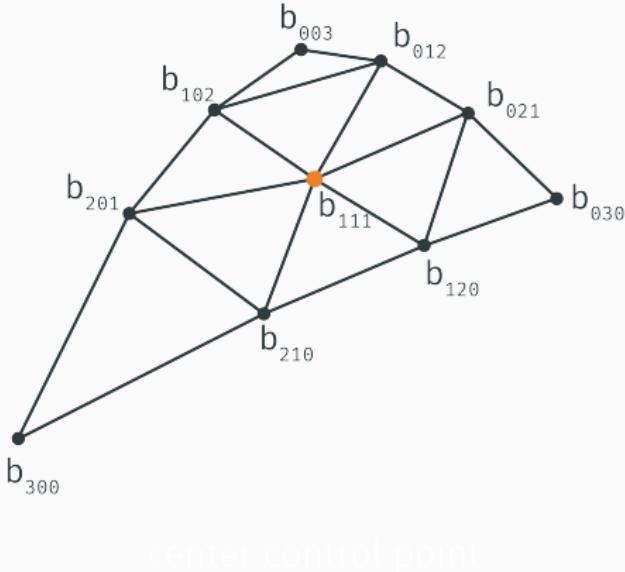
2015-12-11

GEOMETRY - CENTER COEFFICIENT



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

# GEOMETRY - CENTER COEFFICIENT



$$P = (b_{003} + b_{012} + b_{021} + b_{030} + b_{102} + b_{111} + b_{120} + b_{210}) / 8$$

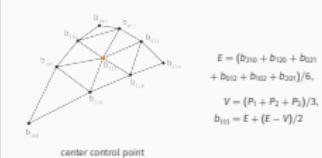
Point Normal triangles

└ Single PN Triangle

└ Geometry - Center Coefficient

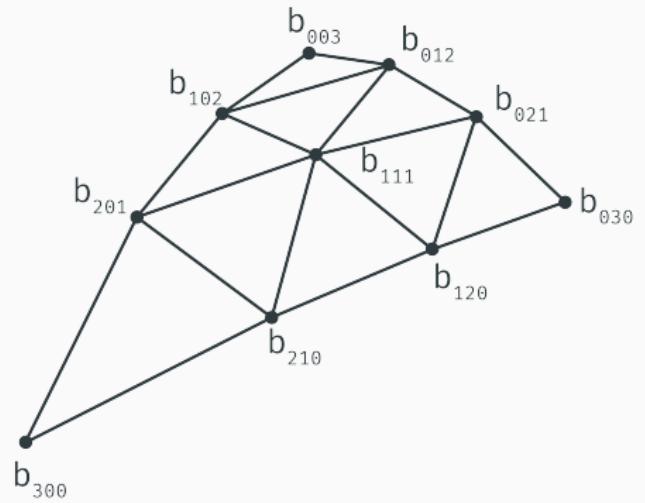
2015-12-11

GEOMETRY - CENTER COEFFICIENT



[Laura] Average of the tangent coefficients plus half the difference between the tangent and vertex coefficients. -> why?

# GEOMETRY - RESULT



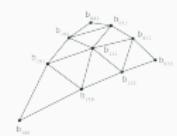
Point Normal triangles

└ Single PN Triangle

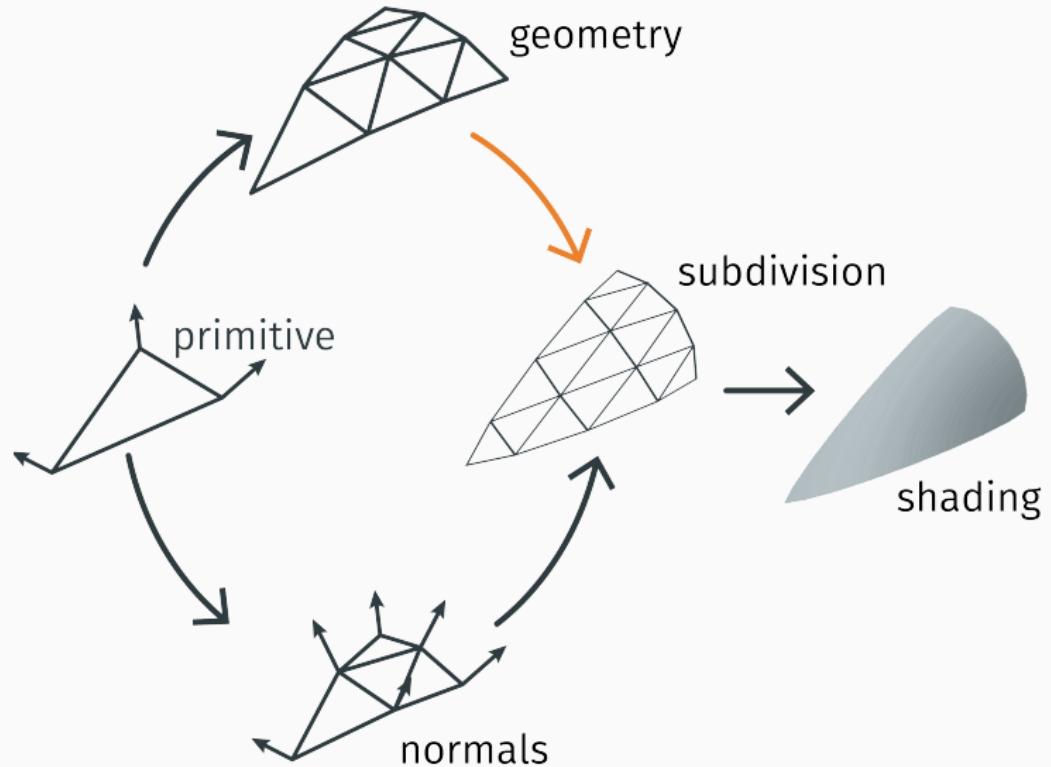
└ Geometry - Result

2015-12-11

[Laura] Results



# OVERVIEW

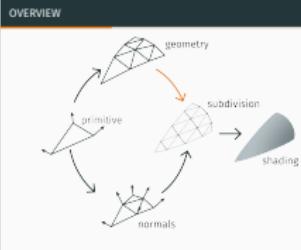


Point Normal triangles

└ Single PN Triangle

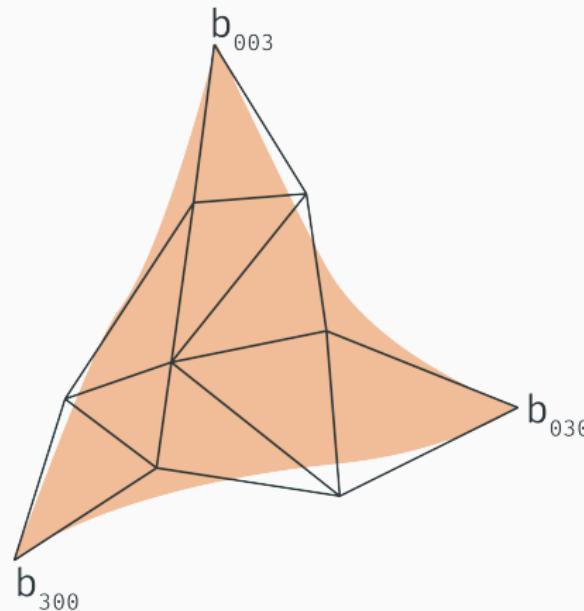
└ Overview

2015-12-11



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

# CUBIC BÉZIER PATCH



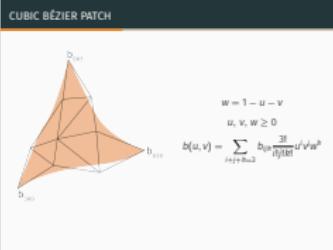
$$b(u, v) = \sum_{i+j+k=3} b_{ijk} \frac{u^i v^j w^k}{i! j! k!}$$

Point Normal triangles

└ Single PN Triangle

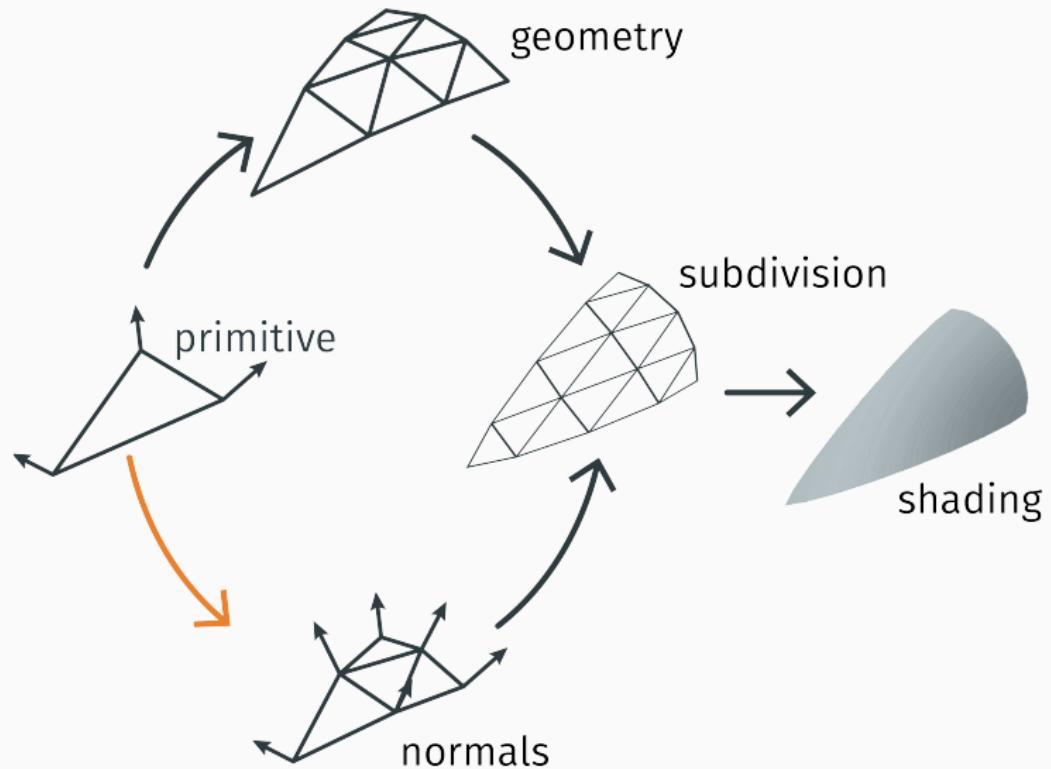
└ Cubic Bézier patch

2015-12-11



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

# OVERVIEW



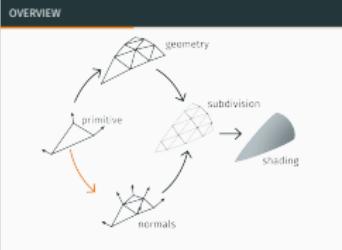
Point Normal triangles

└ Single PN Triangle

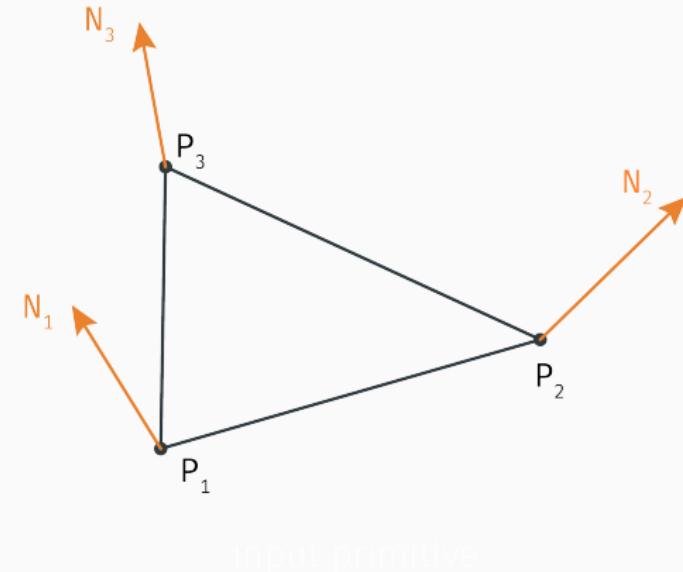
└ Overview

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

2015-12-11



# NORMALS



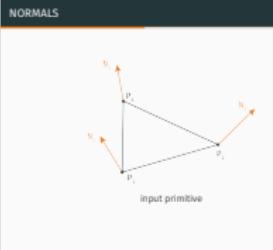
Point Normal triangles

└ Single PN Triangle

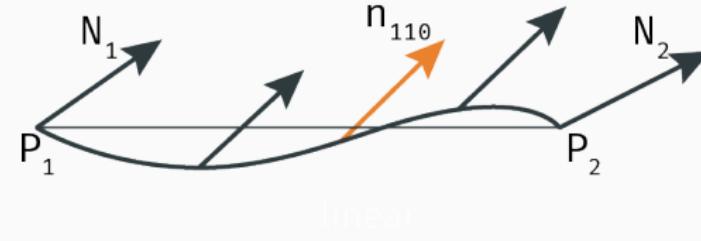
└ Normals

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

2015-12-11



# NORMALS - THEORY



quadratic

Point Normal triangles  
└ Single PN Triangle

└ Normals - theory

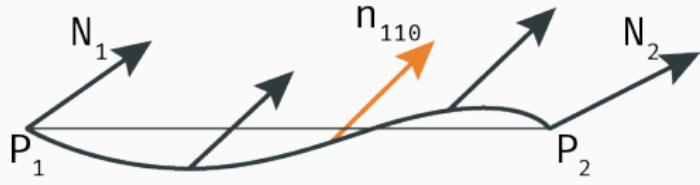
2015-12-11

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

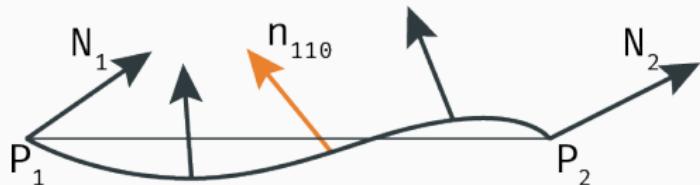
NORMALS - THEORY



# NORMALS - THEORY



linear



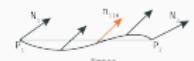
quadratic

Point Normal triangles  
└ Single PN Triangle

└ Normals - theory

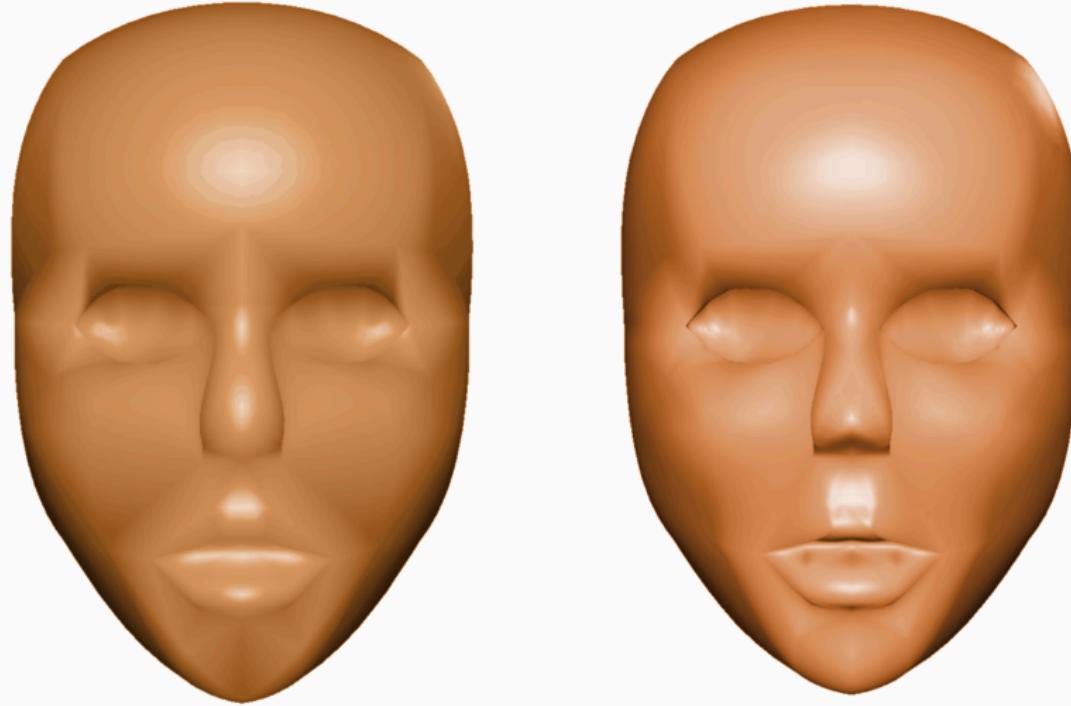
2015-12-11

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

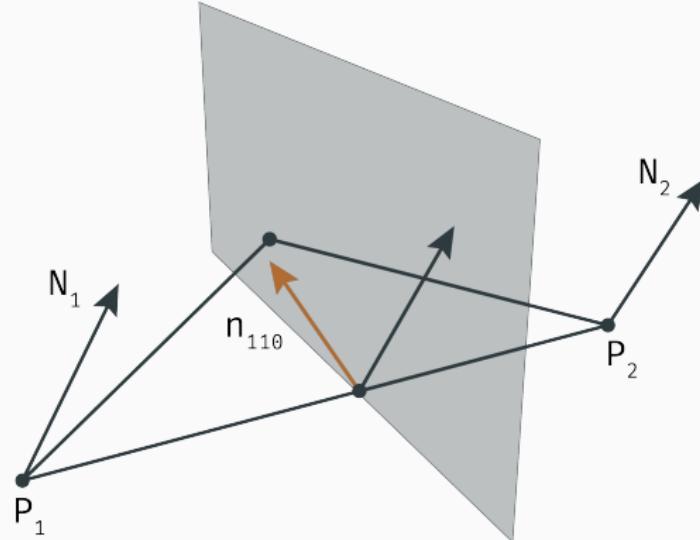
[Rick] Look how pretty.

2015-12-11

NORMALS - EXAMPLE



# NORMALS - THEORY



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

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

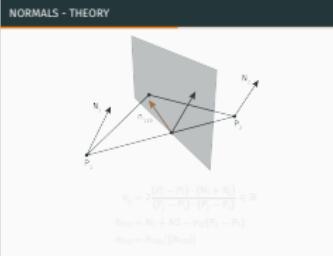
$$n_{110} = h_{110} / \|h_{110}\|$$

Point Normal triangles

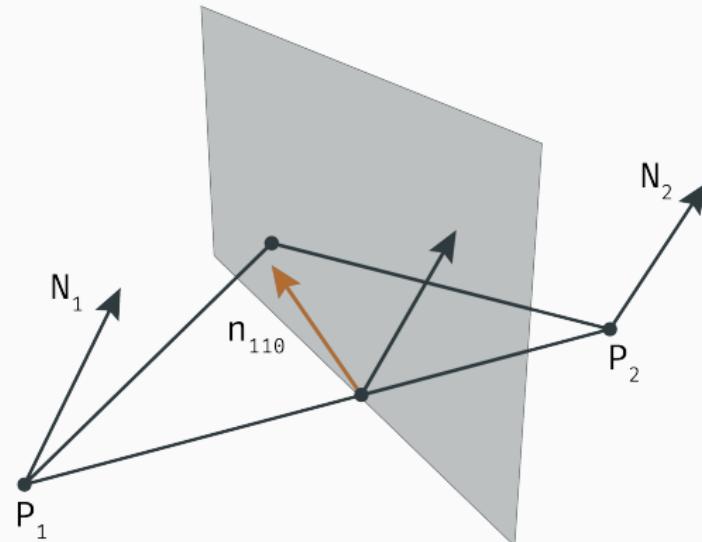
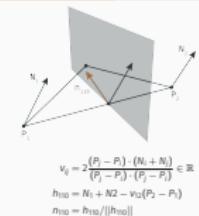
└ Single PN Triangle

└ Normals - theory

2015-12-11



[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.



$$\begin{aligned} v_{ij} &= \frac{(P_j - P_i) \cdot (N_i + N_j)}{\|(P_j - P_i)\| \cdot \|(N_i + N_j)\|} \in \mathbb{R} \\ h_{110} &= N_1 + N_2 - v_{ij}(P_j - P_i) \\ n_{110} &= h_{110} / \|h_{110}\| \end{aligned}$$

# NORMALS - THEORY

Point Normal triangles

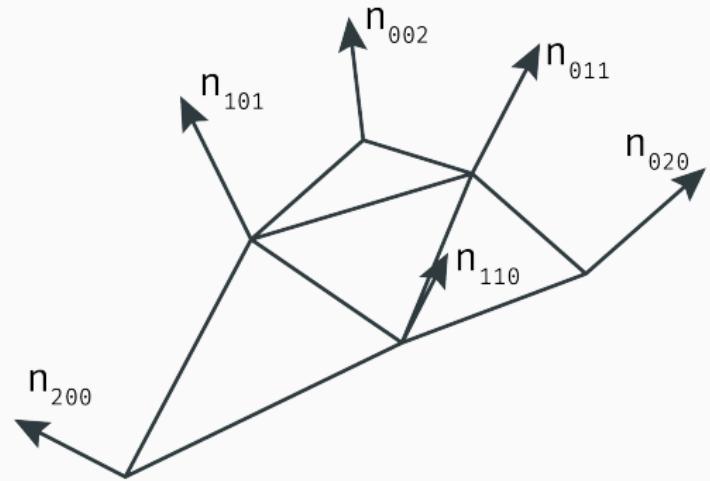
└ Single PN Triangle

└ Normals - theory

2015-12-11

[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

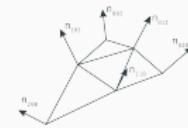


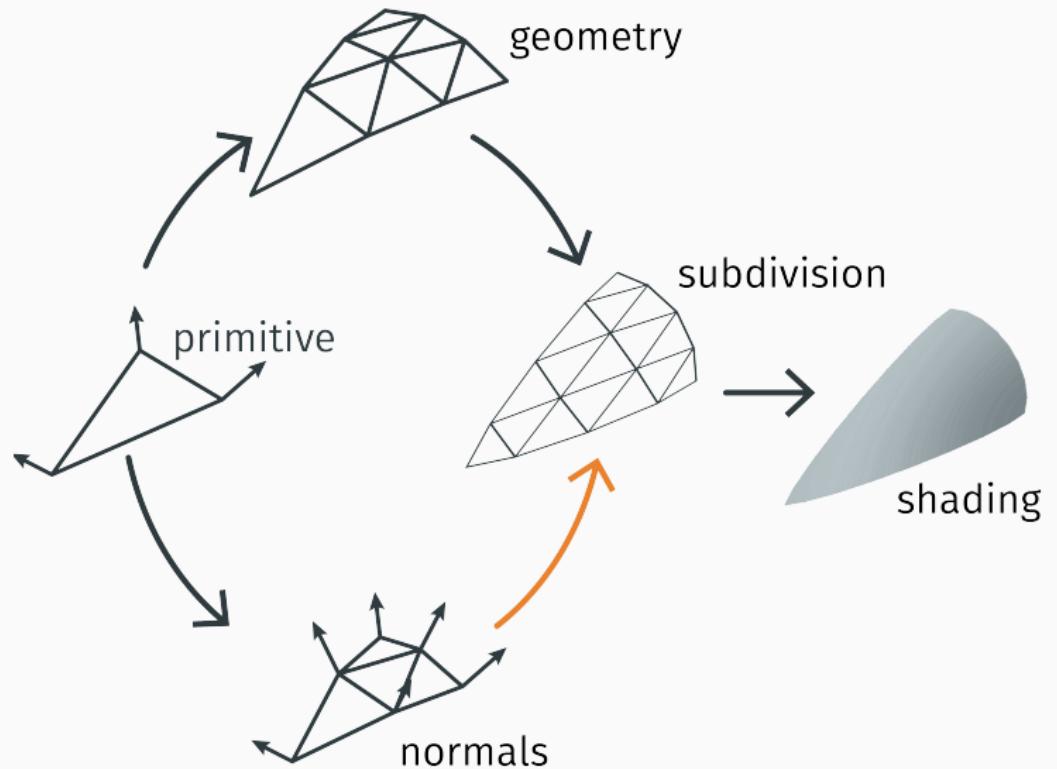
Point Normal triangles

└ Single PN Triangle

└ Normals - result

[Rick] Result





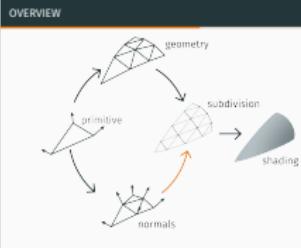
## Point Normal triangles

## └ Single PN Triangle

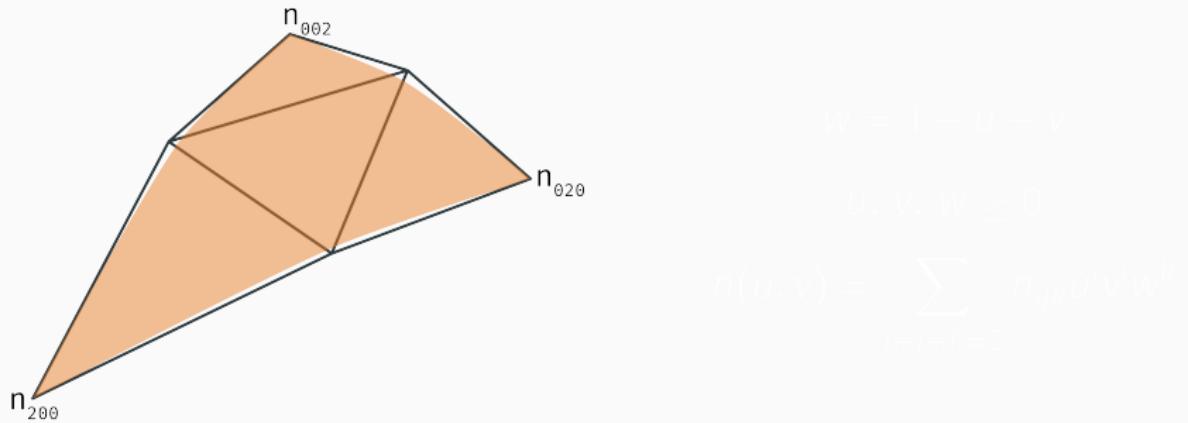
## └ Overview

2015-12-11

[Laura] 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.



# QUADRATIC PATCH



$$w = 1 - u - v$$

$$u, v, w \geq 0$$

$$\mathbf{n}(u, v) = \sum_{i+j+k=0}^2 n_{ijk} u^i v^j w^k$$

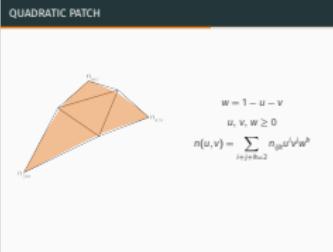
Point Normal triangles

└ Single PN Triangle

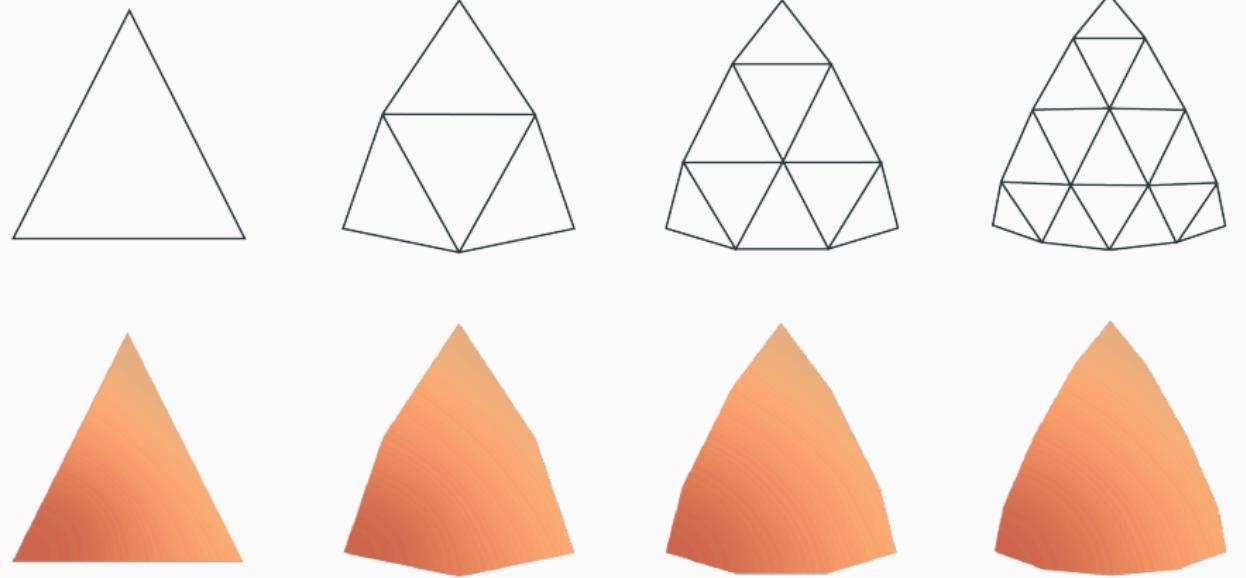
└ Quadratic Patch

2015-12-11

[Laura]  $u, v$  and  $w$  are convex combinations



# LEVEL OF DETAIL



Point Normal triangles

└ Single PN Triangle

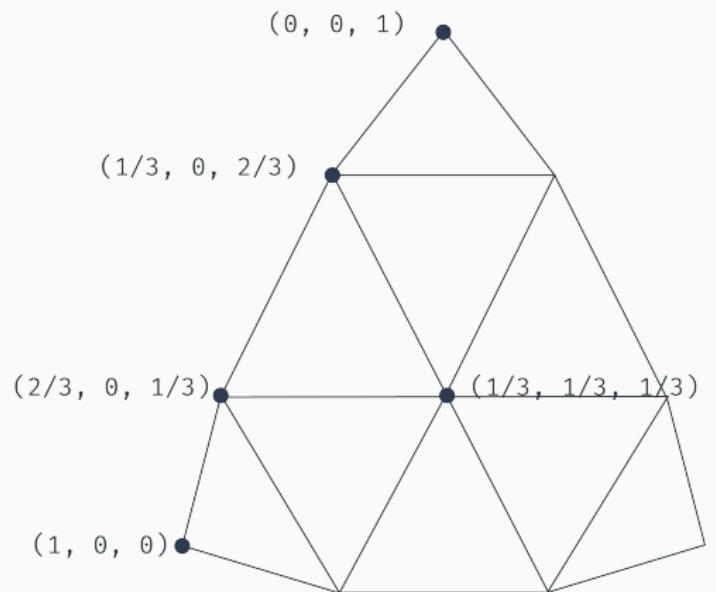
└ Level Of Detail

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



2015-12-11

# LEVEL OF DETAIL

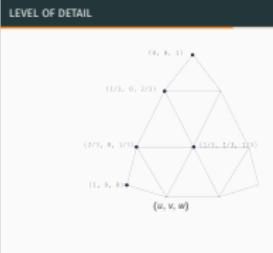


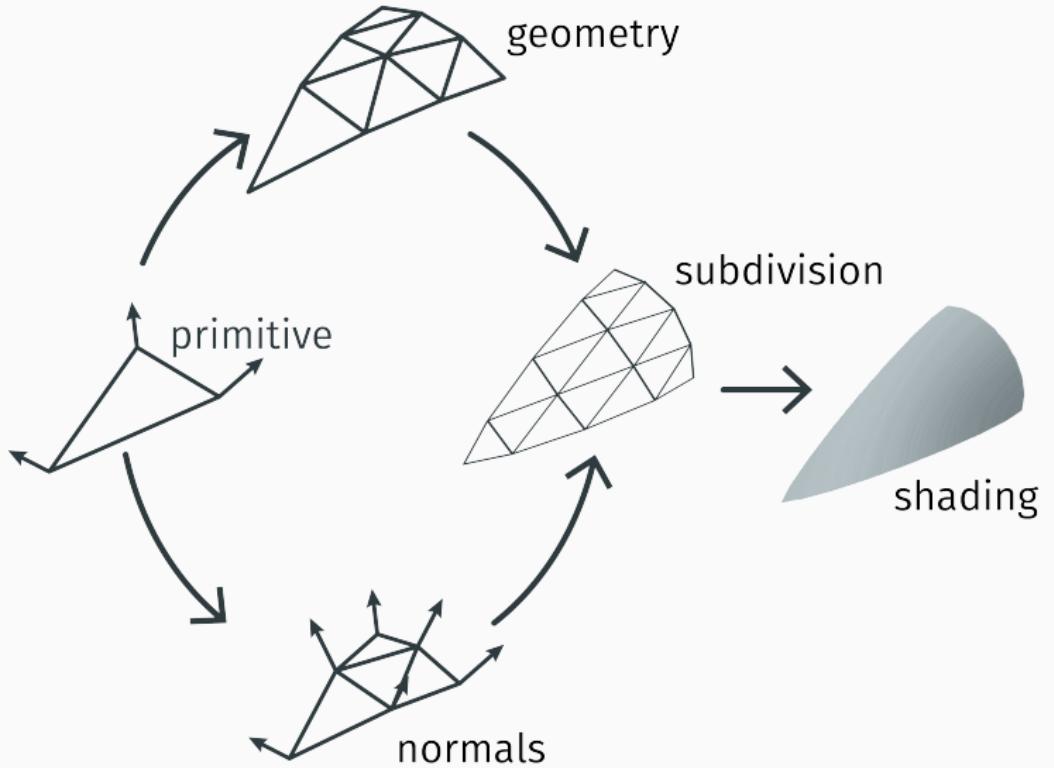
Point Normal triangles  
└ Single PN Triangle

2015-12-11

└ Level Of Detail

[Laura] blaat

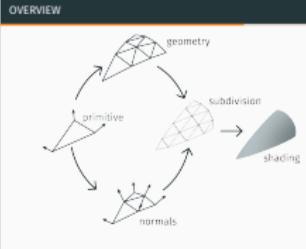




2015-12-11

Point Normal triangles

- └ Single PN Triangle
- └ Overview



[Laura] Shading out of the scope of this presentation [Laura] Why quadartic patch for normals, why cubic patch for geometry

# A TRIANGLE MESH

---

Point Normal triangles  
└ A Triangle Mesh

2015-12-11

A TRIANGLE MESH

[rick] blaat

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

---

<sup>1</sup>Vlachos et al.

## Point Normal triangles

### └ A Triangle Mesh

### └ Properties

2015-12-11

**[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.”<sup>1</sup>

<sup>1</sup>Marchis et al.

# CONTINUITY

PN triangles have<sup>2</sup>

- $C^1$  continuity in the vertex points
- $C^0$  continuity along the edges
- $C^\infty$  everywhere else

---

<sup>2</sup>Jiao and Alexander

## Point Normal triangles

### └ A Triangle Mesh

### └ Continuity

2015-12-11

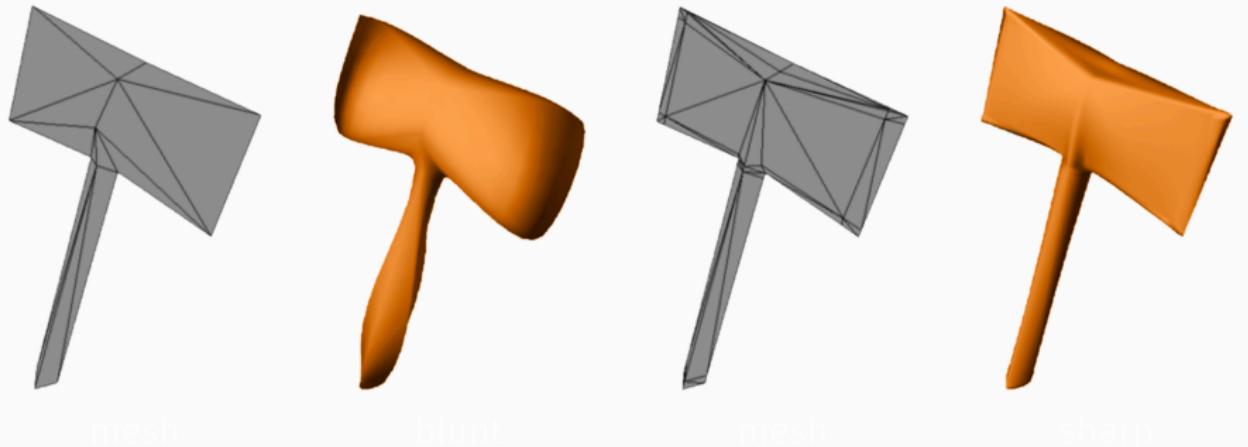
[Rick] Continuity  $C^0$  is important -> no gaps. Higher is better because this gives a more smooth result.

PN triangles have<sup>2</sup>

- $C^1$  continuity in the vertex points
- $C^0$  continuity along the edges
- $C^\infty$  everywhere else

<sup>2</sup>Jiao and Alexander

# SHARP EDGES



Point Normal triangles

└ A Triangle Mesh

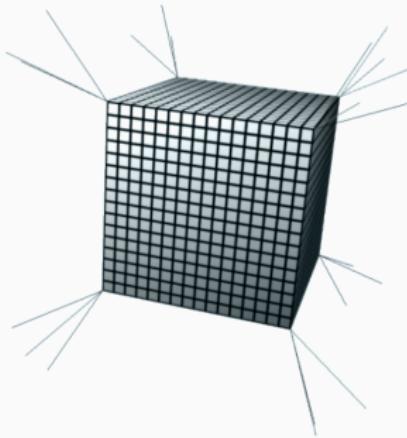
└ Sharp Edges

2015-12-11



[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

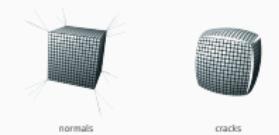
Point Normal triangles

└ A Triangle Mesh

└ Separate Normals

2015-12-11

SEPARATE NORMALS



## GRAPHICS PIPELINE

---

Point Normal triangles  
└ Graphics Pipeline

2015-12-11

GRAPHICS PIPELINE

---

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

# HARDWARE - PIPELINES



2015

2015

Point Normal triangles

└ Graphics Pipeline

└ Hardware - Pipelines

2015-12-11

HARDWARE - PIPELINES



2001

# HARDWARE - PIPELINES

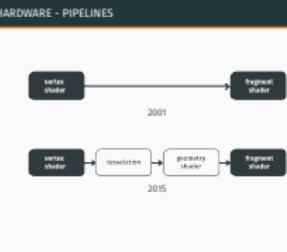


Point Normal triangles

└ Graphics Pipeline

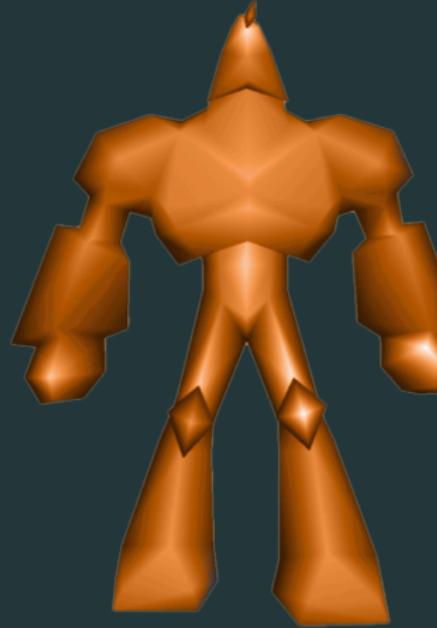
└ Hardware - Pipelines

2015-12-11

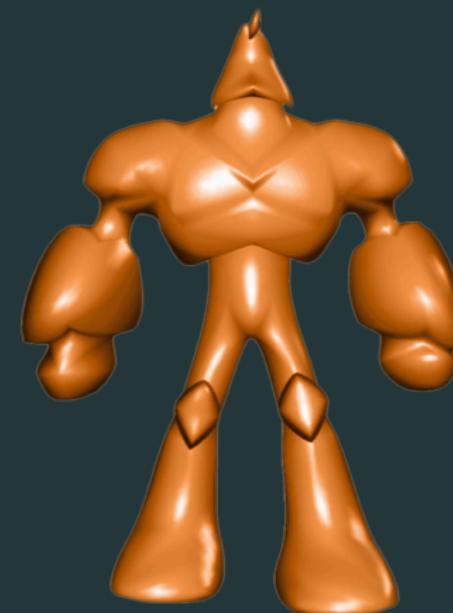


[Laura] 2015 we have OpenGL 4.5 with more programmable shaders and the whole process can be done on the GPU. Since PN triangles only uses the primitive, no neighboring primitives, easy in shaders.

# QUESTIONS?



TRIANGLES



PN TRIANGLES

Point Normal triangles  
└ Graphics Pipeline

2015-12-11

Laura blaat



## Point Normal triangles

### └ Graphics Pipeline

### └ References

2015-12-11

- 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.