POINT NORMAL TRIANGLES

Rick van Veen Laura Baakman

December 14, 2015

Advanced Computer Graphics

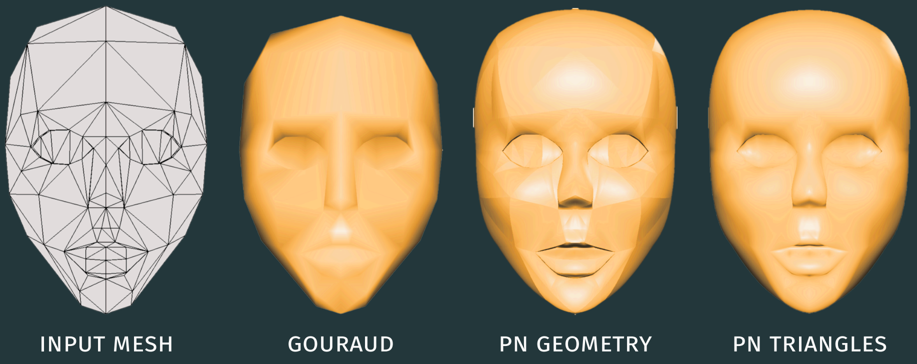
2015-12-12

Point Normal triangles

POINT NORMAL TRIANGLES

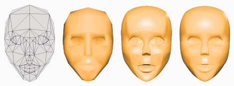
Rick van Veen Laura Baakman
December 14, 2015
Advanced Computer Graphics

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



2015-12-12

Point Normal triangles



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

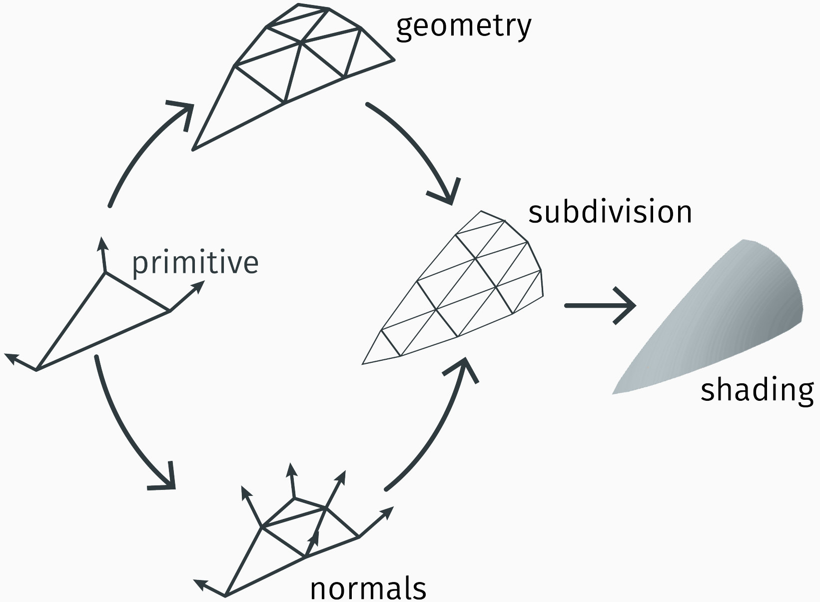
SINGLE PN TRIANGLE

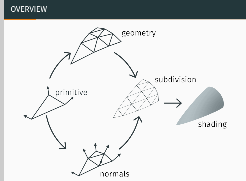
SINGLE PN TRIANGLE

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

Overview on the next slide

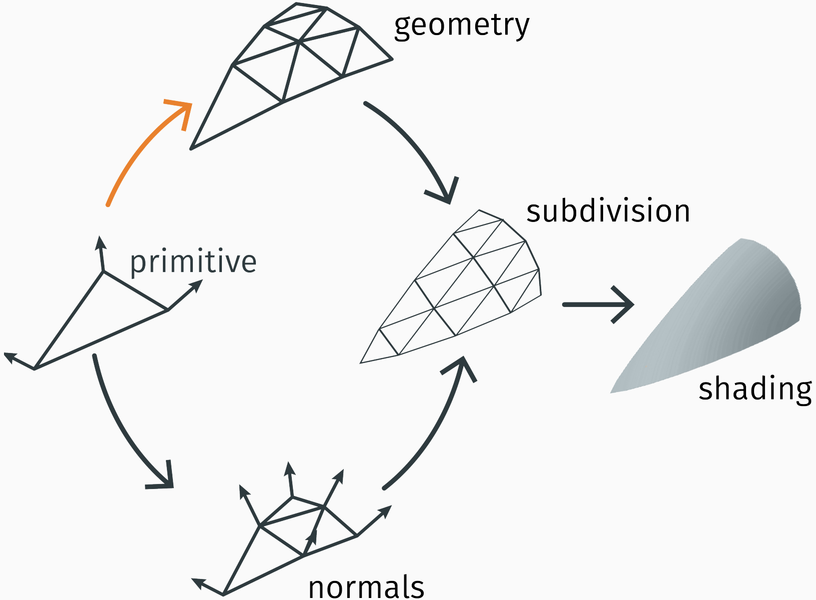
OVERVIEW

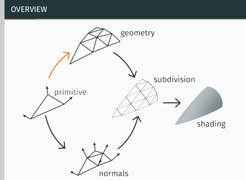




[Laura] PN triangle is defined by geometry and normal component.

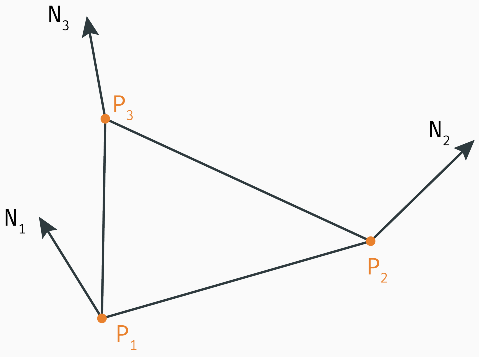
OVERVIEW



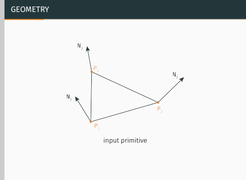


Laura From geometric component of input primitive to geometric component of PN triangle

GEOMETRY



input primitive

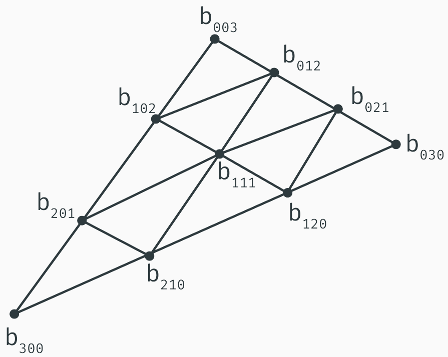


[Laura] This a standard triangle primitive, defined by its vertices and normals.

Focus on getting the different control primitives.

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

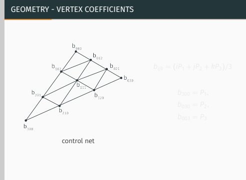
GEOMETRY - VERTEX COEFFICIENTS



control net

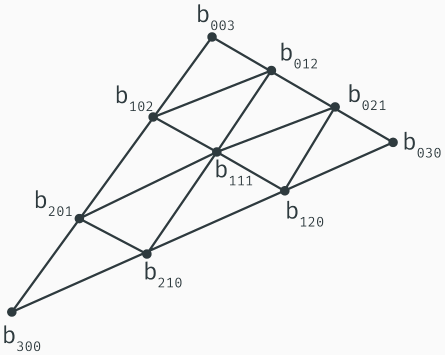
$$b_{ijk} = (iP_1 + jP_2 + kP_3)/3$$

$$b_{300} = P_1,$$
$$b_{030} = P_2,$$
$$b_{003} = P_3$$



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

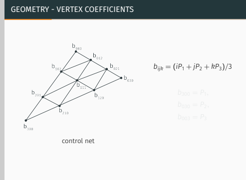
GEOMETRY - VERTEX COEFFICIENTS



control net

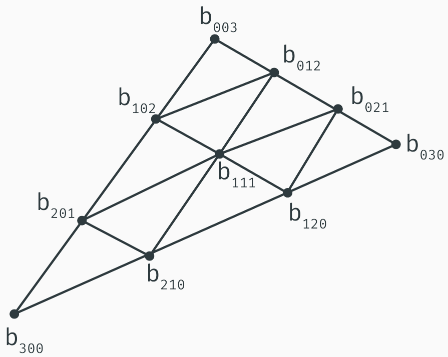
$$b_{ijk} = (iP_1 + jP_2 + kP_3)/3$$

$$b_{300} = P_1,$$
$$b_{030} = P_2,$$
$$b_{003} = P_3$$



[Laura] Nice formula

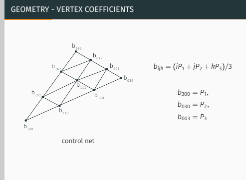
GEOMETRY - VERTEX COEFFICIENTS



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

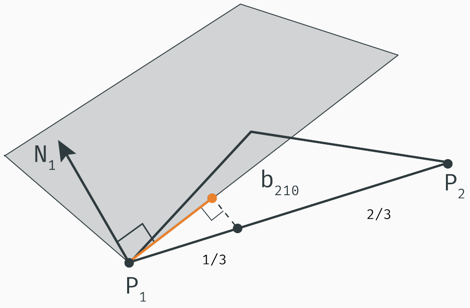
$$b_{300} = P_1,$$
$$b_{030} = P_2,$$
$$b_{003} = P_3$$

control net



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

GEOMETRY - TANGENT COEFFICIENTS



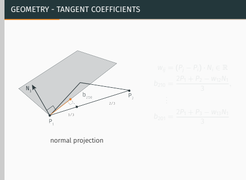
normal projection

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

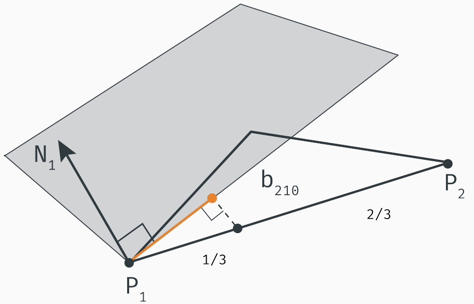
$$b_{210} = \frac{2P_1 + P_2 - w_{12}N_1}{3},$$

$$\vdots$$

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



[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.



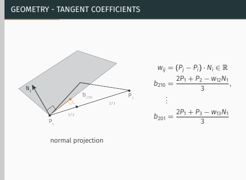
$$w_{ij} = (P_j - P_i) \cdot N_i \in \mathbb{R}$$

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

$$\vdots$$

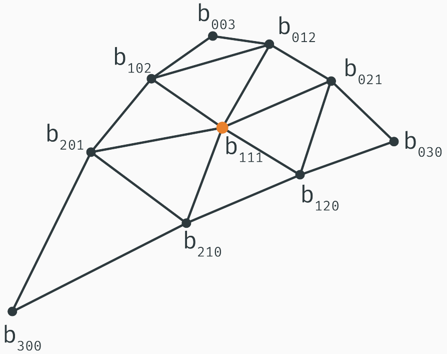
$$b_{201} = \frac{2P_1 + P_3 - w_{13}N_1}{3}$$

normal projection



[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 - CENTER COEFFICIENT

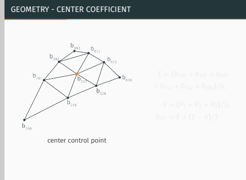


$$E = (b_{210} + b_{120} + b_{021} + b_{012} + b_{102} + b_{201})/6,$$

$$V = (P_1 + P_2 + P_3)/3,$$

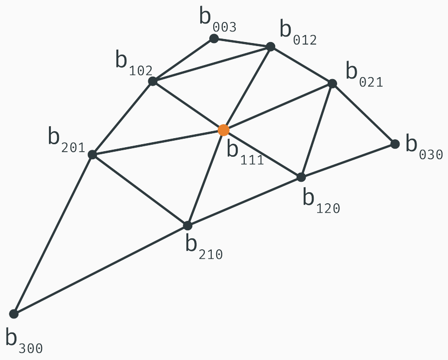
$$b_{111} = E + (E - V)/2$$

center control point



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

GEOMETRY - CENTER COEFFICIENT

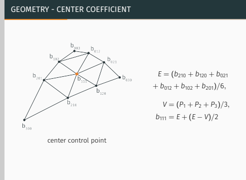


$$E = (b_{210} + b_{120} + b_{021} + b_{012} + b_{102} + b_{201})/6,$$

$$V = (P_1 + P_2 + P_3)/3,$$

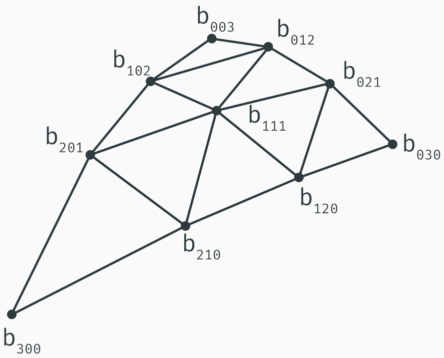
$$b_{111} = E + (E - V)/2$$

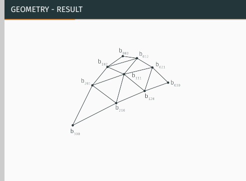
center control point



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

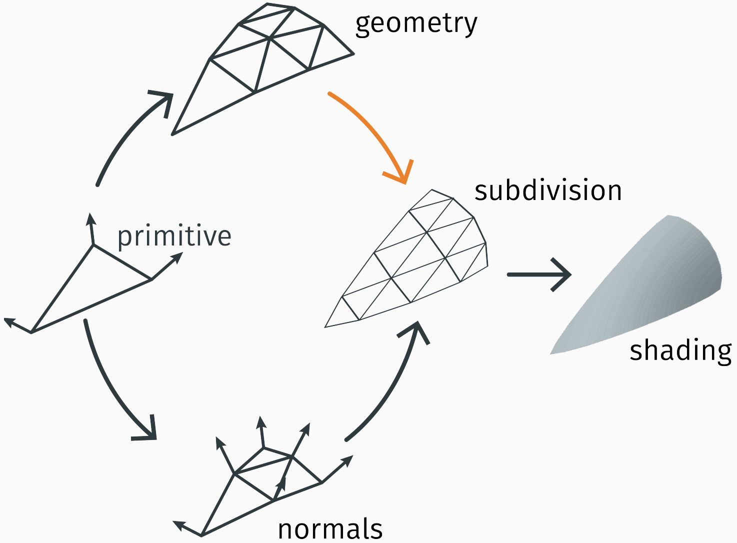
GEOMETRY - RESULT

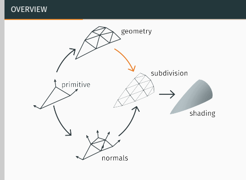




[Laura] Results

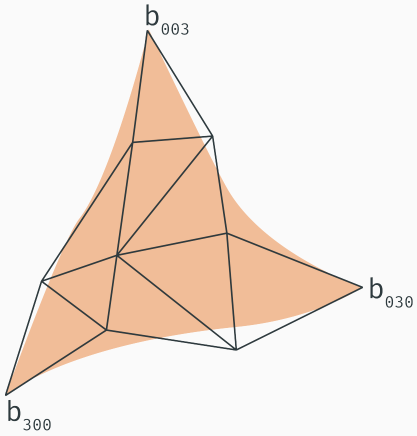
OVERVIEW





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

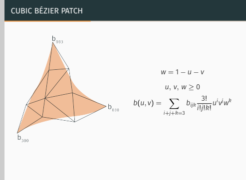
CUBIC BÉZIER PATCH



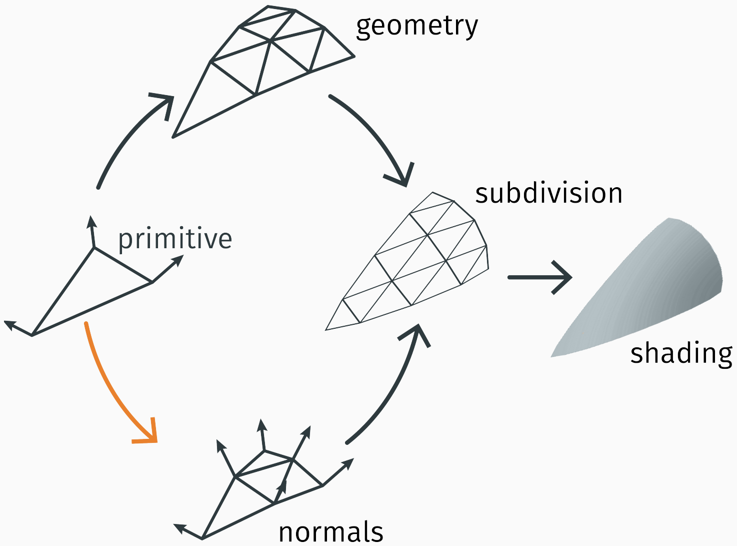
$$w = 1 - u - v$$

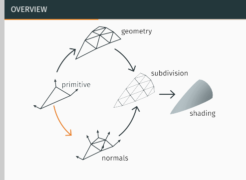
$$u, v, w \geq 0$$

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



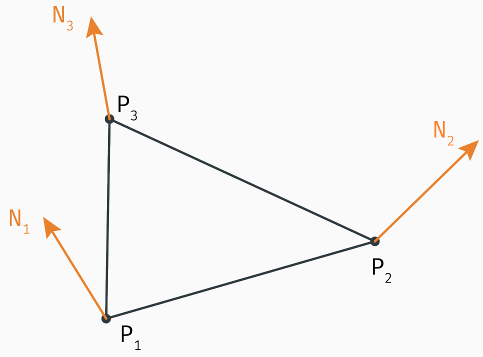
[Rick] Very nice formula with a nice picture. Mention **Barycentric coordinates**u, v, w are a convex combination



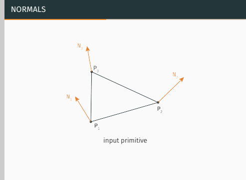


[Rick] From the primitive normals the the PN triangle normals

NORMALS

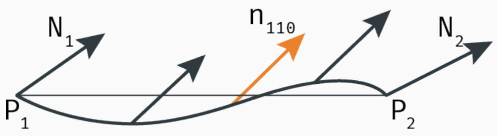


input primitive



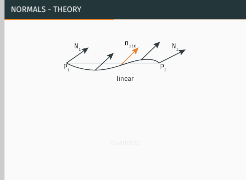
[Rick] Recap input primitive and with emphasis on the normals.

NORMALS - THEORY

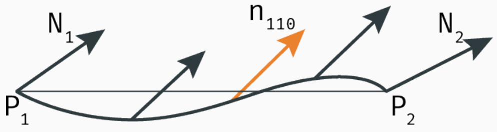


linear

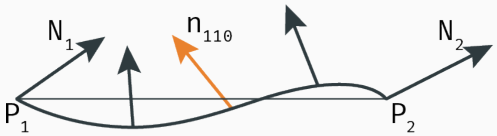
quadratic



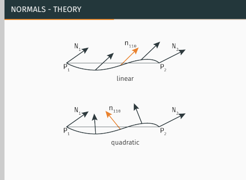
[Rick] Stress that there is a need to capture the cubic bezier curve (inflection points) and that this cannot be



linear

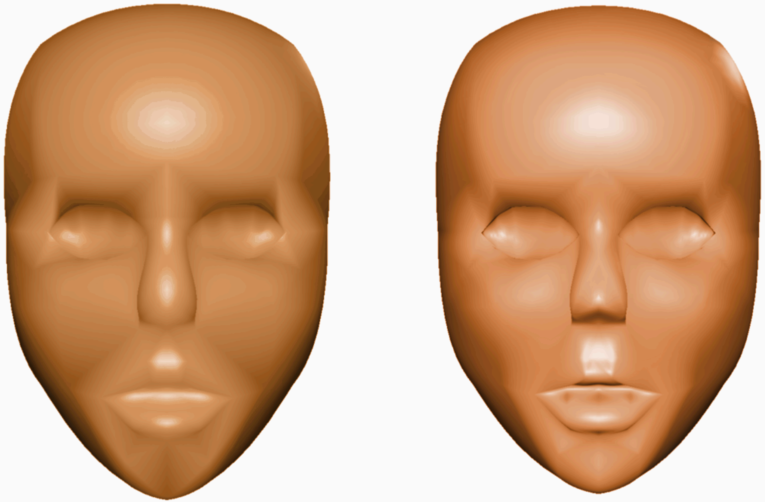


quadratic

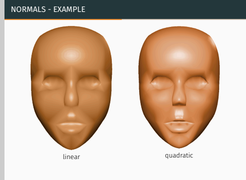


[Rick] Quadratic does capture inflection points. Trade off between performance and result (maybe?)

NORMALS - EXAMPLE

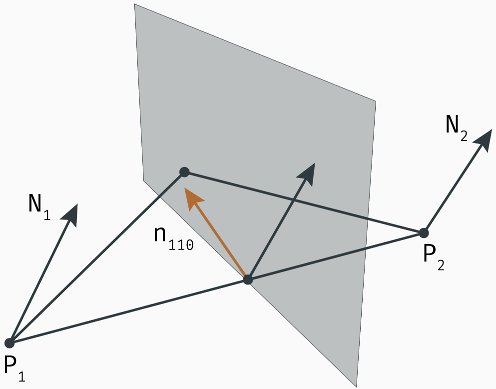


linear quadratic



[Rick] Look how pretty.

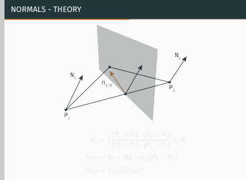
NORMALS - THEORY



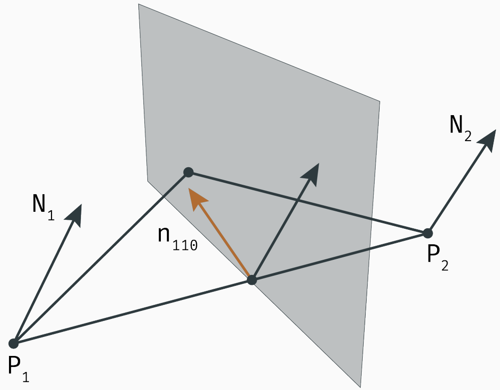
$$v_{ij} = 2\frac{(P_j - P_i)\cdot(N_i + N_j)}{(P_j - P_i)\cdot(P_j - P_i)} \in \mathbb{R}$$

$$h_{110} = N_1 + N2 - v_{12}(P_2 - P_1)$$

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



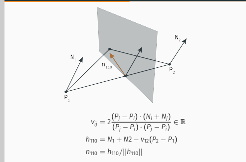
[Rick] Formula in words: reflect the averaged normal (average of N1 and N2) on the plane orthogonal/perpendicular the the edge at the mid point.



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

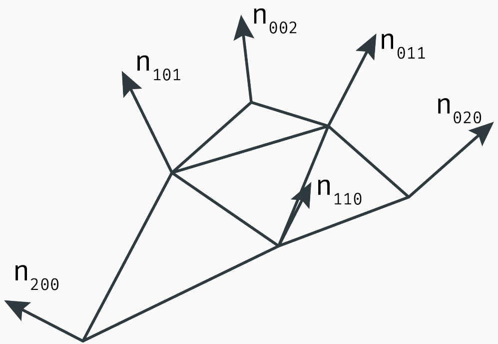
$$h_{110} = N_1 + N2 - v_{12}(P_2 - P_1)$$

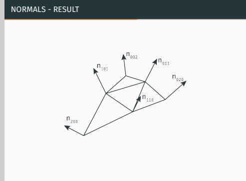
$$n_{110} = h_{110}/||h_{110}||$$



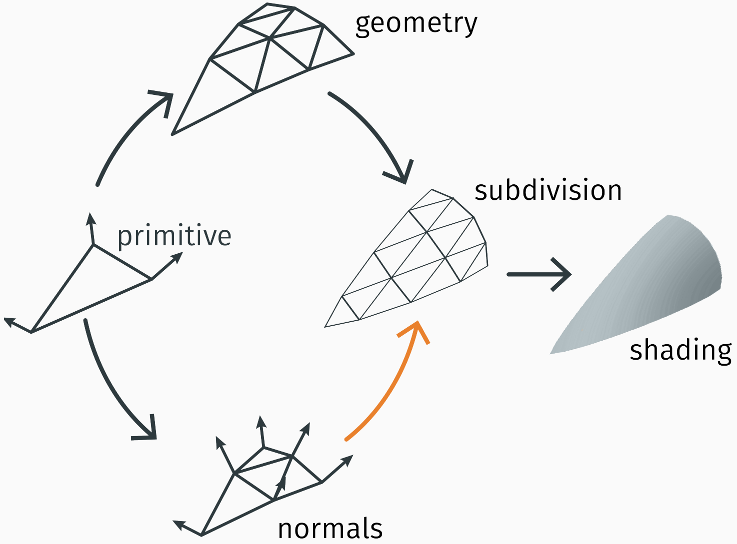
[Rick] Formula in words: reflect the averaged normal (average of N1 and N2) on the plane orthogonal/perpendicular the the edge at the mid point.

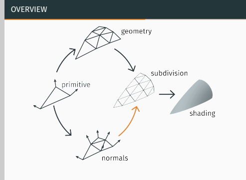
NORMALS - RESULT





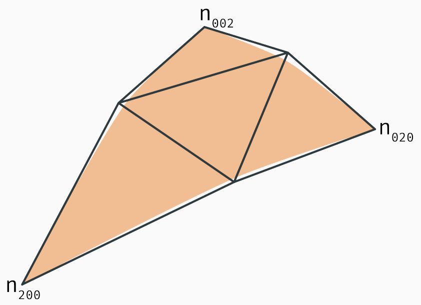
[Rick] Result





laura

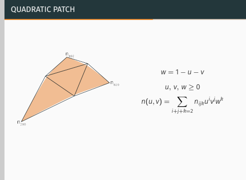
QUADRATIC PATCH



$$w = 1 - u - v$$

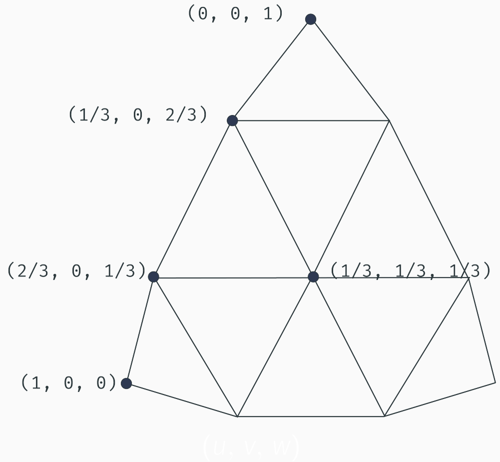
$$u, v, w \geq 0$$

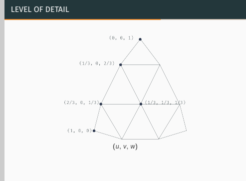
$$n(u, v) = \sum_{i+j+k=2} n_{ijk} u^i v^j w^k$$



[Laura] u, v and w are an convex combination

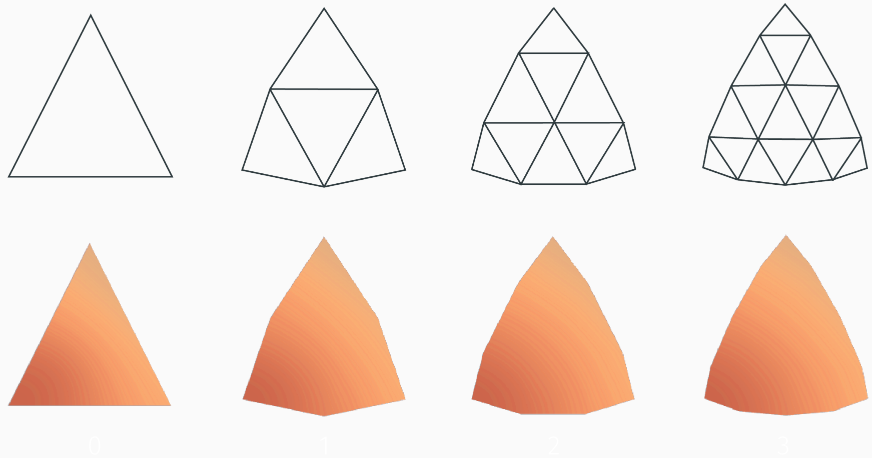
LEVEL OF DETAIL

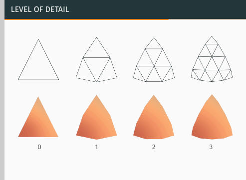




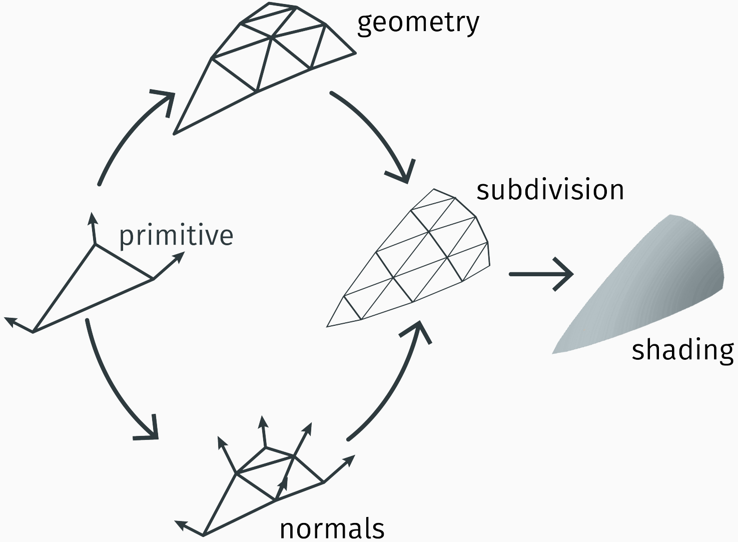
[Laura] LOD = 2

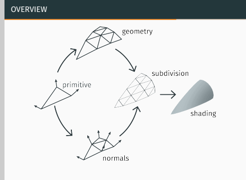
LEVEL OF DETAIL





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





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.Simplicity v.s. modeling range
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

A TRIANGLE MESH

[rick] blaat

PROPERTIES

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

[1] Vlachos et al.

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

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

CONTINUITY

PN triangles have:[2]

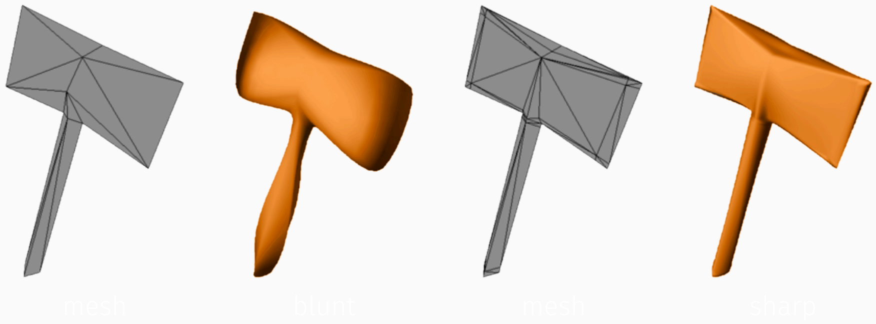
- C^1 continuity in the vertex points
- C^0 continuity along the edges
- C^∞ everywhere else

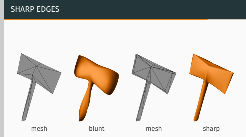
[2] Jiao and Alexander

CONTINUITY
PN triangles have:[2]
- C^1 continuity in the vertex points
- C^0 continuity along the edges
- C^∞ everywhere else
[2] Jiao and Alexander

[Rick] Continuity C0 is important -> no gaps. Higher is better because this gives a more smooth result.

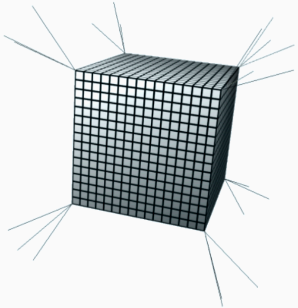
SHARP EDGES



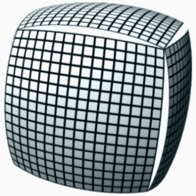


[Rick] Curved triangles do not always give the preferred results -> sharp edges. Solution is to insert more triangles at the sharp edges -> model needs to be changed :(

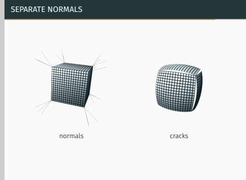
SEPARATE NORMALS



normals



cracks



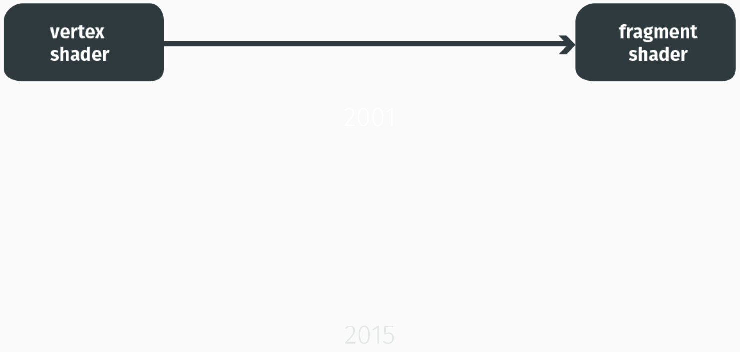
[Rick] Beyond the scope of the paper extension exist to overcome the problem what you have when combining multiple meshes. Story about shared vertices.

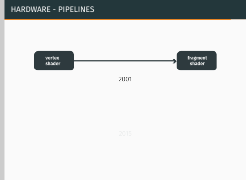
GRAPHICS PIPELINE

GRAPHICS PIPELINE

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

HARDWARE - PIPELINES

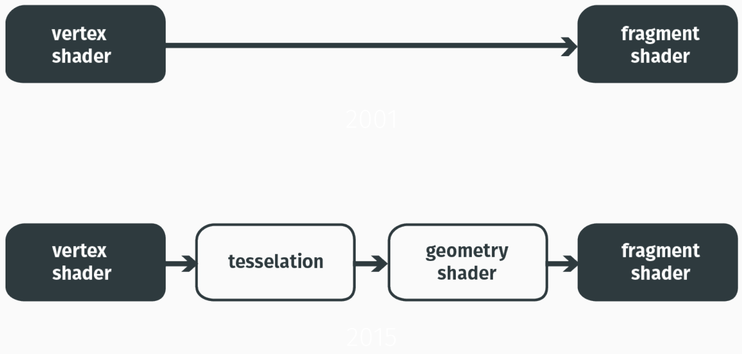


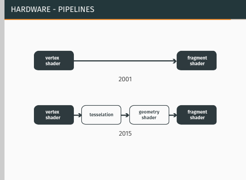


[Laura] Great part of the paper stresses the point that it can easily be implemented as a preprocessing step (CPU.

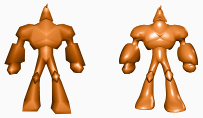
2001 pipeline (OpenGl 1.3)

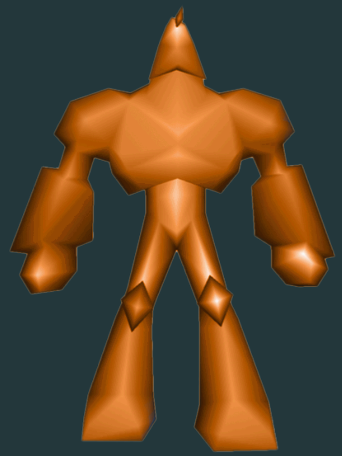
HARDWARE - PIPELINES



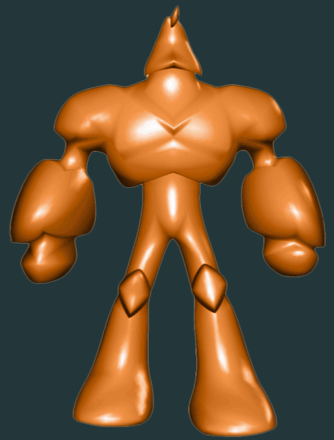


[Laura] 2015 we have OpenGL 4.5 with more programmable shaders and the whole process can be done on the GPU. Since PN triangles only usses the primitive, no neighboring primitives, easy in shaders.



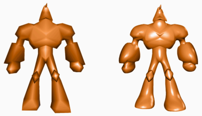


TRIANGLES

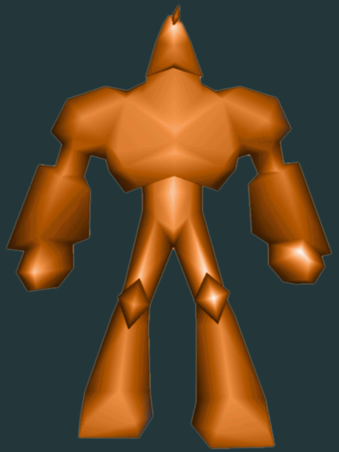


PN TRIANGLES

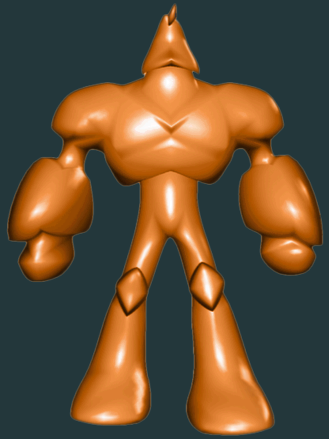
Laura Why PN triangles are not suited to rendering for CAD: On real objects, the normal field is determined by the geometry and thus fixed. If you invent fake normals the rendering looks better but the user gets an unpleasant surprise when the product is actually manufactured.



QUESTIONS?



TRIANGLES



PN TRIANGLES

REFERENCES

- 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.

REFERENCES

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.