POINT NORMAL TRIANGLES

Rick van Veen Laura Baakman

December 14, 2015

Advanced Computer Graphics

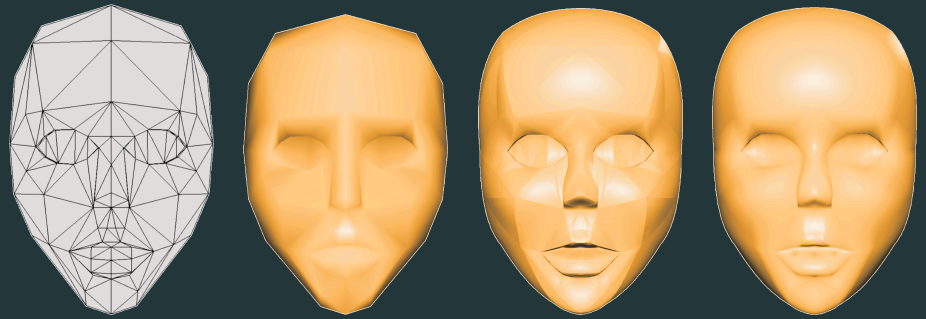
2015-12-09

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.



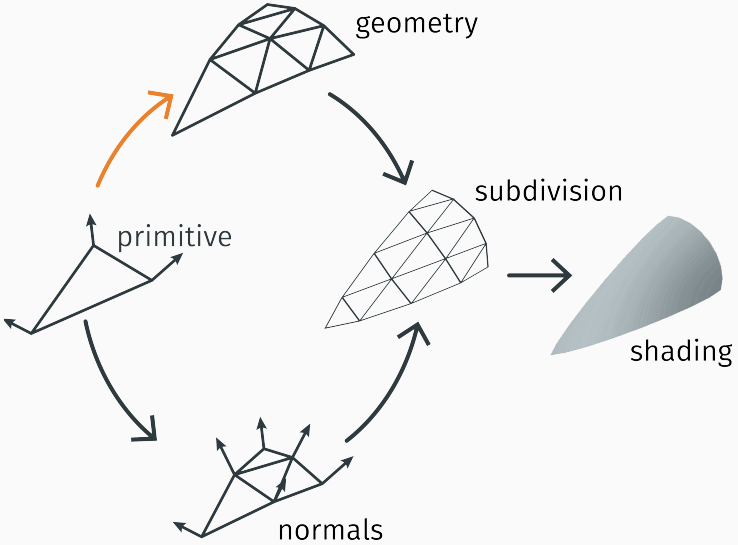
Point Normal triangles

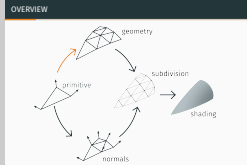
[Name] Why PN triangles? Look at the nice result it gives :-) and we will see that it easy to extend it to the ‘existing’ pipeline.

SINGLE PN TRIANGLE

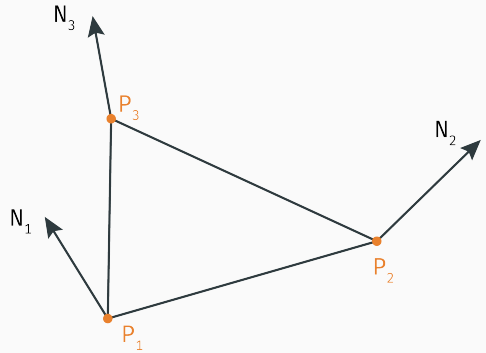
[Name] How does one construct a single PN triangle?
Overview on the next slide

OVERVIEW

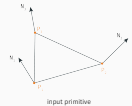




[Name] Why PN triangles? Look at the nice result it gives :-) and we will see that it easy to extend it to the ‘existing’ pipeline. Story about Bezier patches...



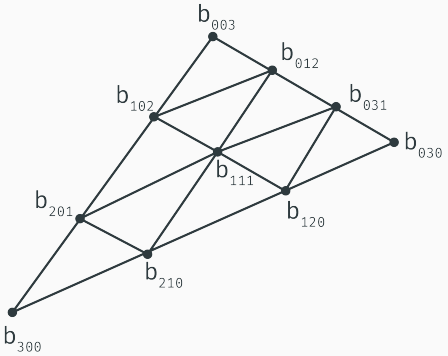
input primitive



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

Focus on getting the different control primitives.

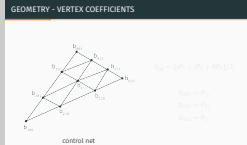
GEOMETRY - VERTEX COEFFICIENTS



control net

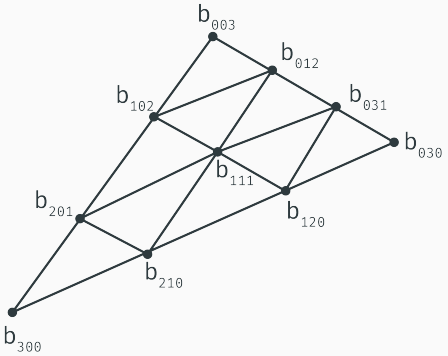
$$b_{ijk} = (iP_1 + jP_2 + kP_3)/3$$

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

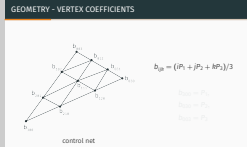


[Name] These are all the initial control point. Evenly divided on the triangle. -> formula

GEOMETRY - VERTEX COEFFICIENTS

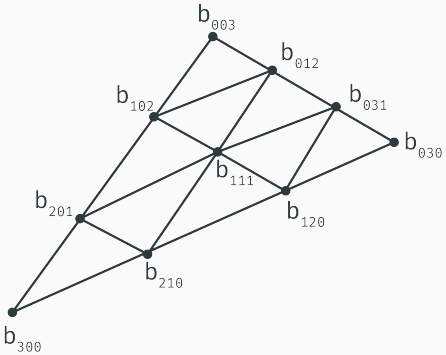


control net



[Name] Nice formula

GEOMETRY - VERTEX COEFFICIENTS



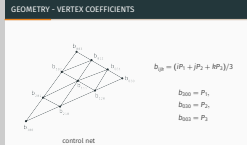
control net

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

$$b_{300} = P_1,$$

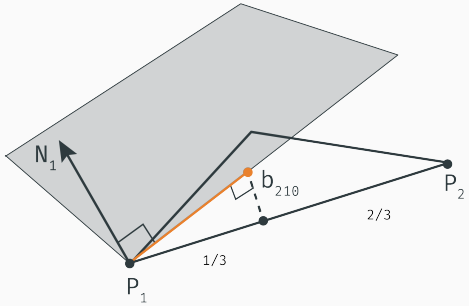
$$b_{030} = P_2,$$

$$b_{003} = P_3$$



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

GEOMETRY - TANGENT COEFFICIENTS



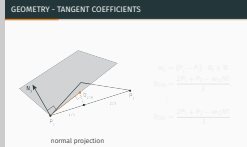
normal projection

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

$$b_{210} = \frac{2P_1 + P_2 - w_{12}N1}{3},$$

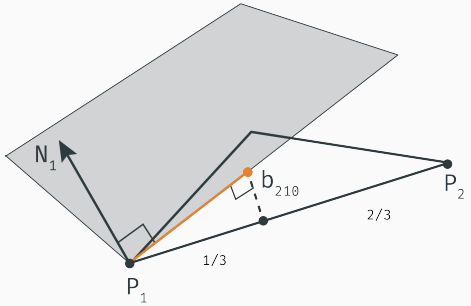
$$\vdots$$

$$b_{201} = \frac{2P_1 + P_3 - w_{13}N1}{3}$$



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

GEOMETRY - TANGENT COEFFICIENTS



normal projection

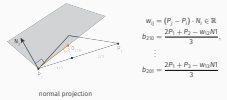
$$w_{ij} = (P_j - P_i) \cdot N_i \in \mathbb{R}$$

$$b_{210} = \frac{2P_1 + P_2 - w_{12}N1}{3},$$

$$\vdots$$

$$b_{201} = \frac{2P_1 + P_3 - w_{13}N1}{3}$$

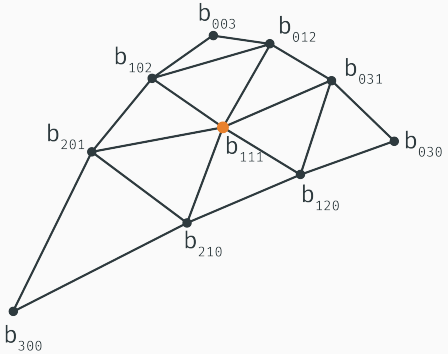
GEOMETRY - TANGENT COEFFICIENTS



$w_{ij} = (P_j - P_i) \cdot N_i \in \mathbb{R}$
$b_{210} = \frac{2P_1 + P_2 - w_{12}N1}{3},$
$\vdots$
$b_{201} = \frac{2P_1 + P_3 - w_{13}N1}{3}$
normal projection

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

GEOMETRY - CENTER COEFFICIENT

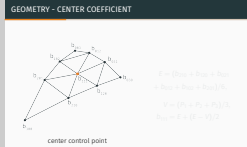


center control point

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

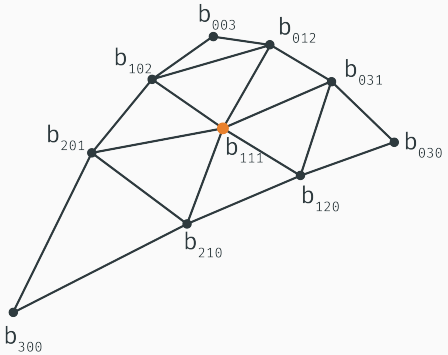
$$V = (P_1 + P_2 + P_3)/3,$$

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



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

GEOMETRY - CENTER COEFFICIENT

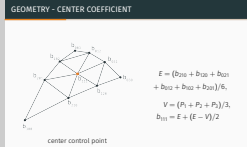


center control point

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

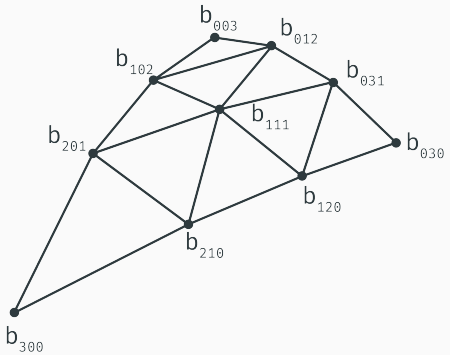
$$V = (P_1 + P_2 + P_3)/3,$$

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

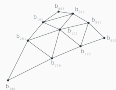


[Name] Average of the tangent coefficients plus half the difference between the tangent and vertex coefficients. -> why?

GEOMETRY - RESULT

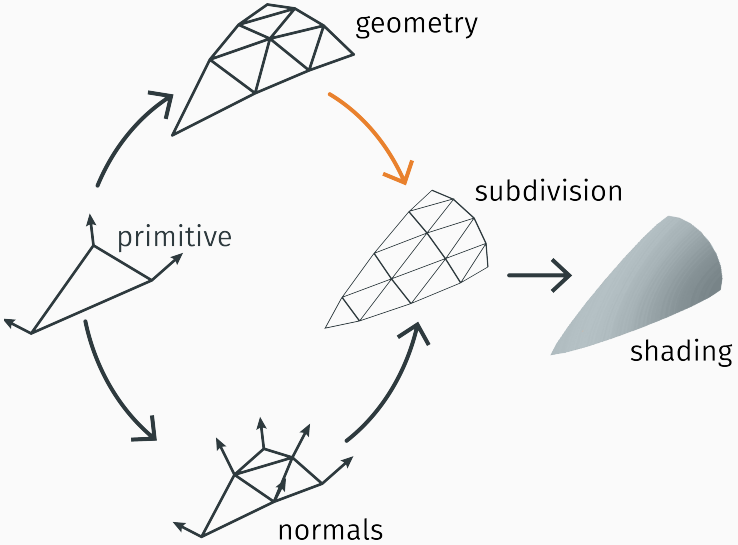


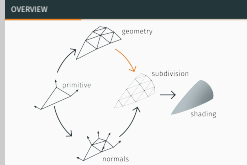
GEOMETRY - RESULT



[Name] Results

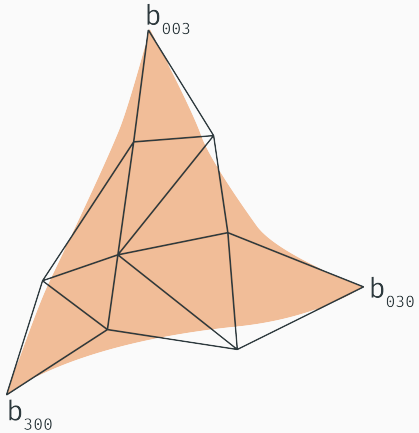
OVERVIEW





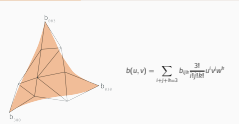
[Name] Overview -> how to get from this to shading.
Sample/subdivide with formula on following slide.

CUBIC PATCH



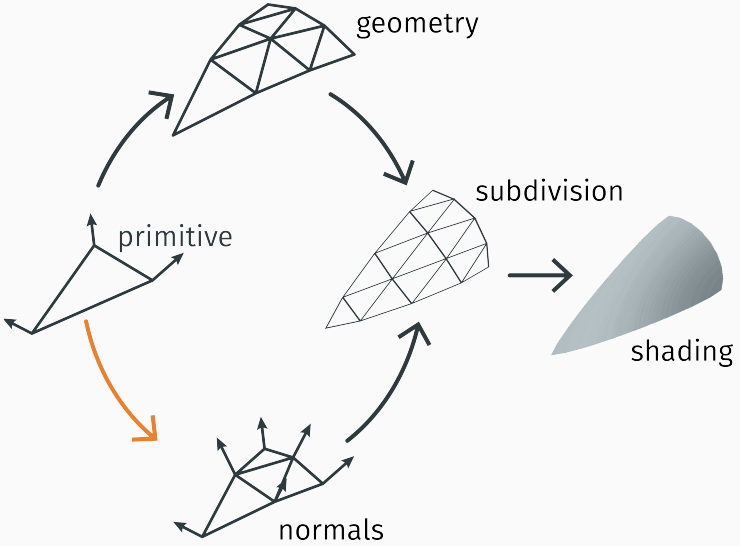
$$b(u,v) = \sum_{i+j+k=3} b_{ijk} \frac{3!}{i!j!k!} u^i v^j w^k$$

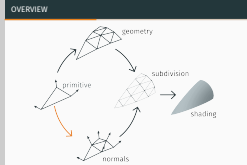
CUBIC PATCH



u, v, w are a convex combination**[Name]** Very nice formula with a nice picture.

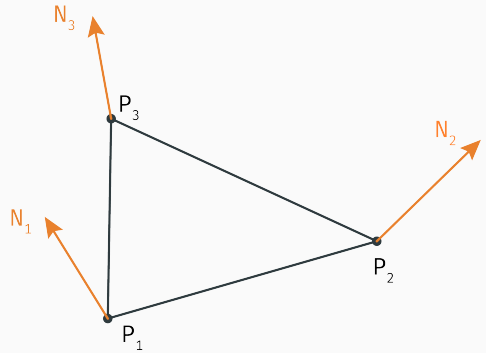
OVERVIEW





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

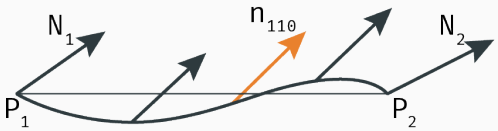
NORMALS



input primitive

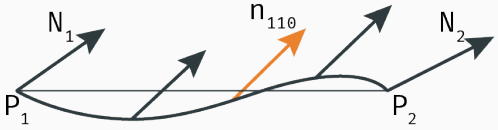
[Name] Recap input primitive and with emphasis on the normals.

NORMALS - THEORY

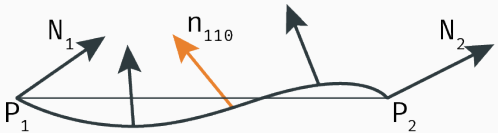


[Name] Stress that there is a need to capture the cubic bezier curve (inflection points) and that this cannot be

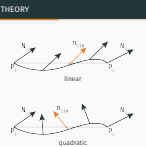
NORMALS - THEORY



linear



quadratic

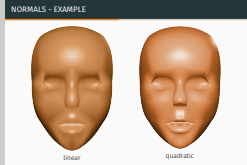


[Name] Quadratic does capture inflection points. Trade off between performance and result (maybe?)

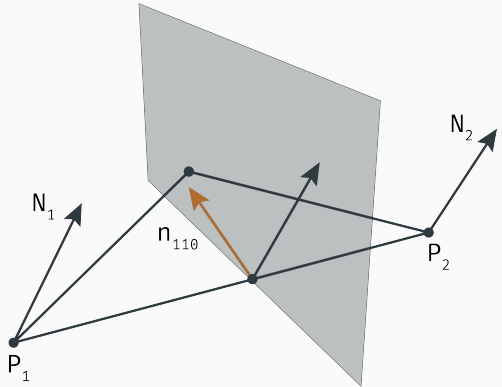
NORMALS - EXAMPLE

linear

quadratic



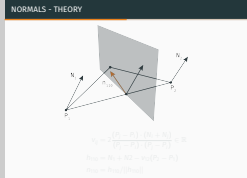
[Name] Look how pretty.



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

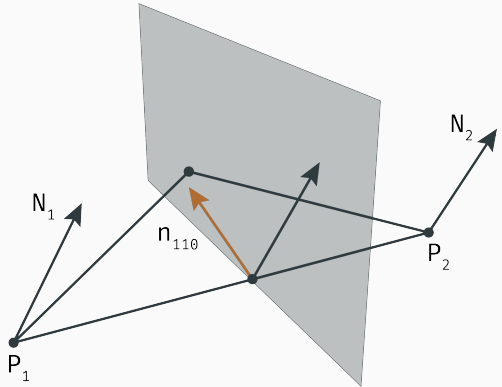
$$h_{110} = N_1 + N2 - v_{12}(P_2 - P_1)$$

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



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

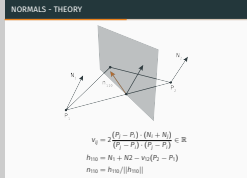
NORMALS - THEORY



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

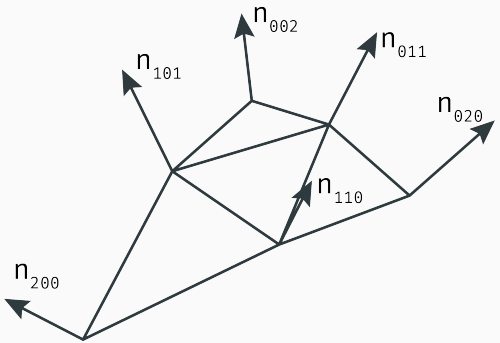
$$h_{110} = N_1 + N2 - v_{i1}(P_2 - P_1)$$

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

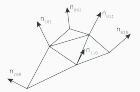


[Name] Formula in words: reflect the averaged normal (average of N1 and N2) on the plane orthogonal/perpendicular the the edge at the mid point.

NORMALS - RESULT

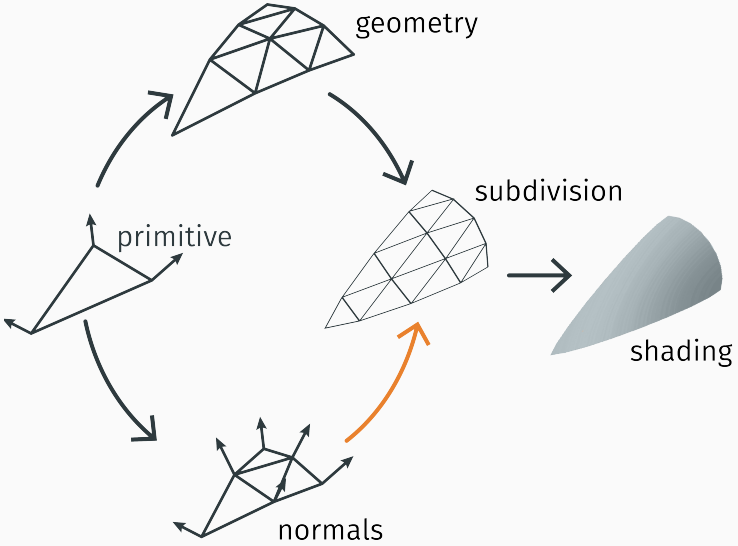


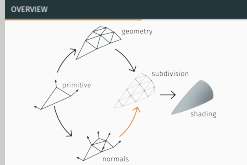
NORMALS - RESULT



[Name] Result

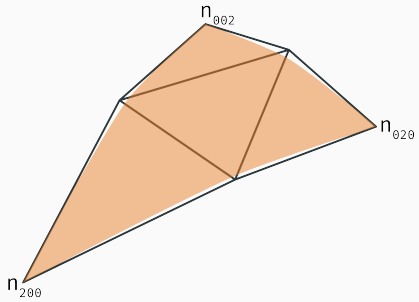
OVERVIEW





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

QUADRATIC PATCH

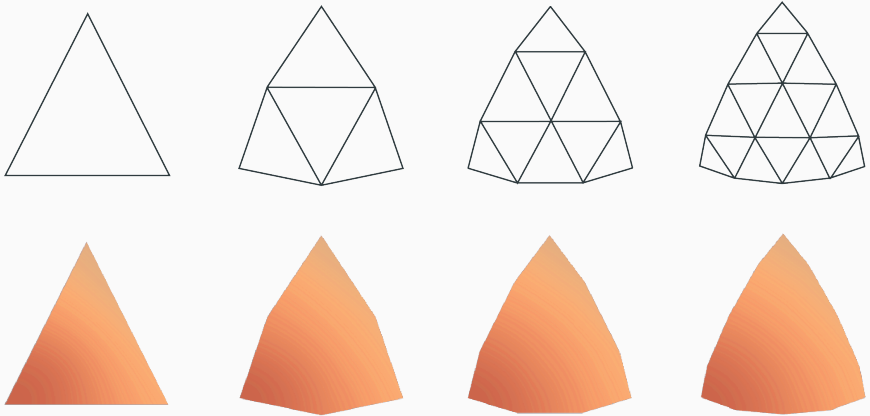


$$n(u,v) = \sum_{i+j+k=2} n_{ijk} u^i v^j w^k$$

QUADRATIC PATCH

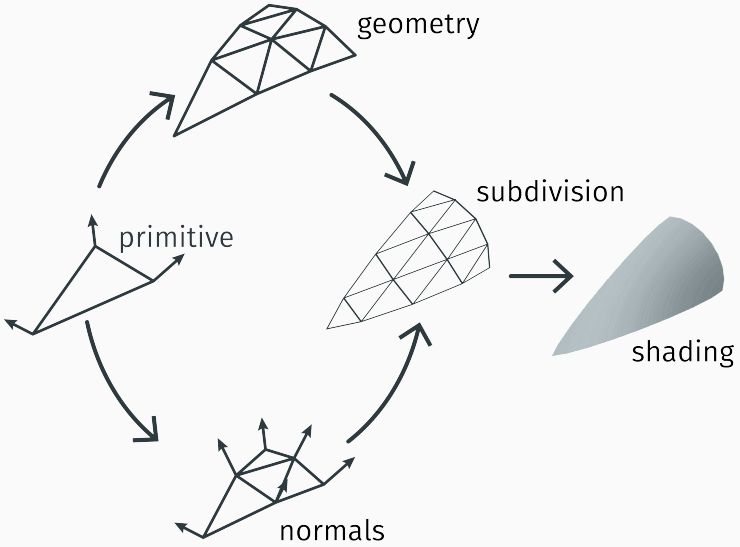
[Name] u, v and w are an convex combination

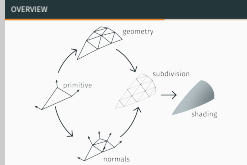
LEVEL OF DETAIL



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

OVERVIEW





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

A TRIANGLE MESH

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.

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

CONTINUITY

PN triangles have[2]

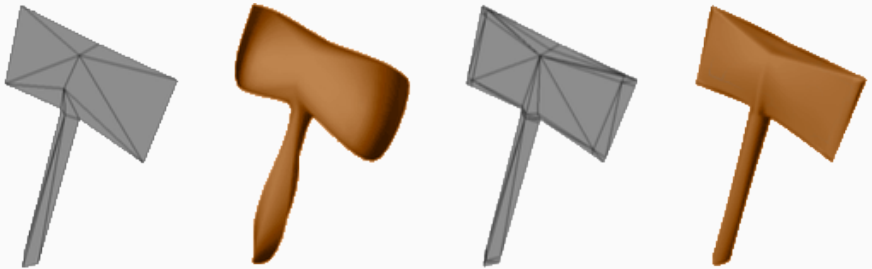
- C^1 continuity in the vertex points
- C^0 continuity everywhere else

[2] Jiao and Alexander

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

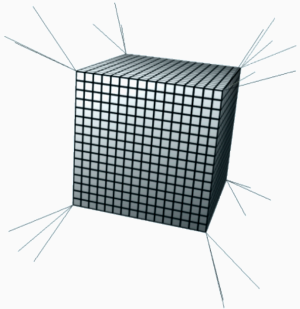
[Name] Continuity C0 is important -> no gaps. Higher is better because this gives a more smooth result.

SHARP EDGES

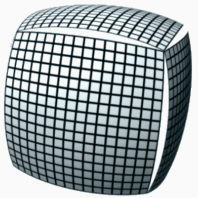


[Name] Curved triangles do not always give the preferred results -> sharp edges. Solution is to insert more triangles at the sharp edges -> model needs to be changed :(

SEPARATE NORMALS



normals



cracks

SEPARATE NORMALS
normals
cracks

[Name] 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

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

HARDWARE - PIPELINES

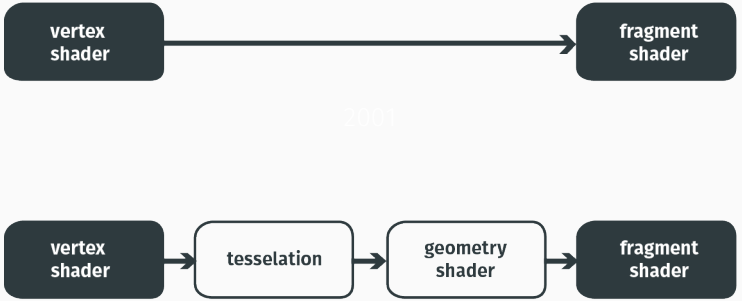
2001

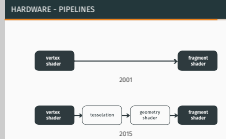
2015

[Name] 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





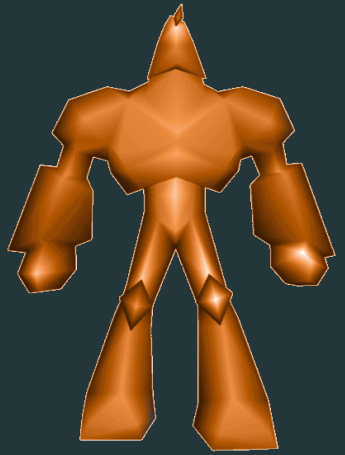
[Name] 2015 we have OpenGL 4.5 with more programmable shaders and the whole process can be done on the GPU. Since PN triangles only usses the primitive, no neighboring primitives, easy in shaders.

CONCLUSION

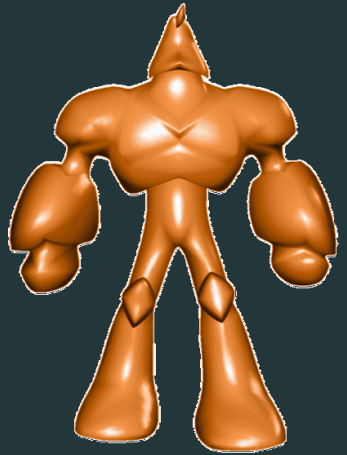
CONCLUSION

[Name] Conclusion?

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.