POINT NORMAL TRIANGLES

Rick van Veen Laura Baakman

December 14, 2015

Advanced Computer Graphics

POINT NORMAL TRIANGLES

Rick van Veen Laura Baakman

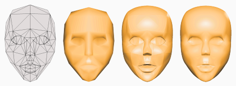
December 14, 2015

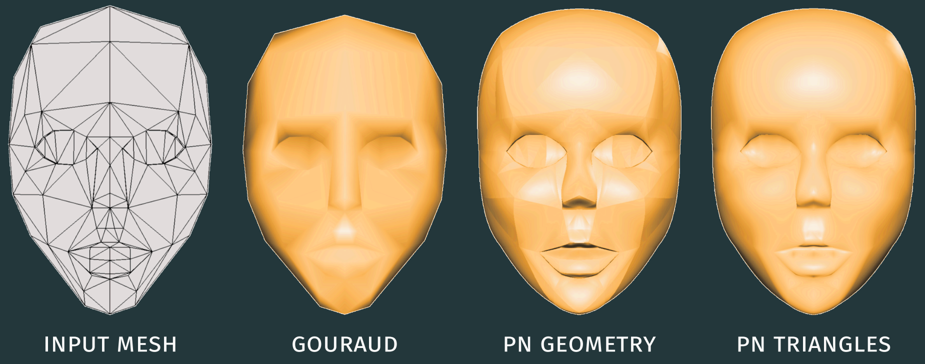
Advanced Computer Graphics

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

2015-12-09

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.

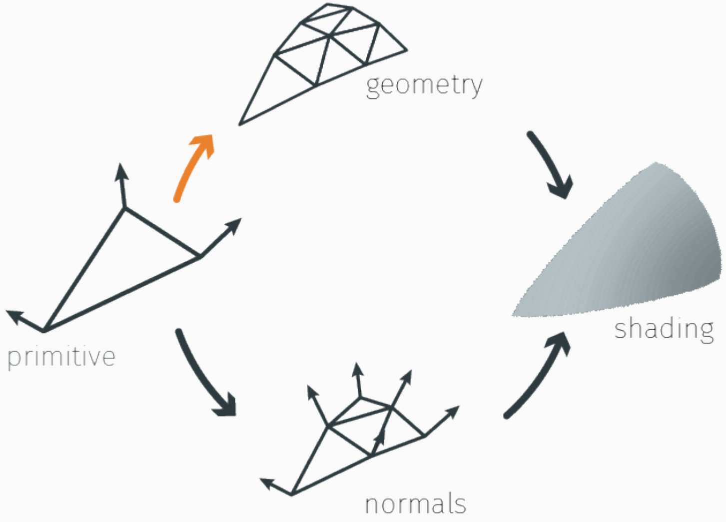
SINGLE PN TRIANGLE

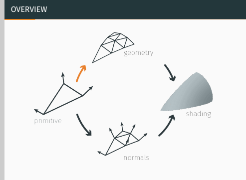
SINGLE PN TRIANGLE

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

Overview on the next slide

OVERVIEW

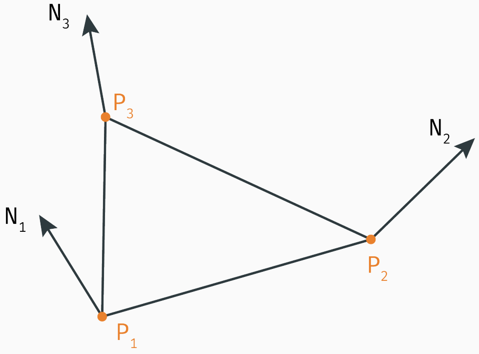




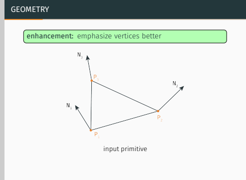
[Name] Why PN triangles? Look at the nice result it gives :-) and we will see that it easy to extend it to the 'existing' pipeline. Story about Bezier patches...

GEOMETRY

enhancement: emphasize vertices better



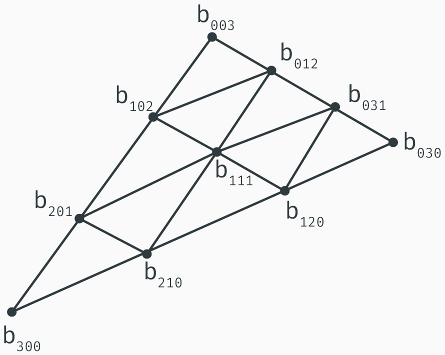
input primitive



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

Focus on getting the different control primitives.

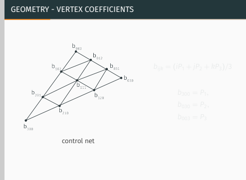
GEOMETRY - VERTEX COEFFICIENTS



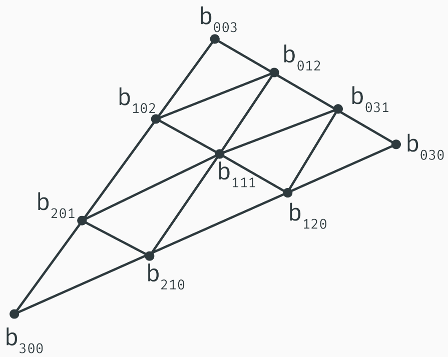
control net

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

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



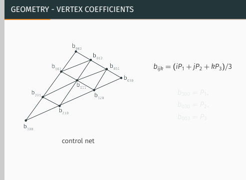
[Name] These are all the initial control point. Evenly divided on the triangle. -> formula



control net

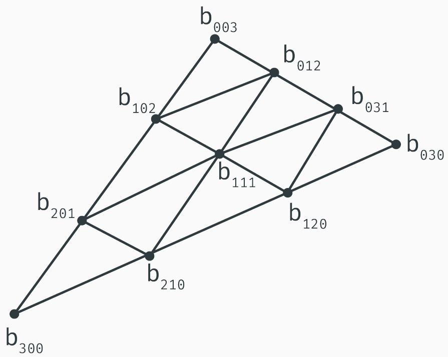
$$b_{ijk} = (iP_1 + jP_2 + kP_3)/3$$

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



[Name] Nice formula

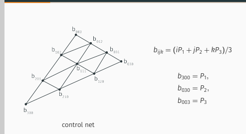
GEOMETRY - VERTEX COEFFICIENTS



control net

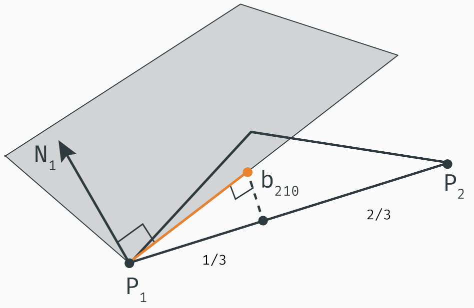
$$b_{ijk} = (iP_1 + jP_2 + kP_3)/3$$

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



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

GEOMETRY - TANGENT COEFFICIENTS



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

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

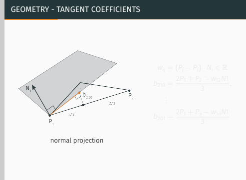
$$\vdots$$

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

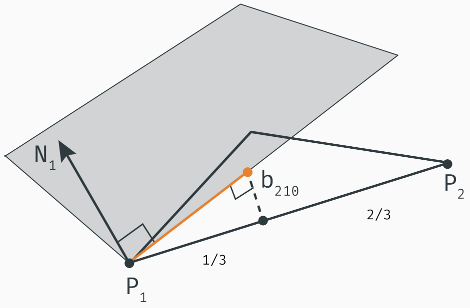
normal projection

2015-12-09

Point Normal triangles
└Single PN Triangle
└Geometry - Tangent Coefficients



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



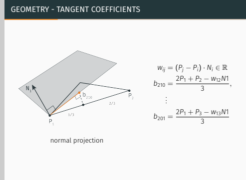
normal projection

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

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

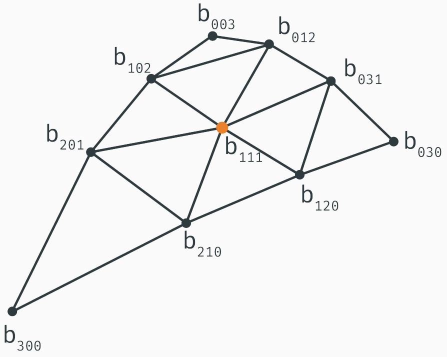
$$\vdots$$

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



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

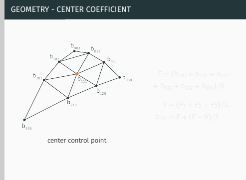
GEOMETRY - CENTER COEFFICIENT



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

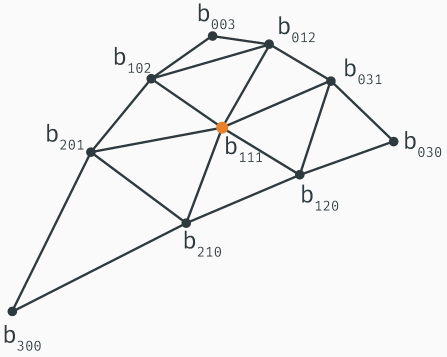
$$V = (P_1 + P_2 + P_3)/3,$$
$$b_{111} = E + (E - V)/2$$

center control point



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

GEOMETRY - CENTER COEFFICIENT

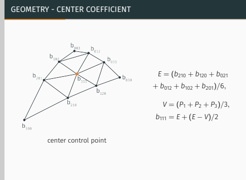


$$E = (b_{210} + b_{120} + b_{021} + b_{012} + b_{102} + b_{201})/6,$$

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

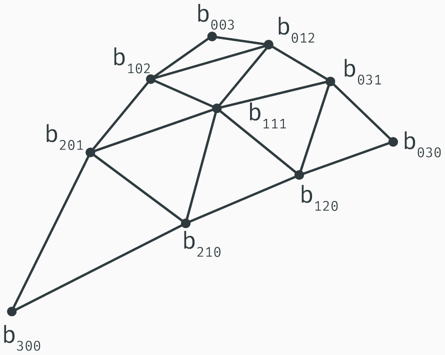
$$b_{111} = E + (E - V)/2$$

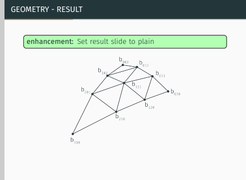
center control point



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

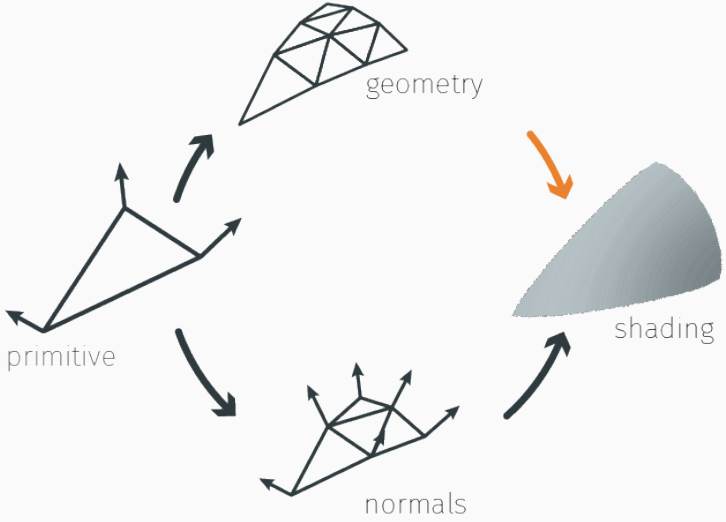
enhancement: Set result slide to plain

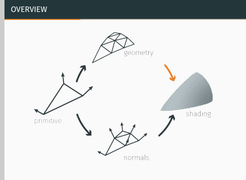




[Name] Results

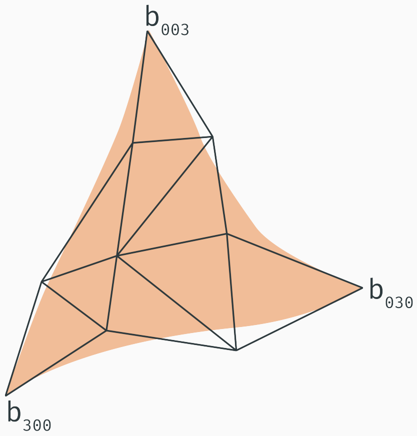
OVERVIEW



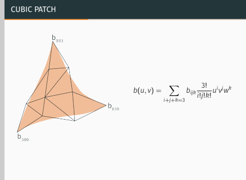


[Name] Overview -> how to get from this to shading.
Sample/subdivide with formula on following slide.

CUBIC PATCH

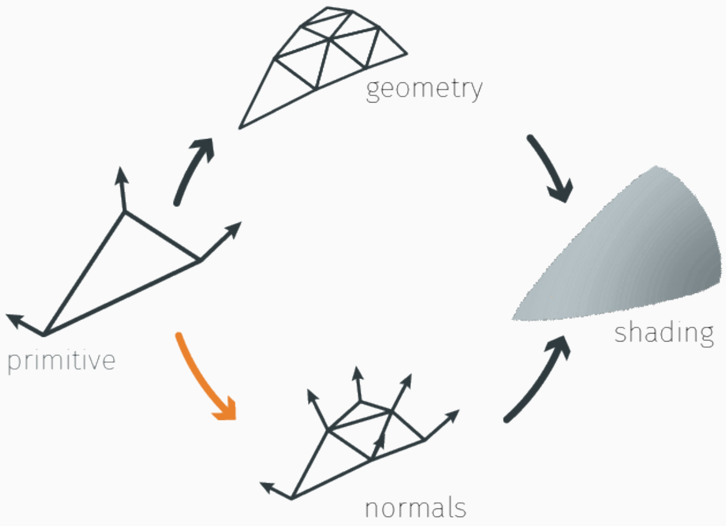


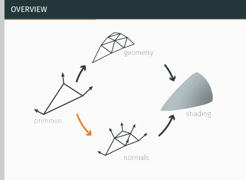
$$b(u,v) = \sum_{i+j+k=3} b_{ijk} \frac{3!}{i!j!k!} u^i v^j w^k$$



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

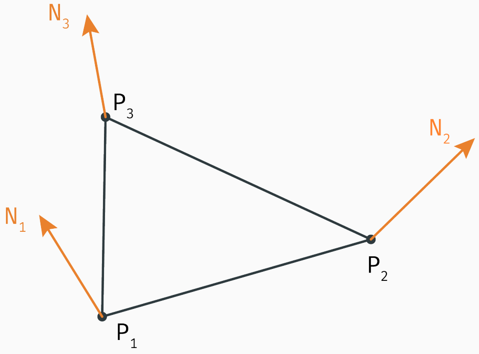
OVERVIEW



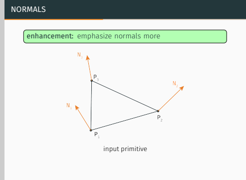


[Name] From the primitive normals the the PN triangle normals

enhancement: emphasize normals more

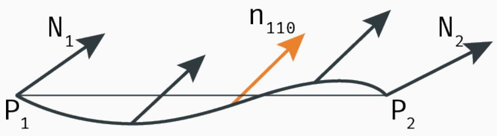


input primitive



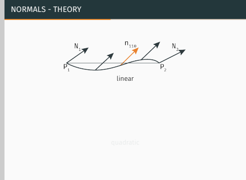
[Name] Recap input primitive and with emphasis on the normals.

NORMALS - THEORY



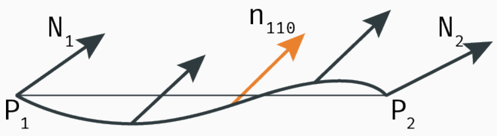
linear

quadratic

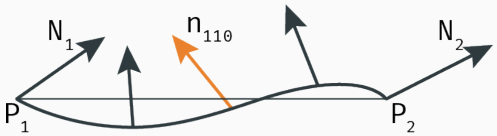


[Name] Stress that there is a need to capture the cubic bezier curve (inflection points) and that this cannot be

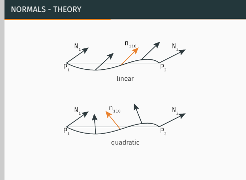
NORMALS - THEORY



linear

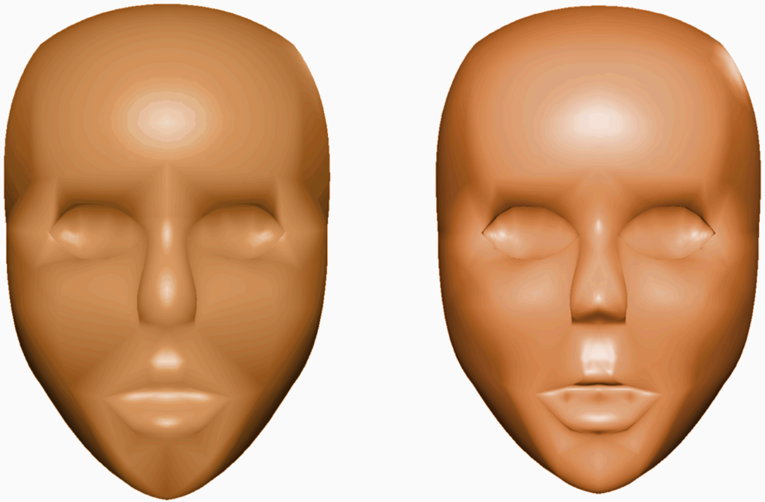


quadratic

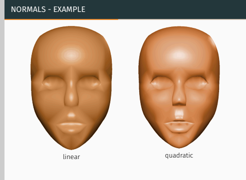


[Name] Quadratic does capture inflection points. Trade off between performance and result (maybe?)

NORMALS - EXAMPLE

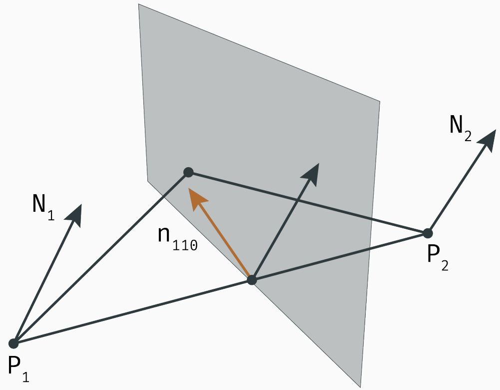


linear quadratic



[Name] Look how pretty.

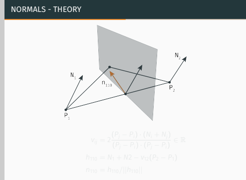
NORMALS - THEORY



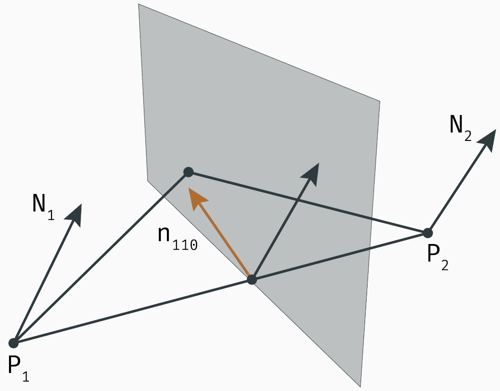
$$v_{ij} = 2\frac{(P_j - P_i) \cdot (N_i + N_j)}{(P_j - P_i) \cdot (P_j - P_i)} \in \mathbb{R}$$

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

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



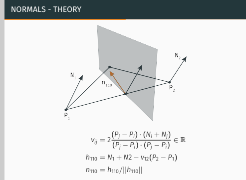
[Name] Formula in words: reflect the averaged normal (average of N1 and N2) on the plane orthogonal/perpendicular the the edge at the mid point.



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

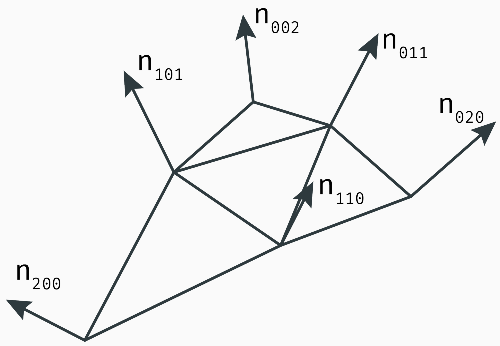
$$h_{110} = N_1 + N2 - v_{12}(P_2 - P_1)$$

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



[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.

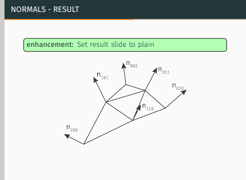
enhancement: Set result slide to plain



2015-12-09

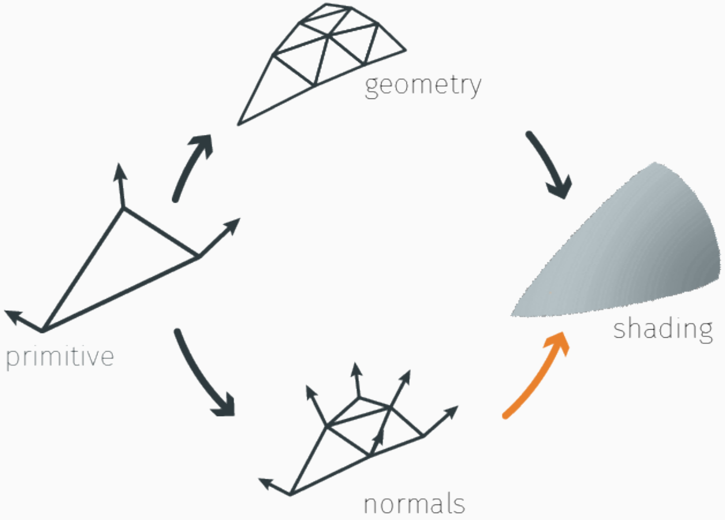
Point Normal triangles
└Single PN Triangle

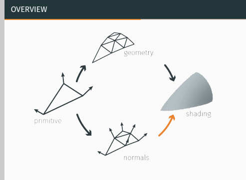
└Normals - result



[Name] Result

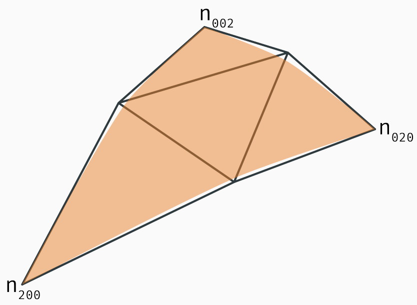
OVERVIEW



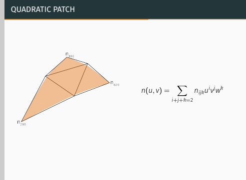


[Name] Why PN triangles? Look at the nice result it gives :-) and we will see that it easy to extend it to the ‘existing’ pipeline.

QUADRATIC PATCH

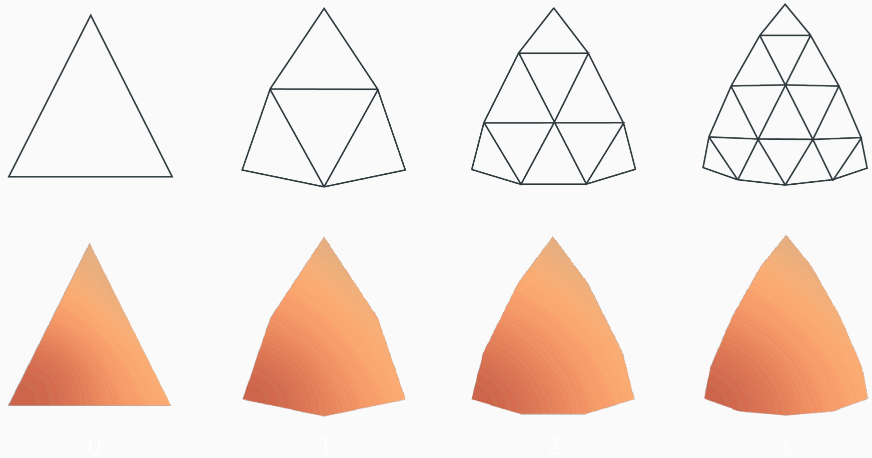


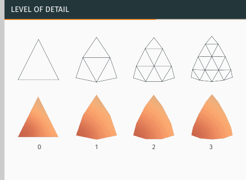
$$n(u,v) = \sum_{i+j+k=2} n_{ijk} u^i v^j w^k$$



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

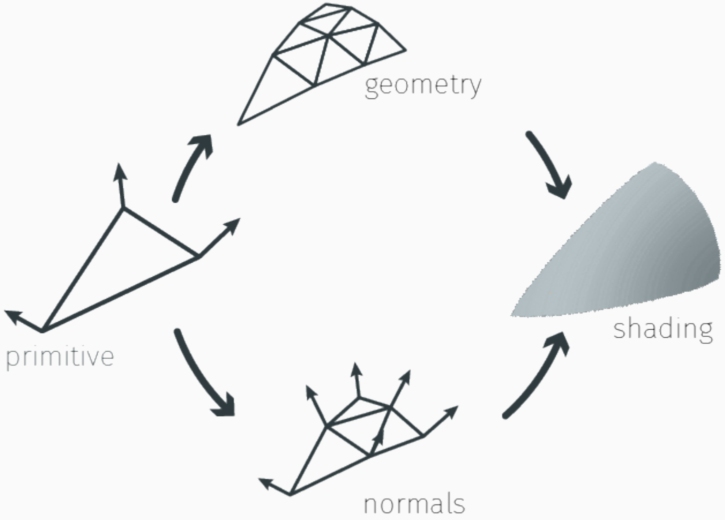
LEVEL OF DETAIL

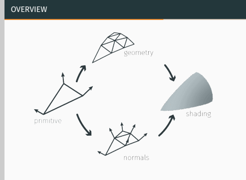




[Name] Level of detail -> subdivision -> how many triangles go through to the next shaders.

OVERVIEW





[Name]

A TRIANGLE MESH

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.

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

PN triangles have:[2]

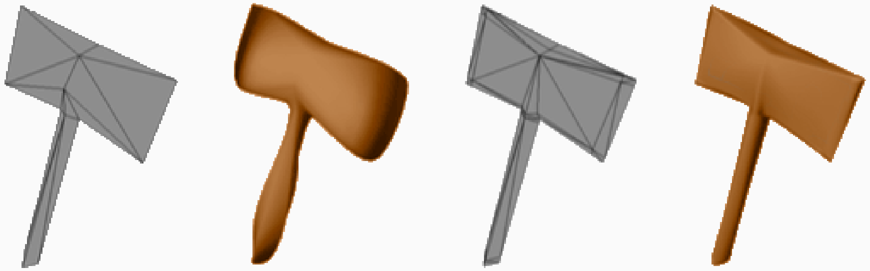
- C^1 continuity in the vertex points
- C^0 continuity everywhere else

[2] Jiao and Alexander

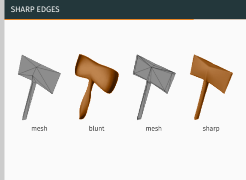
CONTINUITY
PN triangles have:[2]
- C^1 continuity in the vertex points
- C^0 continuity everywhere else
[2] Jiao and Alexander

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

SHARP EDGES

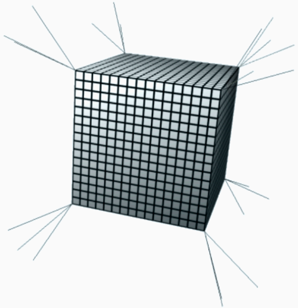


mesh blunt mesh sharp

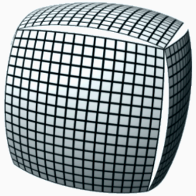


[Name] Curved triangles do not always give the preferred results -> sharp edges. Solution is to insert more triangles at the sharp edges -> model needs to be changed :(

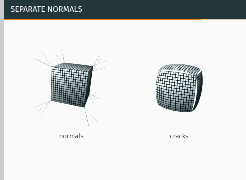
SEPARATE NORMALS



normals



cracks



[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

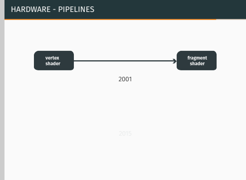
GRAPHICS PIPELINE

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

HARDWARE - PIPELINES

2001

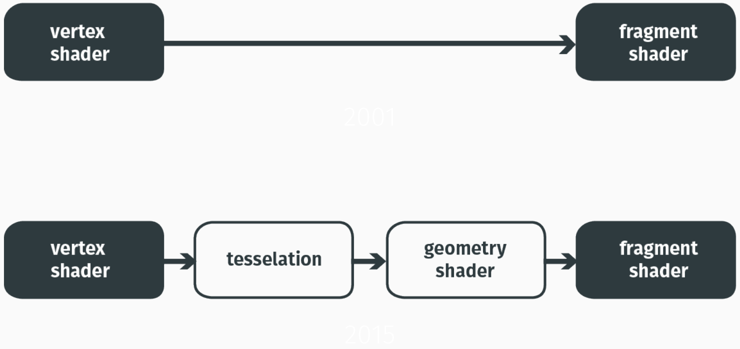
2015

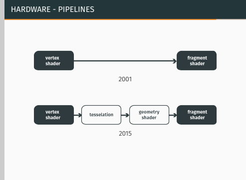


[Name] Great part of the paper stresses the point that it can easily be implemented as a preprocessing step (CPU.

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.

CONCLUSION

CONCLUSION

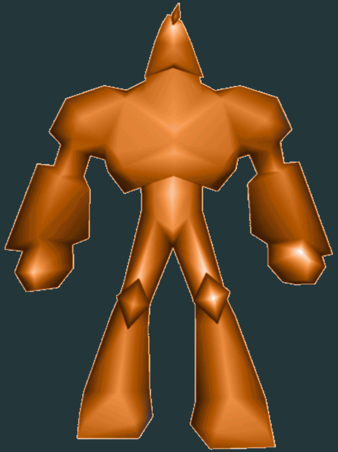
CONCLUSION

Conclusion?

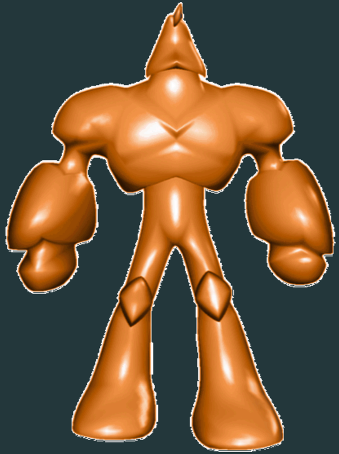
CONCLUSION
Conclusion?

[Name] Conclusion?

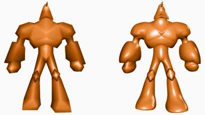
QUESTIONS?



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.