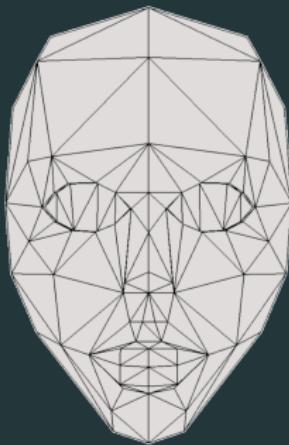


POINT NORMAL TRIANGLES

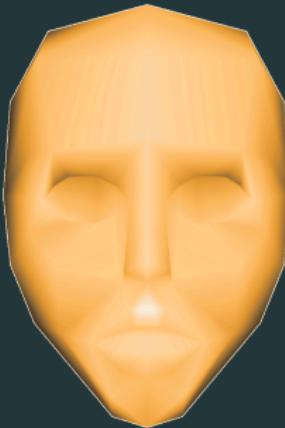
Rick van Veen Laura Baakman

December 14, 2015

Advanced Computer Graphics



INPUT MESH



GOURAUD



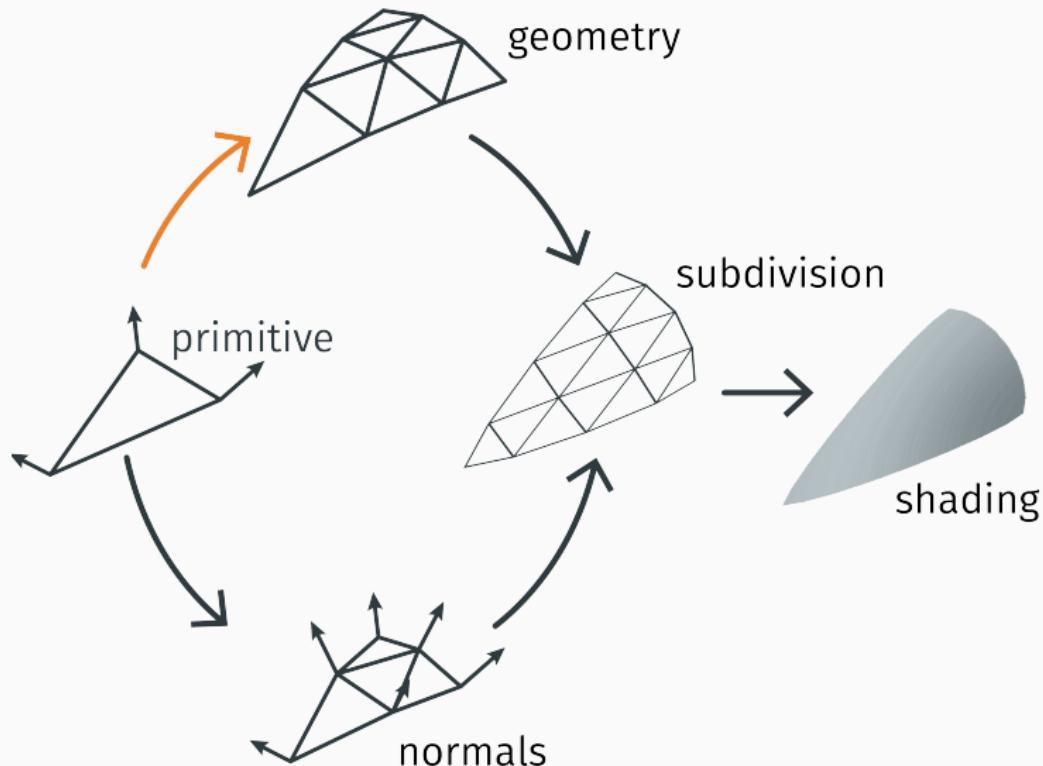
PN GEOMETRY



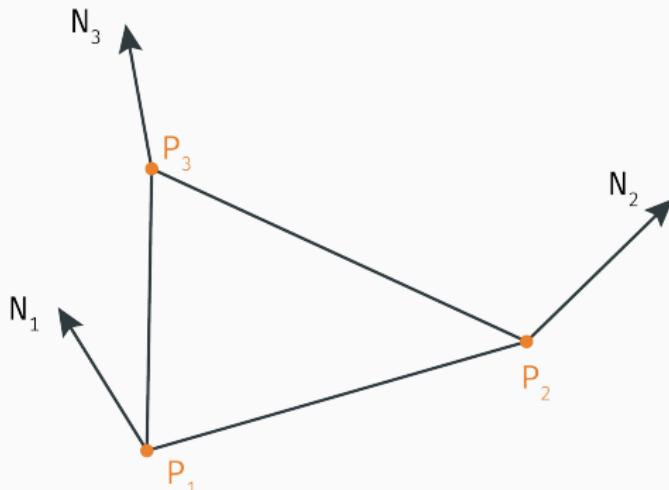
PN TRIANGLES

SINGLE PN TRIANGLE

OVERVIEW

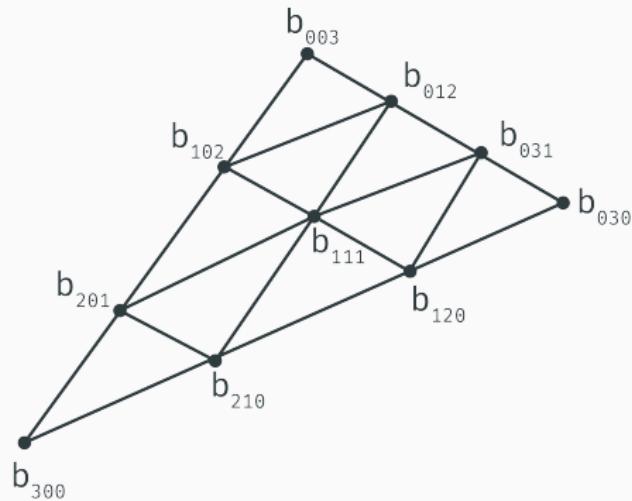


GEOMETRY



input primitive

GEOMETRY - VERTEX COEFFICIENTS



control net

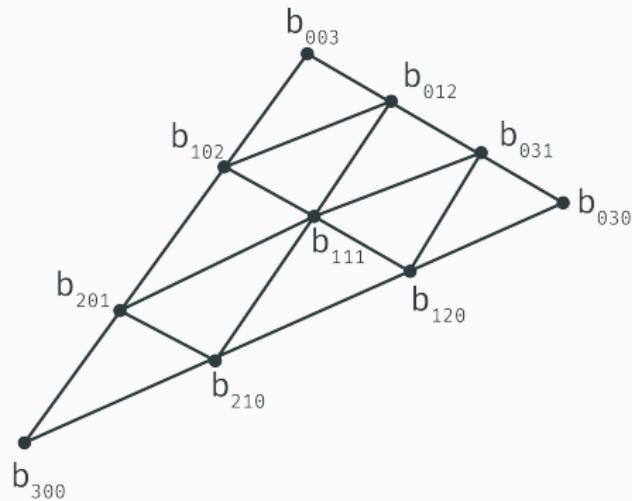
$$b_{ijk} = (iP_1 + jP_2 + kP_3)/3$$

$$b_{300} = P_1,$$

$$b_{030} = P_2,$$

$$b_{003} = P_3$$

GEOMETRY - VERTEX COEFFICIENTS



control net

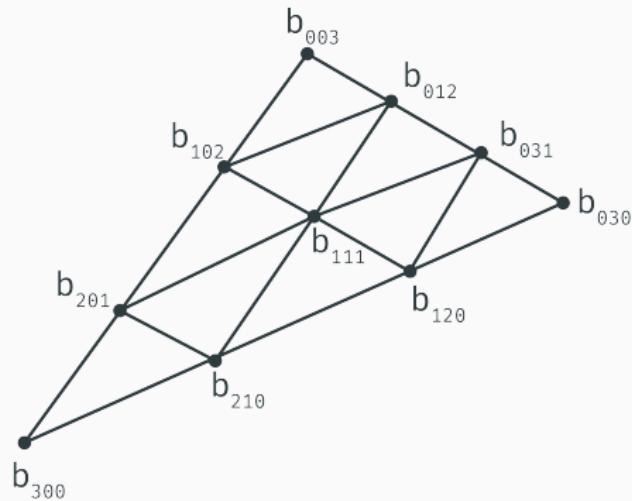
$$b_{ijk} = (iP_1 + jP_2 + kP_3)/3$$

$$b_{300} = P_1,$$

$$b_{030} = P_2,$$

$$b_{003} = P_3$$

GEOMETRY - VERTEX COEFFICIENTS



control net

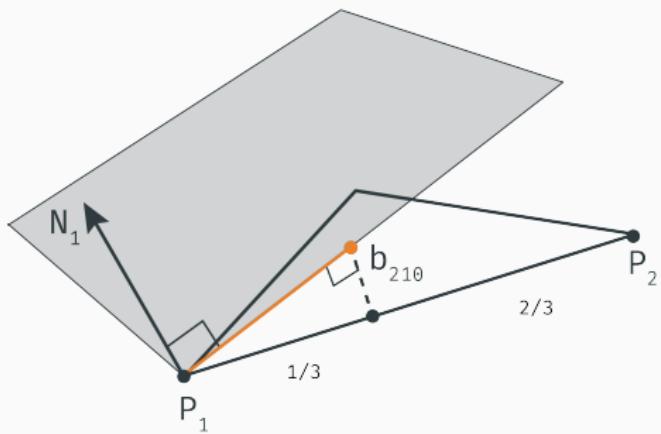
$$b_{ijk} = (iP_1 + jP_2 + kP_3)/3$$

$$b_{300} = P_1,$$

$$b_{030} = P_2,$$

$$b_{003} = P_3$$

GEOMETRY - TANGENT COEFFICIENTS



normal projection

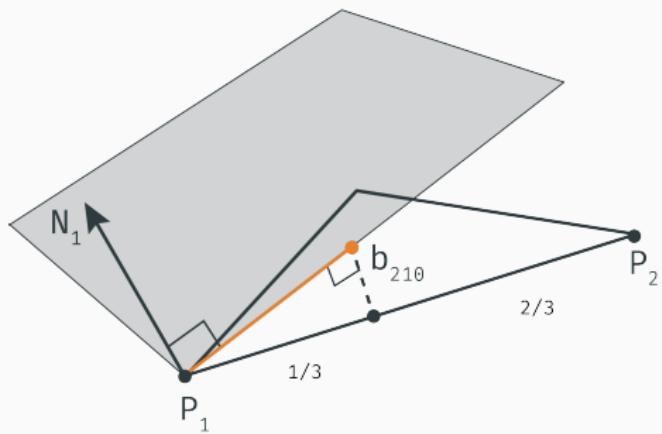
$$w_{ij} = (P_j - P_i) \cdot N_i \in \mathbb{R}$$

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

⋮

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

GEOMETRY - TANGENT COEFFICIENTS



normal projection

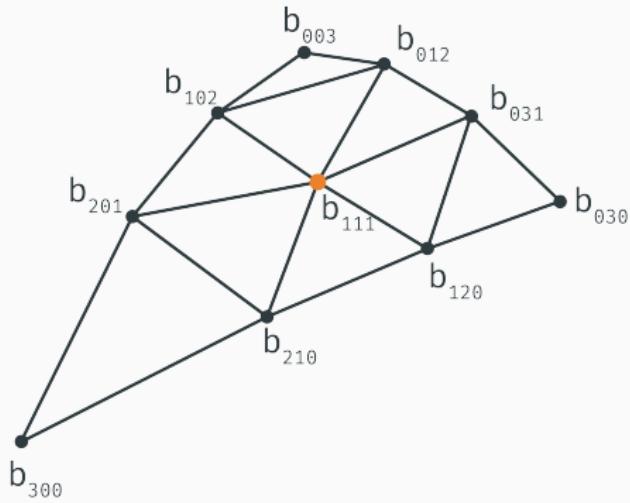
$$w_{ij} = (P_j - P_i) \cdot N_i \in \mathbb{R}$$

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

⋮

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

GEOMETRY - CENTER COEFFICIENT

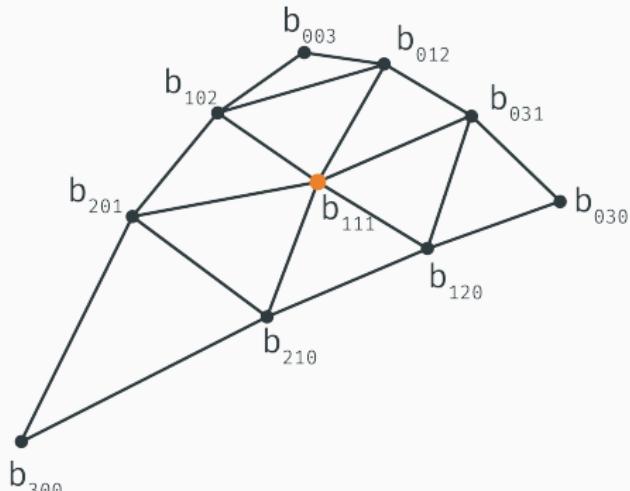


$$E = (b_{210} + b_{120} + b_{021} + b_{012} + b_{102} + b_{201})/6,$$

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

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

GEOMETRY - CENTER COEFFICIENT



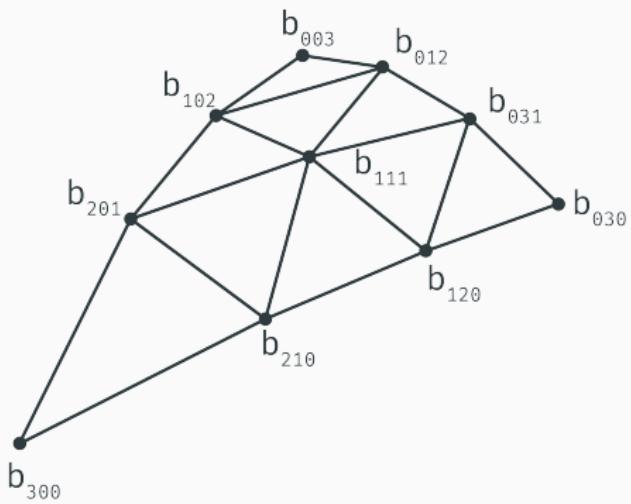
center control point

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

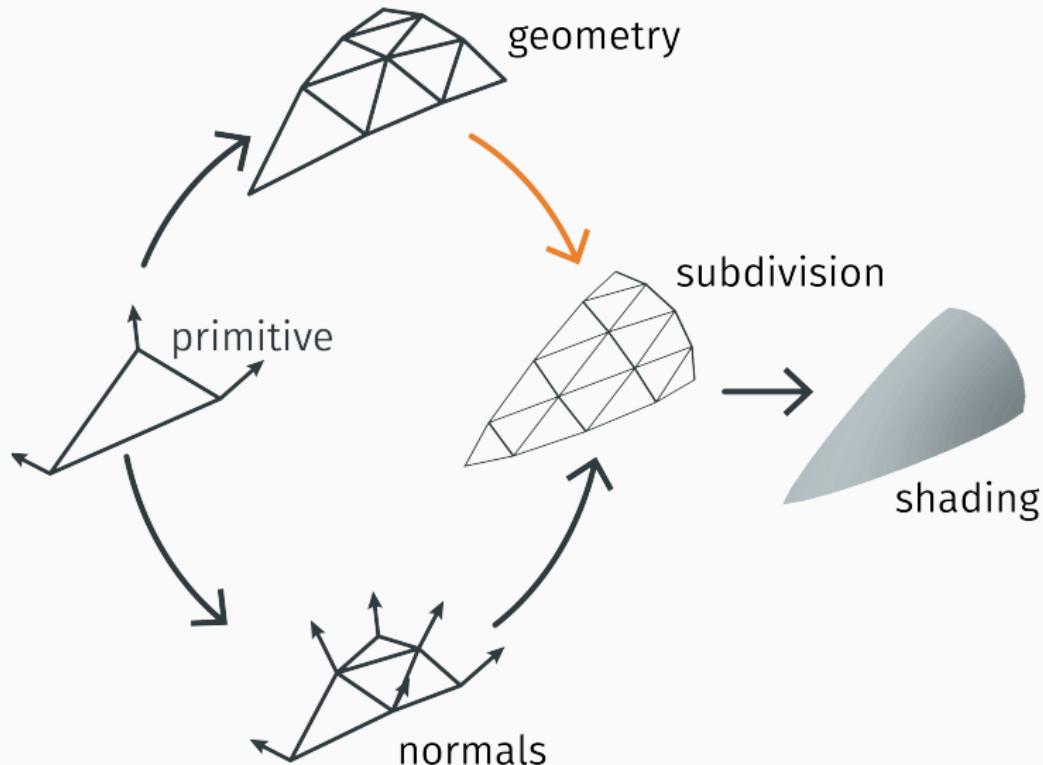
$$V = (P_1 + P_2 + P_3)/3,$$

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

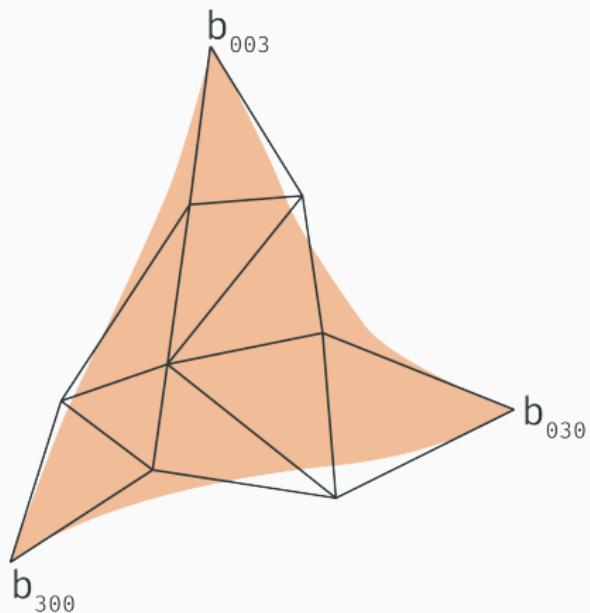
GEOMETRY - RESULT



OVERVIEW



CUBIC BÉZIER PATCH

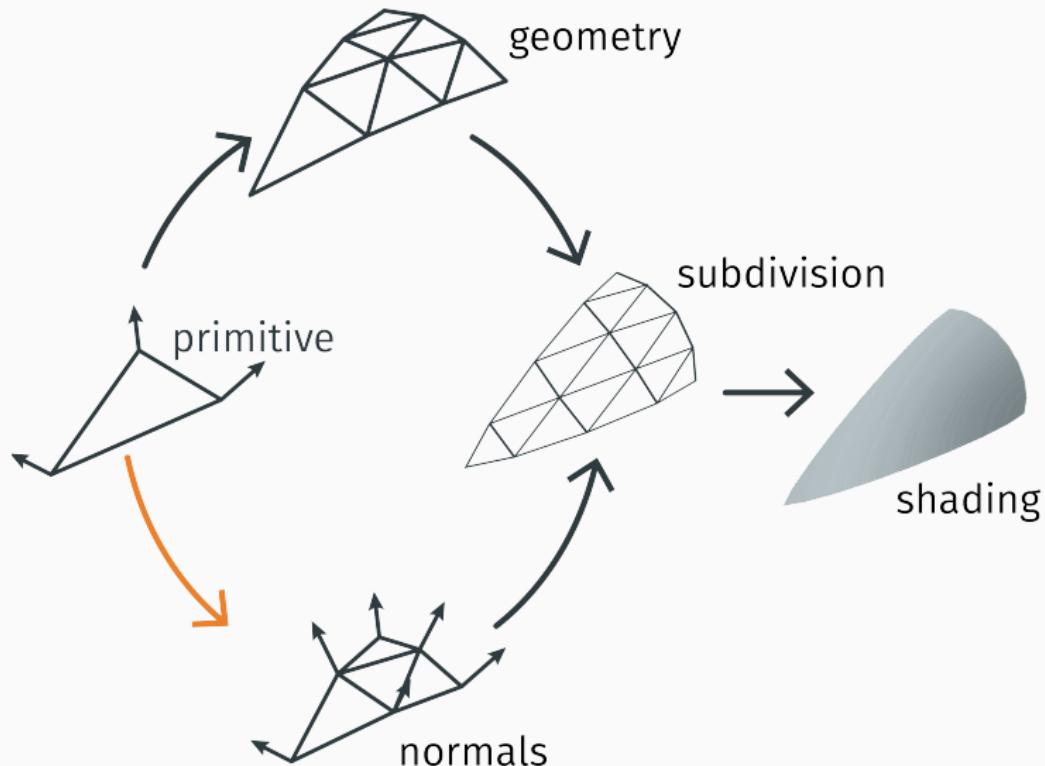


$$w = 1 - u - v$$

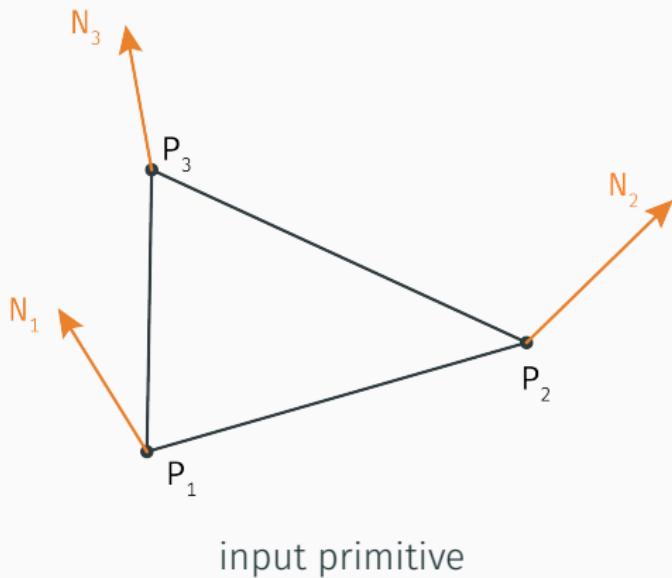
$$u, v, w \geq 0$$

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

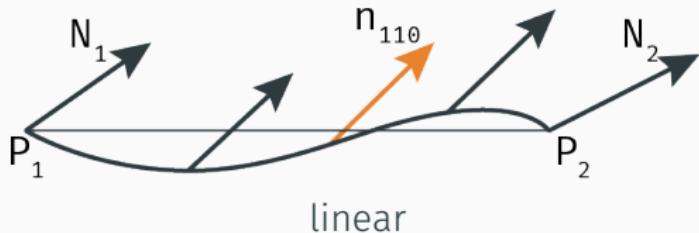
OVERVIEW



NORMALS

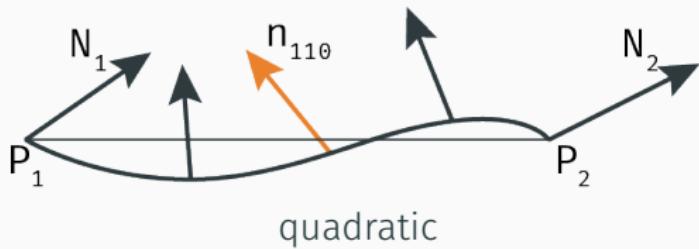
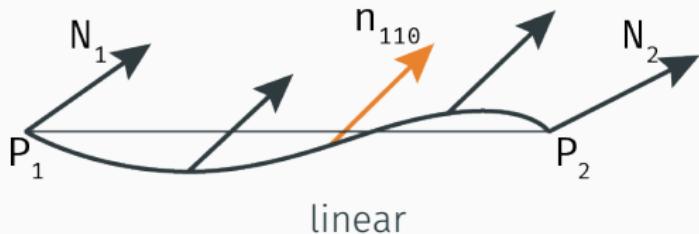


NORMALS - THEORY



quadratic

NORMALS - THEORY



NORMALS - EXAMPLE

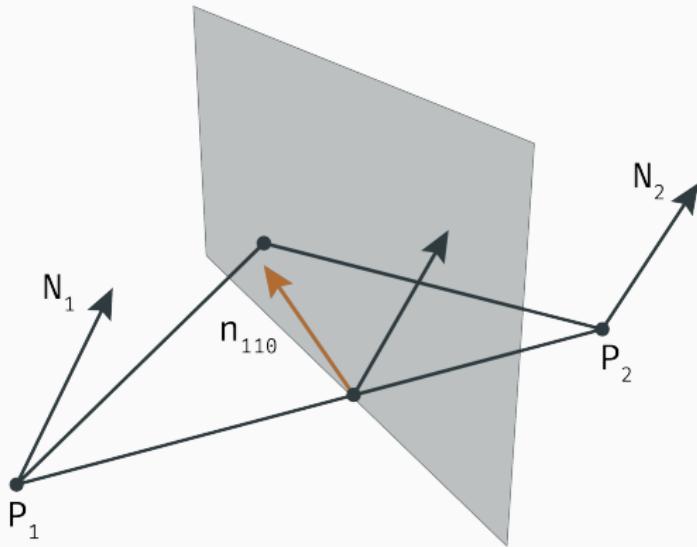


linear



quadratic

NORMALS - THEORY

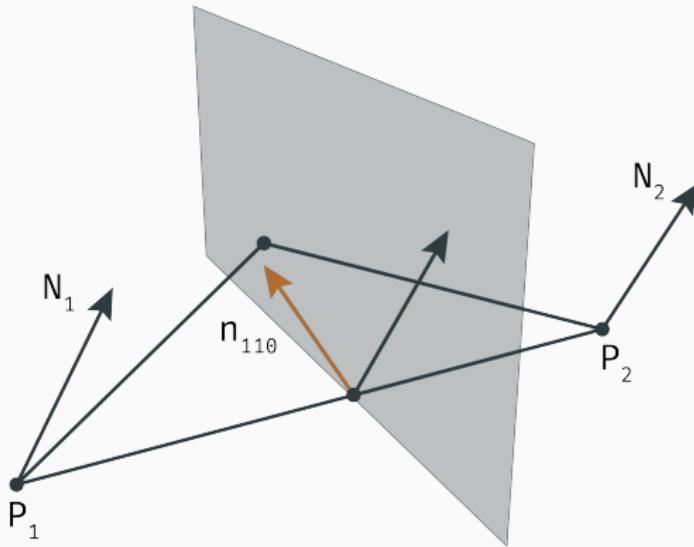


$$v_{ij} = 2 \frac{(P_j - P_i) \cdot (N_i + N_j)}{(P_j - P_i) \cdot (P_j - P_i)} \in \mathbb{R}$$

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

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

NORMALS - THEORY

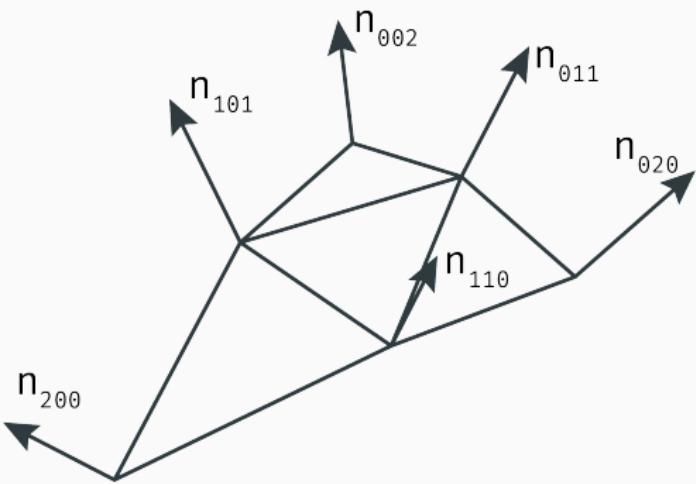


$$v_{ij} = 2 \frac{(P_j - P_i) \cdot (N_i + N_j)}{(P_j - P_i) \cdot (P_j - P_i)} \in \mathbb{R}$$

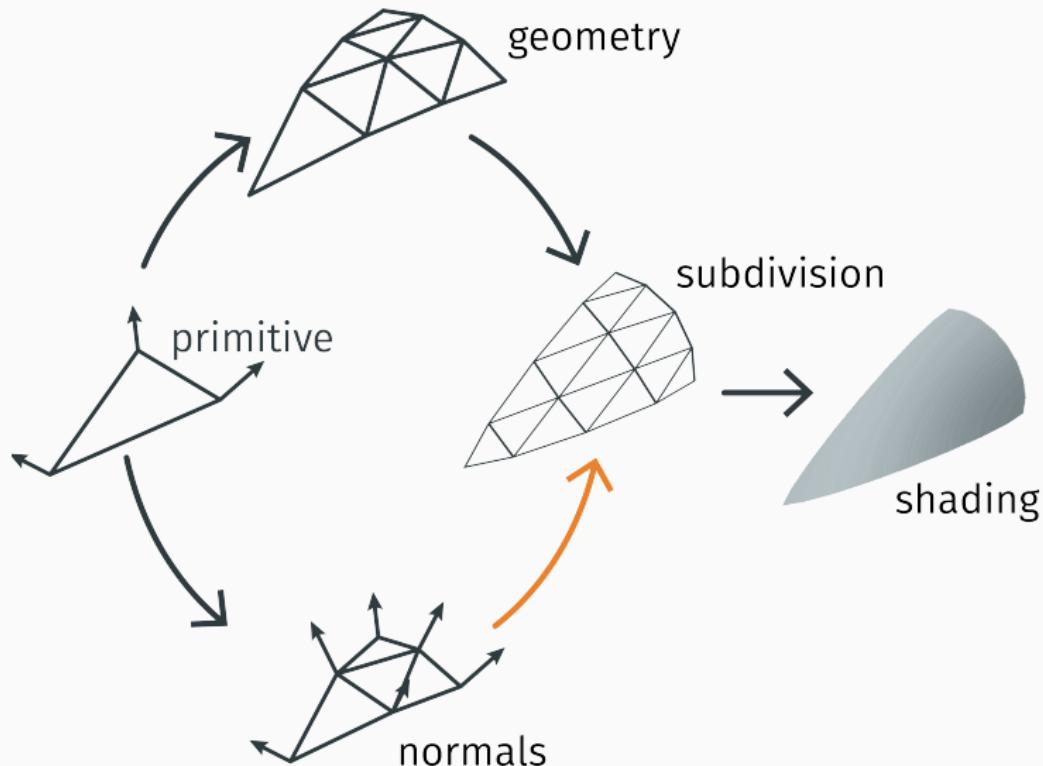
$$h_{110} = N_1 + N_2 - v_{12}(P_2 - P_1)$$

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

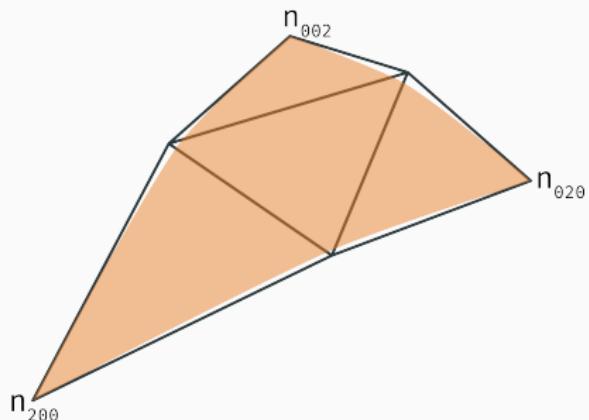
NORMALS - RESULT



OVERVIEW



QUADRATIC PATCH

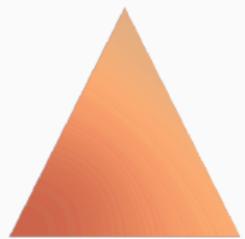
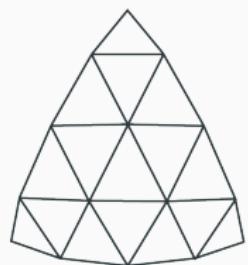
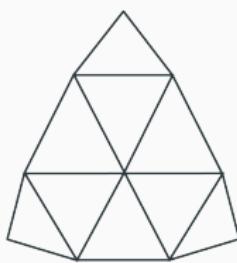
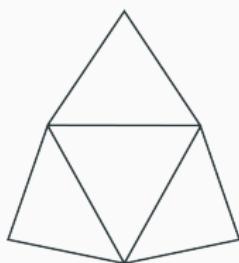
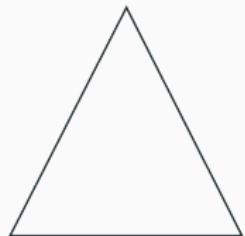


$$w = 1 - u - v$$

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

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

LEVEL OF DETAIL



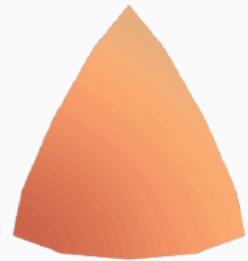
0



1

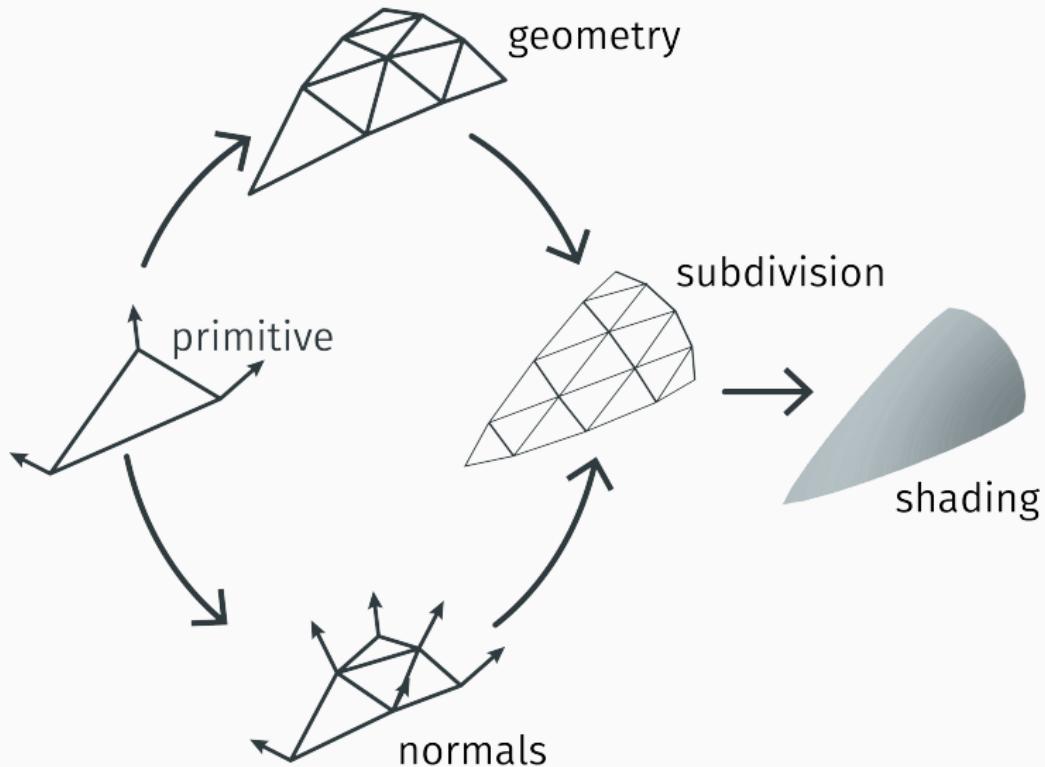


2



3

OVERVIEW



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

¹Vlachos et al.

PN triangles have:²

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

²Jiao and Alexander

SHARP EDGES



mesh



blunt

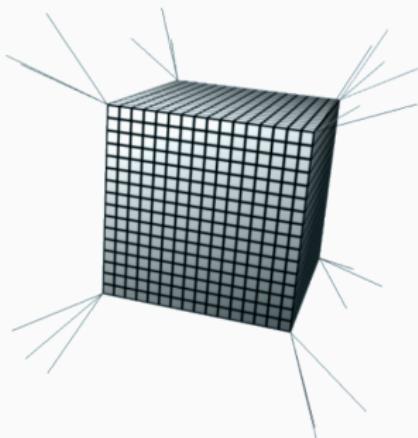


mesh



sharp

SEPARATE NORMALS



normals



cracks

GRAPHICS PIPELINE

HARDWARE - PIPELINES



2001

2015

HARDWARE - PIPELINES



2001

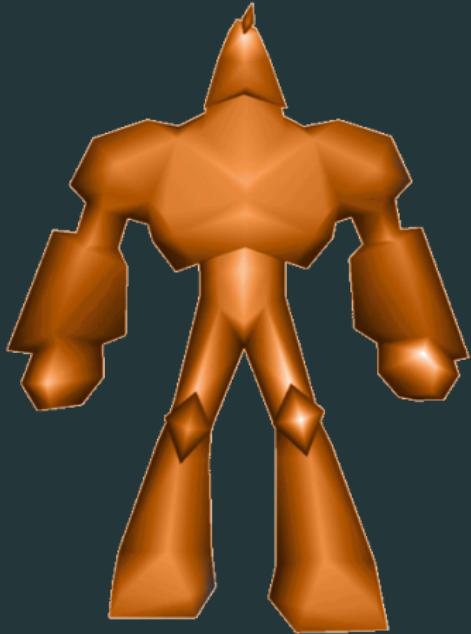


2015

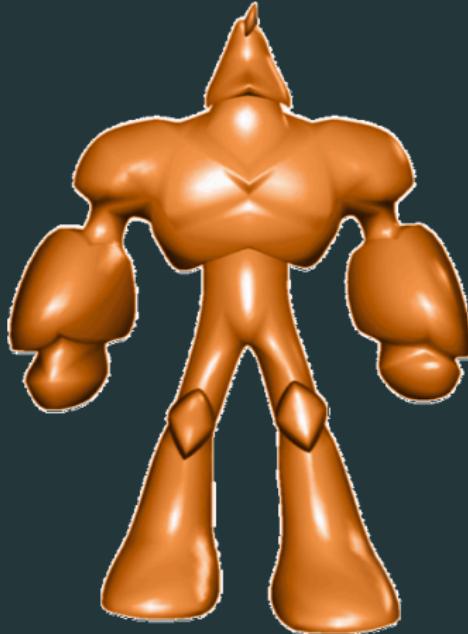
CONCLUSION

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.