2015-12-14

Point Normal triangles

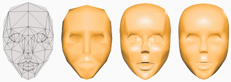
POINT NORMAL TRIANGLES

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

[Rick]

1. Welcome everybody.
2. Abbreviation PN
3. Vlachos and others (2001)
4. Improve **visual quality** of triangle based **real time entertainment**, e.g. Computer games

Point Normal triangles



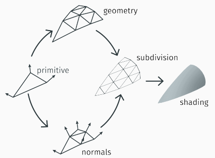
[Rick]

1. Tell what can be seen on the slide.
2. Stand still by the geometry
3. PN triangle -> more **organic shapes** (and **continuous surface**)
4. Relatively easy to extend existing ‘pipeline’

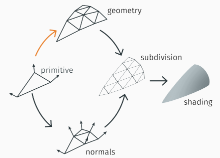
[Rick]

1. Explain (vaguely) process from input to result (prev. slide) -> input -> single -> mesh
2. Start: Construction of single pn triangle -> Laura

Point Normal triangles
└─Single PN Triangle



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

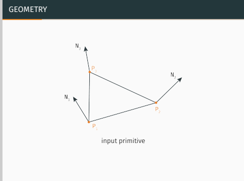


Laura From input primitive to geometric component of PN triangle

2015-12-14

Point Normal triangles

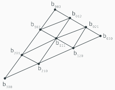
└Single PN Triangle



1. **[Laura]** This a standard triangle primitive, defined by its vertices and normals.
2. **[Laura]** Note that we only have this input primitive, without information about its neighbors.

Point Normal triangles

└─Single PN Triangle



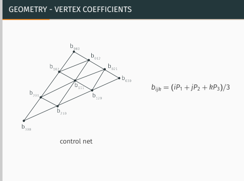
control net

1. [Laura] This is the topology of the final patch
2. [Laura] Mention which vertices are vertex, tangent and center coefficient.

2015-12-14

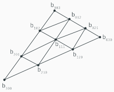
Point Normal triangles

└─Single PN Triangle



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

Point Normal triangles
└Single PN Triangle



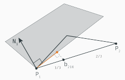
control net

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

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

1. [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

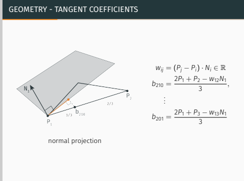
$w_{ij} = (P_j - P_i) \cdot N_i \in \mathbb{R}$

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

$\vdots$

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

[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.

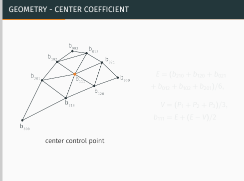


[Laura] Define a plane using the closest vertex and its normal. Find the point on this plane that is closest to the uniformly distributed point.

2015-12-14

Point Normal triangles

└─Single PN Triangle

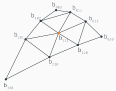


1. [Laura] Note that this is the result of the previous step -> now only center coefficient is left.

2015-12-14

Point Normal triangles
└─Single PN Triangle

GEOMETRY - CENTER COEFFICIENT



center control point

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

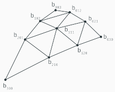
$$V = (P_1 + P_2 + P_3)/3,$$

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

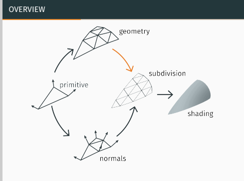
1. **[Laura]** Average of the tangent coefficients plus half the difference between the tangent and vertex coefficients.

2015-12-14

Point Normal triangles
└Single PN Triangle

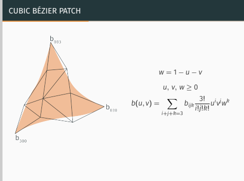


[Laura] Results



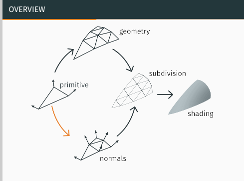
[Rick]

1. From **geometry** to getting a number of **flat triangles**
2. **Number of flat triangles** depends on the **level of detail** -> Later
3. Noticed: Cubic bézier patch control net -> next slide.



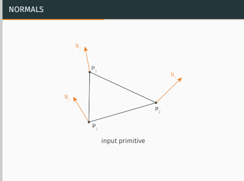
[Rick]

1. Any point is -> a **unique convex combination** of the control vertices. (Barycentric coordinates)
2. Computations done with this **cubic bezier triangle** function in **Bernstein form.**



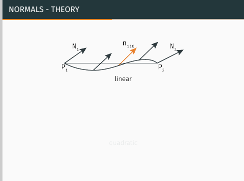
[Rick]

1. **Seen**: Construction of cubic bézier **geometry**
2. Summarize (?)
3. **Now**: Construction of normals -> next slide



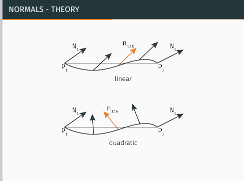
[Rick]

1. This step: Again input primitive -> only vertices and normals are known
2. Stress (again): no knowledge of adjacent triangles



[Rick]

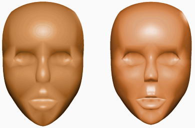
1. Stress: Two (more but ...) methods are possible
2. **First**: <u>Linear</u> -> GOOD for parabolic curve -> BAD with inflection points (**cubic**)



[Rick]

1. **Second** option: <u>quadratic</u> does capture inflection points
2. **Higher order?** not necessary (performance vs quality)

NORMALS - EXAMPLE



linear quadratic

[Rick]

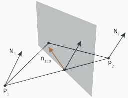
1. Theory is just fine, but what about some results!
2. Which one is more pleasing/better?

2015-12-14

Point Normal triangles

└Single PN Triangle

NORMALS - THEORY



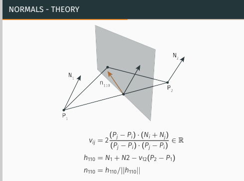
$$v_{ij} = 2\frac{(P_j - P_i)\cdot(N_i + N_j)}{(P_j - P_i)\cdot(P_j - P_i)} \in \mathbb{R}$$

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

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

[Rick]

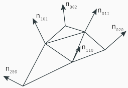
1. Intuitively -> take the average of the two normals.
2. GOOD parabolic (different normals). BAD cubic (same normals)
3. Drawing (?)
4. Solution -> reflect the average.



[Rick]

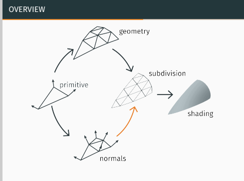
1. Intuitively -> take the average of the two normals.
2. GOOD parabolic (different normals). BAD cubic (same normals)
3. Drawing (?)
4. Solution -> reflect the average.

NORMALS - RESULT



[Rick]

1. The 6 control points (normals) for the parametrization used during subdivision. -> Laura

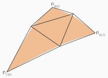

OVERVIEW
geometry
subdivision
primitive
shading
normals

laura

2015-12-14

Point Normal triangles
└Single PN Triangle

QUADRATIC PATCH

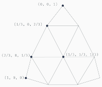


$$w = 1 - u - v$$
$$u, v, w \geq 0$$
$$n(u,v) = \sum_{i+j+k=2} n_{ijk} u^i v^j w^k$$

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

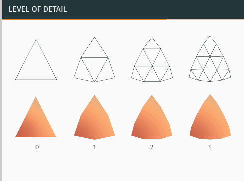
2015-12-14

Point Normal triangles
└Single PN Triangle

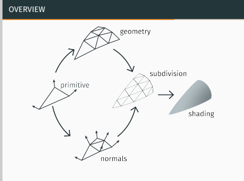


(u, v, w)

[Laura] LOD = 2



[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

[Rick]

1. Seen: Construction and parametrization of a Single PN Triangle
2. Now: How does is work in mesh of multiple triangles

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

[1] vlachos2001curved

[Rick]Stress: important property (of Pn traignle) for a mesh -> else **gaps** and other **artifacts** will emerge -> continuity

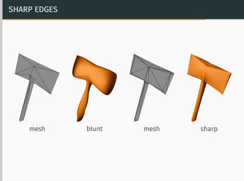
PN triangles have:[2]

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

[2]jiao2005parallel

[Rick]

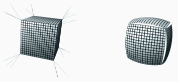
1. Continuity C0 is important -> no gaps
2. Continuity C0 along edges only when shared normals -> later example of separate normals
3. C^2 desirable but not a real problem -> given too us by literature



[Rick]

1. **Problem** Pn triangles perform well, but not always desired results.
2. **Example** Sharp edges.
3. **Solution** insert more triangles at the sharp edges -> model needs to be changed :(

SEPARATE NORMALS

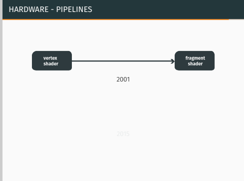


normals cracks

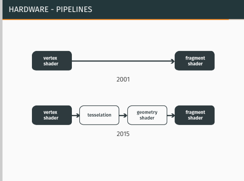
[Rick]

1. Pn triangle work -> restrict the model -> shared normals
2. Describe slides -> Pn triangle can thus not deal with this
3. Extension exists -> PN-AEN (Adjacent edge normals) (Nvidia)

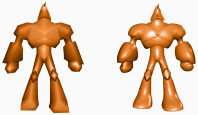
[Laura] Blaat



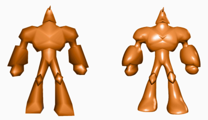
[Laura] Great part of the paper stresses the point that it can easily be implemented as a preprocessing step (CPU.
2001 pipeline (OpenGl 1.3)



[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.



[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.



b