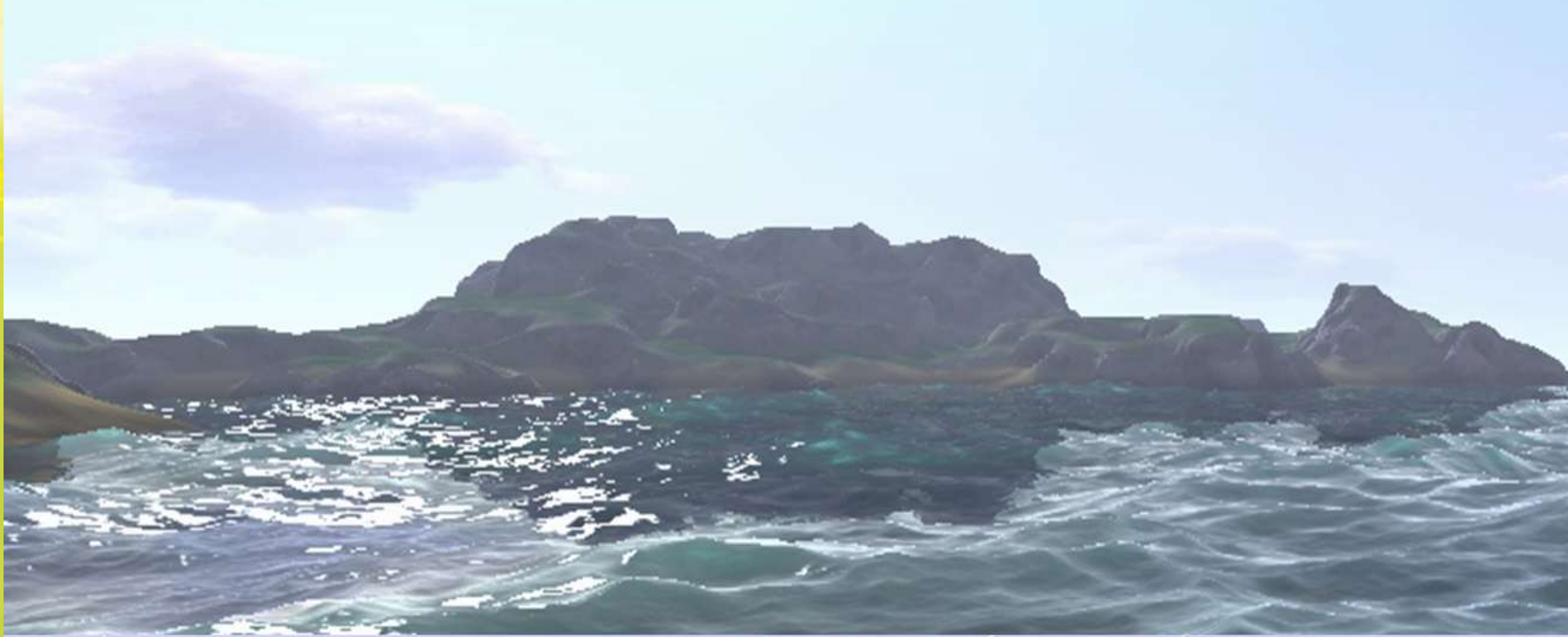




Direct3D 11 曲面细分技术

Tianyun Ni (倪天芸), Nvidia



- 游戏画面的每个像素被精致地渲染, 然而只呈现少量的几何细节。

简介(1)

- 为实时渲染的游戏场景提供类似3D电影中的特有的及其丰富的画面
- 产生更多细节，减少
 - 输入：低精度网格+置换贴图
 - 大量的三角形在GPU上生成
- 大量昂贵的操作可移到低精度网格上，从而降低操作成本

简介(2)

- 灵活掌控 所需的细节程度
 - 在实时运行的过程当中调节适当精度的网格
 - 仅在必需的地方添加细节
 - 不同程度的细节变化是连续顺畅的。
- 可应用于游戏场景中任何需要丰富画面的地方
- 主要探讨在游戏角色上的应用



第三届中国游戏开发者大会
China Game Developers Conference 2010

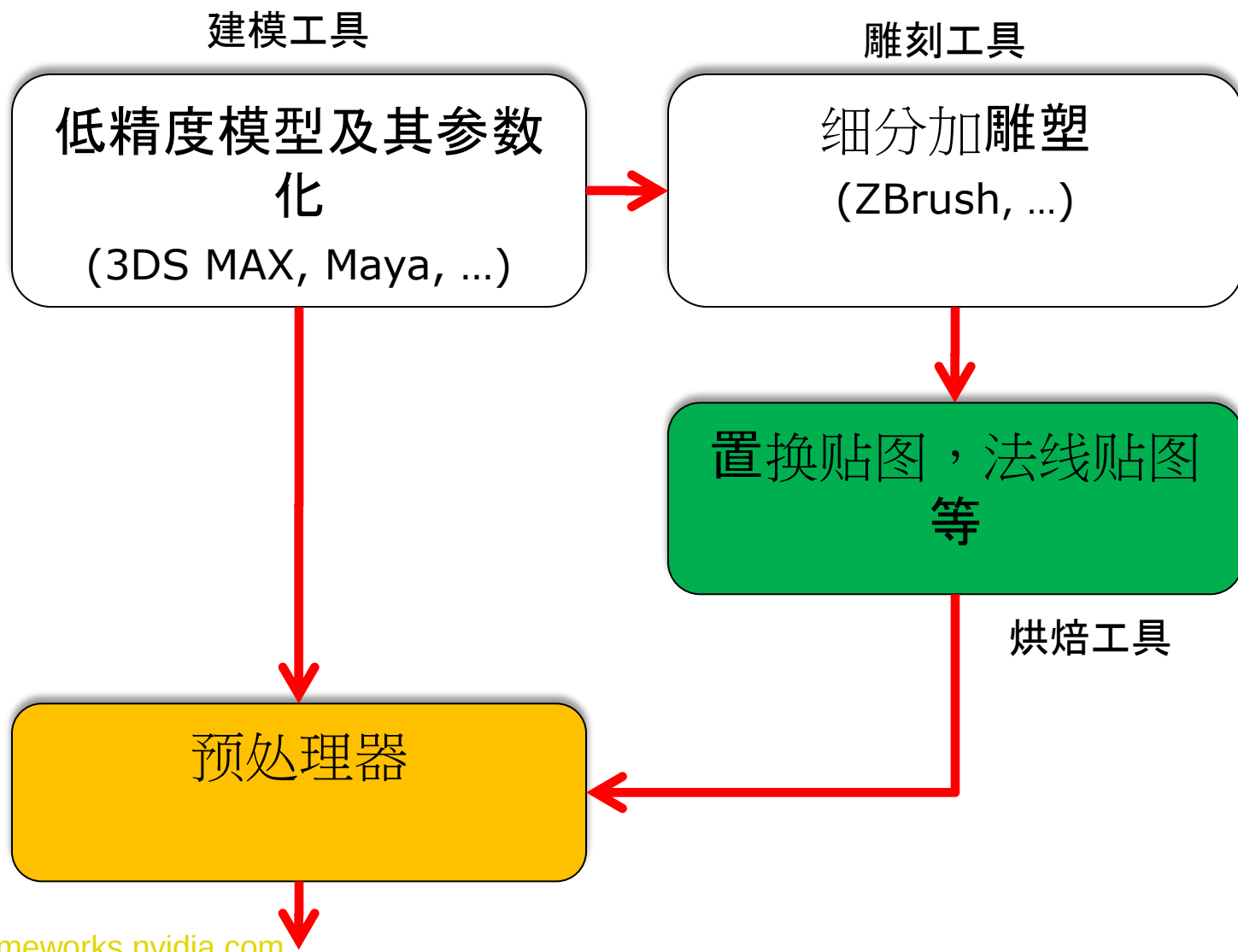
演示Demos

模型打造管线流程

- 现存的 DCC 工具有: 3dsMax, Maya, Blender, Zbrush, Mudbox, Modo, Grazy Bump, 等等。
 - 建模工具
 - 雕刻工具
 - 烘焙工具 (生成置换贴图, 法线贴图, 闭塞贴图, 等等)



模型打造管线流程





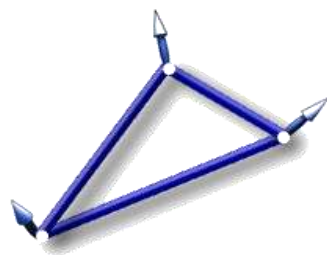
曲面细分算法概观

曲面细分算法

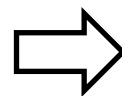
- 为您的游戏模型选择合适的曲面细分算法
 - 网格连接类型
 - 实现的复杂性
 - 在性能速度和视觉质量间权衡
- 线性插值
- 本地构造算法
 - PN , Phong Tessellation
- 近似 Catmull-Clark 细分

PN 细分

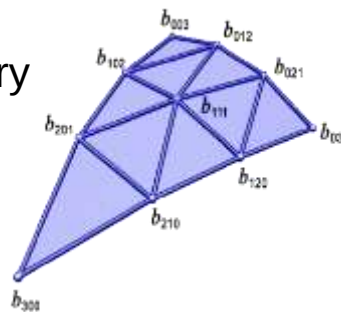
- “Curved PN Triangles”, by Alex Vlachos, Jörg Peters, Chas Boyd, and Jason Mitchell, I3D 2001.



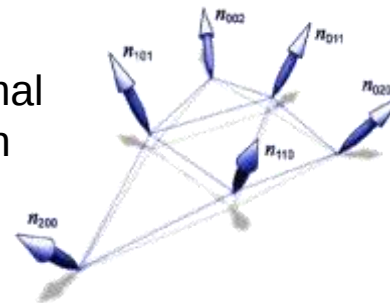
输入三角形



Geometry patch



Normal patch



- “PN Quads”, by Jörg Peters, 2008.

http://www.cise.ufl.edu/submit/files/file_020f70fe71888f602530143e2e326be2.pdf

- 除了计算内部控制点的计算方法不同，其他控制点采用相同的计算公式

Phong 细分

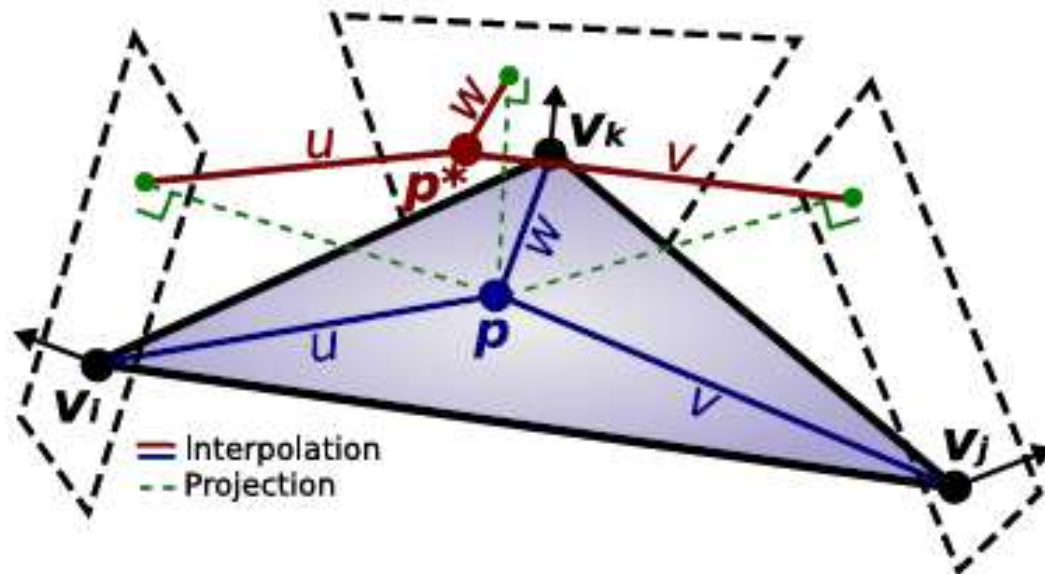


Figure 3: Phong Tessellation principle. Instead of interpolating normals as in Phong Shading, we interpolate projection onto vertices tangent plane to define a curve geometry for each triangle.

- From *Boubekeur and Alexa*, Siggraph Asia 2008

Phong 细分

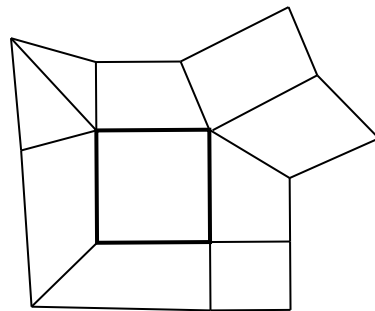


Boubekeur and Alexa, 2008

- 非常简单的算法
- 产生计算错误当曲线上的点改变了曲率（二阶导数）的符号改变
- 当输入网格是相当密集时，计算结果效果很好

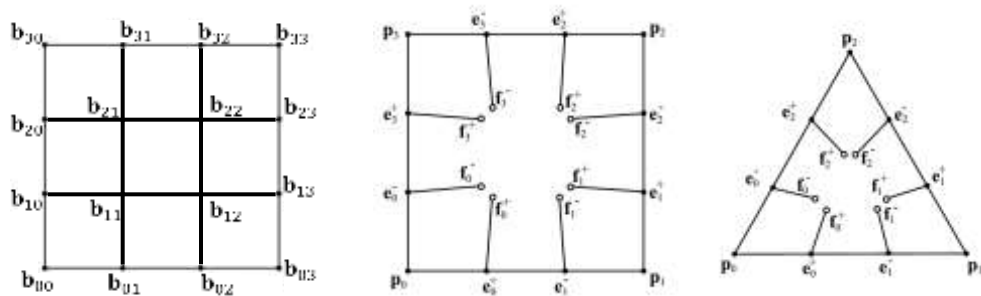
近似 Catmull-Clark 的细分

- 提供电影品质的画面
 - Catmull-Clark 细分广泛应用于电影制作以及雕刻和建模工具中
 - 适合于以四边形为主的网格建模，网格中可含少量的三角形
- 近似而非插值
- 输入不只是一个面，还需要它的所有邻点。



近似 Catmull-Clark 的细分

- 输入网格中的每一个面对应一个曲面段
 - 常规四边形 $\rightarrow$ 三次贝塞尔曲面段
 - 非常规四边形 $\rightarrow$ 四边形的 gregory 曲面段
 - 三角形 $\rightarrow$ 三角形的 gregory 曲面段



细分算法的综合比较

	一个 patch 图元中的顶点数	曲面段所含控制点的数目	输入网格的连接种类	曲面质量及光滑度
Phong	3 Or 4	6 Or 9	 	瑕疵出现在曲率符号变化或有较大曲率变化的区域
PN	3 Or 4	10+6 Or 16+9	 	瑕疵出现在有较大曲率变化的区域
Gregory ACC	16 to 32	15 Or 20	 	类似于 CC 细分生成的曲面

- 不同的曲面细分算法区别在
 - Patch 图元中的顶点数
 - 曲面段控制点的计算 (in Hull Shader)
 - 曲面上的点的计算(in Domain Shade)

DirectX 11 细分管线图

DirectX 11 曲面细分管线图

Patch 图元



Input Assembler

Vertex Shader

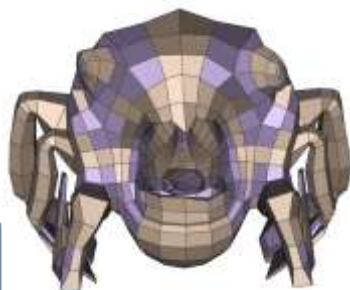
Hull Shader

Tessellator

Domain Shader

Setup/Raster

Pixel Shader



Skining

输入网格
(可分解成多个patch 图元)
置换贴图
法线贴图 (可有可无)

```
struct VERTEX
{
    float3 vPosition           : POSITION;
    float2 vUV                 : TEXCOORD0;
    float3 vTangent            : TANGENT;
    uint4 vBones                : BONES;
    float4 vWeights             : WEIGHTS;
};
```

DirectX 11 曲面细分管线图

Patch 图元



Input Assembler

Vertex Shader

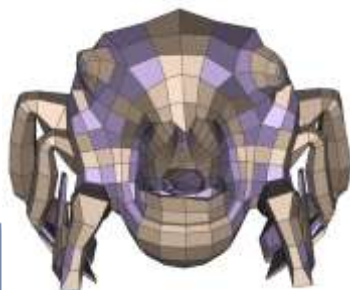
Hull Shader

Tessellator

Domain Shader

Setup/Raster

Pixel Shader



输入网格
(可分解成多个patch primitives)
置换贴图
法线贴图 (可有可无)

Skinning

- 计算控制点
- 计算 LOD

三角形大量生成

DirectX 11 曲面细分管线图

Patch 图元



Input Assembler

Vertex Shader

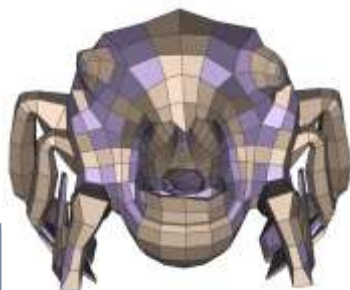
Hull Shader

Tessellator

Domain Shader

Setup/Raster

Pixel Shader



输入网格
(可分解成多个patch primitives)
置换贴图
法线贴图 (可有可无)

Skinning

- 计算控制点
- 计算 LOD

三角形大量生成

- 计算细分后表面上的点
- 用置换贴图里对应的高度值重新表面上点的位置
- 运用法线贴图取得置换贴图后新的法线 (转移到 DS?)
- 着色处理



细分后的光滑曲面



高精度网格

实际运用中的问题与 解决方案

曲面细分因子

- 域类型: 三角形 / 四边形 / 线段
- 间距:
 - 离散的 / 连续的 / 2的乘方
- 在 hull shader 中的 constant phase stage 计算曲面细分因子
 - 边细分因子
 - 面内细分因子
 - ProcessTriTessFactorsMax
 - ProcessTriTessFactorsMin
 - ProcessTriTessFactorsAvg

避免不必要的细节

- 与视角的距离
- 像素空间
 - 注意边可能是曲线而非直线
 - 建议的精度：三角形的面积大于8个像素
- 在轮廓线附近细分
 - 在细分后，平面变成曲面
 - 置换贴图的使用加大了分辨轮廓线所在的难度
- 在置换贴图里高度值得一阶导数

避免不必要的细节

- 在 hull shader 中删除所有背对的，以及在视线之外的三角形
 - 把所有的曲面因子的值设成 0
 - 这些被删除的曲面段不再被送到接下去的管线单元
- 辨别哪个三角形是背对视线的有以下注意事项
 - 在细分后，平面变成曲面
 - 置换贴图的使用加大了分辨轮廓线所在的难度

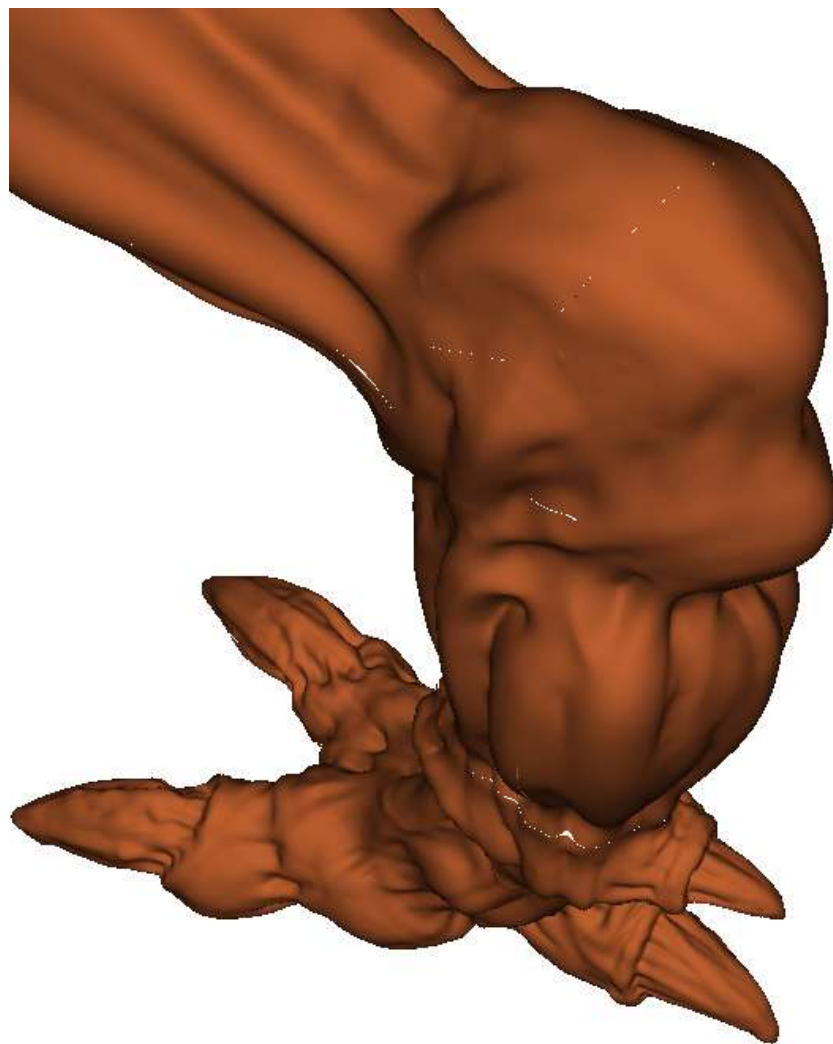
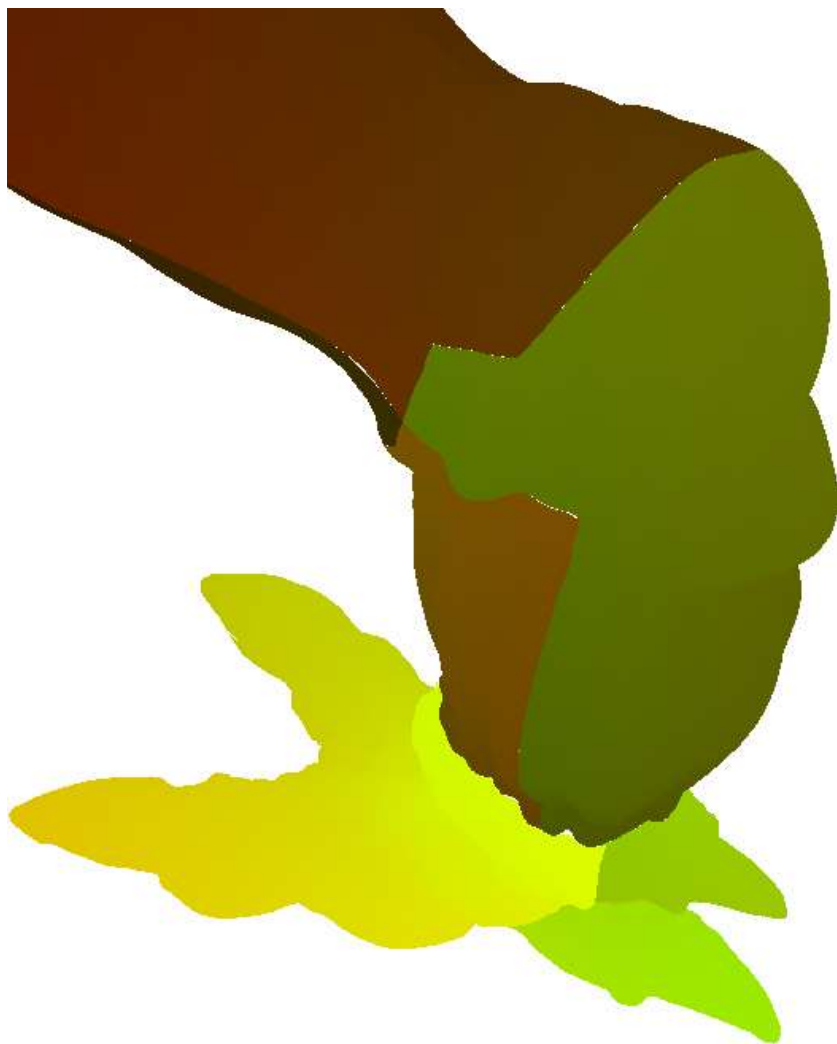
密封的曲面细分

- 未做过密封顶点位置处理的曲面细分会造成高精度网格中细小的洞
- 未做过密封法线处理的曲面细分会引起像素着色的断点
- 起因
 - 浮点数表达的有限精度
 - 纹理接缝在参数化中可能的非连续性

浮点数计算

- 浮点数的加法交换律失效:
 - $A + B + C + D \neq D + C + B + A$
 - $(A + B) + (C + D) \neq (D + C) + (B + A)$
- FMA 不同于 MUL+ADD
 - $A * a + B * b \rightarrow \text{FMA}(A * a, B, b) \neq \text{FMA}(B * b, A, a)$
- 对共享的点（角落上的，边上的），用相同的或对称的计算顺序
 - 控制点
 - 曲面

缝隙:非连续的参数化

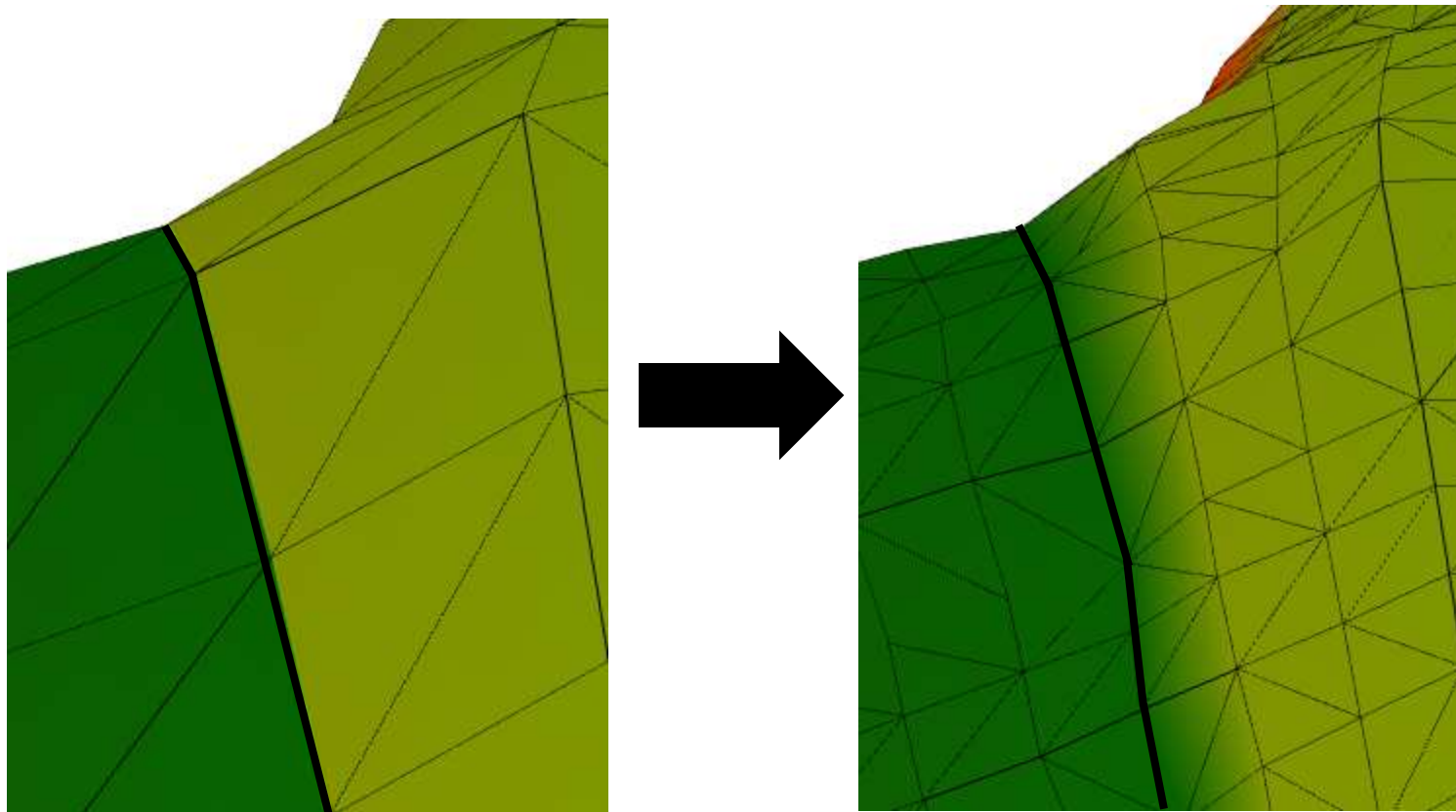


弥补缝隙的方法

- 美工可以做的：
 - 减少，隐藏纹理接缝
 - 在纹理接缝处确保置换高度时零
 - 纹理空间重新参数化，让纹理接缝的长度和方向一致
- 我们建议自动定义纹理所属性。

定义纹理所属性

- 纹理接缝处的两个三角形相邻的边共享一套纹理坐标



密封的位移置换

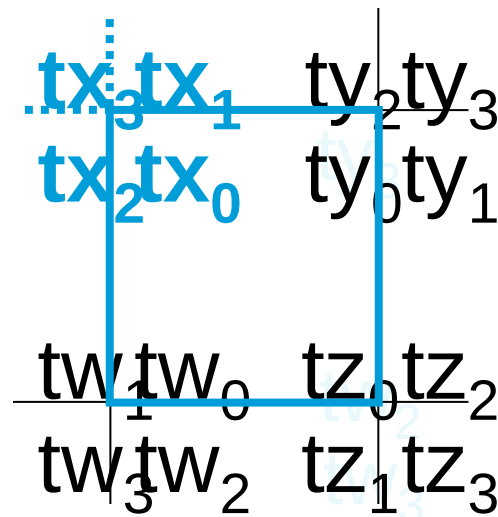
- 在每一个纹理接缝处的顶点
储存4个纹理坐标 (0: 内部;

1,2: 边; 3: 角)

- 16 (四边形), 12 (三角形)

- 根据参数曲面函数域 (u, v) 的值去决定取
哪一个纹理坐标

- 将多边形顶点的纹理坐标进行Barycentric
interpolation



折边和棱角



[KMDZ09]

- 如果两个或多个顶点共享同一位置，却又不同的法线。这些点或者在折边上，或者在棱角上。
- 方法 1: 在折边和棱角出添加小多边形
 - 美工建模时加
 - 写一个程序工具自动添加
- 方法 2: 在此类顶点做上标记
 - 网格预先处理部分加标记
 - 在shader 程式中做相应的改变



游戏实例解析

Metro 2033

Metro 2033: 游戏

- 游戏内容涉及恐怖，生存，角色扮演，射击等
- 故事情节根据Dmitry Glukhovsky的小说改编



采用Tessellation技术处理的怪物截图

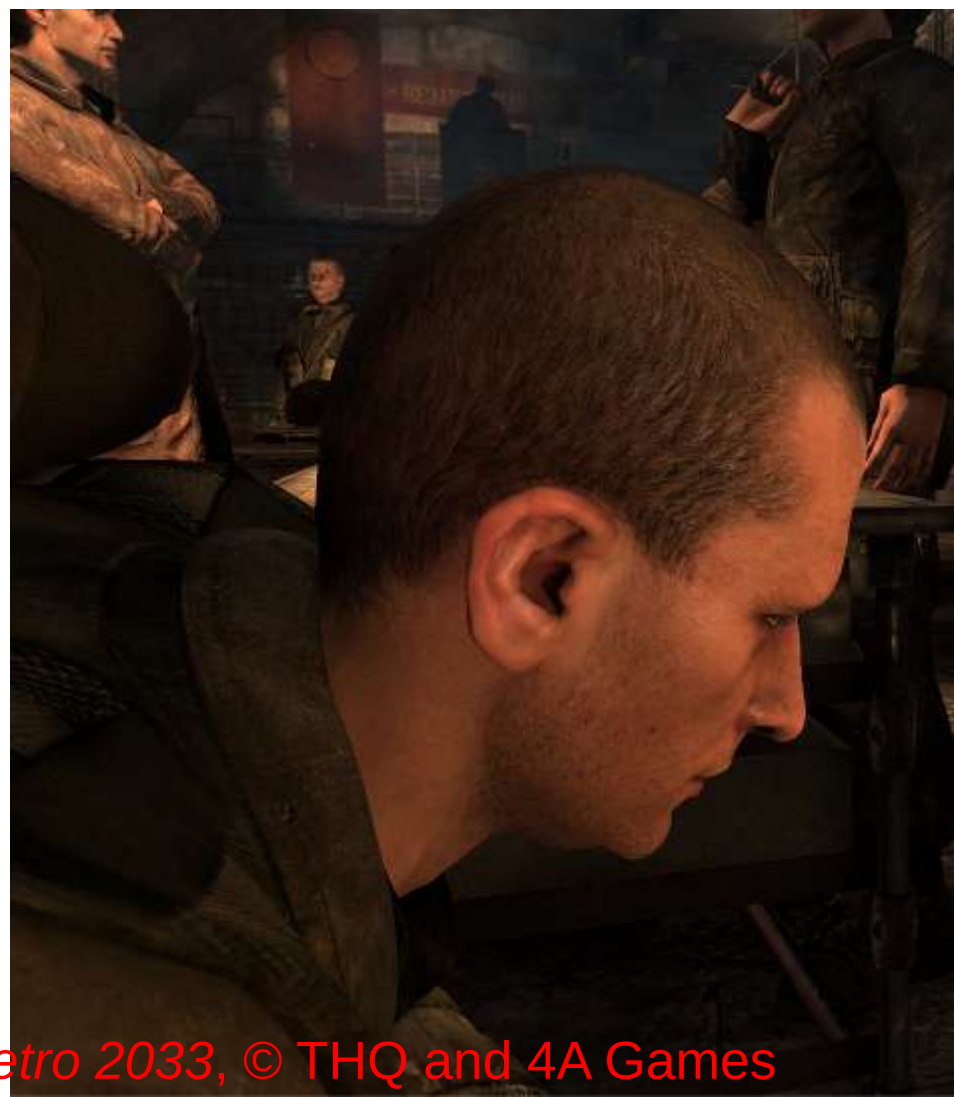
使用 Phong tessellation



更多Metro游戏里的怪物...



... 和人物角色



From Metro 2033, © THQ and 4A Games

... 更多人物角色



始于一个低精度网格



GPU硬件细分

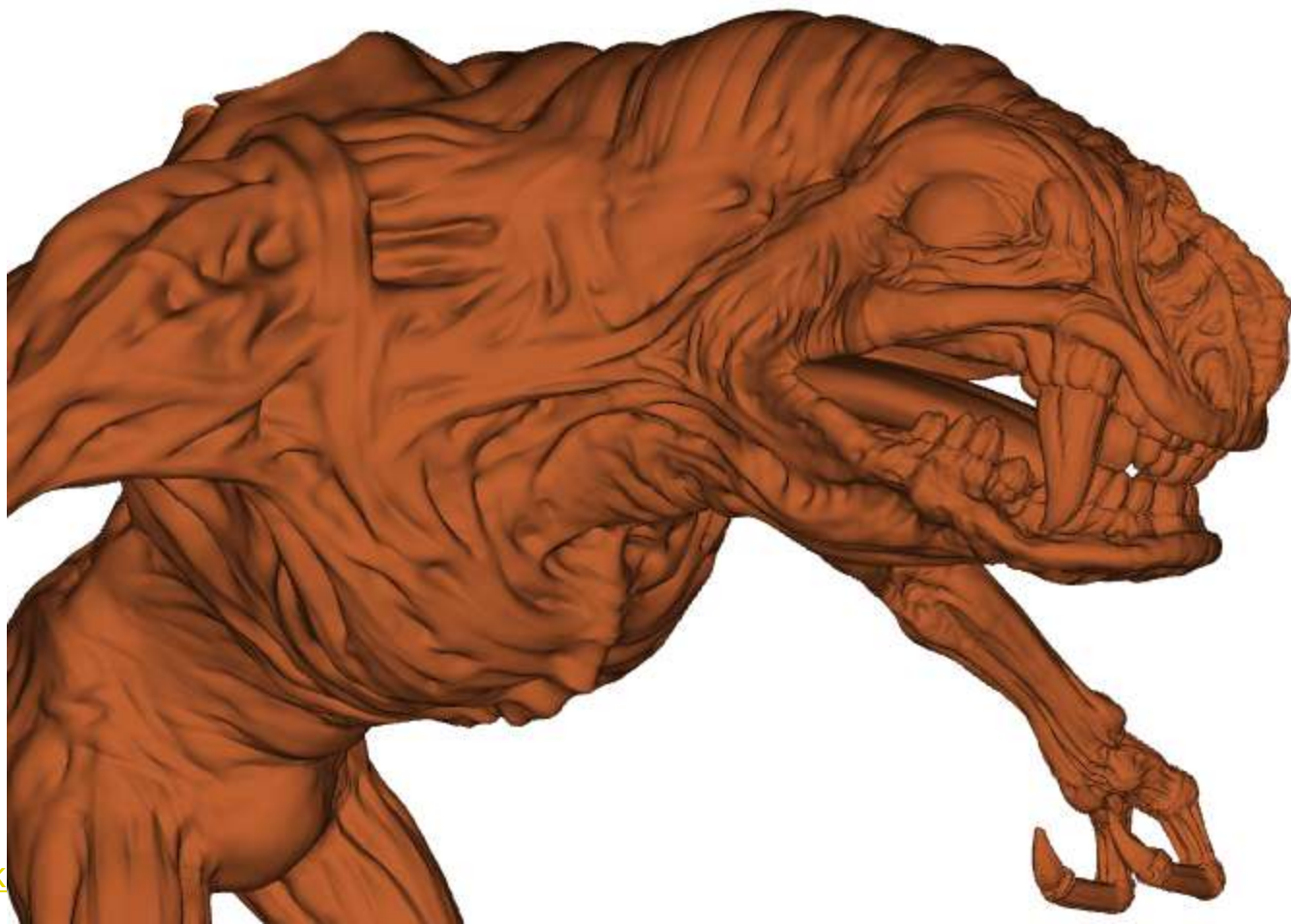


[gamework](#)



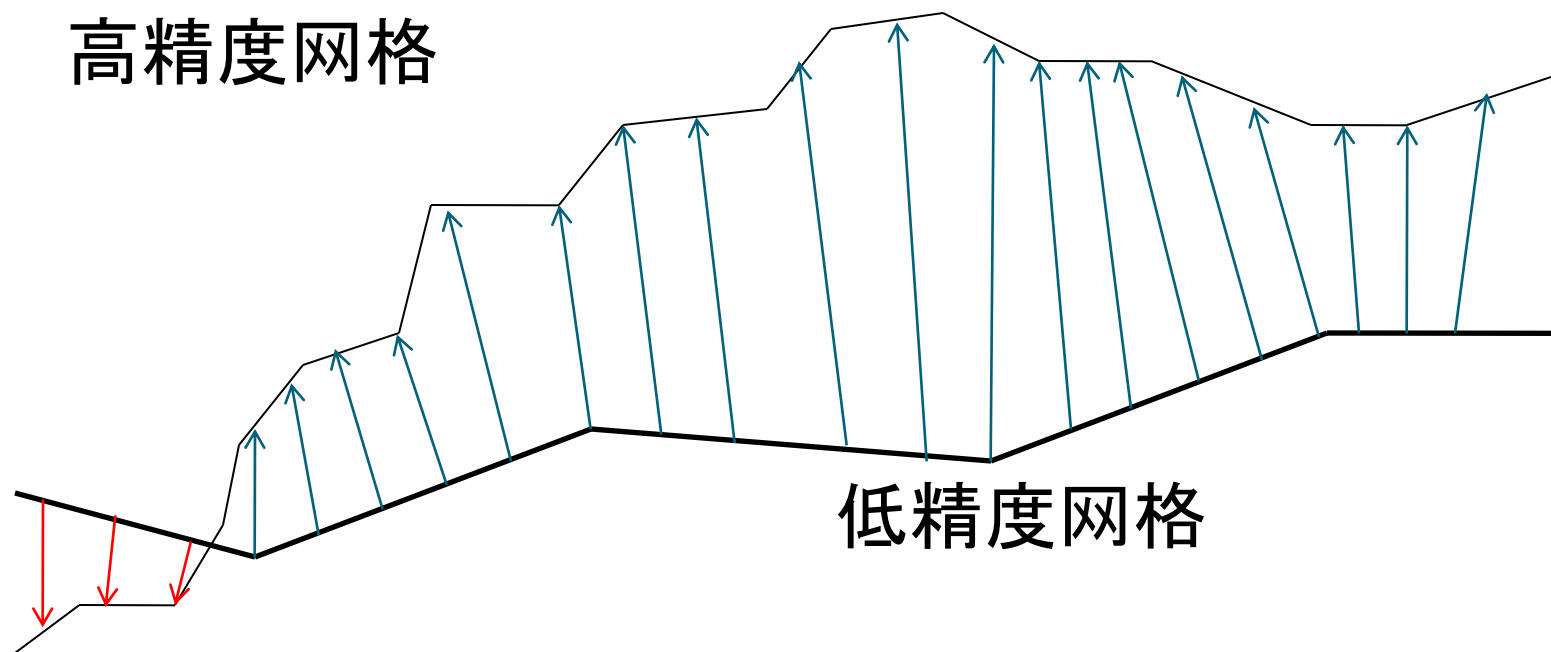
第三届中国游戏开发者大会
China Game Developers Conference 2010

加上置换贴图



计算置换贴图里的高度值

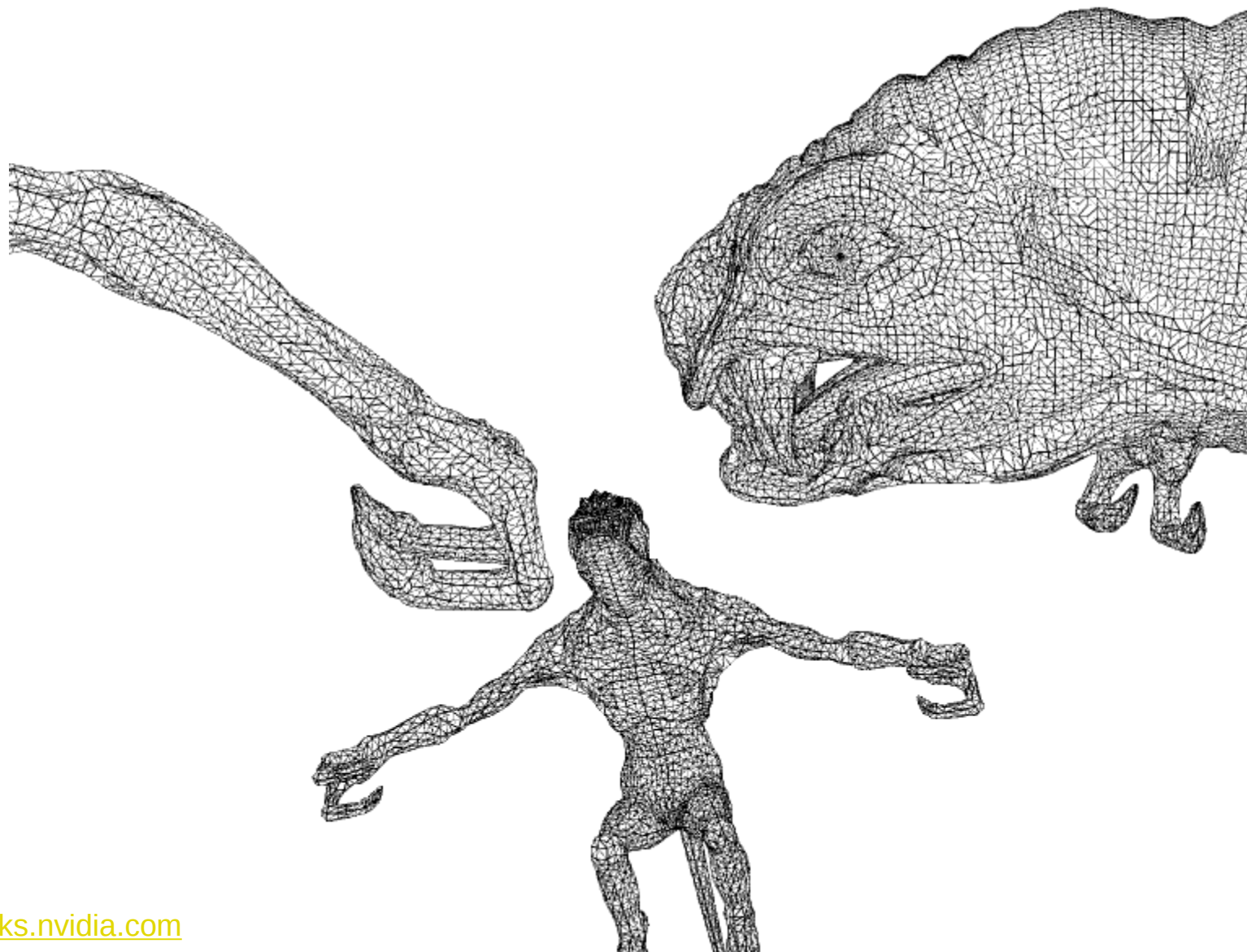
- 需要：低精度网格和最后成形的高精度网格



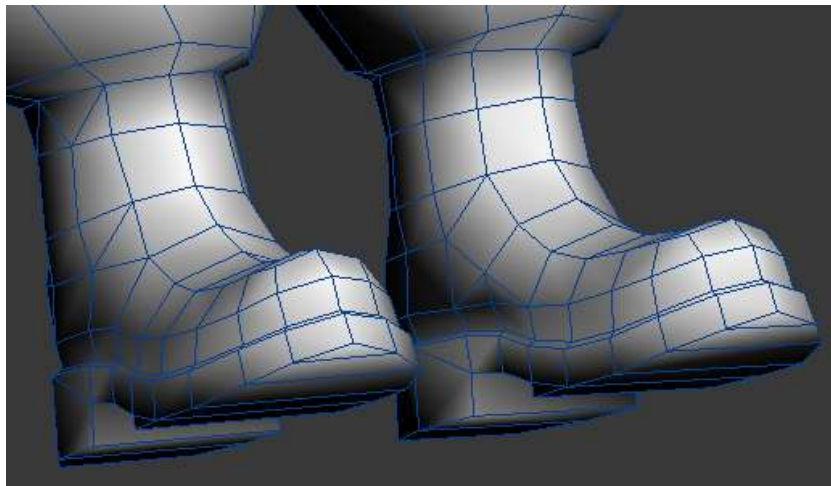
LOD 计算

- $TESS_FACT = LEN * NP * Q / DIST$
- LEN 是指world space里边的长度
- NP: 屏幕上像素的数目
- Q: 一个优化常数
- DIST: 从视角到边的中心的距离

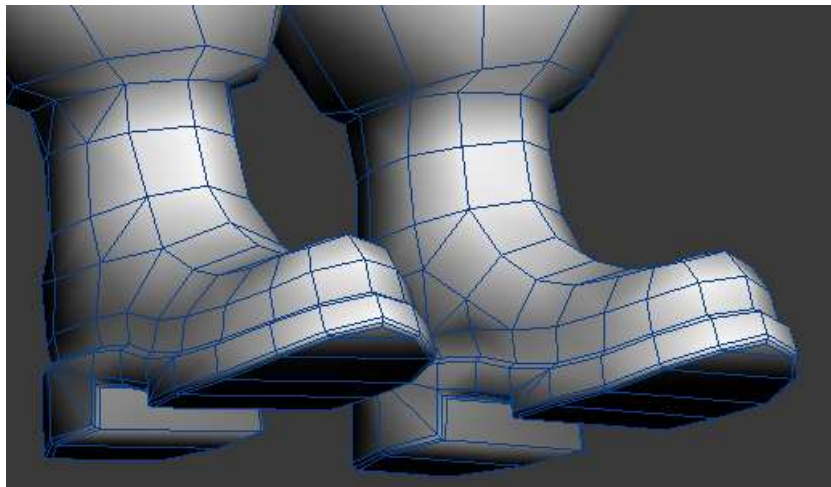
相同大小的三角形



折边细分可能有的问题

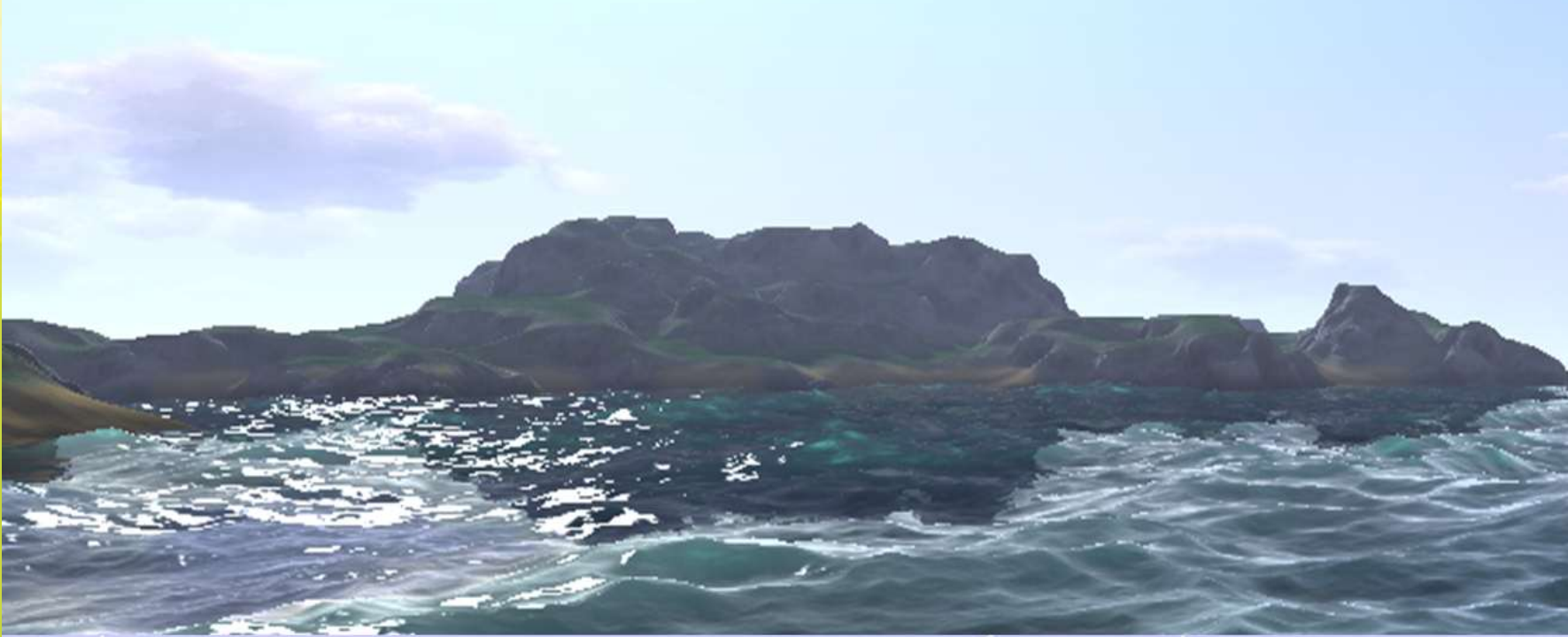


添加细小多边形可改善折边细分后的效果



结束语

- Direct3D11 细分技术灵活有效地全面提升游戏画面的细节和写实感
- 根据实际需要，选择一个适合的细分算法
- 有效地编写程式
- 将游戏的视觉效果推到一个全新的高度



谢谢!

Q&A: tni@nvidia.com