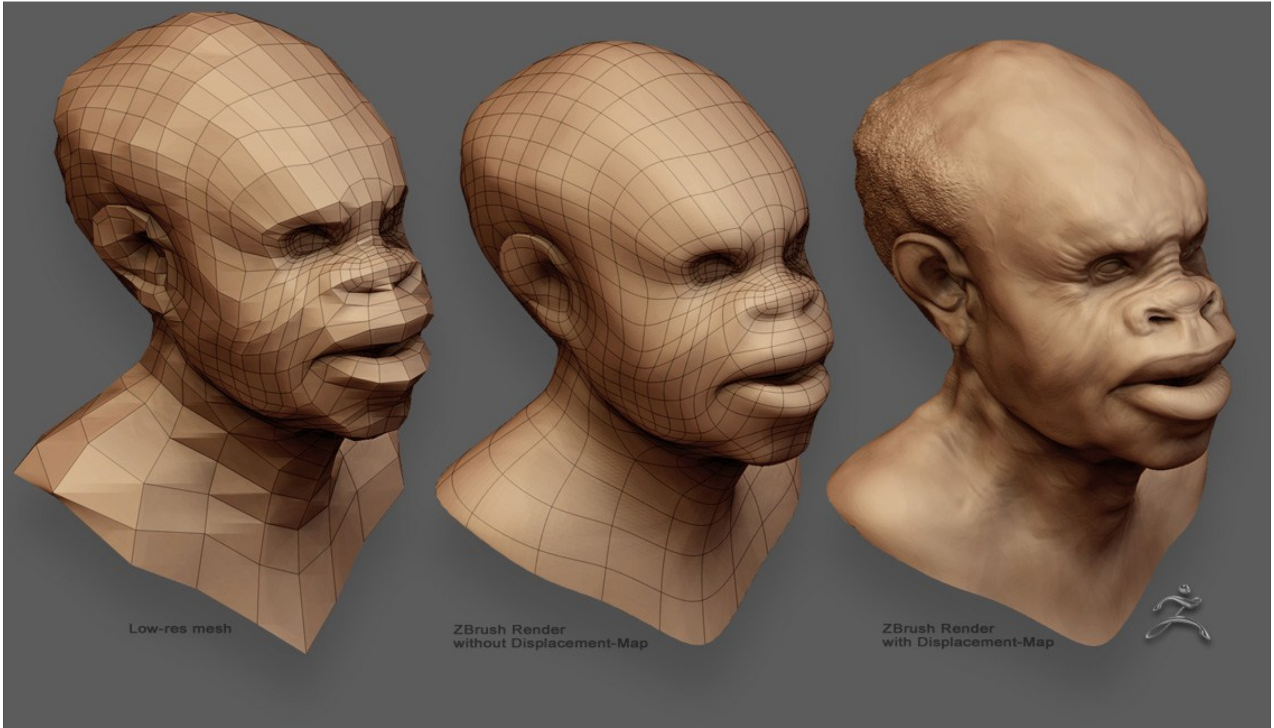
Generating Displacement from Normal Map for use in 3D Games

Kirill Dmitriev

Evgeny Makarov

NVIDIA

Adding Geometric Detail to the Game

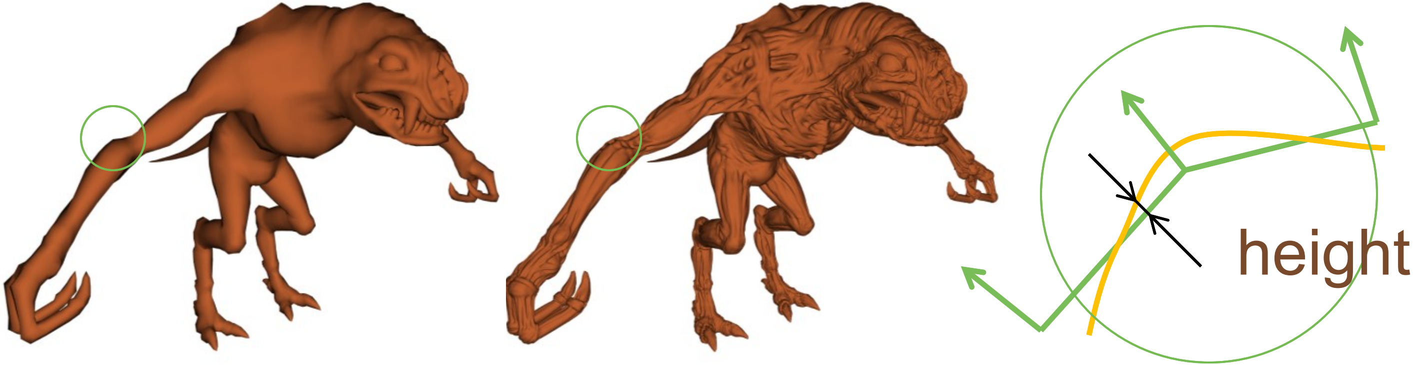

Low-res mesh
ZBrush Render without Displacement-Map
ZBrush Render with Displacement-Map

Displacement vs. Bump or POM

- Can be done using HW Tessellation
- Silhouettes
- Occlusion and self shadowing
- Correct parallax
- Works with multisampling

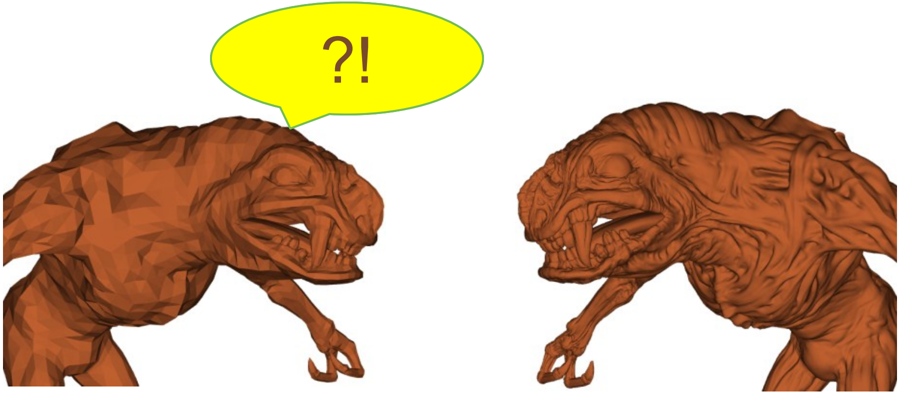
Full Artistic Pipeline

- Design coarse model
- Subdivide and add details
- Calculate displacement as mesh difference



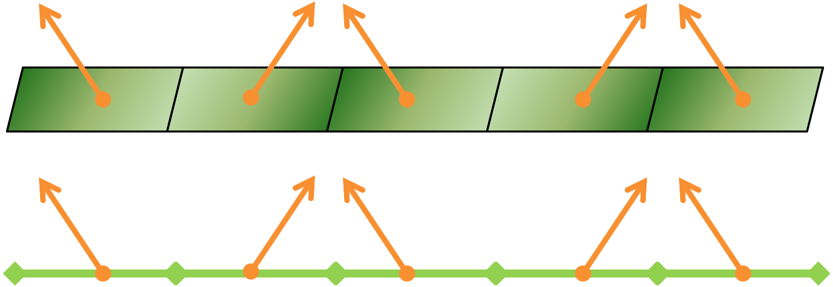
Shortcut possible?

- Have only coarse model and normal map
- Want to compute displacement



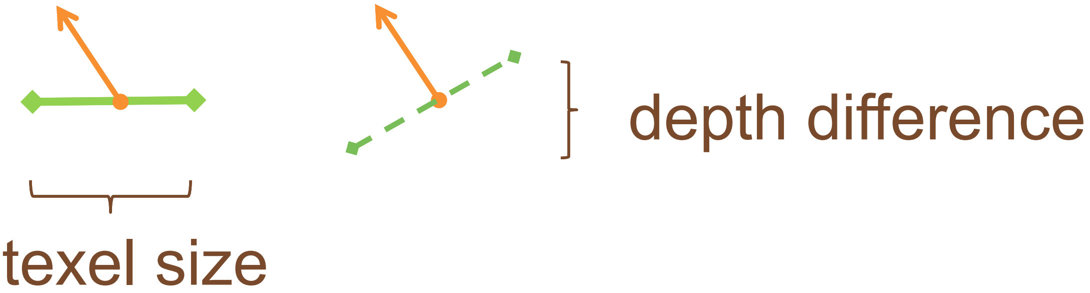
Computing Displacement in 1D

- Normal per texel is known
- Texel world size is known



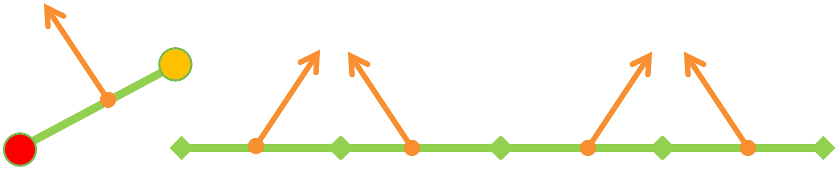
Depth Difference Map (DDM)

- Preprocess normal map to build DDM
- DDM stores height delta when crossing the texel



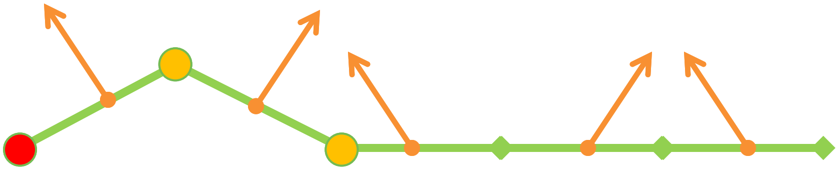
Computing Displacement in 1D

- Get displacement map by integrating depth difference map



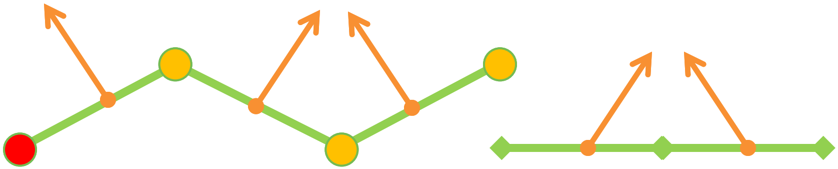
Computing Displacement in 1D

- Get displacement map by integrating depth difference map



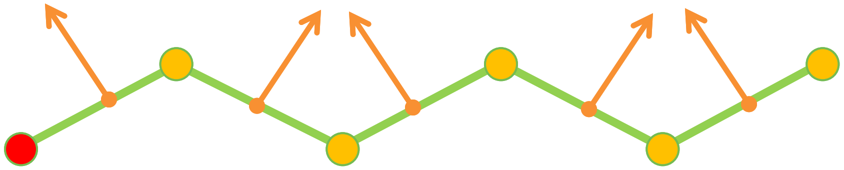
Computing Displacement in 1D

- Get displacement map by integrating depth difference map



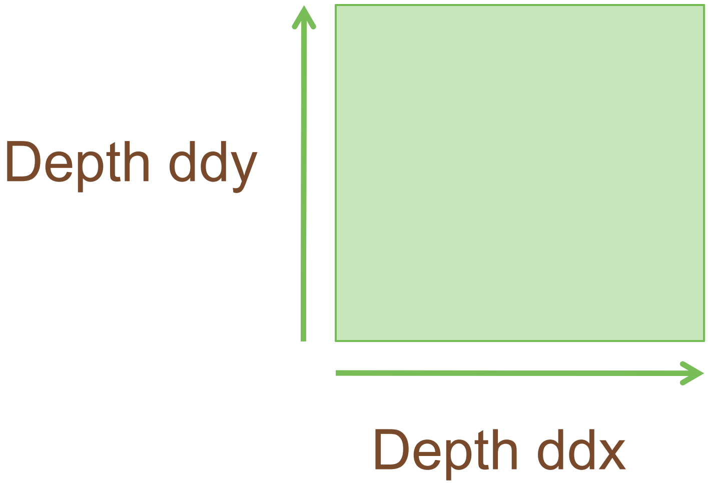
Computing Displacement in 1D

- Get displacement map by integrating depth difference map



Depth Difference in 2D

- 2D DDM stores two height deltas: for horizontal and vertical directions



Depth ddy

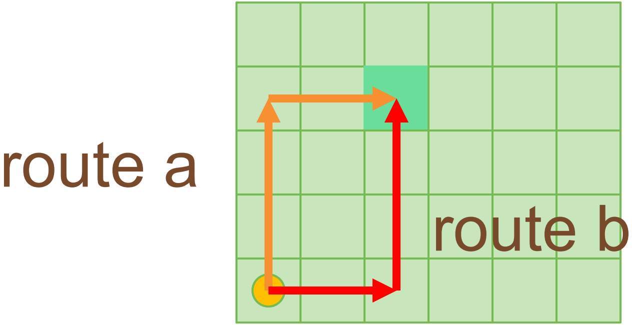
Depth ddx

Computing Displacement in 2D

- Integrate DDM starting from the corner
- Integrate DDM starting from multiple points

Errors will occur …

- Depending on the route to a point we can get different results



height(route a) ≠ height(route b)

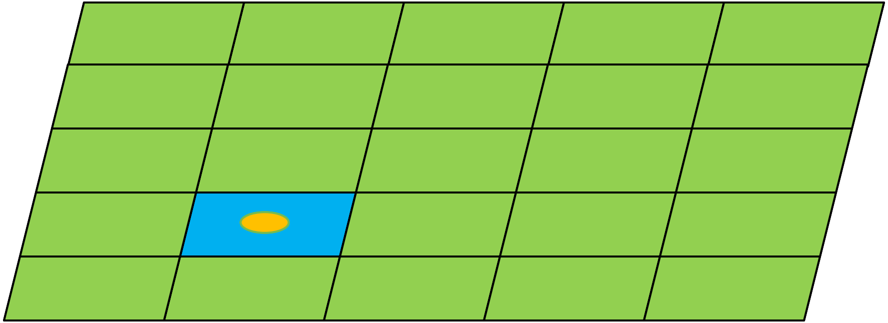
Normal Map is Lossy

- Stores average per texel. Can't reconstruct exact normal in every point
- Does not have information about discontinuities
- Stored in low precision

- All those errors accumulate

Uniform Approach

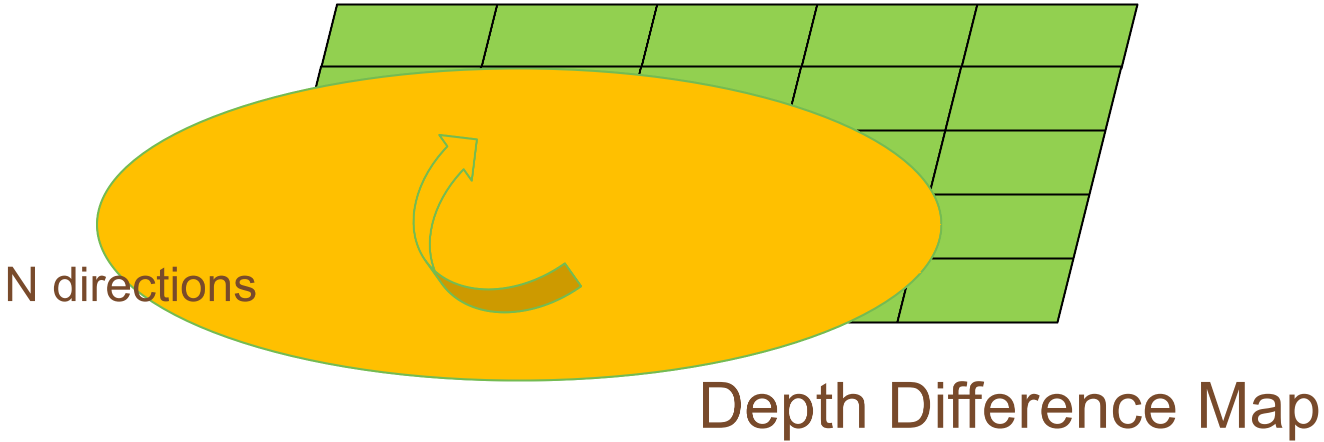
- Compute every texel independently
- No preferred points or directions



Depth Difference Map

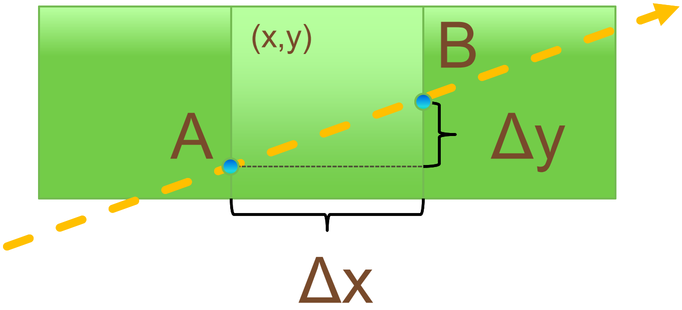
Uniform Approach

- Starting from zero depth integrate depth over circle
- Shoot N rays uniformly distributed over 360 º



Integrating Height

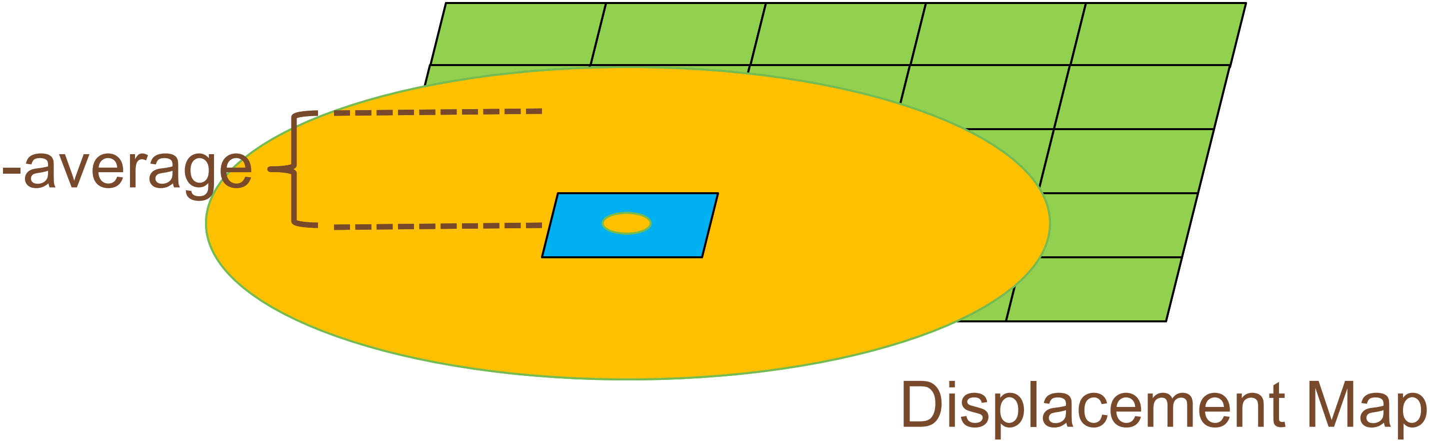
- Height is reconstructed from DDM on the fly and added to integration sum
- Integration in polar coordinates!



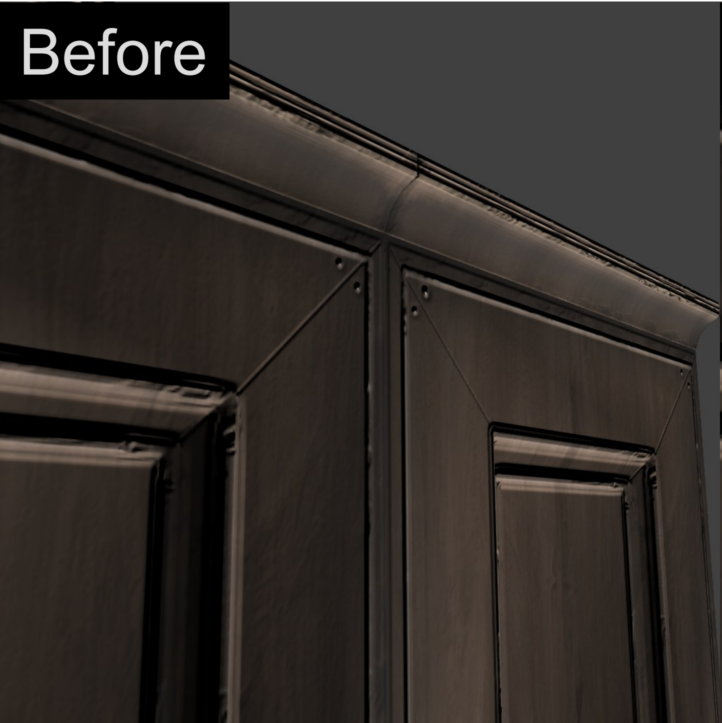
height(B) = height(A) + DDM(x,y) * float2(Δx,Δy)

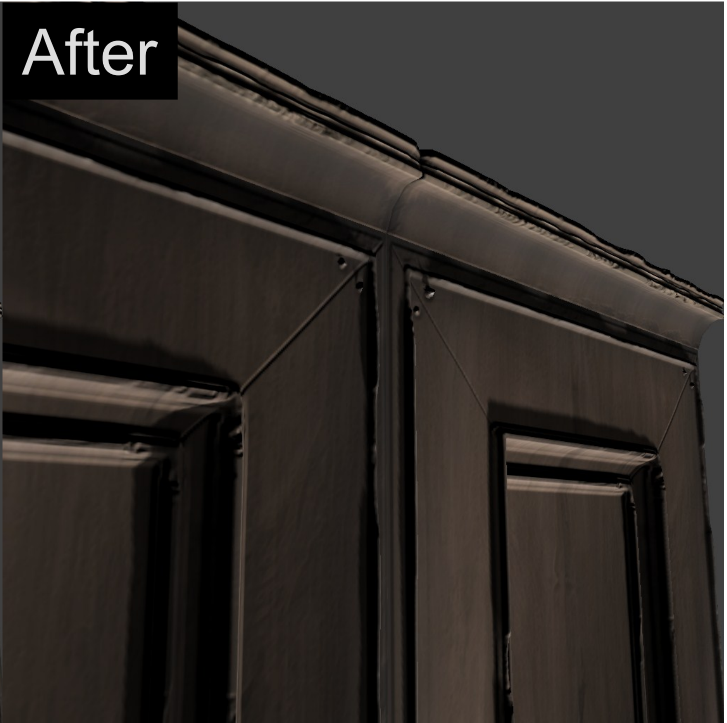
Uniform Approach

- Target zero displacement on average
- Offset texel by computed average

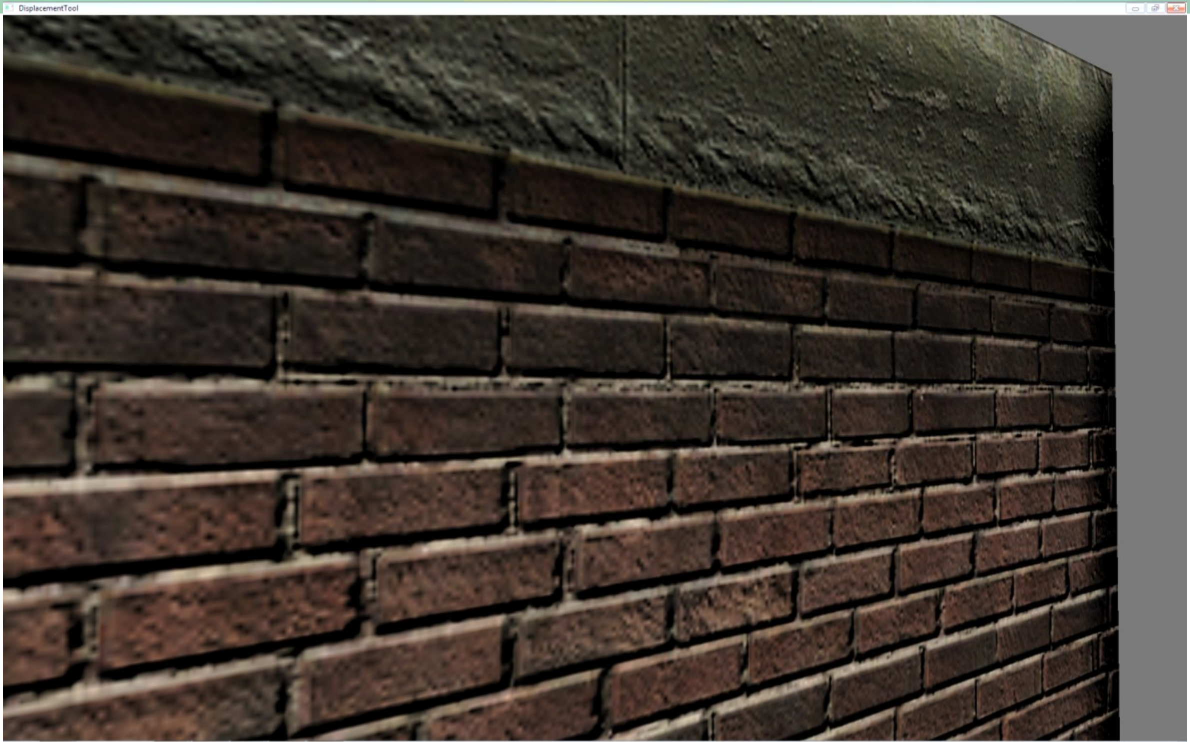


Results

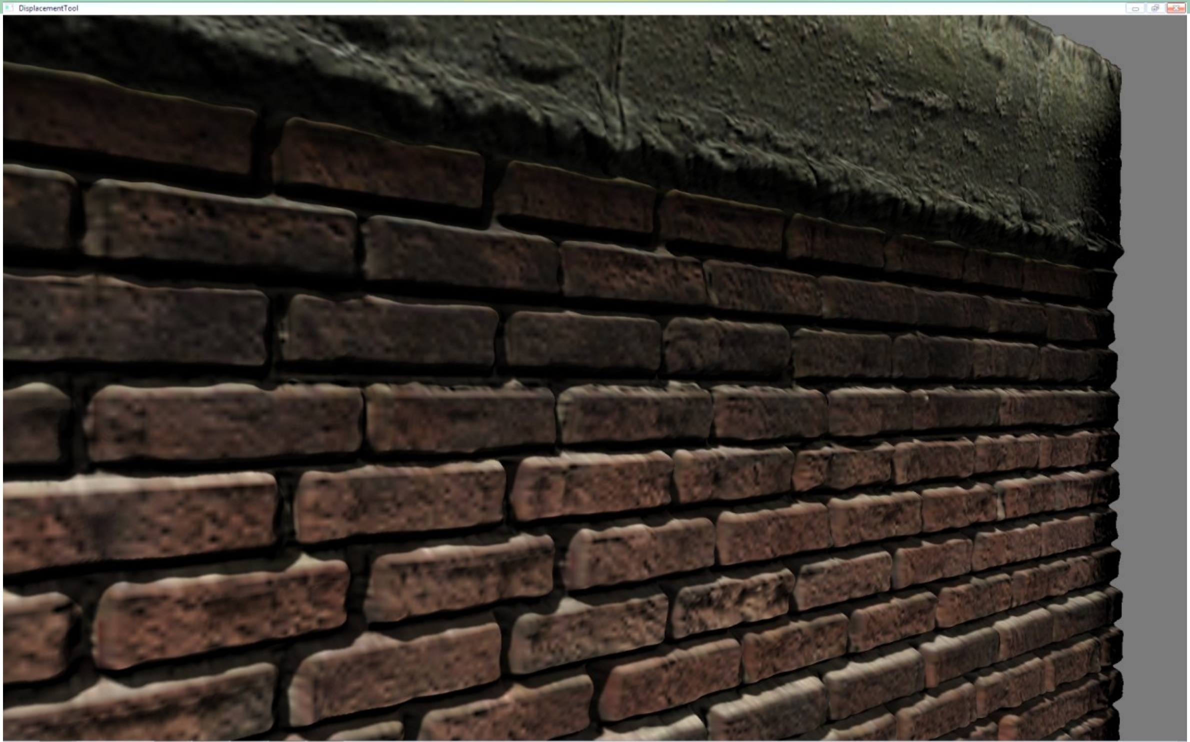




Results



Results



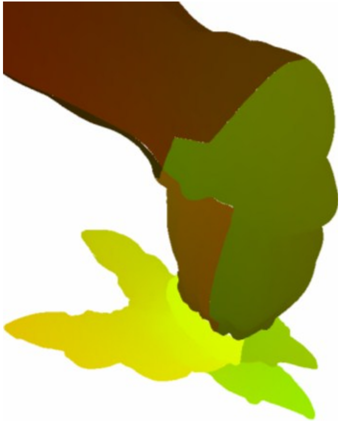
Summary

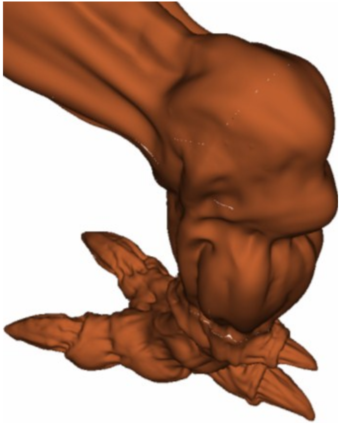
- Fully automatic solution
- Works reasonably well even on ‘painted’ normal maps

- Getting good looking displacement is not the whole story…

Displacement Problems

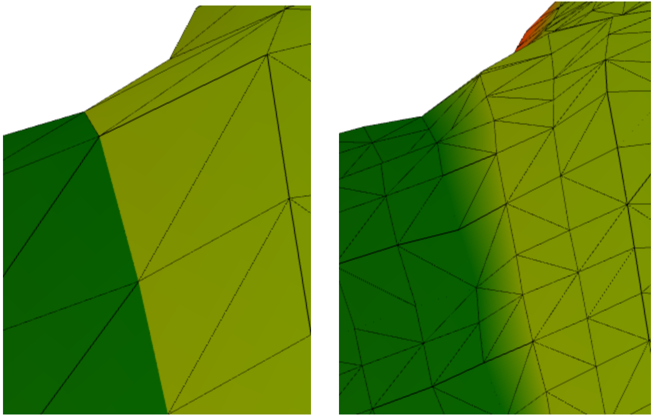
- Cracks
  - Texture coordinates discontinuities
  - Multiple materials assigned to sub-meshes
- Stretching





Texture Coordinates Discontinuities

- Use adjacency information in Domain Shader to stitch the crack
- Each vertex gets assigned 3 extra coordinates
  - 1 for dominant corner
  - 2 for dominant edge



Texture Coordinates Discontinuities

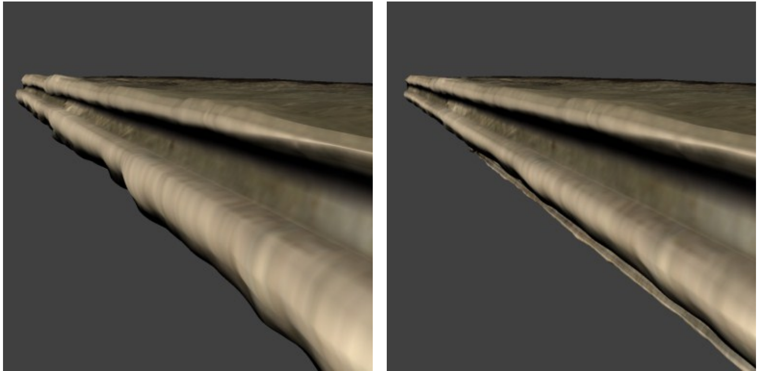
- Reducing discontinuities on the seams
  - For each texel
    - Using adjacency, average height value of connected texels
    - Blend smoothly displacement to the border

Multi Materials

- Mark material seams
- Zero displacement on the seam

Stretching

- 2-pass approach
    - Calculate displacement as usual
    - Measure stretching on real model
    - Reduce weight of texels causing stretching on the second pass



Geometry Collision and Separation

- Displacement changes object sizes
  - Objects may collide or separate from each other
    - Characters flying over tessellated terrain
    - Billboards pierced by tessellated walls

- Zero average displacement helps
- Deferred decals

Questions

kdmitriev

or

emakarov

(at)

nvidia.com