
Game Developers Conference® | March 9-13, 2010 | Moscone Center | San Francisco, CA
THINK SERVICES
A DIVISION OF UNITED BUSINESS MEDIA LLC
GDC 10
Learn. Network. Inspire.
www.GDConf.com

Game Developers Conference®
March 9-13, 2010
Moscone Center
San Francisco, CA
www.GDConf.com

DX11 Effects in Metro 2033: The Last Refuge

Oles Shishkovtsov, 4A Games
Ashu Rege, NVIDIA
Nikolai Sakharnykh, NVIDIA

Metro 2033: the game

- A combination of horror, survival, RPG and shooting
- Based on a novel by Dmitry Glukhovsky

Technology

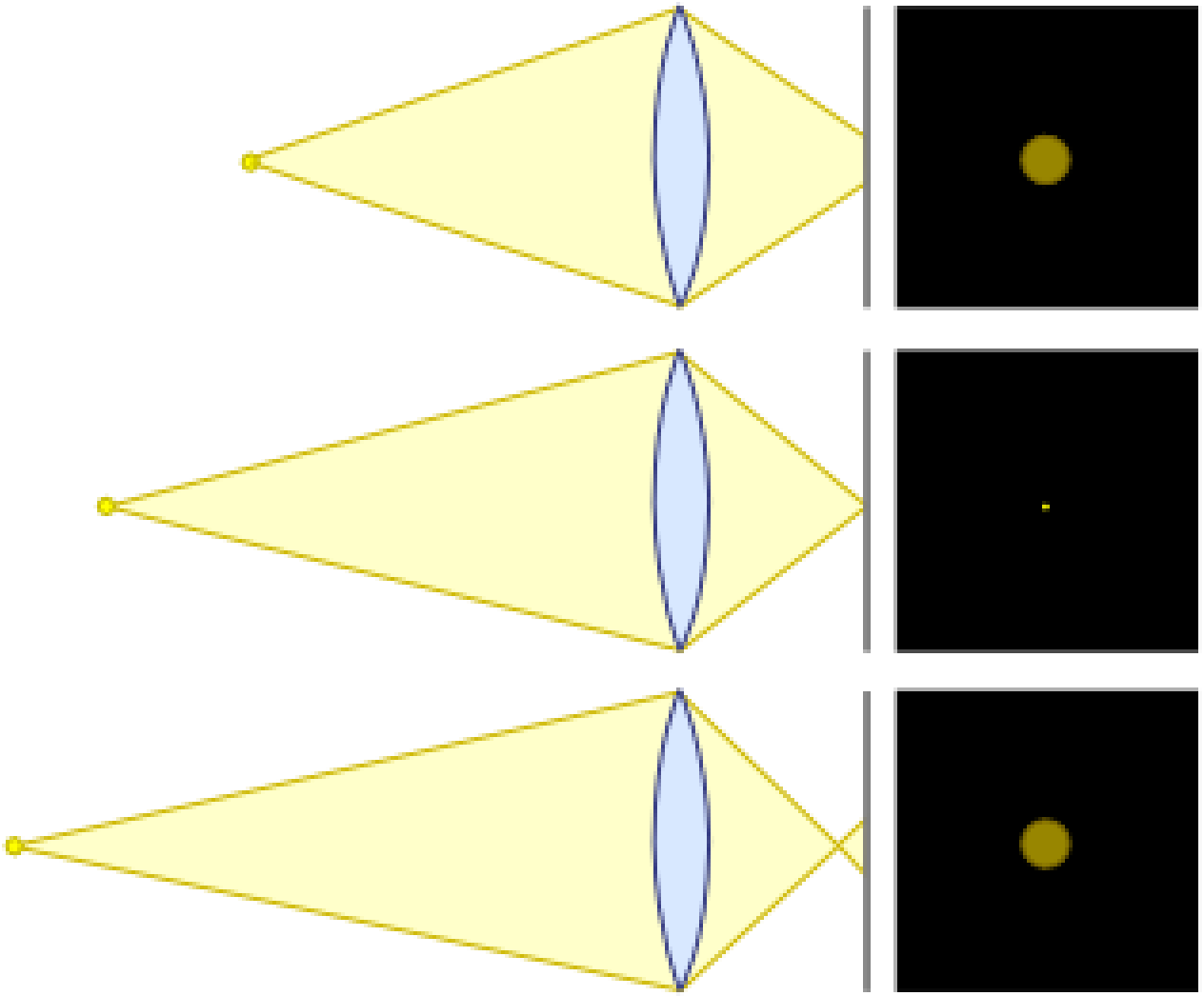
- Developed by Oles Shishkovtsov
  - Lead architect of the STALKER engine
- Metro engine is based on new tech
- Packs a lot of innovation
  - **Pervasive DX11 tessellation**
  - **Advanced post processing using DX Compute**

Game Developers Conference®
March 9-13, 2010
Moscone Center
San Francisco, CA
www.GDConf.com

Depth of field

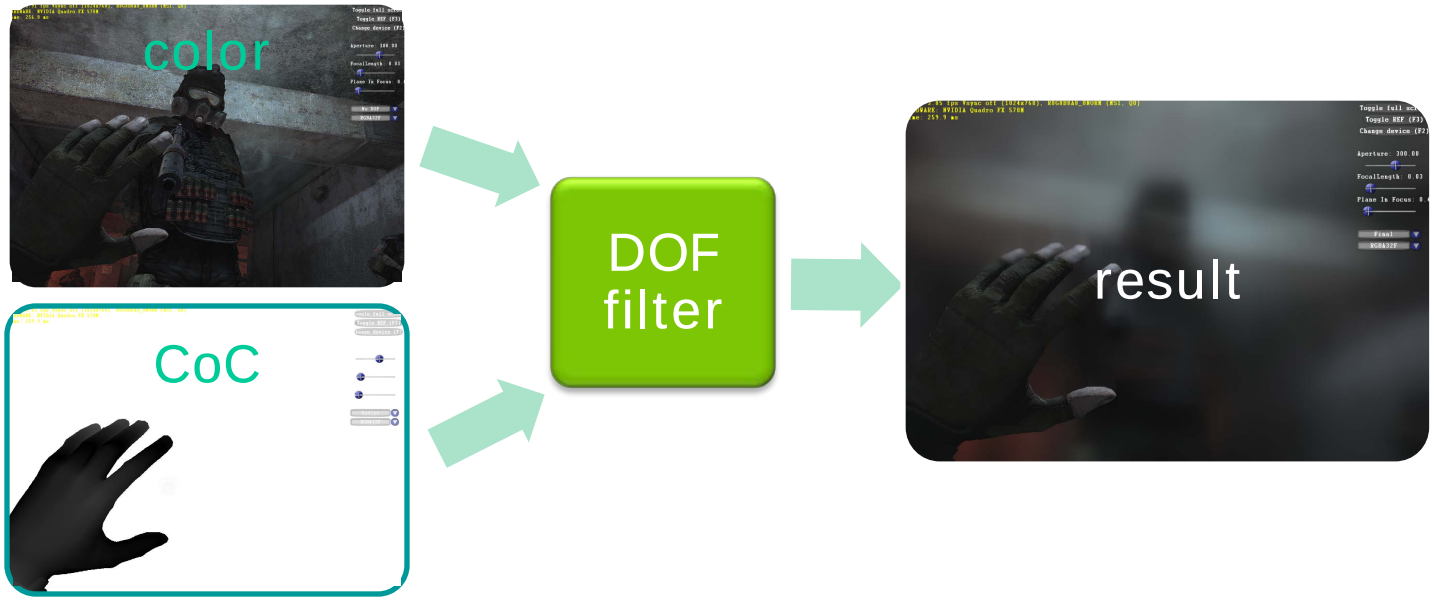
- Common effect in games these days
- Typically post-processing image from a pin-hole camera
- **Key challenge:** Need to keep sharp in-focus objects and blurry backgrounds from bleeding into each other

Circle of Confusion (CoC)

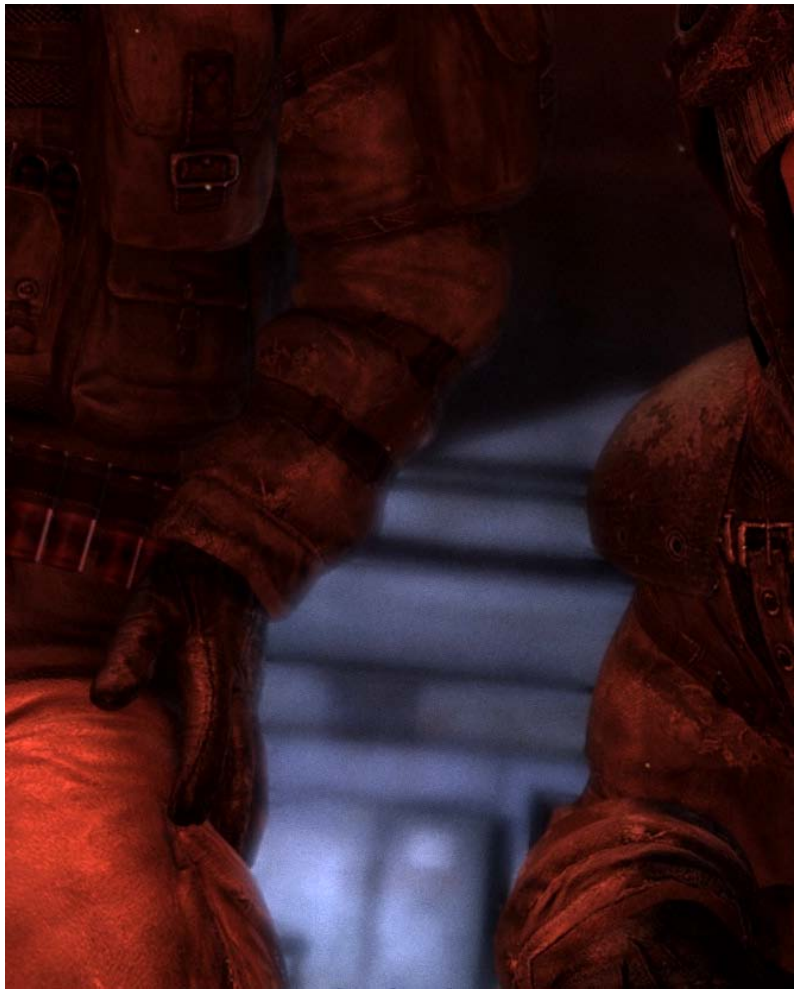


Depth of field effect

- Post-processing input color layer by using depth layer to calculate CoC (circle of confusion)



Bleeding artifacts



From *Metro 2033*, © THQ and 4A Games

Bleeding artifacts

From *Metro 2033*, © THQ and 4A Games

Diffusion DOF in Metro

From *Metro 2033*, © THQ and 4A Games

Diffusion DOF in Metro

From *Metro 2033*, © THQ and 4A Games

Diffusion DOF in Metro

From *Metro 2033*, © THQ and 4A Games

Diffusion-based DoF

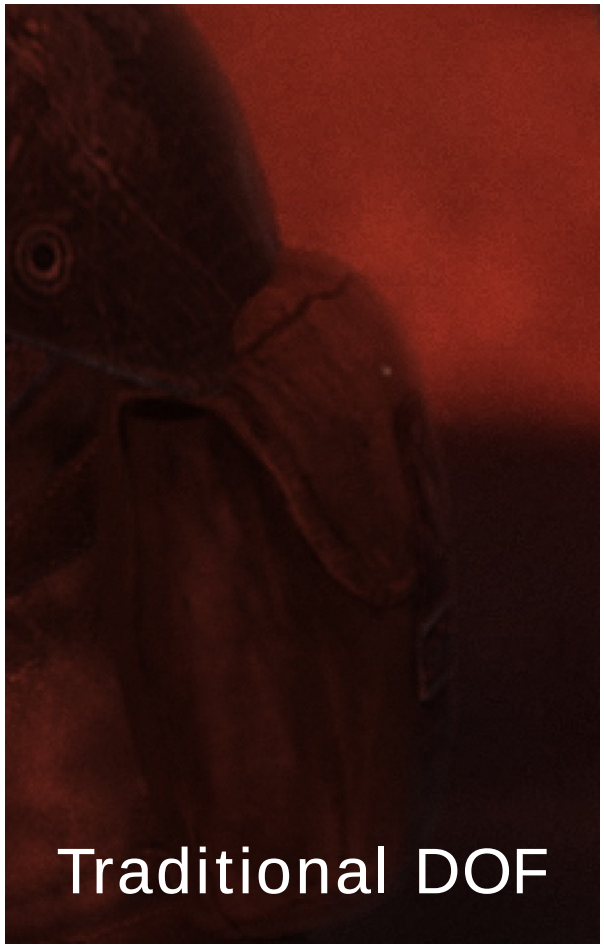
- Main problem:
  - Blur kernel size varies across screen
- Diffusion simulation:
  - Convert CoC into varying heat conductivity and allow colors to *diffuse* as temperature series.
  - Small CoC == lower diffusion
- See *Interactive DOF using Simulated Diffusion on a GPU*, Kass et al.

Benefits

- No color bleeding

From *Metro 2033*, © THQ and 4A Games

Benefits – detail view



From *Metro 2033*, © THQ and 4A Games

Benefits

- Clear separation of sharp in-focus and blurred out-of-focus objects

From *Metro 2033*, © THQ and 4A Games

Cinematic look

From *Metro 2033*, © THQ and 4A Games

Game Developers Conference®
March 9-13, 2010
Moscone Center
San Francisco, CA
www.GDConf.com

Implementation

- We cast DOF problem in terms of differential equation

$$\frac{\partial u}{\partial t} = \nabla \cdot (\beta \cdot \nabla u)$$

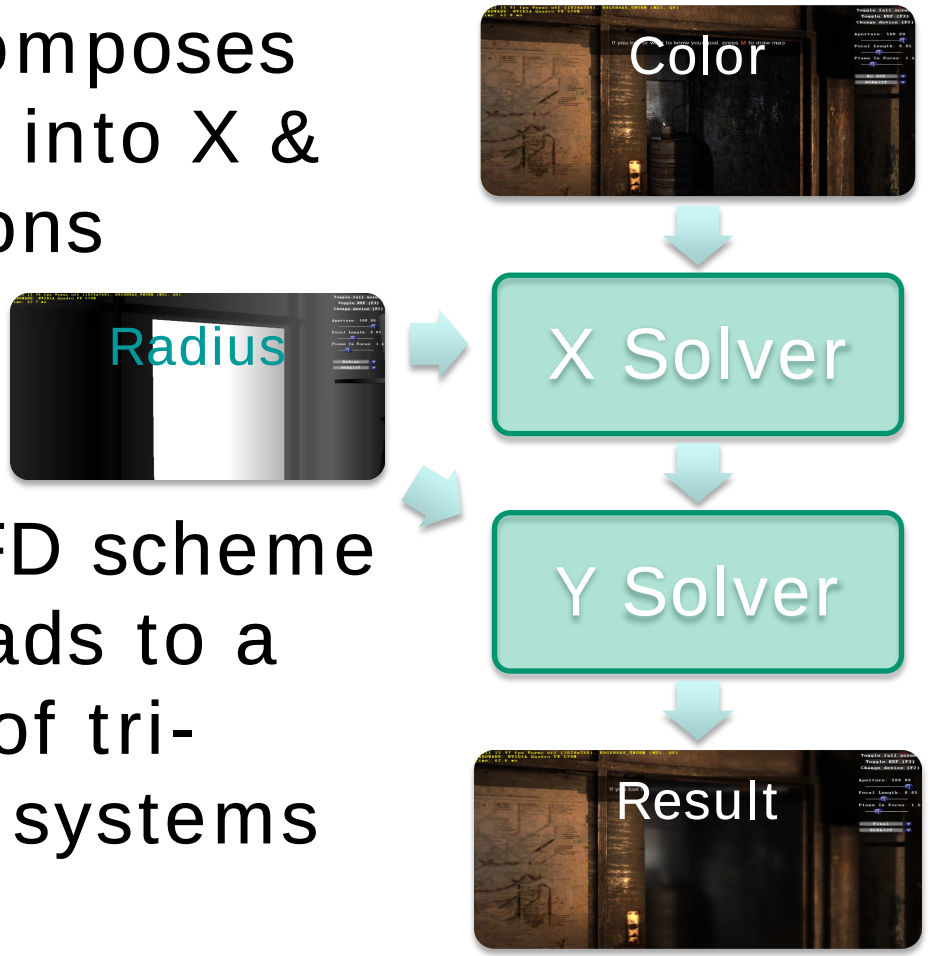
$u(x, y)$ Image color

$\beta(x, y)$ Circle of confusion

- Using Alternate Direction Implicit (ADI) numerical method

Implementation

- ADI decomposes equation into X & Y directions
- Applies FD scheme which leads to a number of tri-diagonal systems



Solving tridiagonal systems

- A number of methods exist:
  - Cyclic reduction (CR)
  - Parallel cyclic reduction (PCR)
  - Simplified Gauss elimination (Sweep)
  - (see references for details)
- We use a new hybrid approach
  - PCR + Sweep

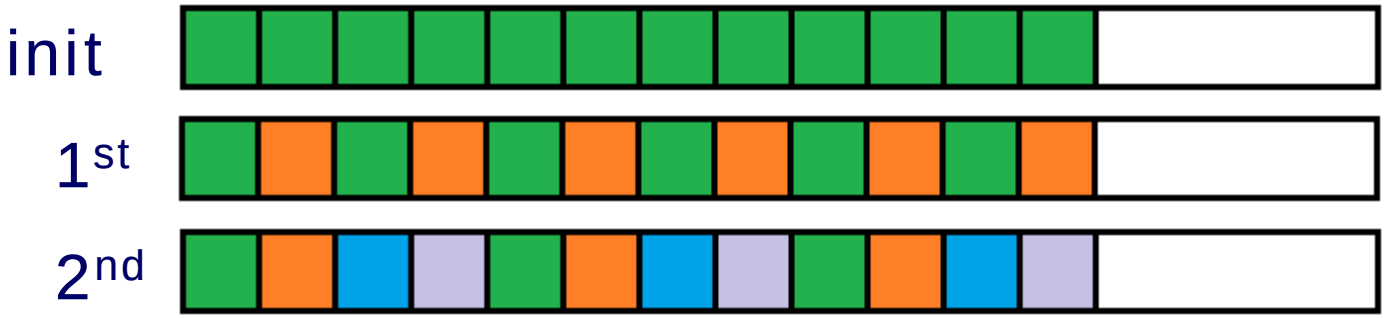
Solving tridiagonal systems (x direction)

$$\begin{bmatrix} b_1 & c_1 & & & 0 \\ a_2 & b_2 & c_2 & & \\ & a_3 & b_3 & \ddots & \\ & & \ddots & \ddots & c_{n-1} \\ 0 & & & a_n & b_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \\ \vdots \\ d_n \end{bmatrix}.$$

- Here x_i are the per-pixel colors
- Each system is of size == WIDTH
- And # of systems == HEIGHT
- d_i are the initial conditions
- We want a GPU friendly version...

Hybrid tridiagonal solver - PCR

- Few steps of parallel cyclic reduction (PCR)
  - Implemented in *pixel shader*



  - At each PCR step we double the number of systems and halve the size of each system → More parallel work to fill GPU.

Game Developers Conference®
March 9-13, 2010
Moscone Center
San Francisco, CA
www.GDConf.com

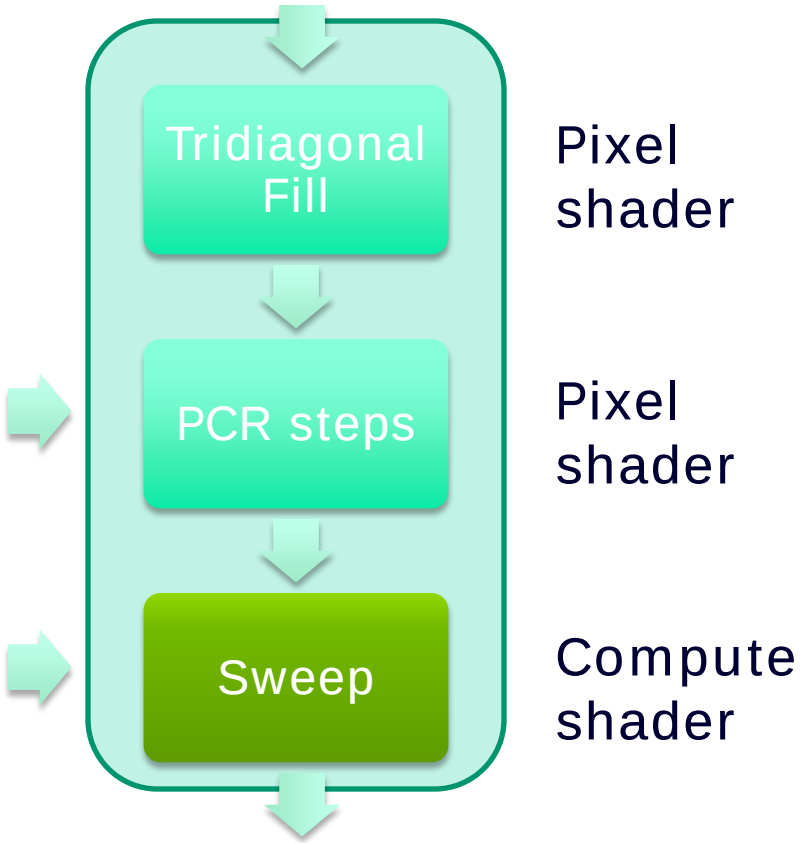
Hybrid tridiagonal solver - Sweep

- Finish with simplified Gauss elimination (Sweep)
  - Implemented in *compute shader*
- Each thread solves one system
  - Forward elimination
  - Backward substitution
  - Complexity O(N)

Tridiagonal solver in DX11

PCR steps = 3

Num systems	System size
Height	Width
Height*8	Width/8



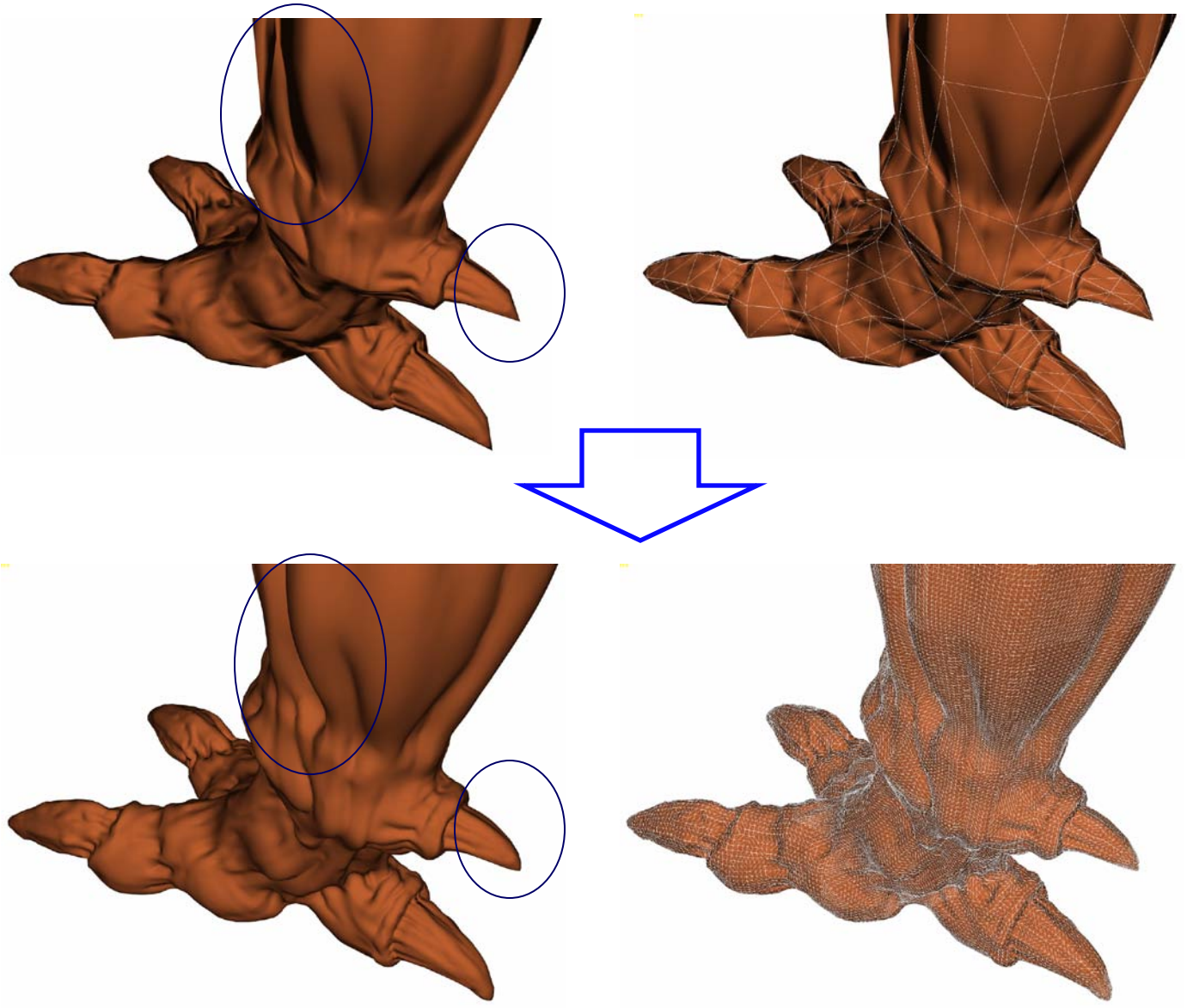
References

- “Interactive depth of field using simulated diffusion on a GPU” Michael Kass, Aaron Lefohn, John Owens, Pixar Animation studios, Pixar technical memo #06-01
- “Tridiagonal solvers on the GPU and applications to fluid simulation” Nikolai Sakharnykh, GTC 2009
- “Fast tridiagonal solvers on the GPU” Yao Zhang, Jon Cohen, John D. Owens, PPoPP 2010

Tessellation

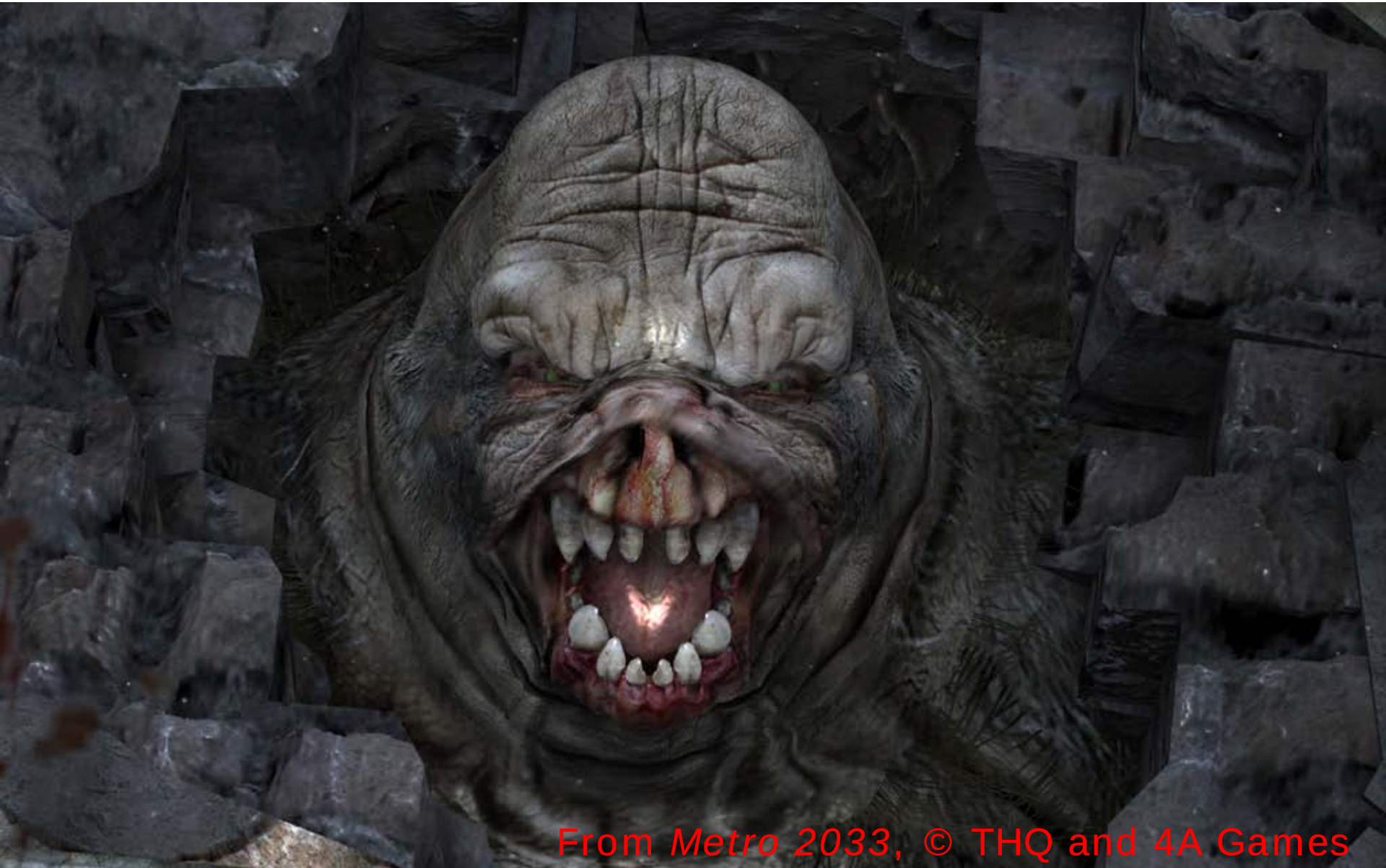
- Enables film-level geometric detail in real-time rendering
- Automatic LOD without popping
- High quality meshes at low cost

Bump Mapping Often not Sufficient



Game Developers Conference®
March 9-13, 2010
Moscone Center
San Francisco, CA
www.GDConf.com

Tessellated Monster in Metro 2033



From *Metro 2033*, © THQ and 4A Games

More Metro Monsters...

From *Metro 2033*, © THQ and 4A Games

… and Characters

From *Metro 2033*, © THQ and 4A Games

… more Characters

From *Metro 2033*, © THQ and 4A Games

Start with a Coarse Base Mesh

HW-based Subdivision

Apply Displacement

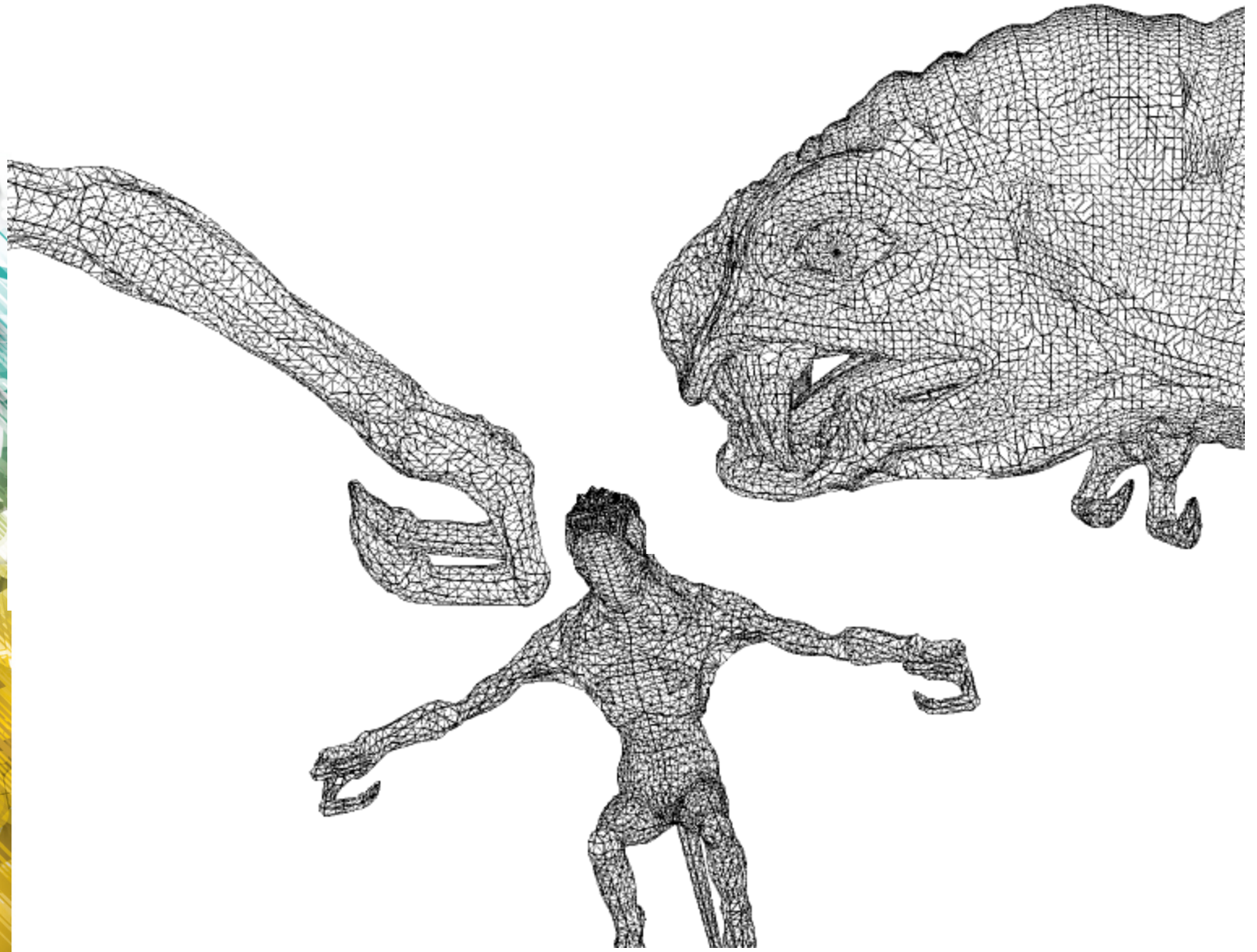
Game Developers Conference®
March 9-13, 2010
Moscone Center
San Francisco, CA
www.GDConf.com

LOD Computation

TESS_FACT = LEN * NP * Q / DIST

- Where LEN is edge length in world space
- NP is number of pixels on the screen
- Q is quality constant
- DIST is distance from observer to edge center

Constant Triangle Size



Subdivision Schemes

- Phong tessellation
- PN-triangles
- Approximate Catmull-Clark
- ...

- Metro uses Phong tessellation

Phong Tessellation

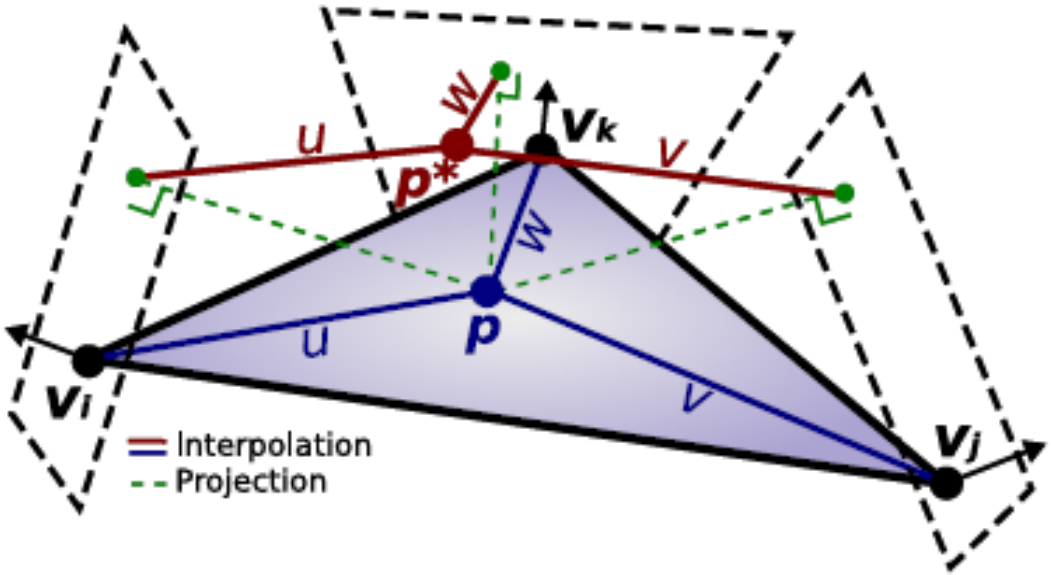
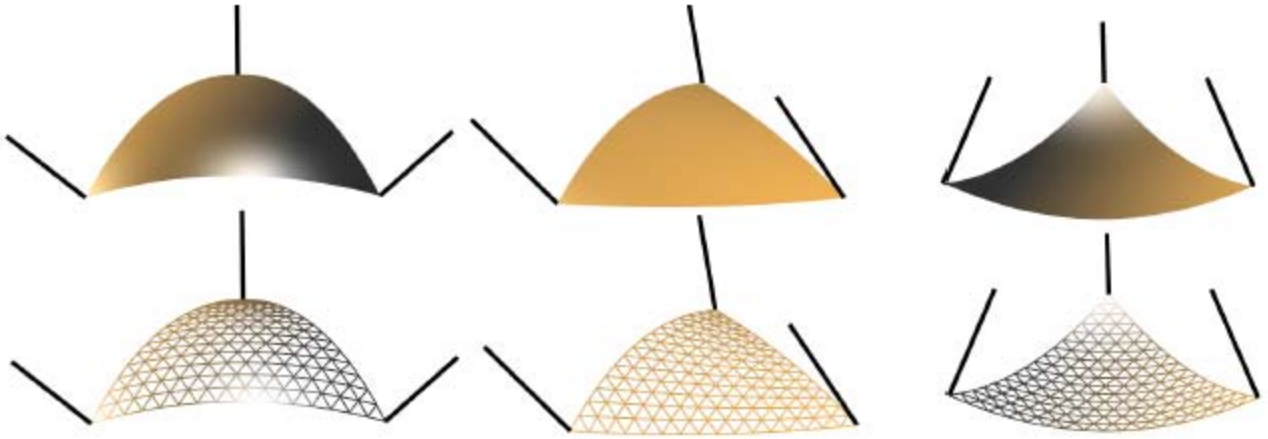


Figure 3: *Phong Tessellation principle. Instead of interpolating normals as in Phong Shading, we interpolate projection onto vertices tangent plane to define a curve geometry for each triangle.*

- From *Boubekeur and Alexa*, Siggraph Asia 2008

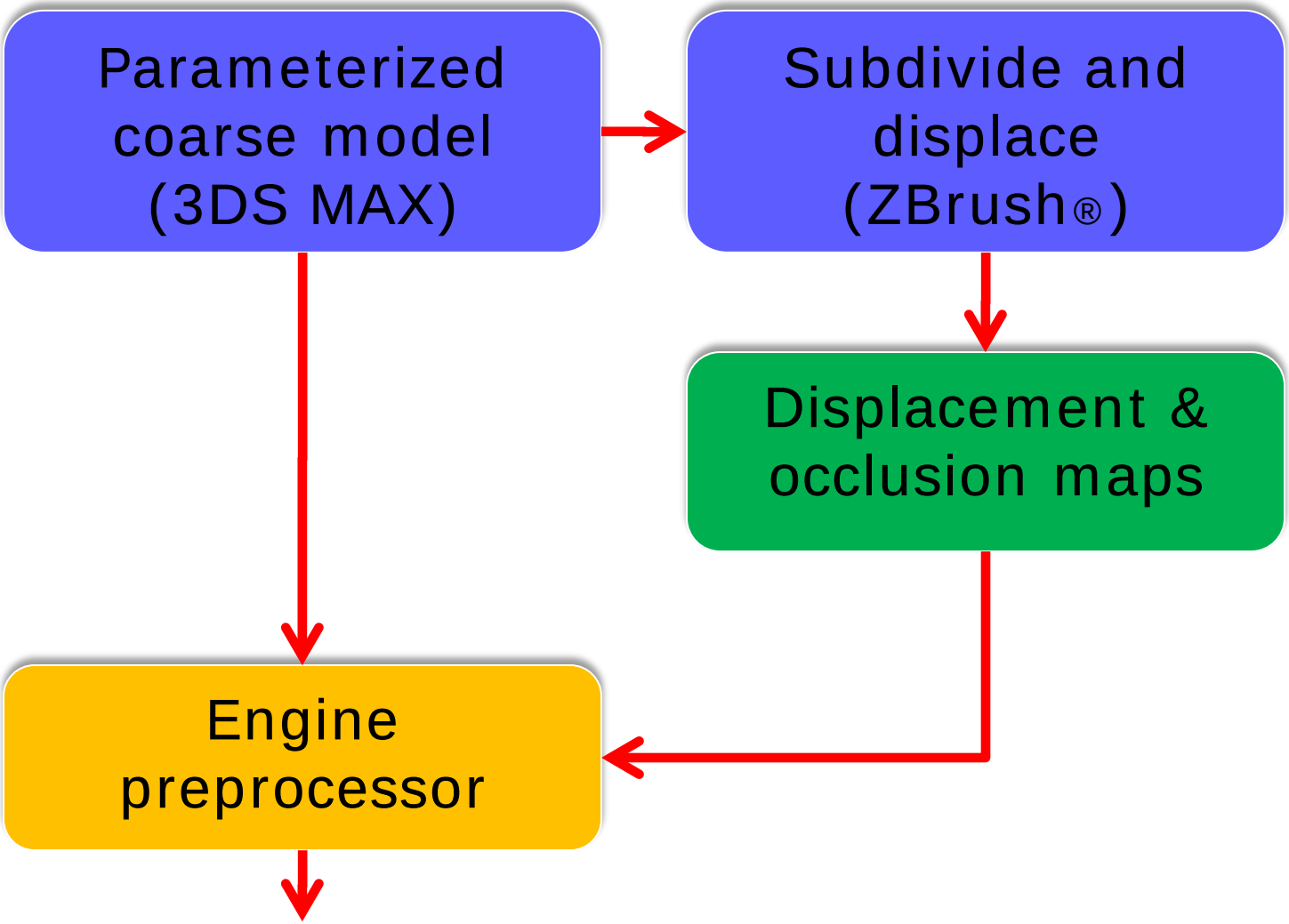
Phong Tessellation



Boubekeur and Alexa, 2008

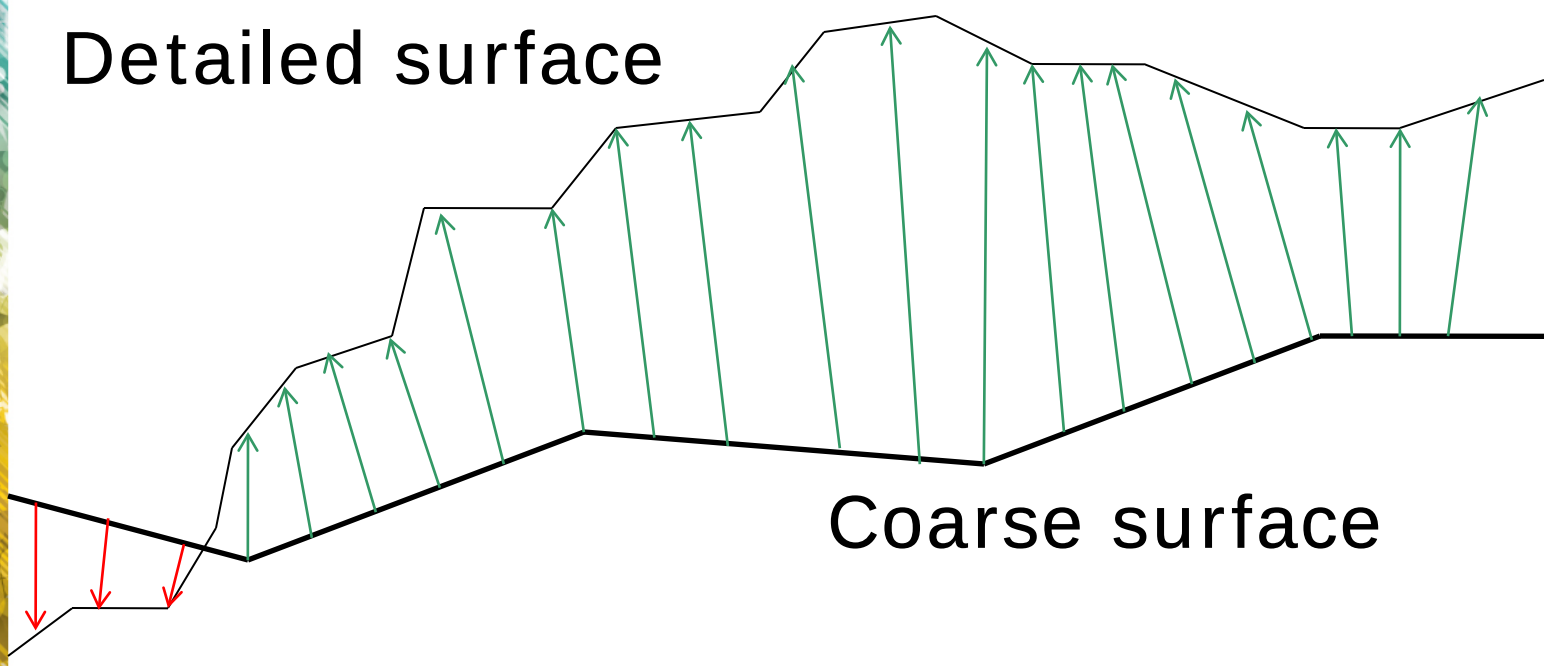
- Simple scheme
- Does not handle inflections well
- Acceptable if initial subdivision is rather dense

Content Creation Pipeline

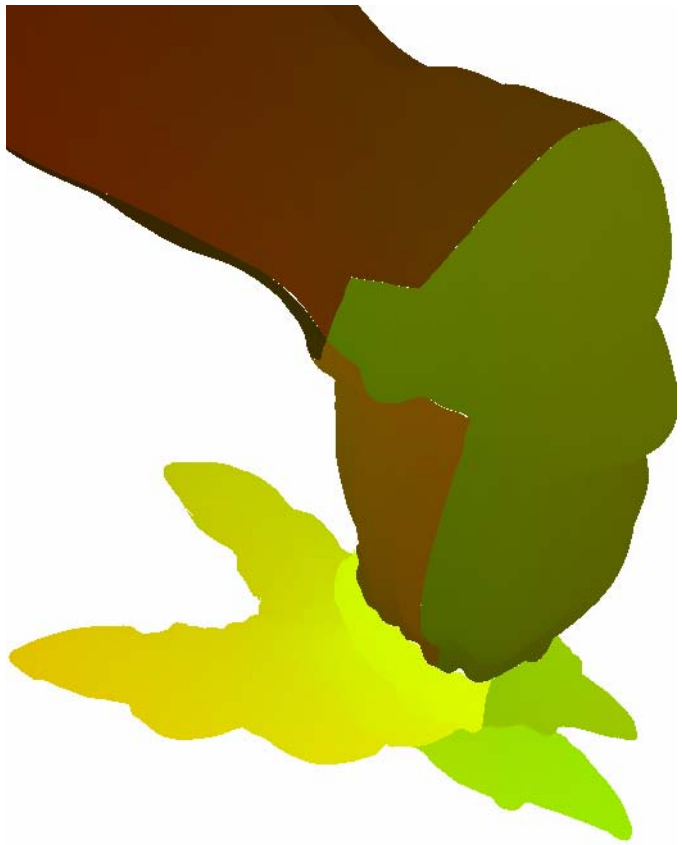


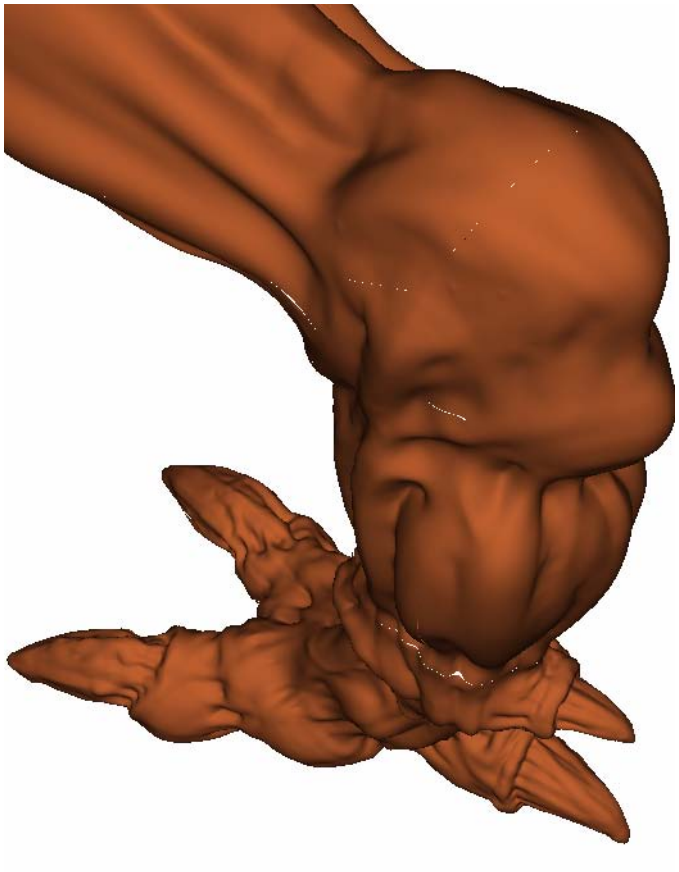
Computing Displacement

- Input: coarse and detailed model



Cracks: Discontinuity in Parametrization

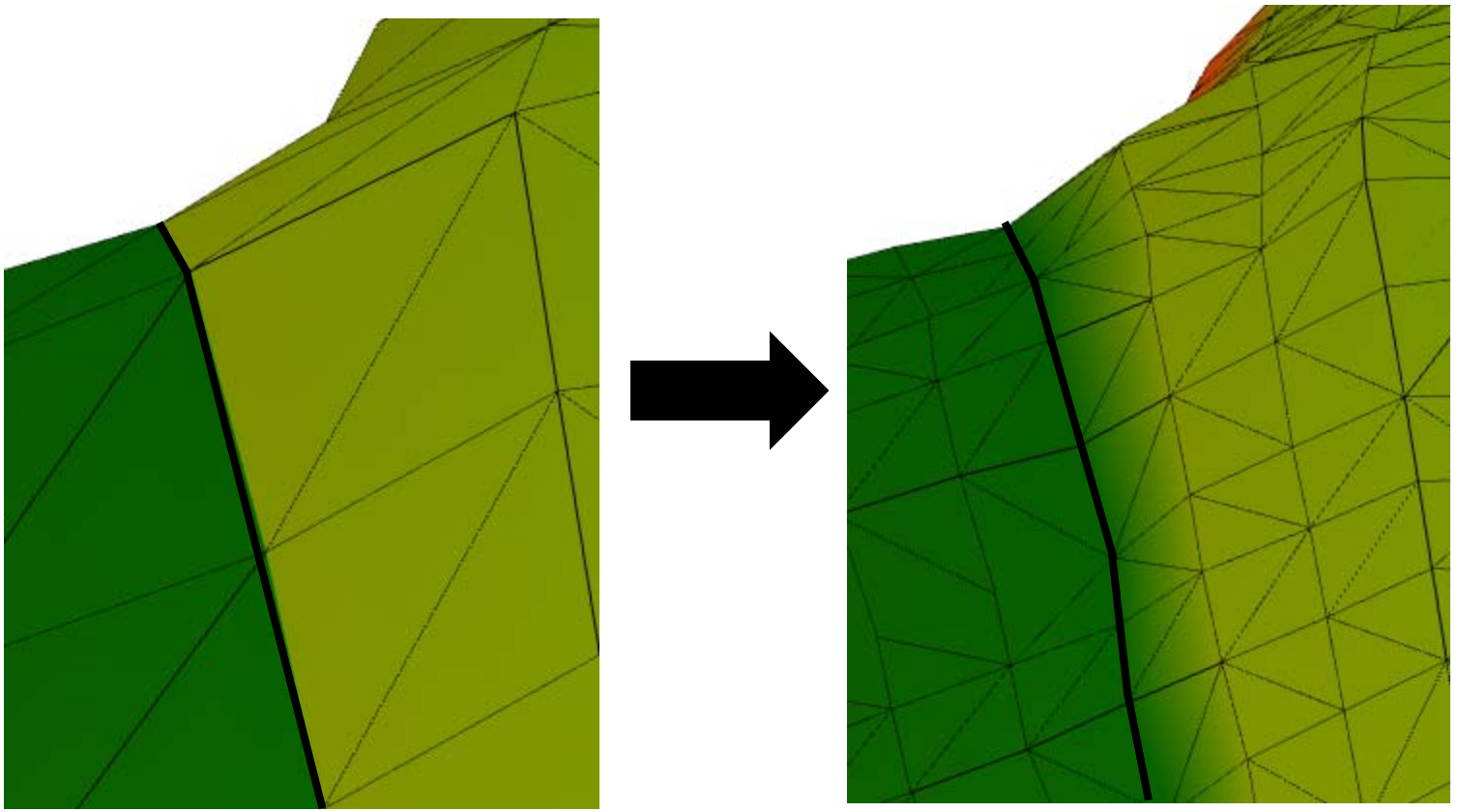




Game Developers Conference®
March 9-13, 2010
Moscone Center
San Francisco, CA
www.GDConf.com

Cracks: Ownership-Based Solution

- Boundary triangles share the same texture coordinates data



Game Developers Conference®
March 9-13, 2010
Moscone Center
San Francisco, CA
www.GDConf.com

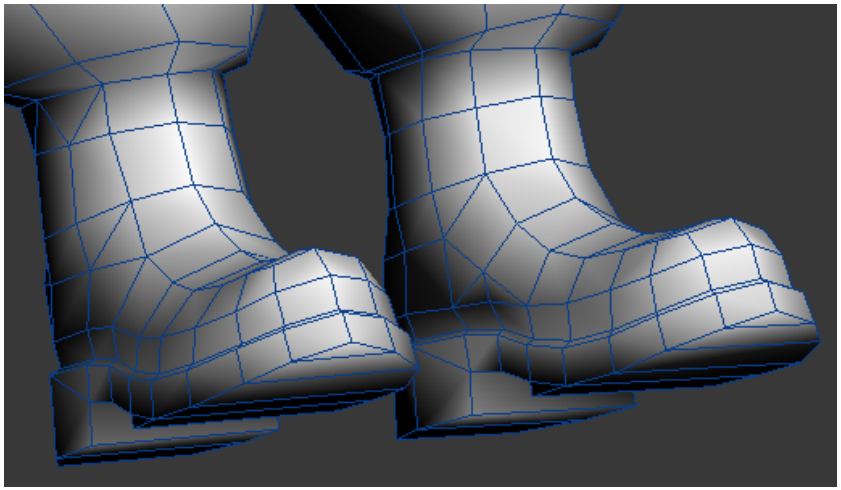
Cracks: Ownership-Based Solution

- Boundary primitives store extra set of texture coordinates

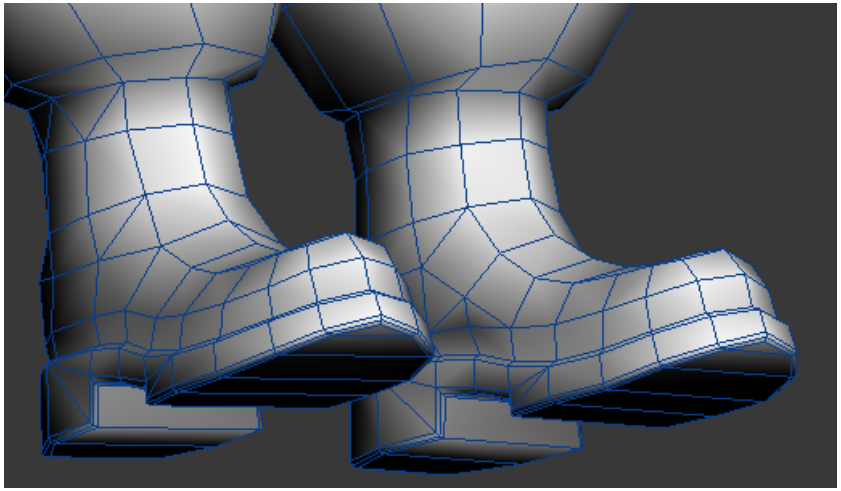
- 1 normal texture coordinate
- 2 extra coordinates per edge
- 1 extra coordinate per corner

  - 12 coordinates per triangle patch
  - 16 coordinates per quad patch

Artifacts on Hard Edges



Transitional Polygons



Game Developers Conference®
March 9-13, 2010
Moscone Center
San Francisco, CA
www.GDConf.com

Conclusion: DX11 in *Metro 2033*

- Tessellation enables a massive increase in geometric fidelity in game graphics
  - Even existing art assets can benefit
- Compute shaders allow first ever implementation of *cinematic* quality post-processing