EGSR 2008

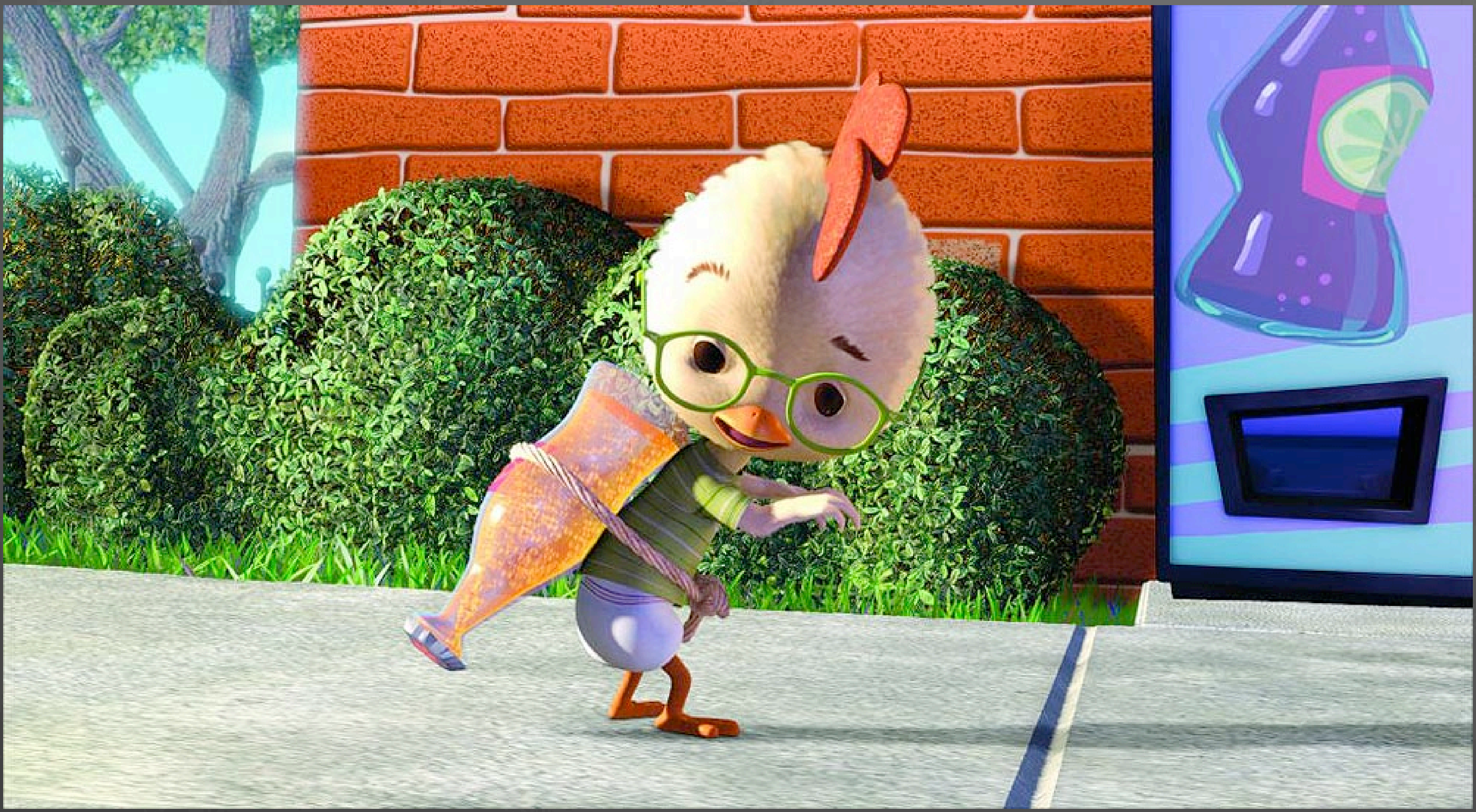
Ptex: Per-face Texture Mapping for Production Rendering

Brent Burley and Dylan Lacewell
Walt Disney Animation Studios

(See attached slide notes for details)

Texture Mapping at Disney

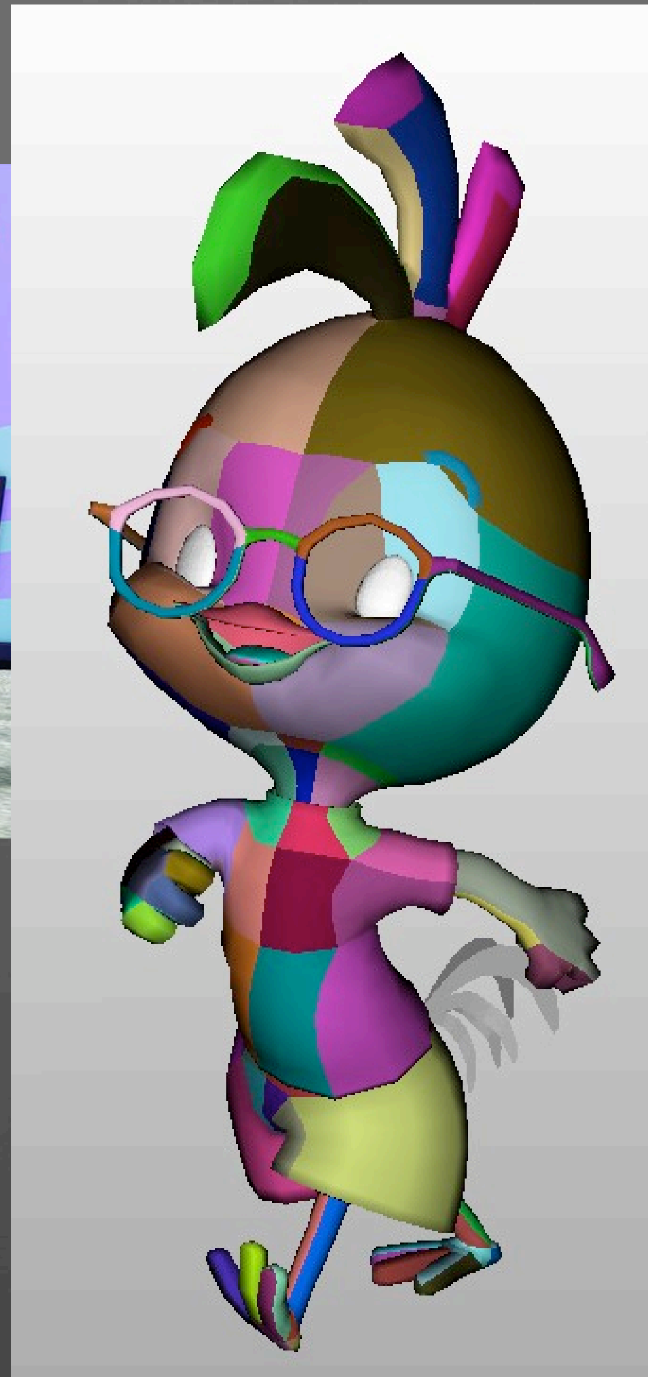
Chicken Little 2003



Art-directed look = lots of textures

First Disney film to use subds.
No more stitching!
But nurbs made texture mapping easy.
Subds?

Chicken Little 2003



170 textures

Approach chosen: Decompose subds into rectangular patches.

Texturing *almost* as easy as nurbs!

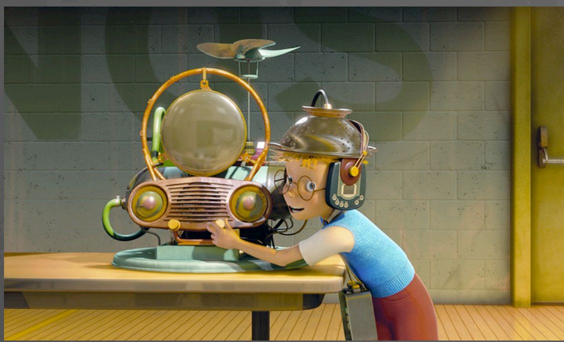
Meet the Robinsons 2005

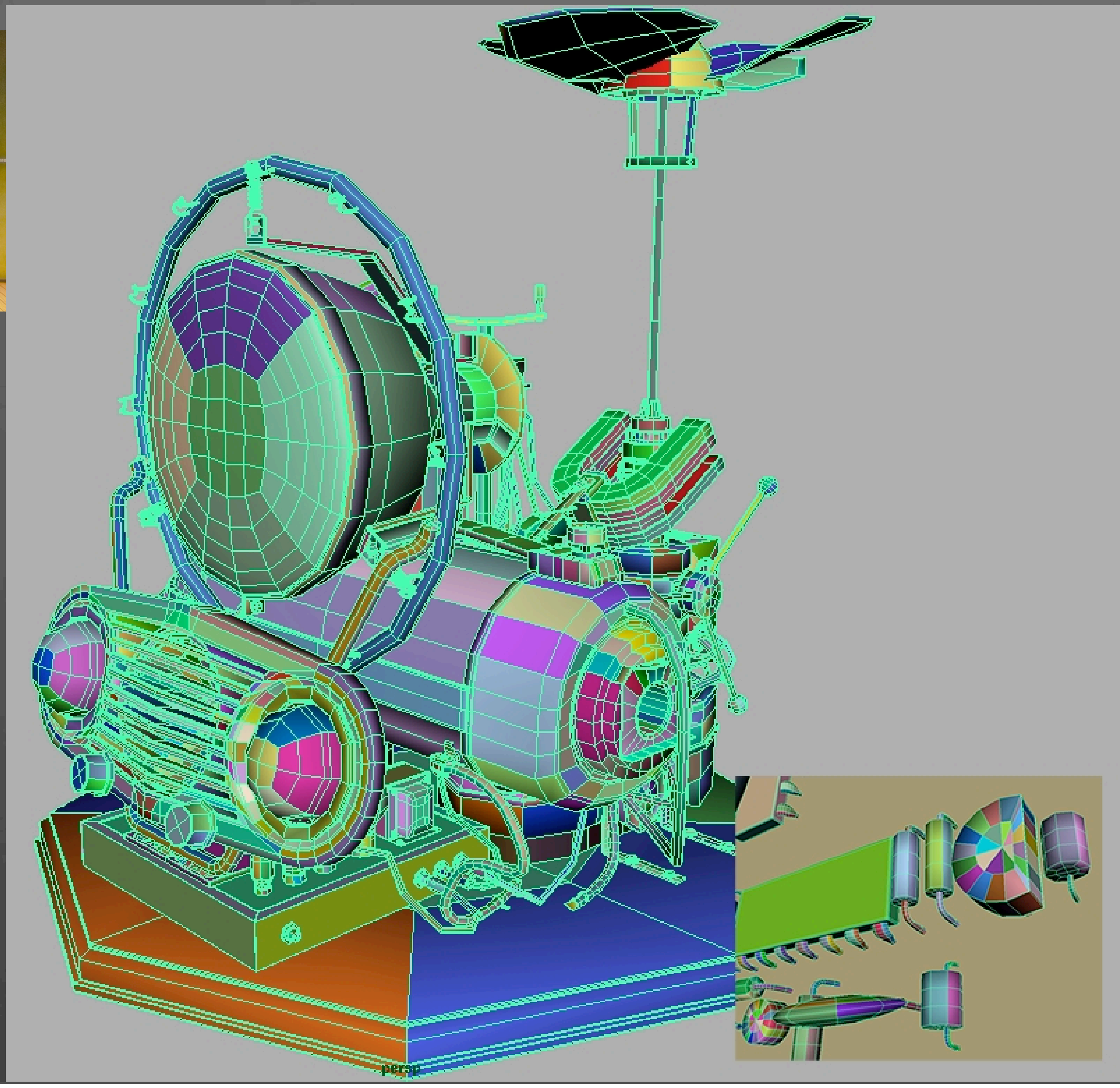
Raised complexity bar.

No more procedurals (now baking to avoid aliasing), just tons o' maps.

Lesson learned: Geometry not always patch-friendly.

Meet the Robinsons 2005



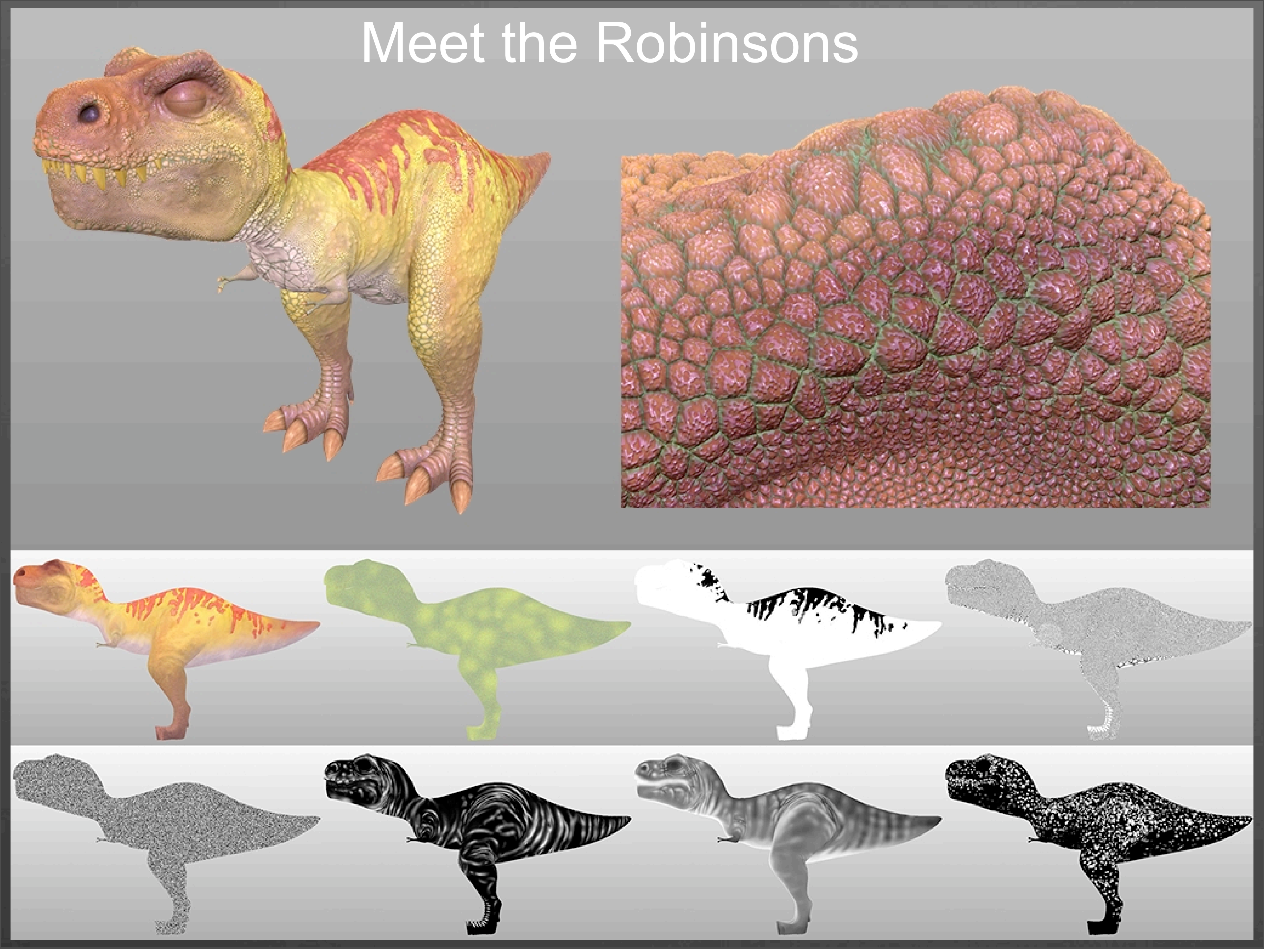


7637 Textures

Case in point!

7637 textures *per layer* on this element.

Up to a dozen layers on some elements.



About ½ of the texture layers for the T-Rex are shown here.

Scales layer had a billion texels.

Many patches covered just a single face and used 2k*2k textures.

Even noise is baked (lower left).

Meet the Robinsons

- 6 million texture files
- 100,000+ textures per render
- Mean file size 30 KB
- Median file size 2.5 KB

- I/O Bound!

Moral: Lots of little files make systems engineers grumpy.

Bolt 2008

Pre-production estimate: 5x complexity of Meet the Robinsons
Already at limits of texture system

Reference Painting

3d Render

Bolt’s painterly approach placed additional demands on texture system.

Two full “looks” on everything: one detailed, one “loose” / less-resolved.

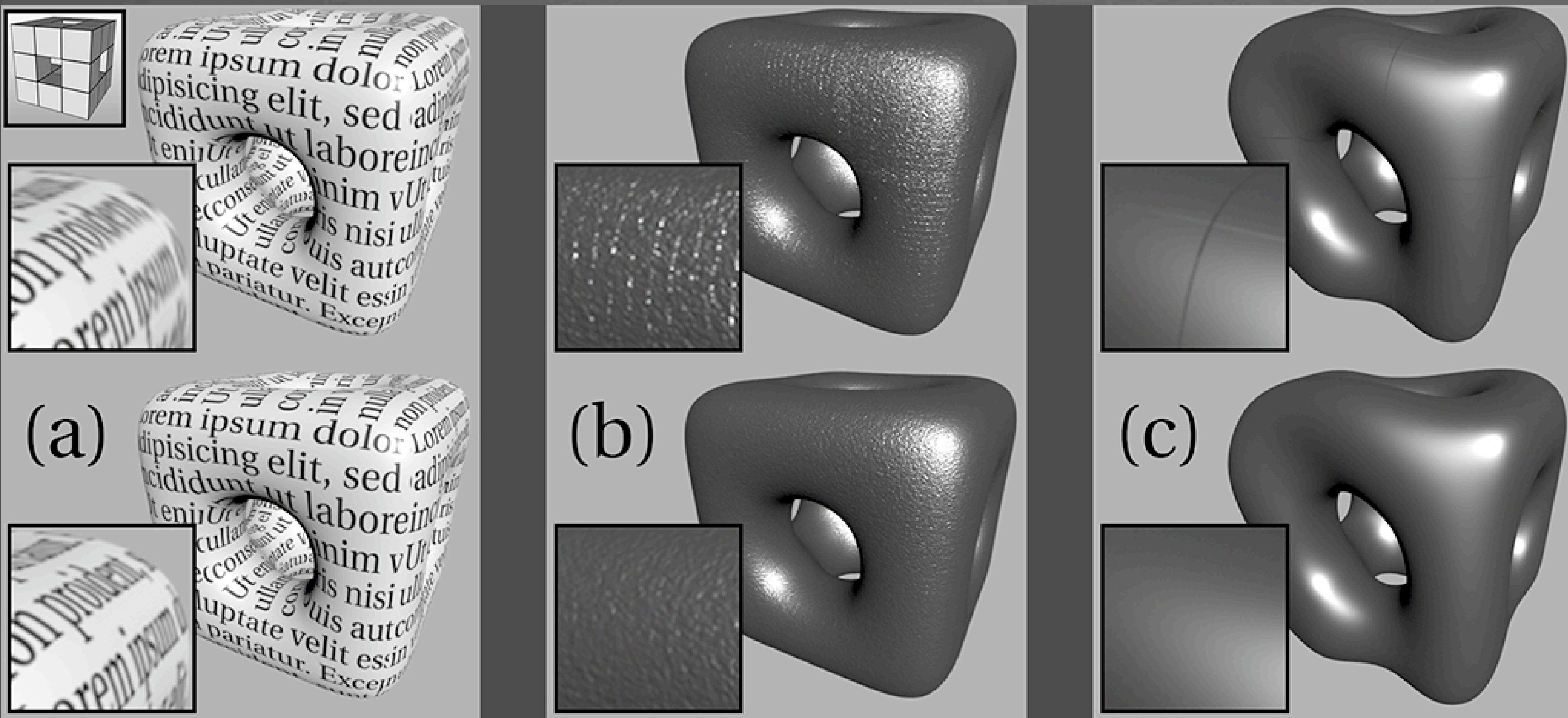
1. Hand-painted normal maps on top of displacement maps are used to make things look “banged up”.
2. Hidden “fins” use hand-painted opacity maps to break up silhouettes.

= many more layers than before

Requirements for new texture system

- Film-quality
- General
- Efficient
- Setup-free

Filtering Requirements



Anisotropic Filtering

Aliasing-free

Seamless

Top row uses:

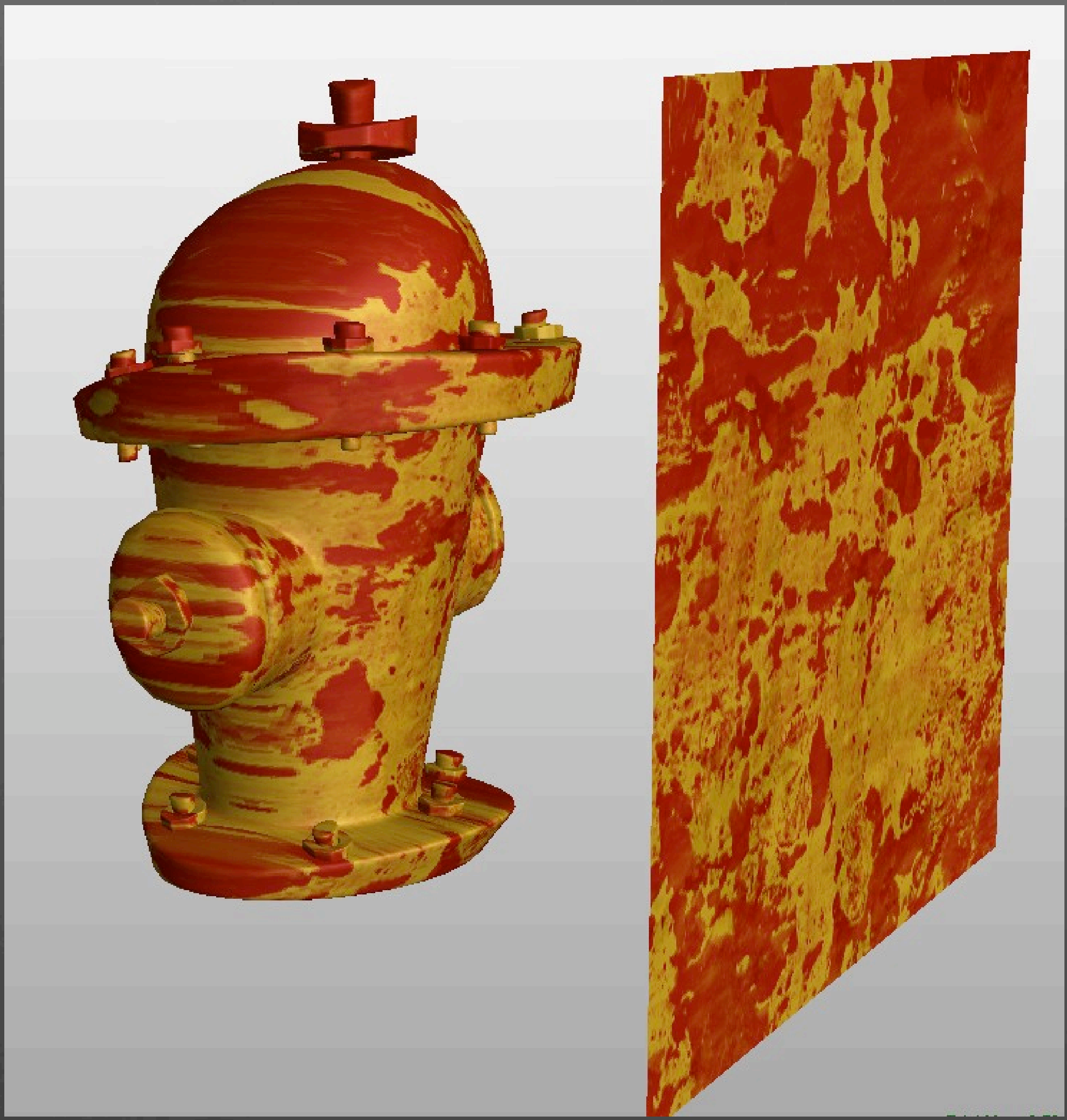
a) isotropic filtering
b) C^1 cubic filter
c) no cross-face filtering w/ large smooth displacements.

Bottom row is what we want:

a) sharpening, anisotropic filter
b) C^2 cubic filter for displacements
c) cross-face filtering

Previous Texture Mapping Methods

Projection Painting



In widespread use.

Depth maps and multiple projections needed to cover complex surfaces.

Cumbersome and expensive on intricate geometry.

Doesn’t cover surface uniquely (can’t bake).

Pelting

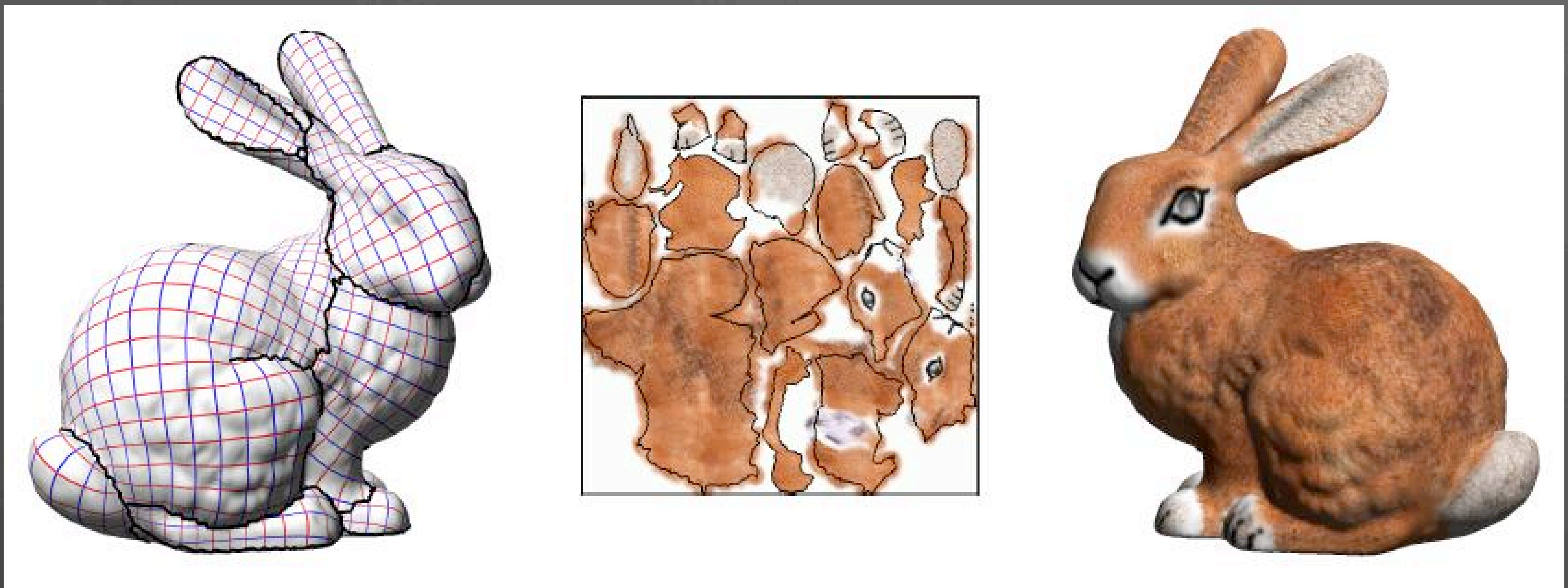
Seamless texture mapping of subdivision surfaces by model pelting and texture blending. Piponi and Borshukov, Siggraph 2000.

Commonly used on characters, but not at Disney.

Labor intensive, and there are filtering challenges.

Not general. Would be hard to pelt a building.

Atlas Methods



Least squares conformal maps for automatic texture atlas generation. Levy et al., Siggraph 2002.

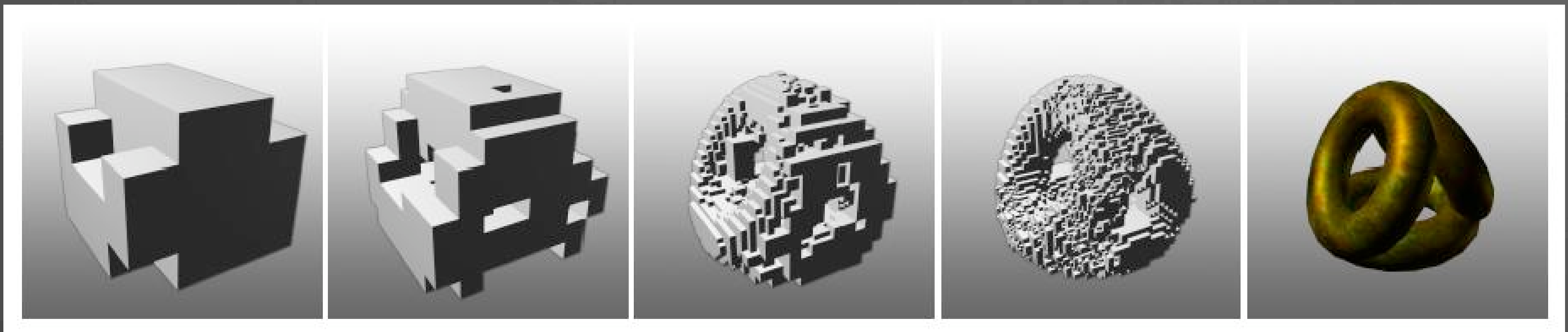
Automated, but …

Filtering challenges.

Limited resolution.

Not general.

Volumetric Methods



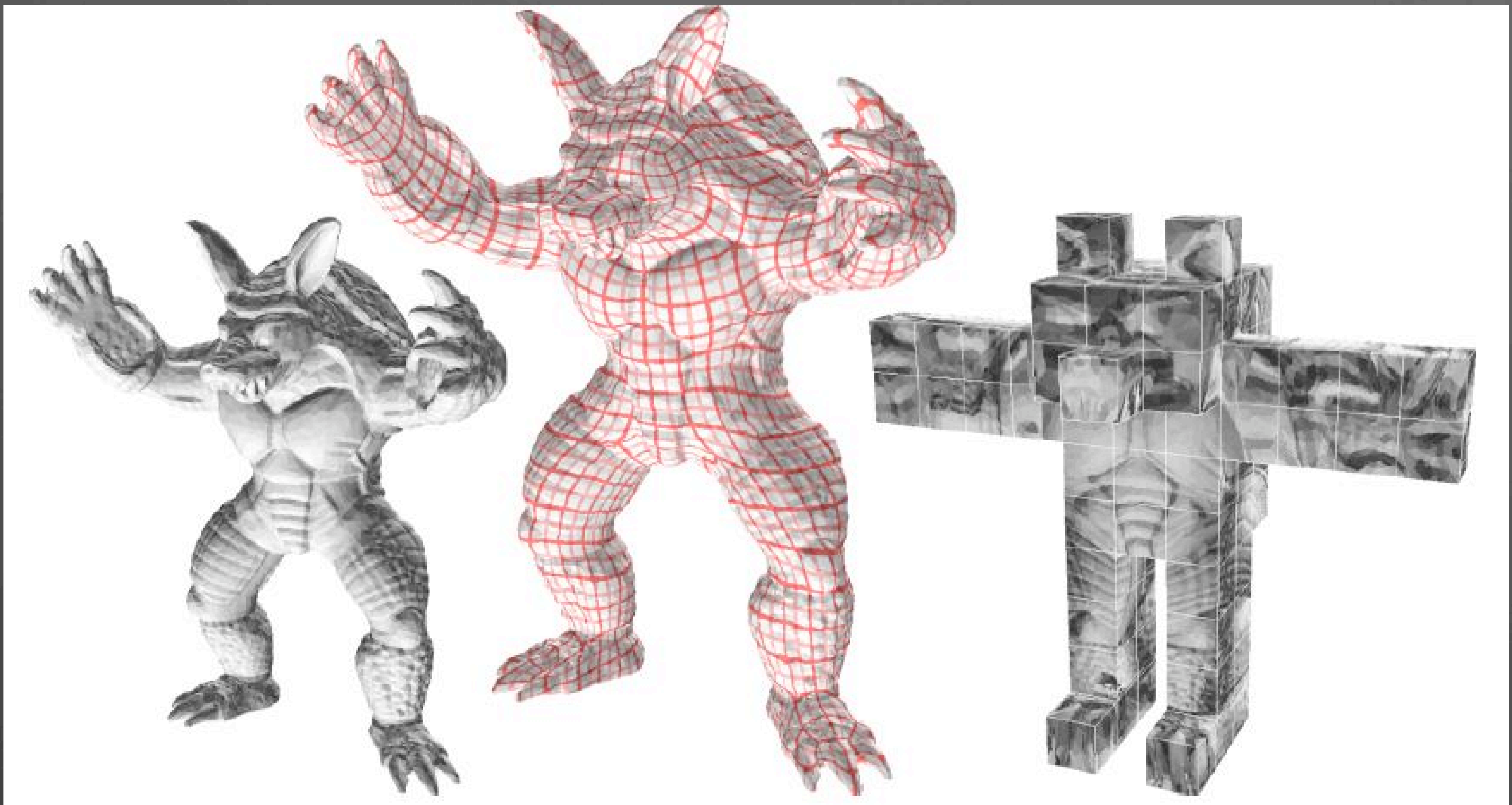
Painting and Rendering textures on unparameterized models.
DeBry et al., Siggraph 2002.

No setup, but…

Filtering issues (e.g. bleeding).

Inefficient storage.

Per-face Methods



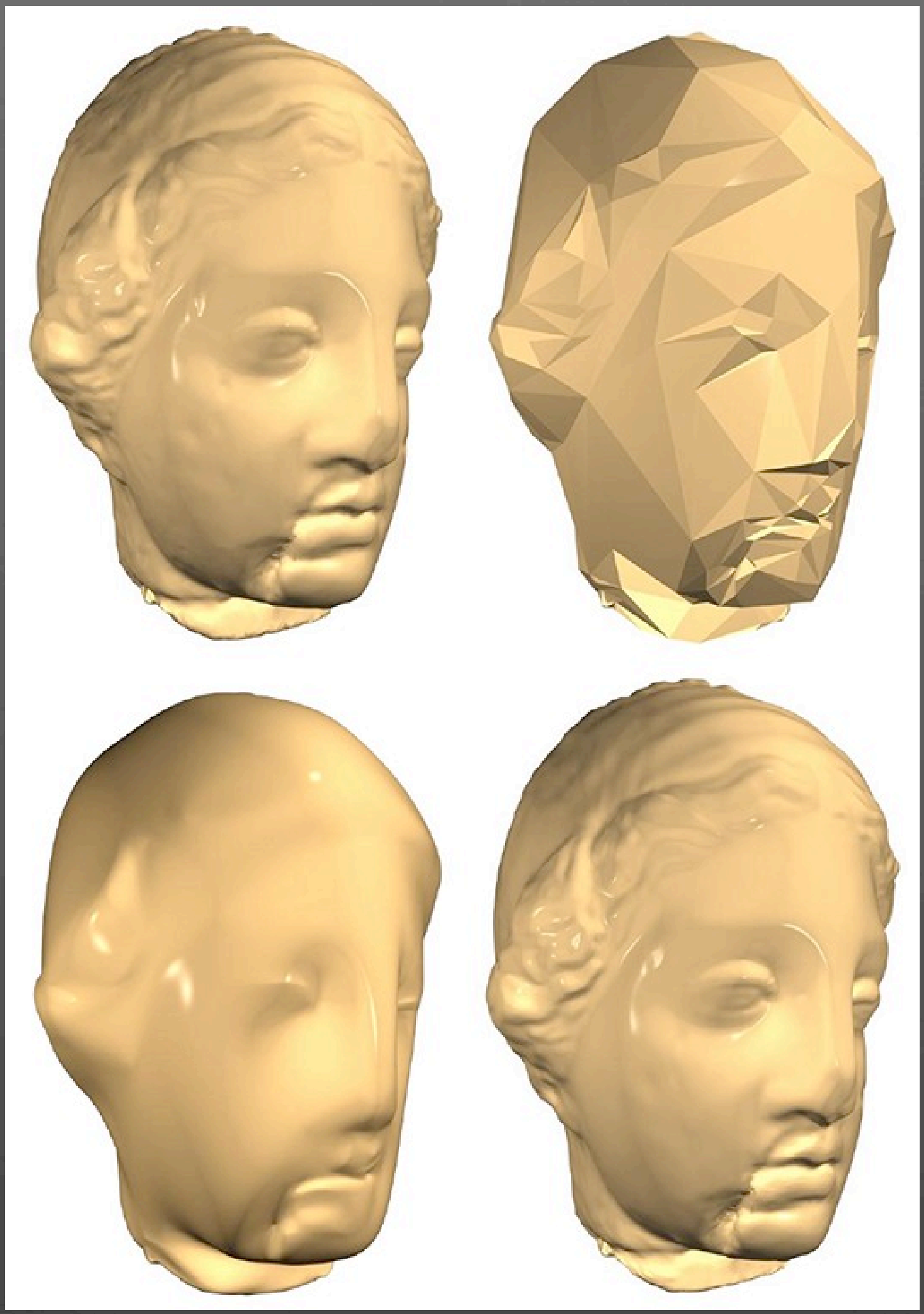
Polycube-maps, Tarini et al., Siggraph 2004.

Novel idea!

Lots of small textures w/ ability to filter across seams.

Not general.

Per-face Methods (cont.)



Displaced Subdivision Surfaces.
Lee et al., Siggraph 2000.

Top row: original surface, simplified control mesh.

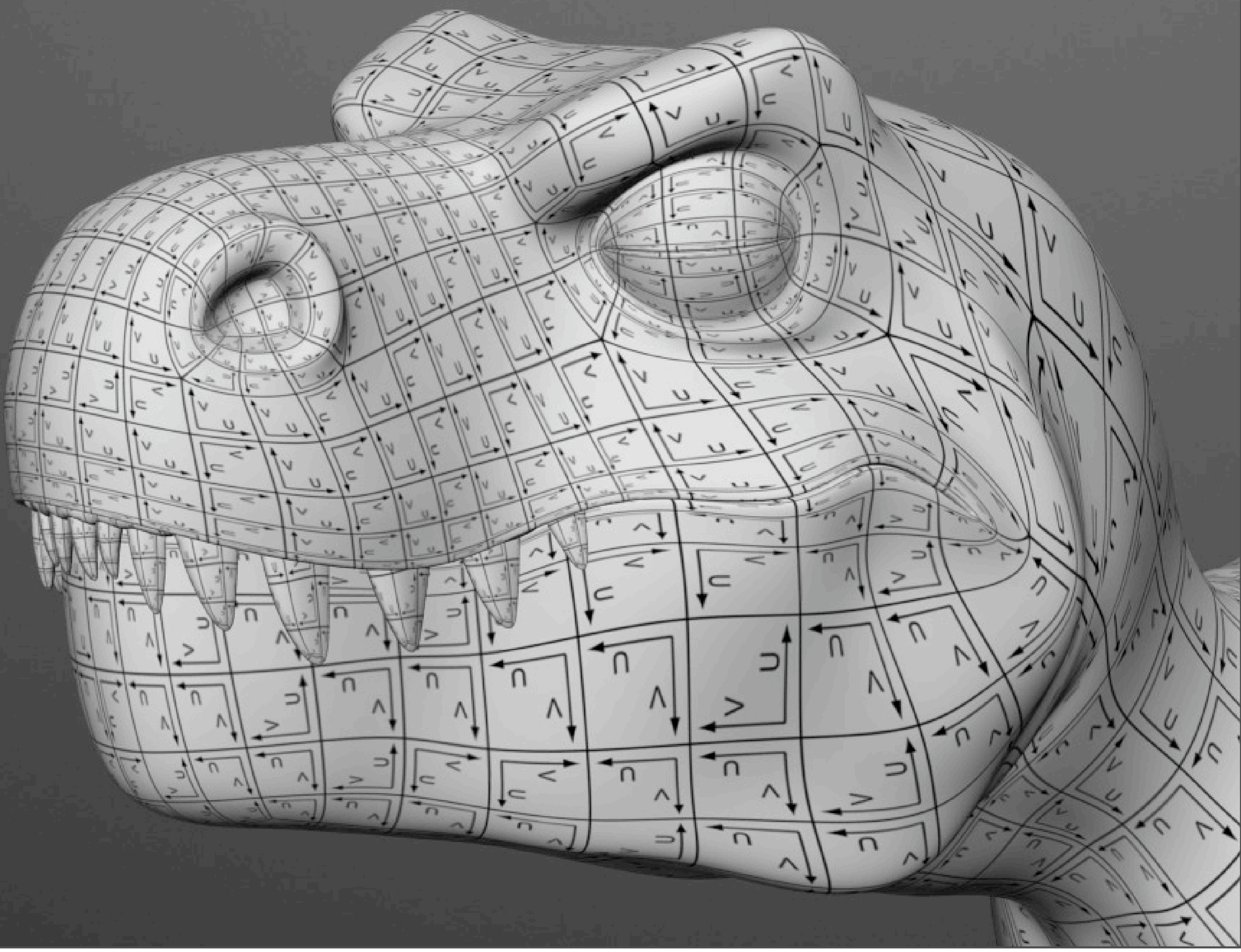
Bottom row: Loop subd surface, reconstructed surface.

Interesting aspect (wrt texture mapping): storing attributes per-face in intrinsic subd domain. Uses subd rules to interpolate values.

But, no multi-res or anisotropic filtering.

Ptex

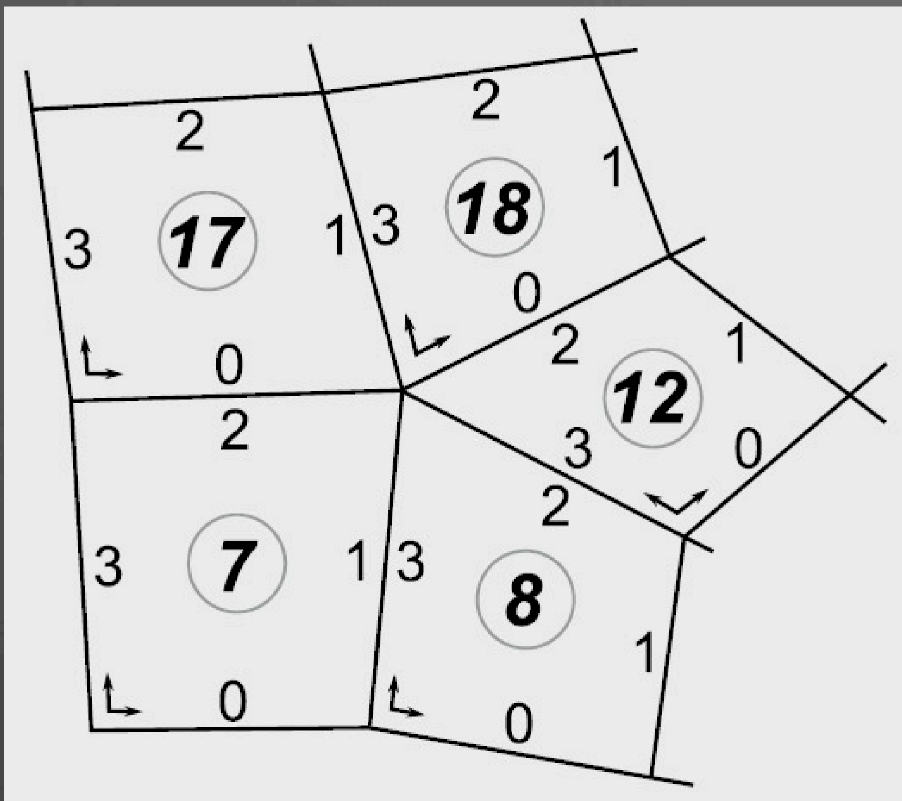
Per-face textures

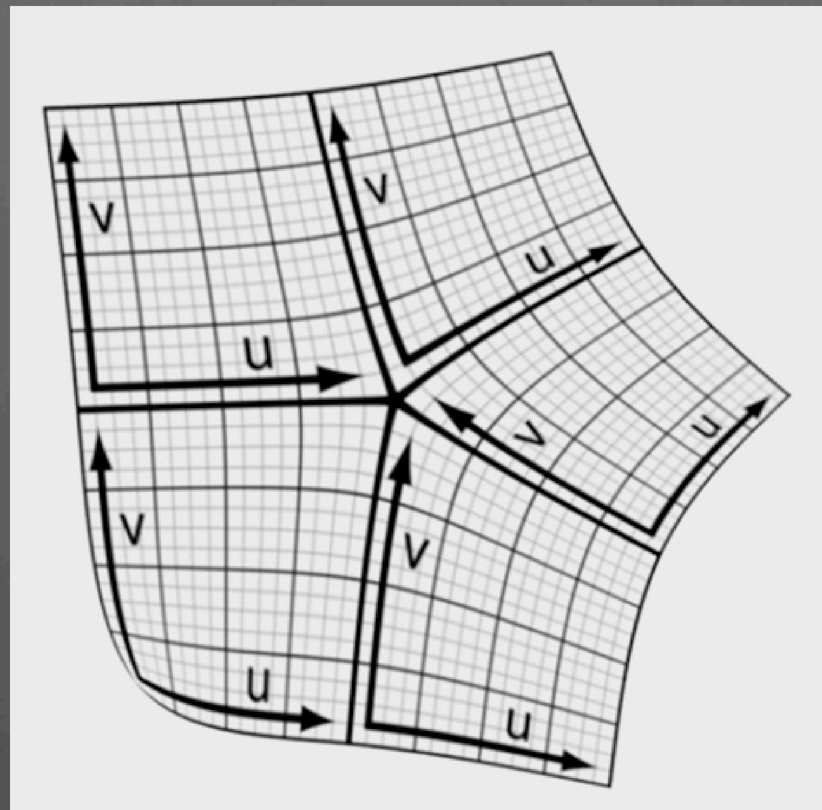


Textures stored per-face in Catmull-Clark subd domain.

Big challenge is filtering.

Per-face UVs and adjacency data





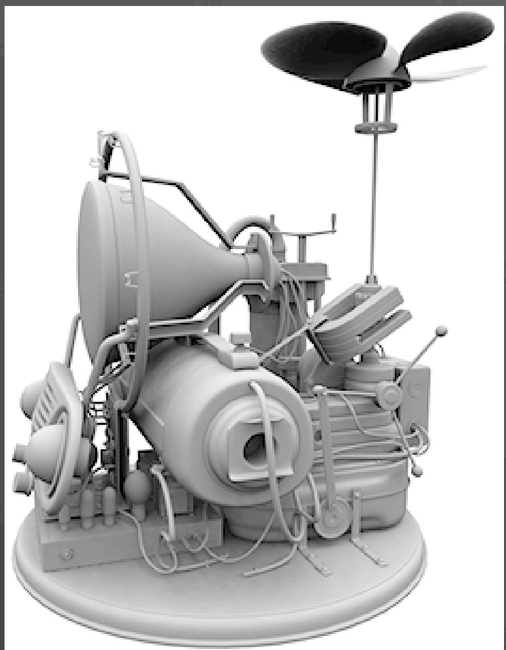
Face 7:
adj. faces: {-1,8,17,-1}
adj. edges: {x,3,0,x}, *x=don't care*

Adjacency data used to filter across face boundaries.

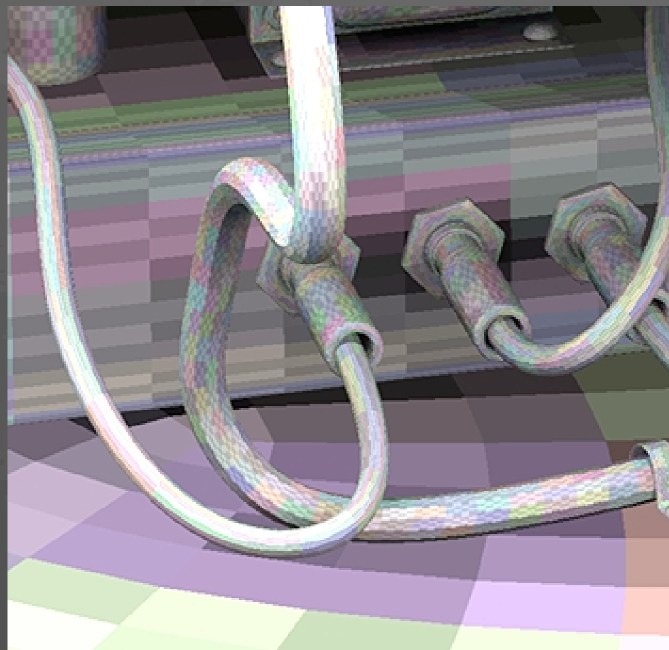
Painting

- Artists paint on limit surface in 3d painting system (tumble, paint, project).
- Auto-size feature compensates for parametric distortion.
- See video demonstration.

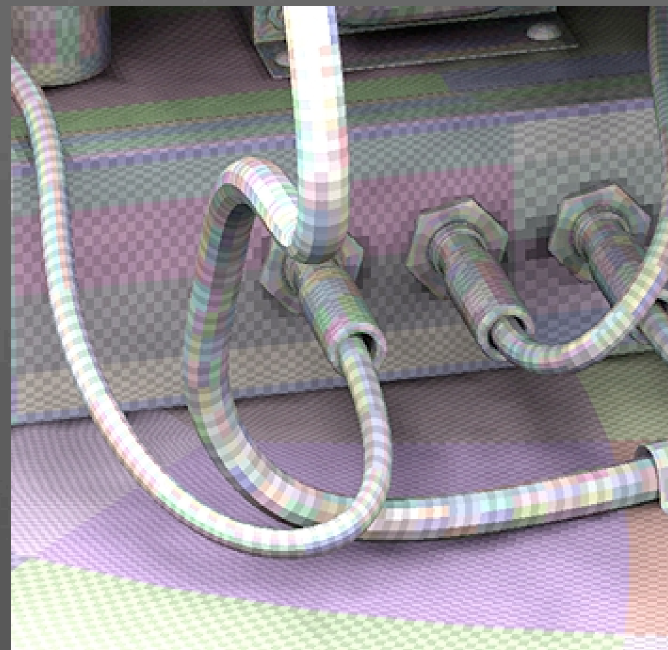
Texture sizing



212,536 faces
1 ptex file



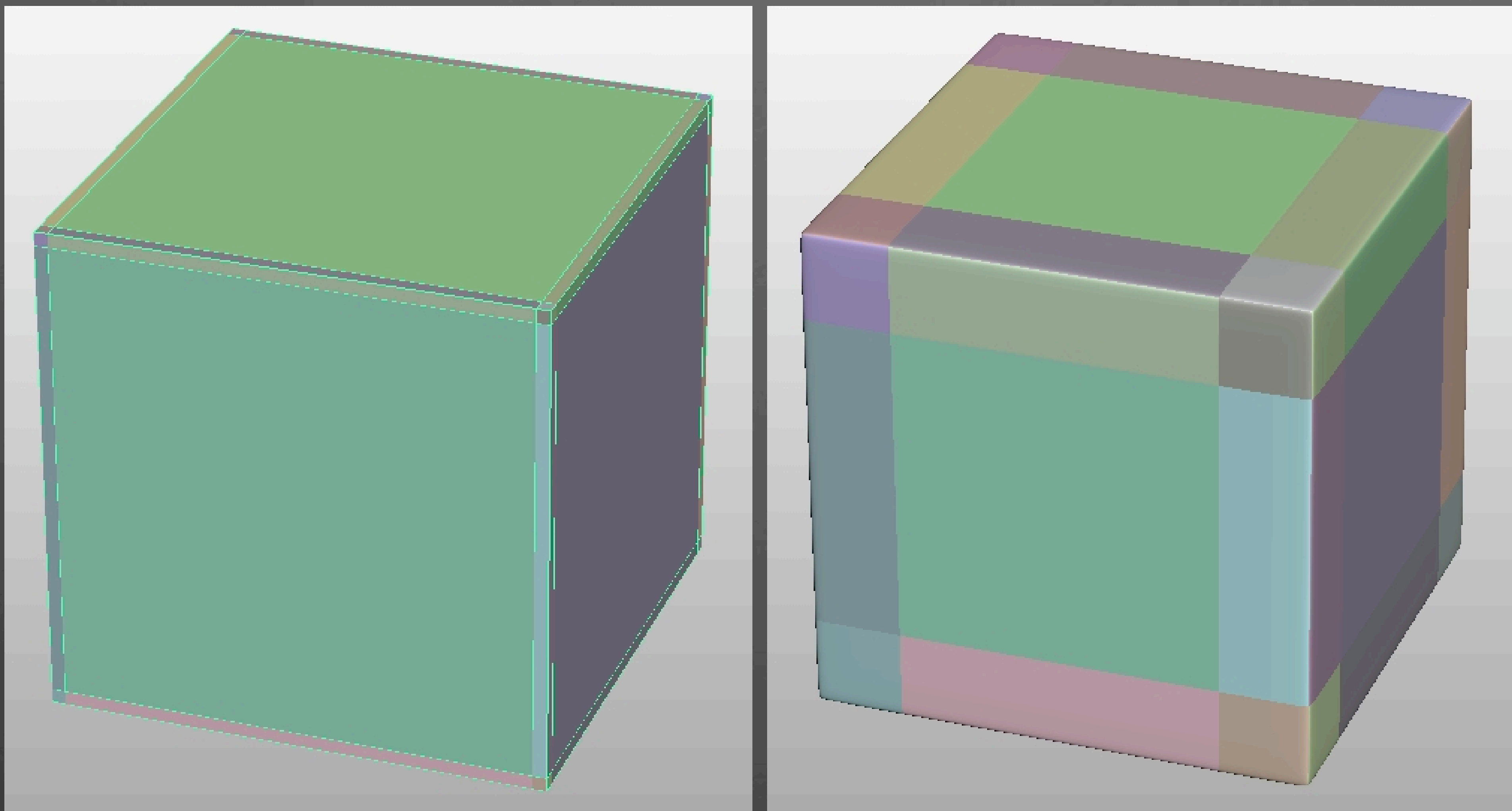
4x4 texels per face
3.4 million texels



Auto-sized
3.4 million texels

Uniform-res would simplify things, but would be inefficient (purple face at bottom has $1/256^{\text{th}}$ the texels of the auto-sized version for same texel budget).

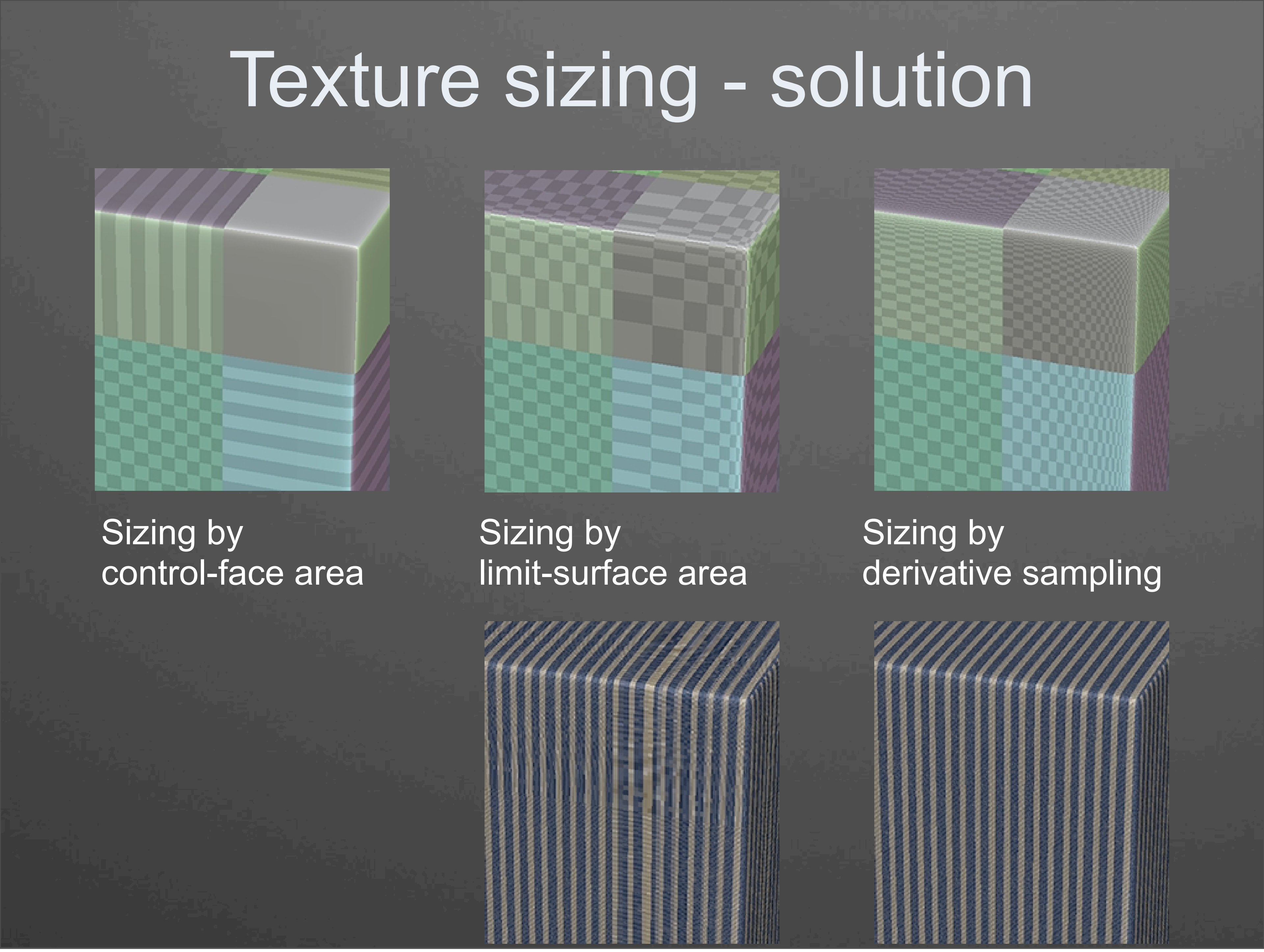
Texture sizing - problem case



Subd cage　　Limit surface

Naïve approach: size textures based on control-face area.

Problem: small faces spread out on limit surface; texels stretched too thin.



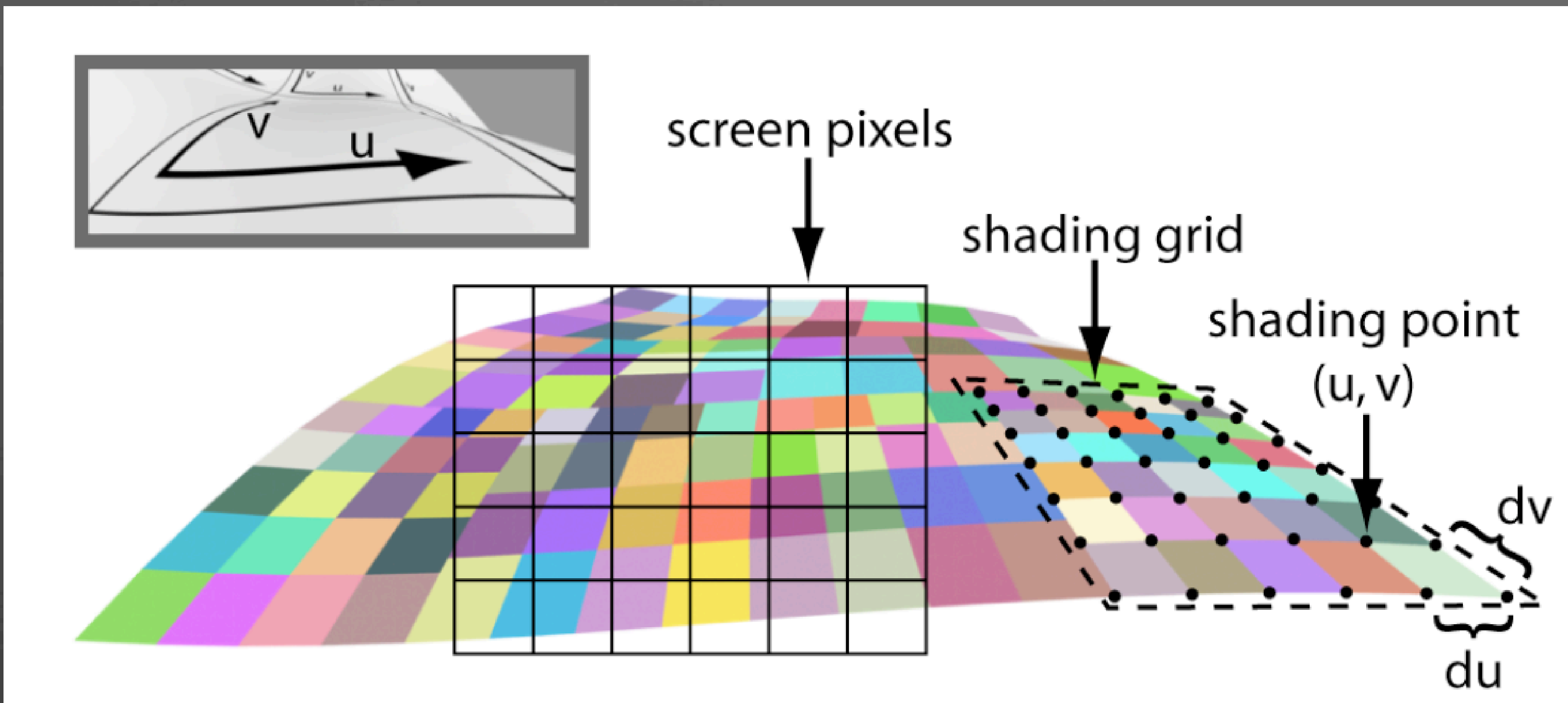
Sizing by limit-surface area *almost* works. But texels can be non-uniformly distributed over face.

Sampling derivatives over face and sizing based on max stretch point gives optimal results.

(Be careful of infinite derivative at EVs.)

Filtering

Filtering in PRMan

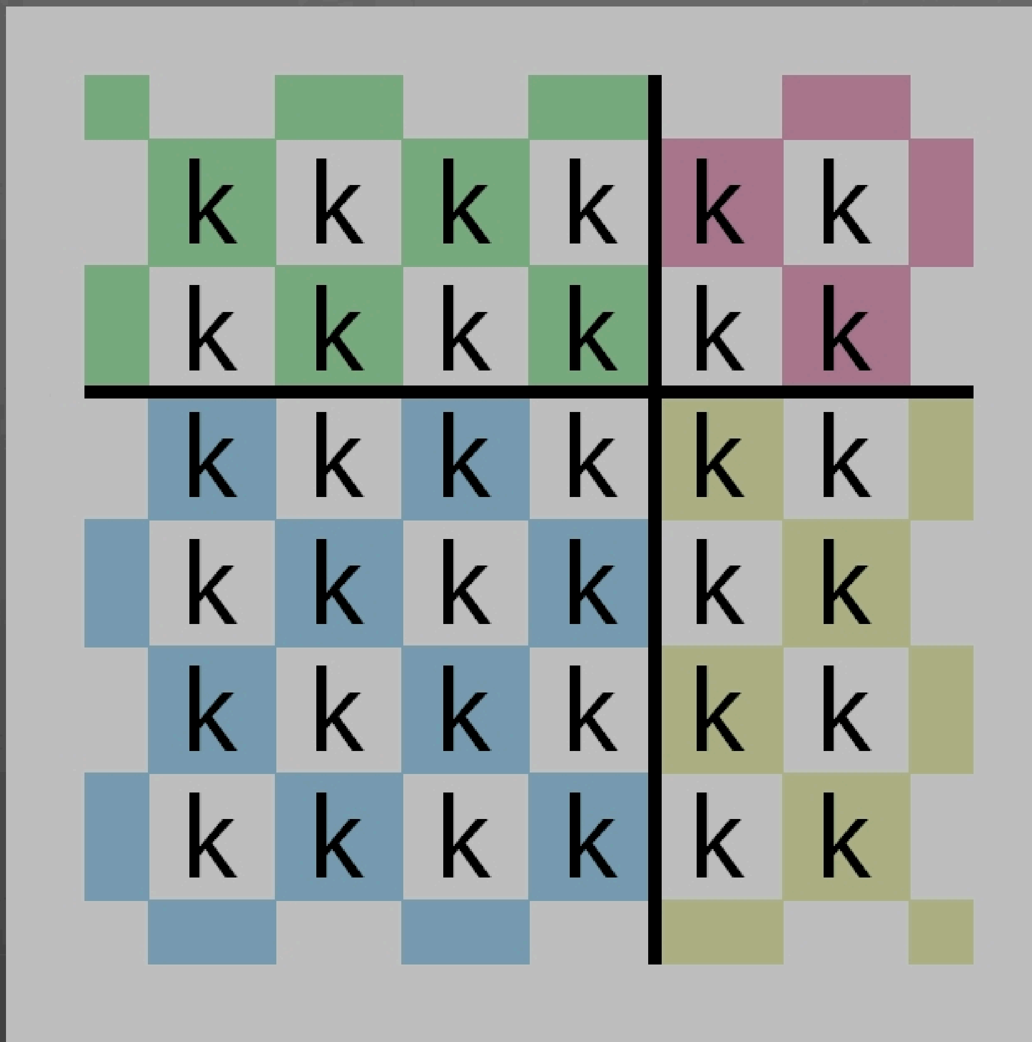


Shading grid spacing determines filter region.

Filter size is du by dv.

Filter rectangle always aligned with per-face textures. Easy anisotropic filtering!

Filtering

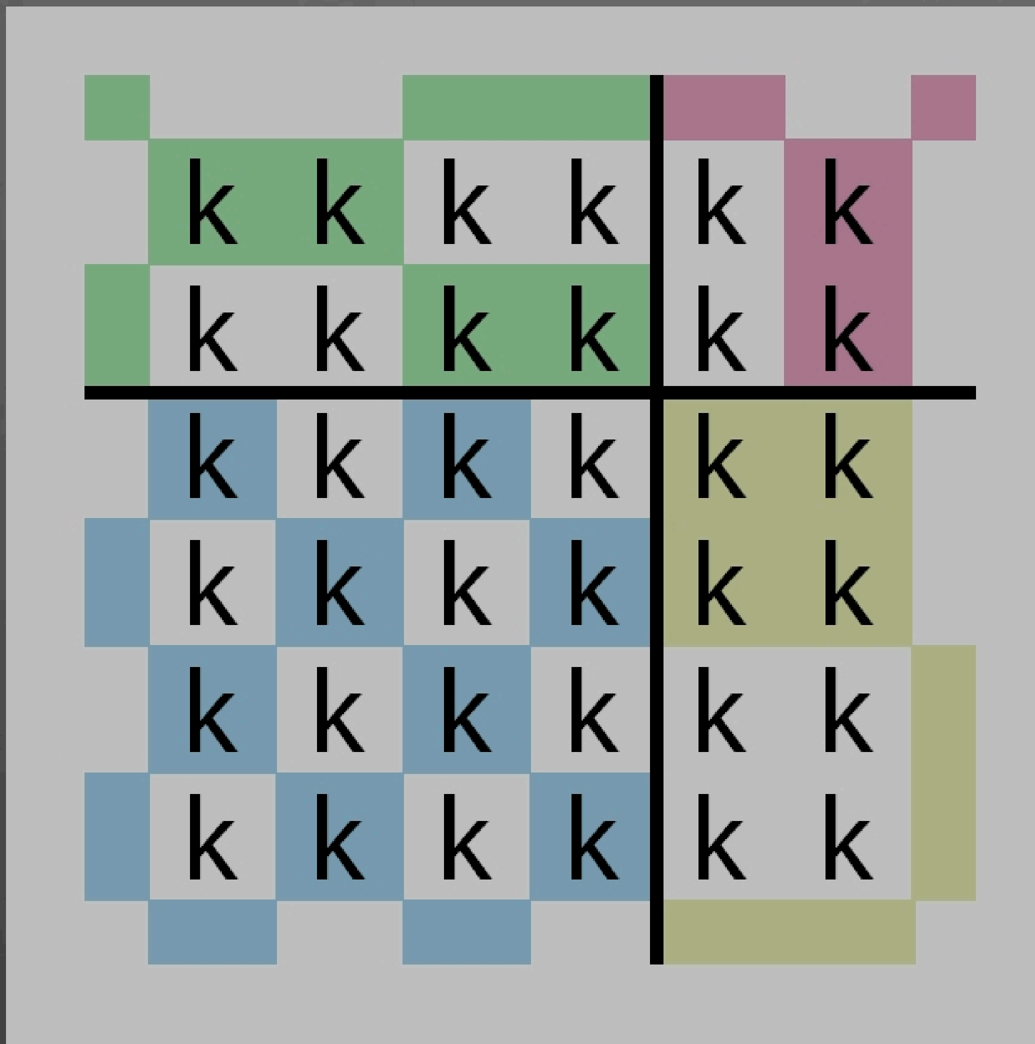


First, we choose the optimal texture res and clamp it against the max res for the local face.

Stored mipmaps and dynamic anisotropic reductions keep kernel size reasonable.

If the overlapped faces have enough res, just apply to each face.

Resolution Mismatch



When insufficient res is available for a neighboring face, just apply to nearest texel.

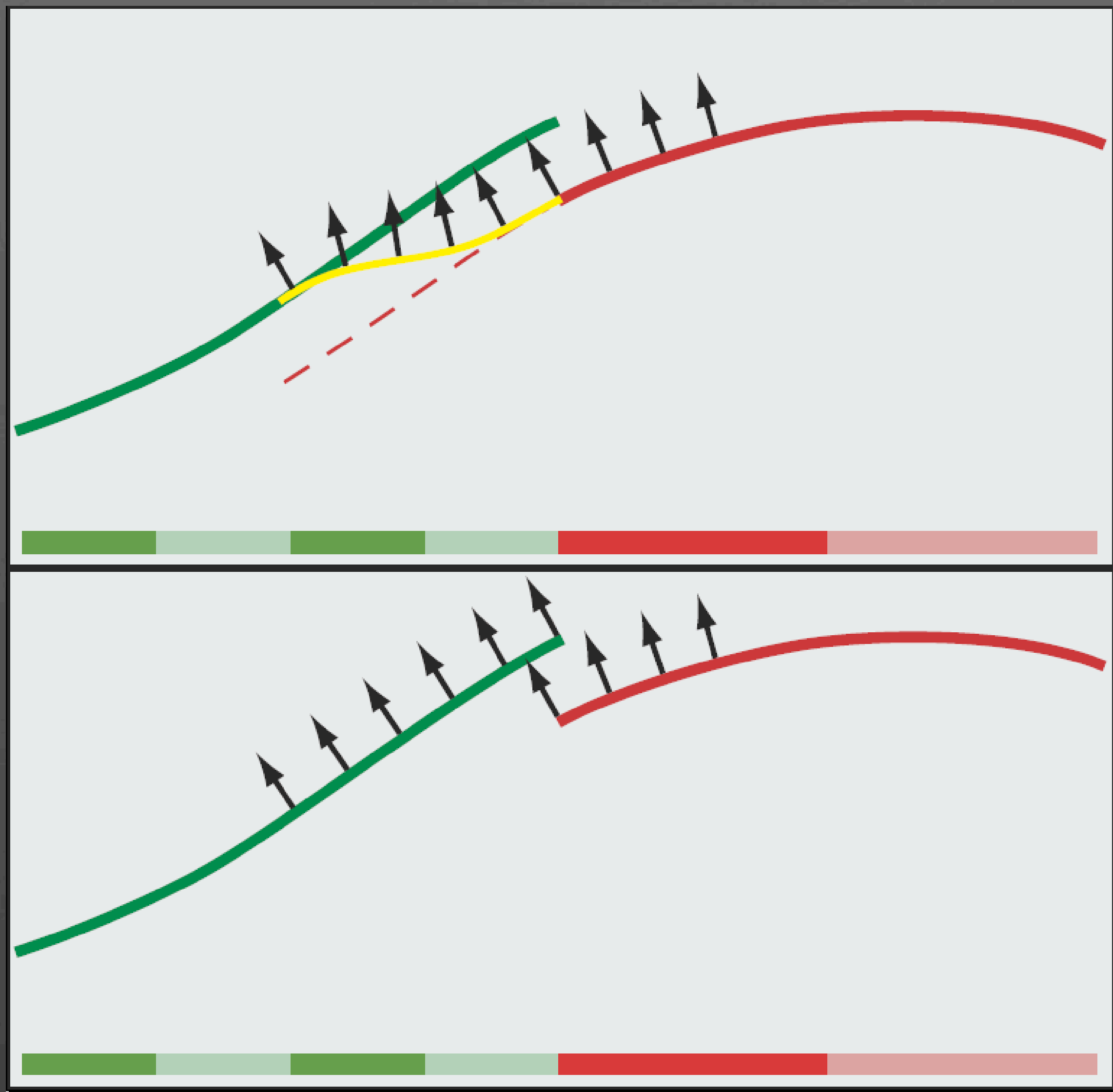
Works because we don't normally paint low-res textures – no extreme magnifications.

Resolution Mismatch (cont.)

Smooth surface

vs.

Smooth normals

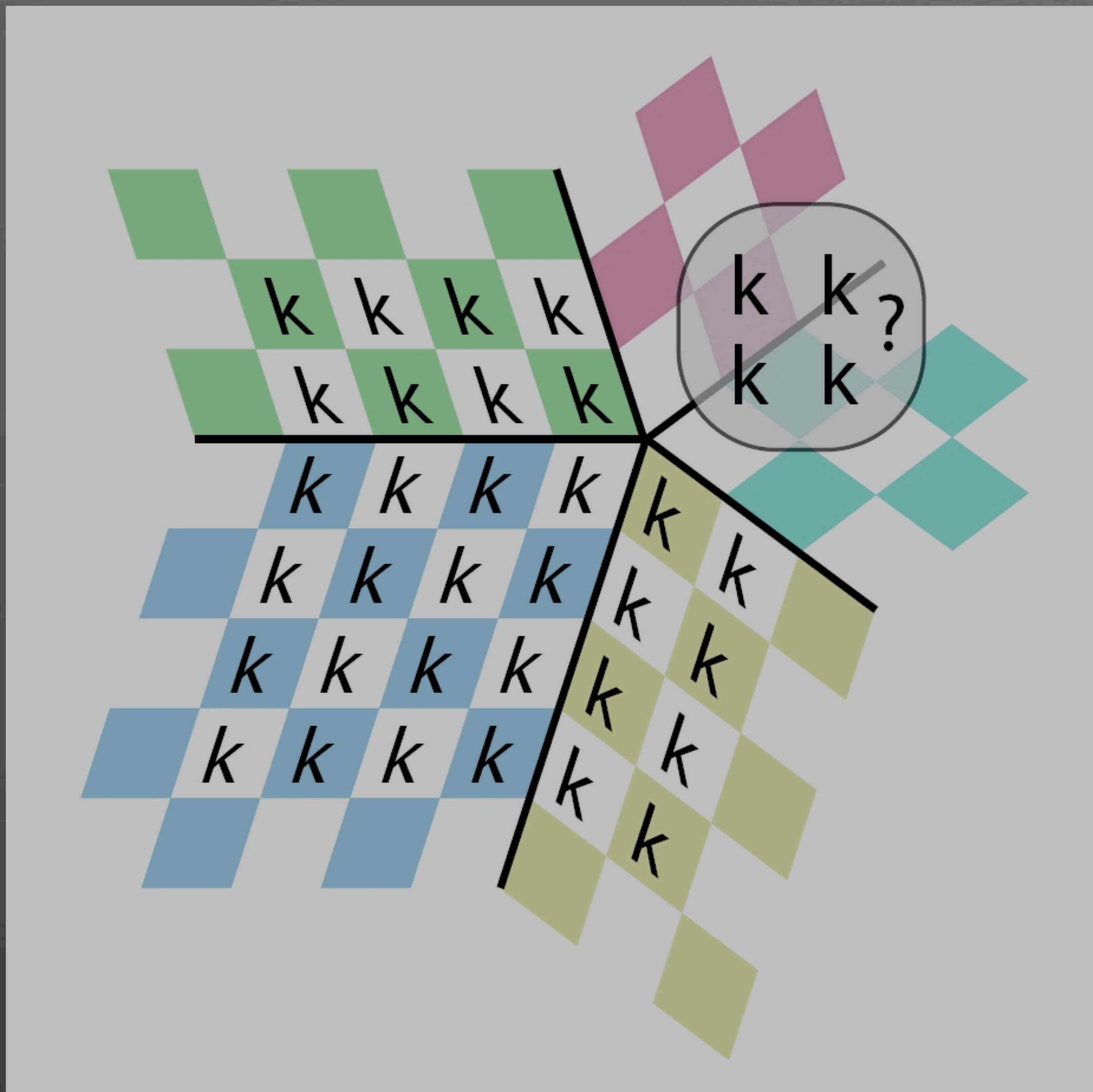


Displaced surface in cross-section:

Clamping is a problem, not because of discontinuity, but because the normals bend towards the undisplaced surface.

Discontinuity is ok as long as normals point in roughly the same direction. PRMan will fill the gap but the sideways poly won’t affect the normals.

Extraordinary Vertex



Can’t apply kernel in usual way. Just ignore corner weights and renormalize.

But discontinuity is still ok.

Results

- Artists love it
- More detail than ever before
- No aliasing problems
- I/O vastly improved

Artists love it

- Modelers don’t worry about painting.
- Painters don’t worry about UVs. Just load and go.

More detail than ever before

- Sharpness control and anisotropic filtering resolve more detail than standard PRMan textures.
- Extra detail *allowed coarser shading for final 2k renders. This gave us a big speedup!*

Results

<u>1024x1024 pixels</u>	Per-patch textures	Ptex
CPU seconds	171	141
# I/O calls	18,581,058	9,209

<u>20x20 pixels</u>	Per-patch textures	Ptex
CPU seconds	27	2
# I/O calls	4,218,733	84

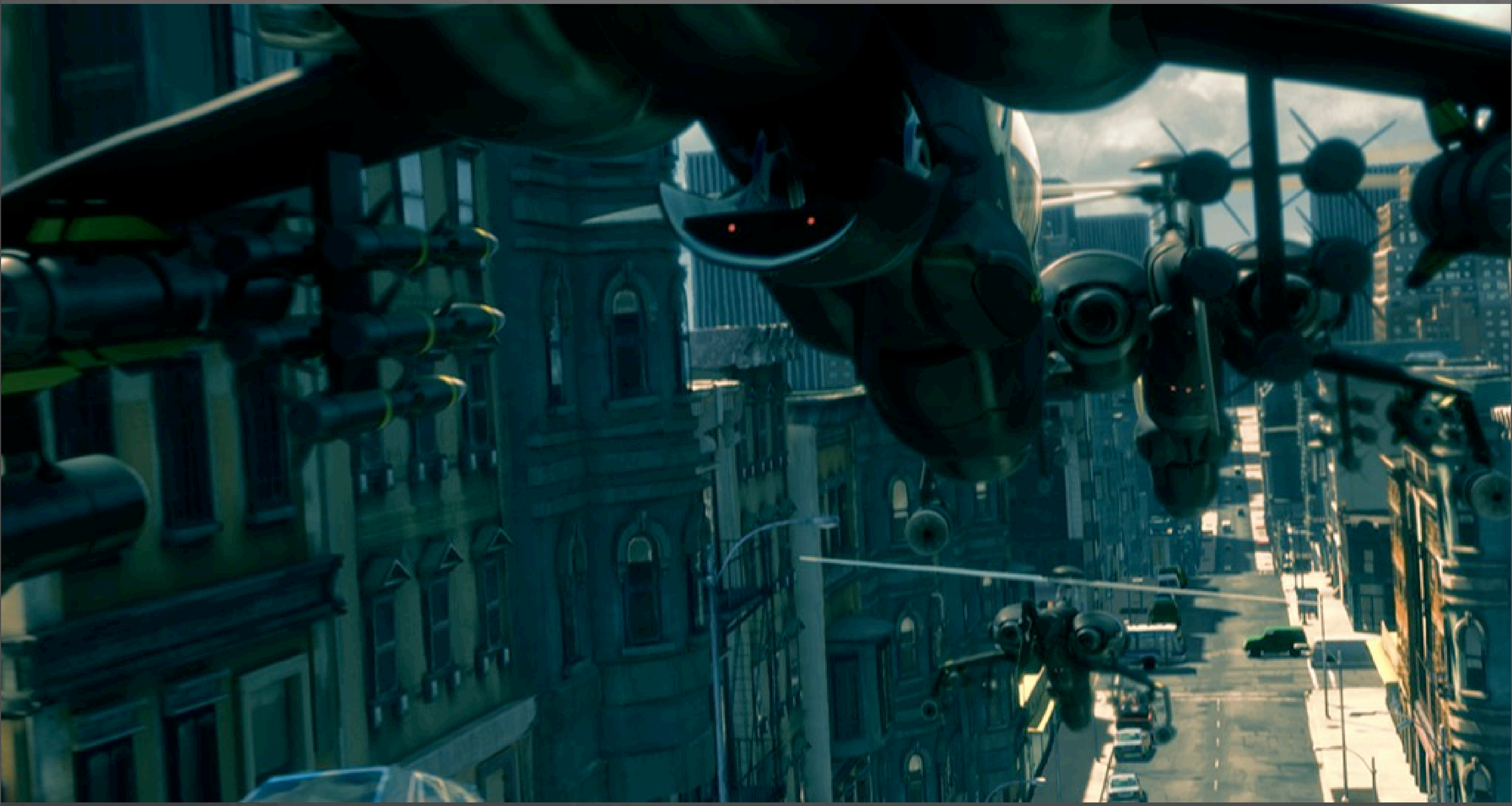
Glago's Guest 2008

Short film. First production to use Ptex.

Wanted it for ease of use.

Appreciated it for efficiency – rendered entire short every night as regression test!

Bolt



Bolt has nearly completed production. Ptex has been an unqualified success.

Ptex was used on virtually every surface in Bolt.

Future

- Currently adding non-quad support (straightforward, described in paper).
- Considering Loop subd support. Interesting question - what shape should texels be on a triangle mesh (triangular, quadrilateral, hexagonal)?