

*On Realistic Industrial
Rendering: Let it go*

Eugene Fiume

University of Toronto &

Alias|wavefront

elf@dgp.toronto.edu

Thanks to A|w Rendering Dudes

- Andrew Woo.
- Kelvin Sung.
- Jim Craighead.
- Sanjay Bakshi.
- Changyaw Yang.
- Andrew Pearce.
- John Dila.
- Ryan Meredith-Jones

Punchlines

- Don't time the market: It's good for research to be ahead of the curve.
- Think instead of defining the possibilities.
- Still room for realistic rendering in industry.
- Need to attend to integration, workflow, generality, scaling and robustness.
- Partnerships are possible, especially through new API's.

One Size Does Not Fit All

- Film and video effects and animation.
- Games.
- Scientific Visualisation.
- Conceptual Design.
- Industrial Design.
- Manufacturing Design.

Film and Video Renderers

- Special effects, complex shaders, tricks.
- Animation, natural phenomena, compositing.
- High quality, consistent, robust, hardware unaccelerated, performance aware.
- Big featured, BIG footprint, BIG data.
- Integrates everything you've ever seen.
- Problematic multiplatform software architectures.
- Huge cost pressures (rendering as a commodity).

Games (authoring and play)

- Small data, lots of texture, low image res.
- Extremely low cost, low margin.
- Pre-rendering combined with runtime rendering (games box or PC graphics card).
- Emphasize real-time performance over image quality.
- Very strange geometry that takes advantage of specific rendering quirks (especially in games boxes).

Design

- Many different rendering techniques (wireframe, strokes, pen and ink, hardware shaded, batch rendered).
- Match rendering quality to appropriate level of conceptual design.
- Rendering in closed loop with modelling.
- Space here for “realistic” rendering.
- But needs hardware acceleration too.
- Less intense cost pressures.

UI and Workflow

- All rendering controls must be accessible.
- An inaccessible feature is a non-feature, or worse, impedes productivity/satisfaction.
- Users don't know physics.
- Rendering is part of a closed loop system.
- Integration with user task flow is essential.
- Rendering choices must scale: e.g., depth map resolution, tessellation level.

Performance

- Things move. Animation rules.
- Caching, tiling, lazy eval., and high up-front costs amortized over many frames (time is the only coherence that's left).
- Hardware rendering used less often than it should be due to platform issues.
- Fast, complex, efficient shading and texturing and all things in between are critical tools to fill in modelling gaps.
- Performance over quality and realism: even ray tracing is often too expensive.

Realism

- It's not passe, but it must interoperate with existing features and code base.
- Better if it can be used selectively: e.g., realistic smoke, cloth, caustics, anisotropic illumination, extended light sources.
- Quality of direct lighting, cinematography, animation consistency, are more important than full transport sol'n.
- Full physical solutions are likely only in high end design or other special settings.

IBR

- Straight IBR is already useful in games and film (no surprise there, since environment maps are thin edge of the wedge).
- Quality and scaling concerns.
- Acquisition of moving objects?
- Sampling issues.
- Full light fields are problematic.

Is Research Failing Industry?

- So, research is not solving today's industrial problems. I would be worried if it were.
- Do long term stuff: find challenging research that answers specific questions.
- Technology transfer will happen in its time.
- Renderers of tomorrow are not likely to be physically based, but more effects will be.
- Need to learn more about software architectures.
- Consider partnerships with industry.

Some Open Problems

- Mesh parameterization and approximation.
- Interactive high-quality rendering.
- Multi-rate sampling and reconstruction.
- Fast approximate visibility and shadows.
- Device independent rendering solutions.
- Better quadrature and error bounds.
- Better reflectance and lights in non-physical renderer (and nonphysical effects in opposite).
- Many systems issues to deal with complexity.