Production Perspectives on High Performance Graphics

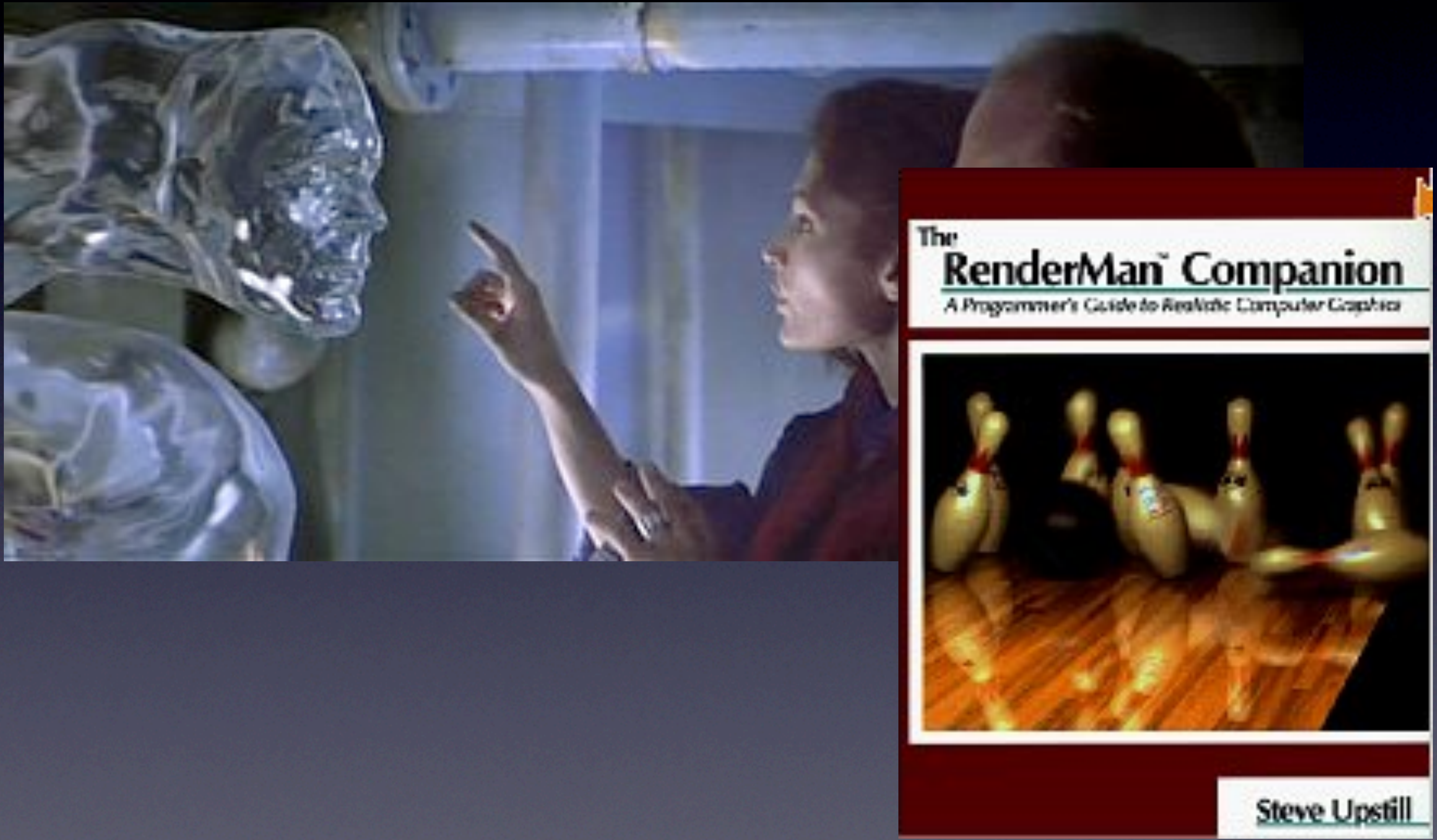
Larry Gritz

sony pictures imageworks

My perspective

- Been on all sides: research, SW development, production artist, HW company, vendor, in-house, user
- I’m a noncombatant now

1989: 20 years ago...



Film rendering

- 24 fps, 20-200 frames/shot, 100-2000 shots
- Must have NO artifacts (AA, MB, DOF, smooth surfs)
- 2-10 GB geometry input, > 100 GB texture
- Want 4-5 hours, accept 10-12 hours
- Typical machine: 8 core, 16-32 GB
- Imageworks: > 5000 cores total

Film vs games

Games

- Render 10^5 frames x many games x 10^6 users
- Fixed frame rate, quality negotiable

Film vs games

Film

- Render *once*, deliver film/digital
- Playback rate unrelated to render time
- Quality fixed, render time negotiable
- Optimize for development & art
- Humans are bottleneck

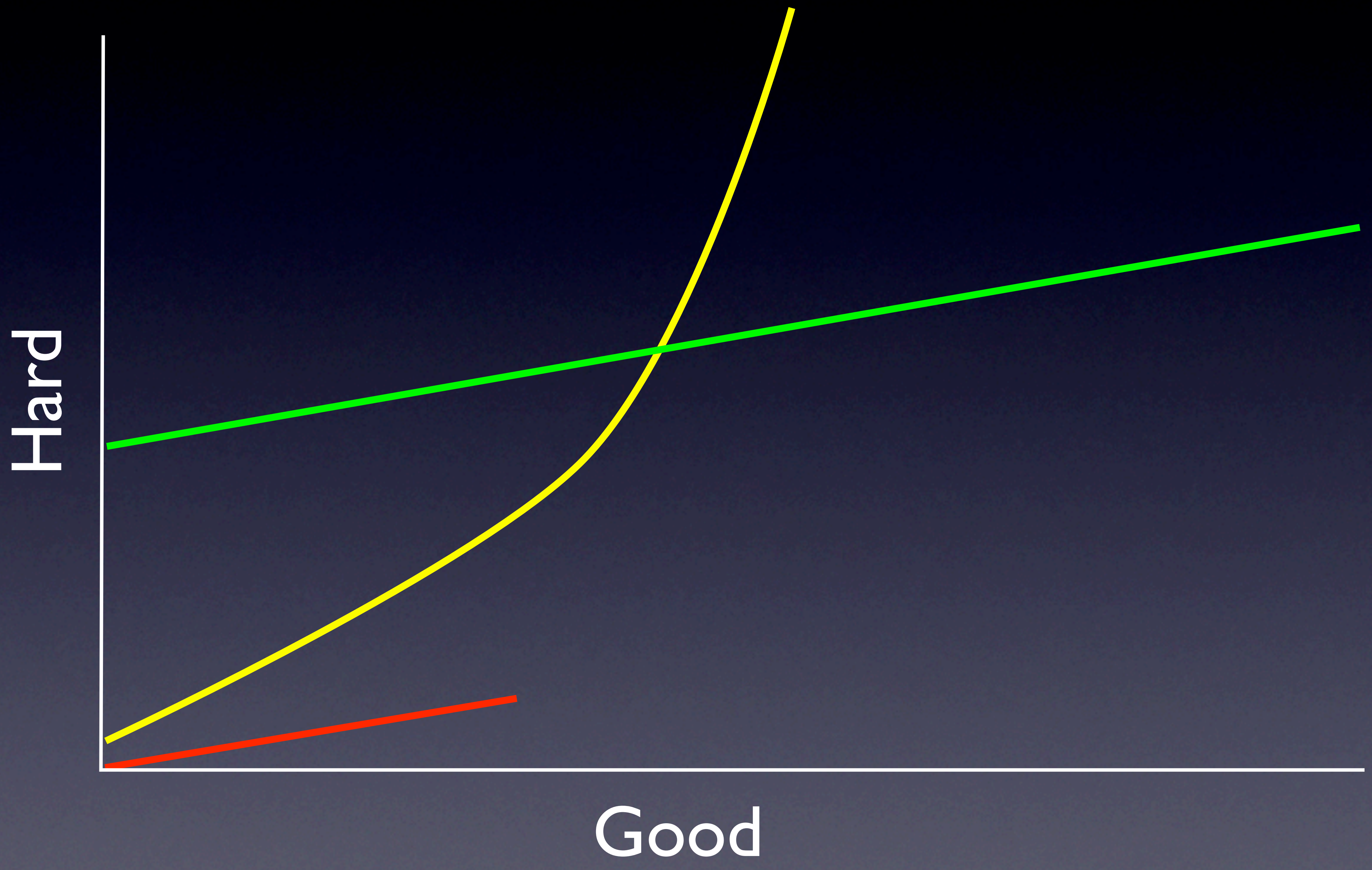
Why film will never be realtime

- Games lie
  - not really 60Hz; doesn't count level load
- Amdahl's Law
  - fixed costs: disk I/O, network, upstream stuff
  - 10-100 frames to amortize over, not 10,000
- Blinn's Law

Why little GFX HW in VFX

- Physics: power, heat/AC, space
- Availability in our blades
- Economics
- Developer time
- Chicken and egg

Design Space for Complexity



Larry Gritz
High Performance Graphics 2009

sony pictures imageworks

Conventional Wisdom

- Ray tracing, GI could be used sparingly
- Not necessary, maybe undesirable
- Too slow to be practical

New Wisdom

- Ray tracing, GI needed *everywhere*
- Frame time less important than pipe time
- Must slash through complexity

Problems with Reyes

- Dense meshes overshade
- Redundant/useless shading
- Harder to parallelize
- Tricky for non-rectangular geometry
- Hodgepodge of techniques for effects
- Most advantages gone with RT/GI

Rendering at Imageworks

- Traditionally used a Reyes-based product
- Wanted GI look for *Monster House* (2006)
- Licensed “Arnold” from Marcos Fajardo
- Arnold co-developed by SPI and SolidAngle
- Arnold is an unbiased path tracer, no scanline front end

“Monster House” footage here.
Sorry, I can’t distribute this.
Rent the video!

Rendering at Imageworks

- After *Monster House,* Arnold used for auxilary rendering, further development
- Sole renderer for *Cloudy With a Chance of Meatballs*
- VFX: Eagle Eye (2009), some *G-Force*
- Now: primary SPI renderer
- In progress: Alice in Wonderland, 2012, Arthur Christmas, Hotel Transylvania, ...

“Cloudy With a Chance of Meatballs” footage here.

Sorry, I can’t distribute this.

View the trailer on the web, see the film when it comes out (Sept 18, 2009)

Rent the video if you are reading this in the future.

Restaurant Stats

- Beauty: 5.8 hours @ 4.1 GB
- Outside: 18 min @ 1.0 GB
- Atmosphere: 32 min @ 2.2 GB

Ray tracing advantages

- Single pass, simpler compositing
- Artists: Much less data/pass management
- More intuitive, predictable, accurate
- Fewer lights, more reuse of lighting rigs
- Development: simpler code, no longer a clutter of separate techniques
- Easy interaction / progressive refinement
- Down-side: displacement no longer "free," lots of re-training

Ray tracing speed

- RT beauty render > Reyes beauty
- RT total < Reyes total (shadows, reflections, ambocc)
- RT artist time << Reyes artist time
  - Pass and data management were dominating artist time
  - CLO lighting throughput doubled

Inspired by real lighting

Alice in Wonderland

“Alice in Wonderland” footage here.
Sorry, I can’t distribute this.
View the trailer on the web, see the film when it comes out (March 5, 2010)
Rent the video if you are reading this in the future.

2012

“2012” footage here.
Sorry, I can’t distribute this.
View the trailer on the web, see the film when it comes out (Nov 13, 2009)
Rent the video if you are reading this in the future.

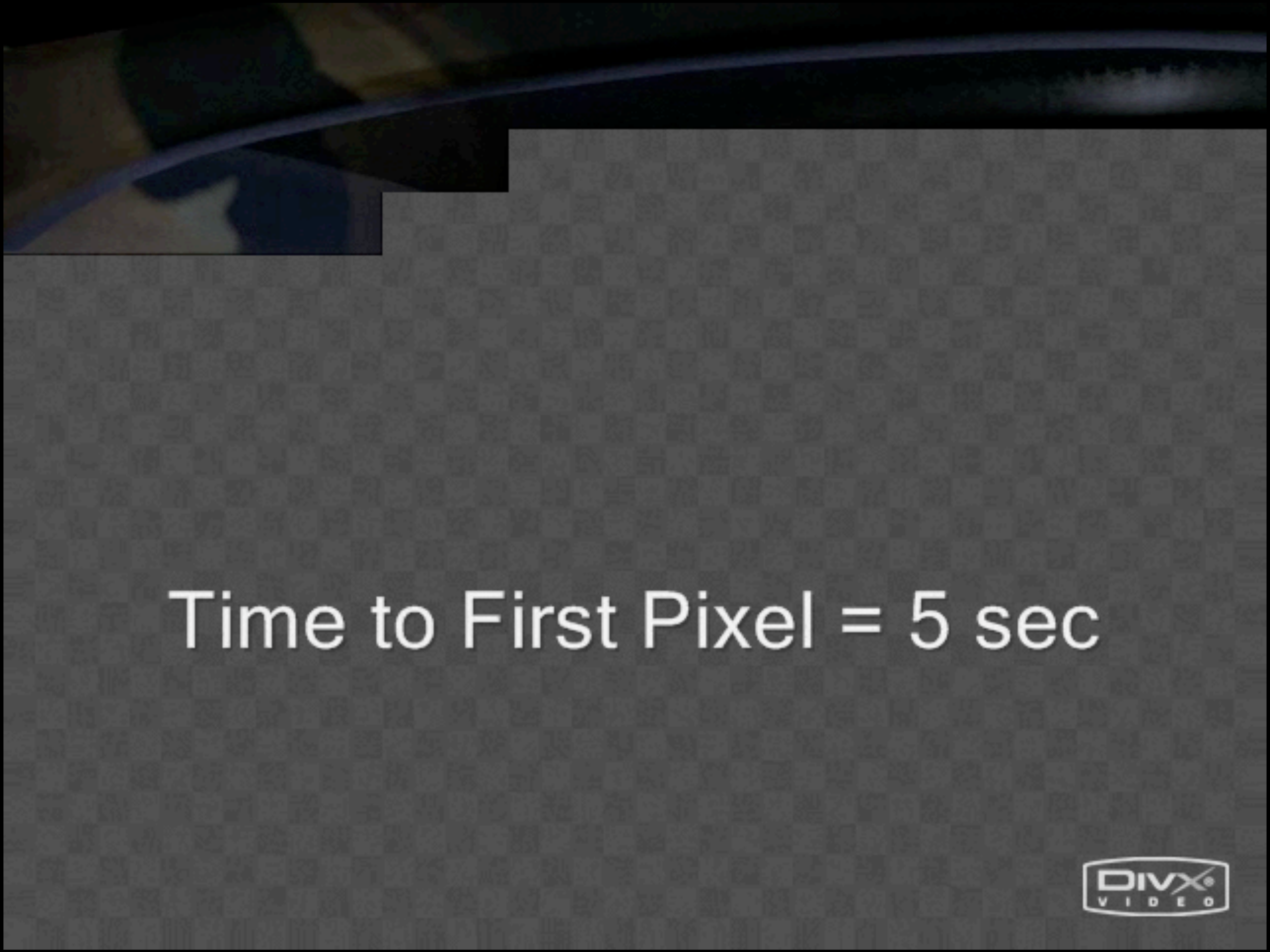
2012 war story

Relighting Strategies

- Just submit another frame
- Light in composite
- "Deep buffer" (with or without GPU)
- 3D GPU
- Ray trace coarse to fine

Sorbetto

- Gelato 2.1 relighting engine
- Cache partially-shaded grids
- Reshade things that changed
- Blast shaded grids at GPU



Time to First Pixel = 5 sec

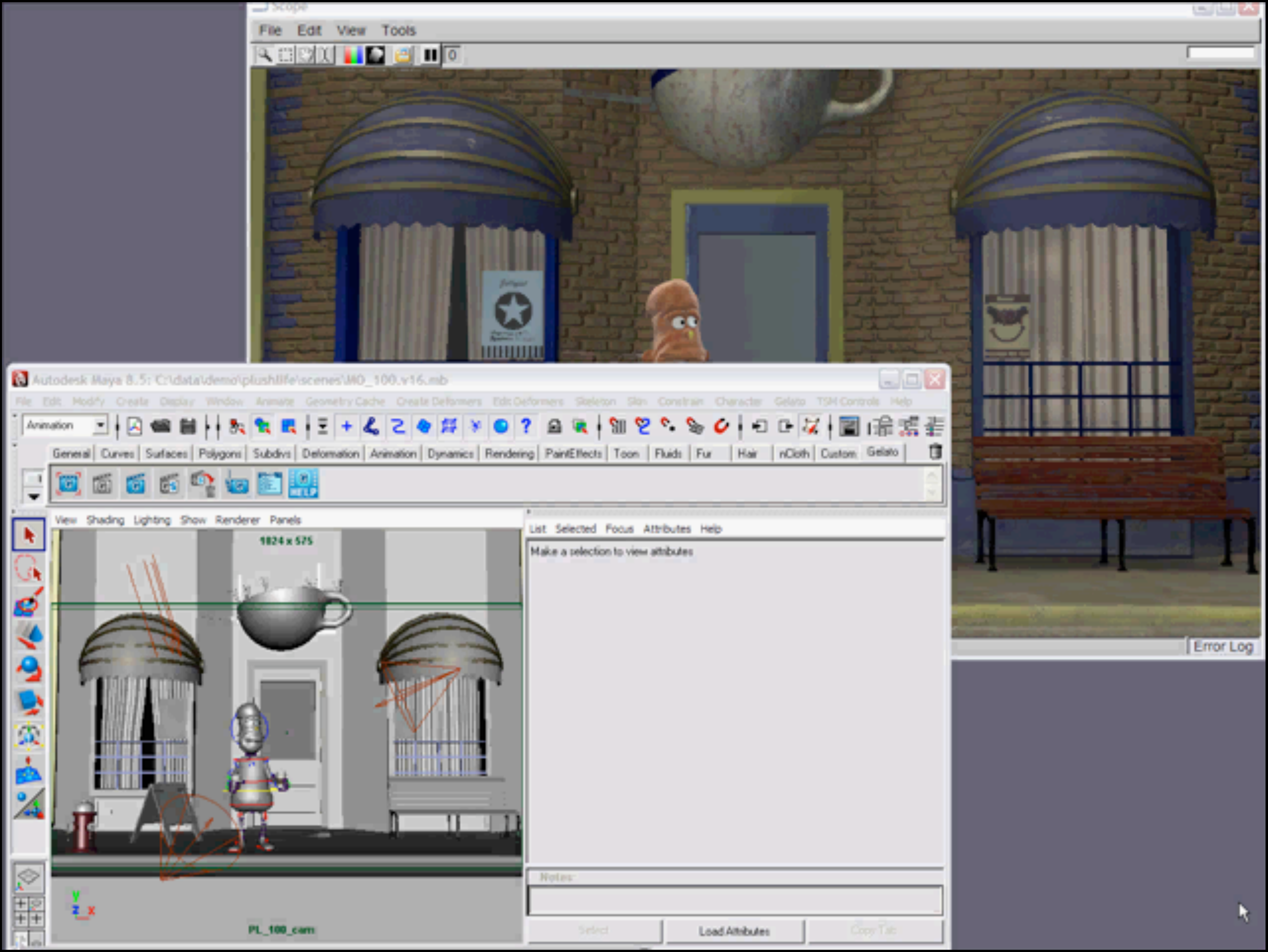
Sorbetto postmortem

- 10x sounded good on paper
- Not interactive enough for complex frames
- Complexity limits

Mocha

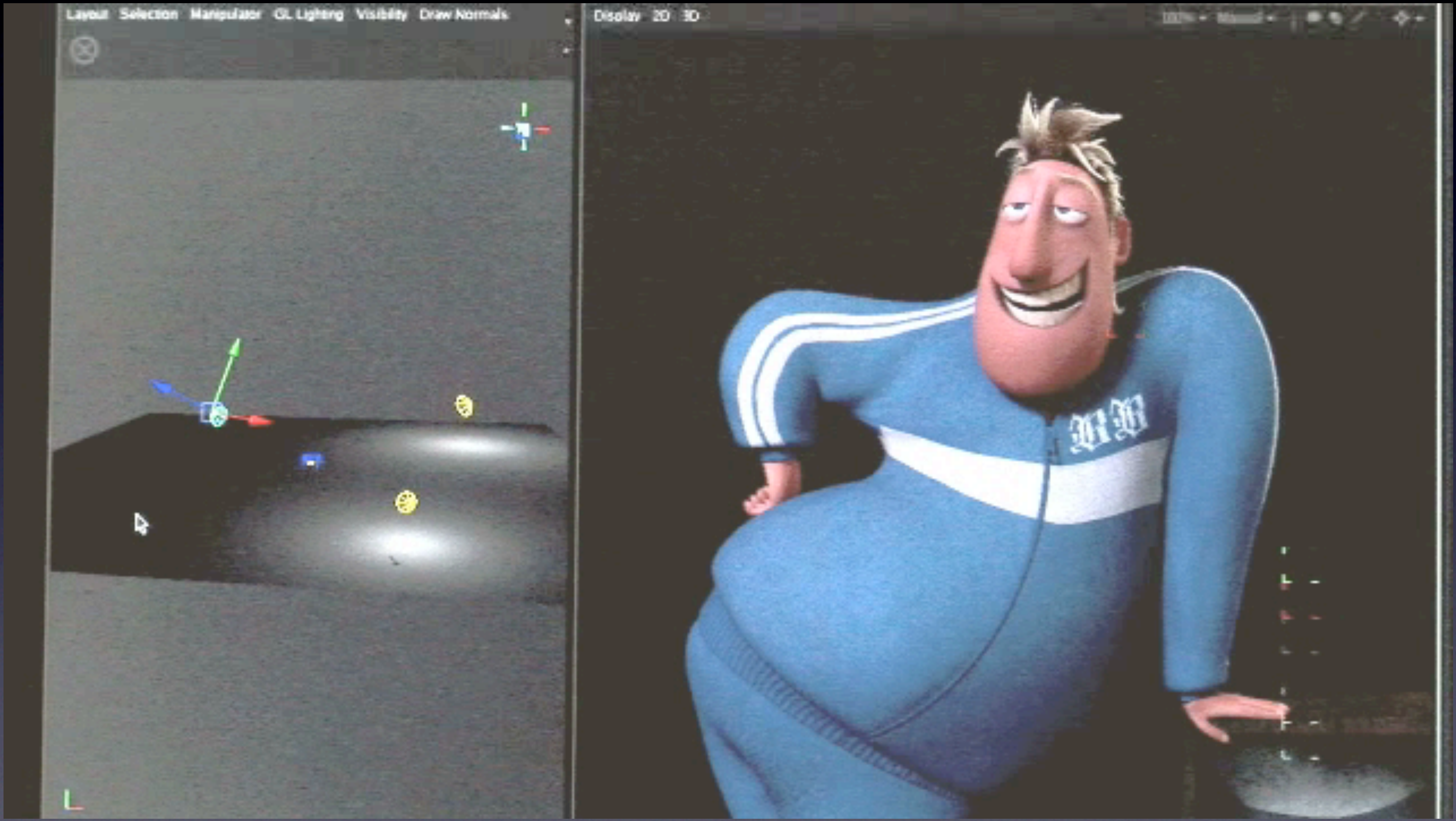
- Gelato 3.0 AKA “Mocha”
- Even more GPU-centric
- Full translation of shaders to Cg on the fly
- ~1 Hz update rate
- Previewed @ SIGGRAPH 2007
- Killed before shipping

the plush life
Rx
Love



Make a selection to view attributes

Arnold


Layout Selection Manipulator GL Lighting Visibility Draw Normals
Display 2D 3D

Miscellaneous

What’s wrong with shaders

- Black boxes
- Can’t sample
- Can’t defer
- Can’t reorder
- Can’t reason about them in any way
- Suboptimal for a modern ray tracer

New Shading Language

- Similar to RSL/GSL, but evolved
- Ray tracing deferred (except when needed)
- Shaders compute “closure” of outgoing radiance
  - Can “evaluate”, “sample” ... later
  - Closures can be combined
- Defer evaluation, batch rays, reorder, importance sample, bidirectional, ...

Open Shading Language

- We're open-sourcing the whole thing
  - language spec, compiler, runtime implementation
- Part of bigger open source effort
  - http://opensource.imageworks.com
- Sorry, OSL not quite ready
  - hope to have specs up in weeks, code in months

Impediments to HPG

Larry Gritz
High Performance Graphics 2009

Which HW/SW stack?

- Dizzying array: muti-CPU GPU {which?} Cell LRB Caustic OpenGL DirectX Cg HLSL GLSL Cuda CTM Sh OpenCL LRB/C++ Pthreads SSE AVX etc.
- Which choices will be relevant in 5 years? 10? 2?
- Cannot afford to code multiple times, or bet on standards that die or stagnate (or evolve too rapidly)
- Need: multi-platform, multi-OS, vendor-neutral
- Want to write once, run on many HW.

Game consoles vs PC

- Console completely static for 5 years
- Paradox: why do games keep getting better?
- Answer: developers *learn.*
- PC GFX: new hardware every ~6 months, new architecture every ~18 months
- Not enough time to learn how to best use it
- Also makes it *very* hard for long-lifetime apps to jump on the train

What we want

- Multi-platform, multi-OS, vendor-neutral
- SW (not to mention developer knowledge) must last years
- Write once, run on many HW.
- Better/newer HW -> faster.
- NOT Better/newer HW -> recode application with new APIs
- NOT Older HW -> app doesn't run

Hey, HW Guys:

- Pick a single standard way to program
  - or a few cross-vendor stable standards
- Let us program once, run anywhere
  - without relinking would be best
- Make your hardware good
  - we'll buy the best !/$ running our SW
  - we won't/can't speculatively port to every HW/API to see what's best, or change every 2 years

Hey, Researchers:

- Equally dizzying array of new techniques
- Which is really faster: CPU, GPU, LRB?
- Paper X, Y, or Z?
- For real-world uses? Do you even know?
- Let's get serious about benchmarking
- We'll try to do our part, too (scenes, etc.)
- I admit: benchmarks are really hard

Conclusions

- Film deals with vast complexity
- Human bottlenecks > machine time
- Want to use HPG, but many hurdles
- Chicken and egg
- Would be easier with:
    - Cross-platform, stable standards for programming
    - Ways to know if/when it will pay off

Acknowledgements

- Marcos Fajardo
- Sony Pictures Imageworks
- NVIDIA
- Film clips: Sony Pictures, Disney
- Gelato: Eric Enderton, Dan Wexler, Jonathan Rice, Philip Nemec, Radomir Mech, John Schlag, Eduardo Bustillo, etc.
- Arnold: Marcos Fajardo, Cliff Stein, Chris Kulla, Jesse Andrewartha, Rene Limberger, Angel Jimenez, etc.
- Media support: Rachel Falikoff, Nikki Bell, Danny Dimian, Rob Bredow

Q&A and Reminders

- Me: lg@imageworks.com
- Cloudy release date: September 18, 2009
- Other Cloudy presentations:
  - "Cloudy" Course Wed 1:45
  - Rendering sketch 8:30 Fri
  - Deconstructing Watchmen panel Thu 8:30
- OSL, etc: http://opensource.imageworks.com
- http://openimageio.org