blue-steel
distributed

real-time

ray tracer

blue-steel
Ray Casting

For every pixel
- Construct a ray from the eye
- For every object in the scene
- Find intersection with the ray
- Keep if closest
- Shade depending on light and normal vector

Finding the intersection and normal is the central part of ray casting

MIT EECS 6.837, Durand
Ray Tracing

- Secondary rays (shadows, reflection, refraction)

MIT EECS 6.837, Durand
Recursion For Reflection

0 recursion

1 recursion

2 recursions

MIT EECS 6.837, Durand
Resolution
1024x1024
Recursion depth
100
Rendering time
2.6 seconds
optimization

SIMD SIMD SIMD SIMD

pipelining

no inheritance in rendering
Regular Grid

MIT EECS 6.837, Durand
Gizmoball
3D physics
Parallelization

task is intrinsically parallelizable

each SPU renders every sixth line of screen

SPUs DMA pixel data directly to frame buffer

PPU controls physics, program initiation, scene parsing
performance

linear speedup with respect to number of cores used for rendering measured in length of time taken to generate a given frame final framerate with reflection, refraction, procedural shading, bump mapping, shadows, and blending: 10 frames per second total rays traced: over 5,000,000 per second
problems
team

scott fisher
michael d'ambrosio
brian sweatt
natalia chernenko
rj ryan
leevar williams