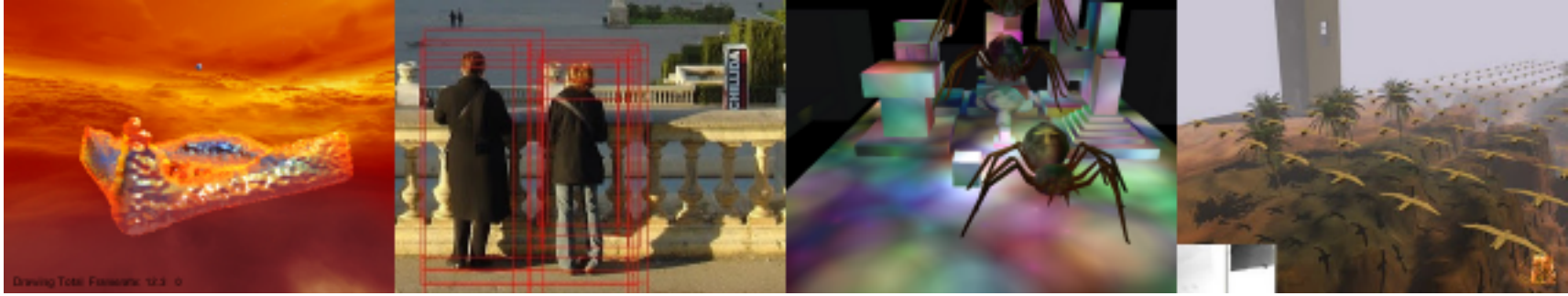


CIS 565 Project Ideas

Patrick Cozzi
University of Pennsylvania
Spring 2012

CUDA Rasterization Pipeline

- Implement a rasterization pipeline in CUDA or OpenCL. How can we make it faster than the OpenGL pipeline?
  - Change sort-first, sort-middle, etc. based on load
  - Relax draw order constraints when it makes sense
  - Removed stages when not necessary
  - Hierarchical culling

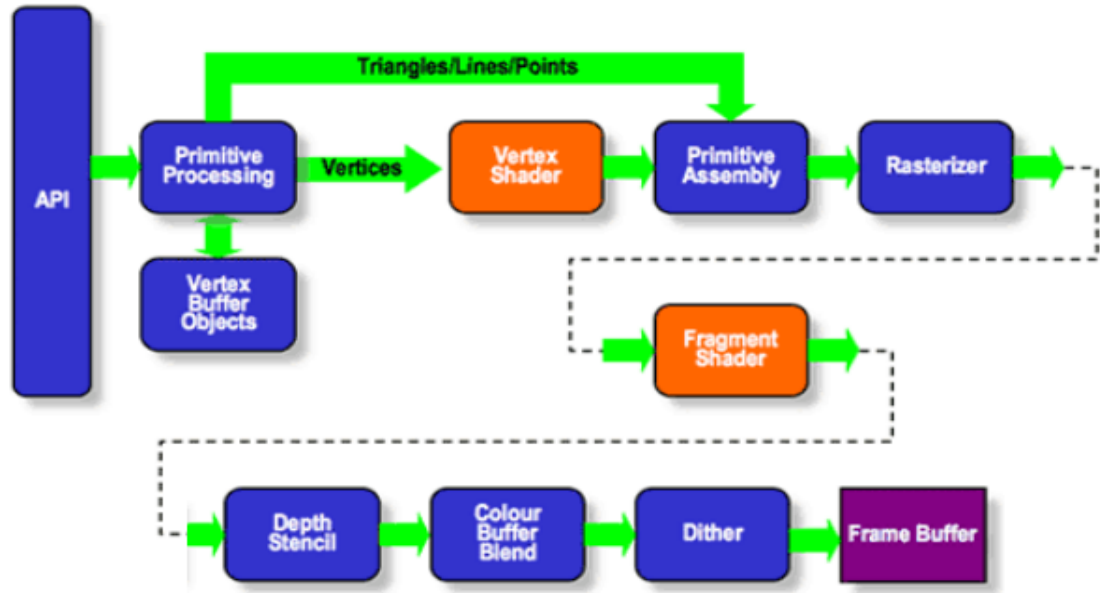
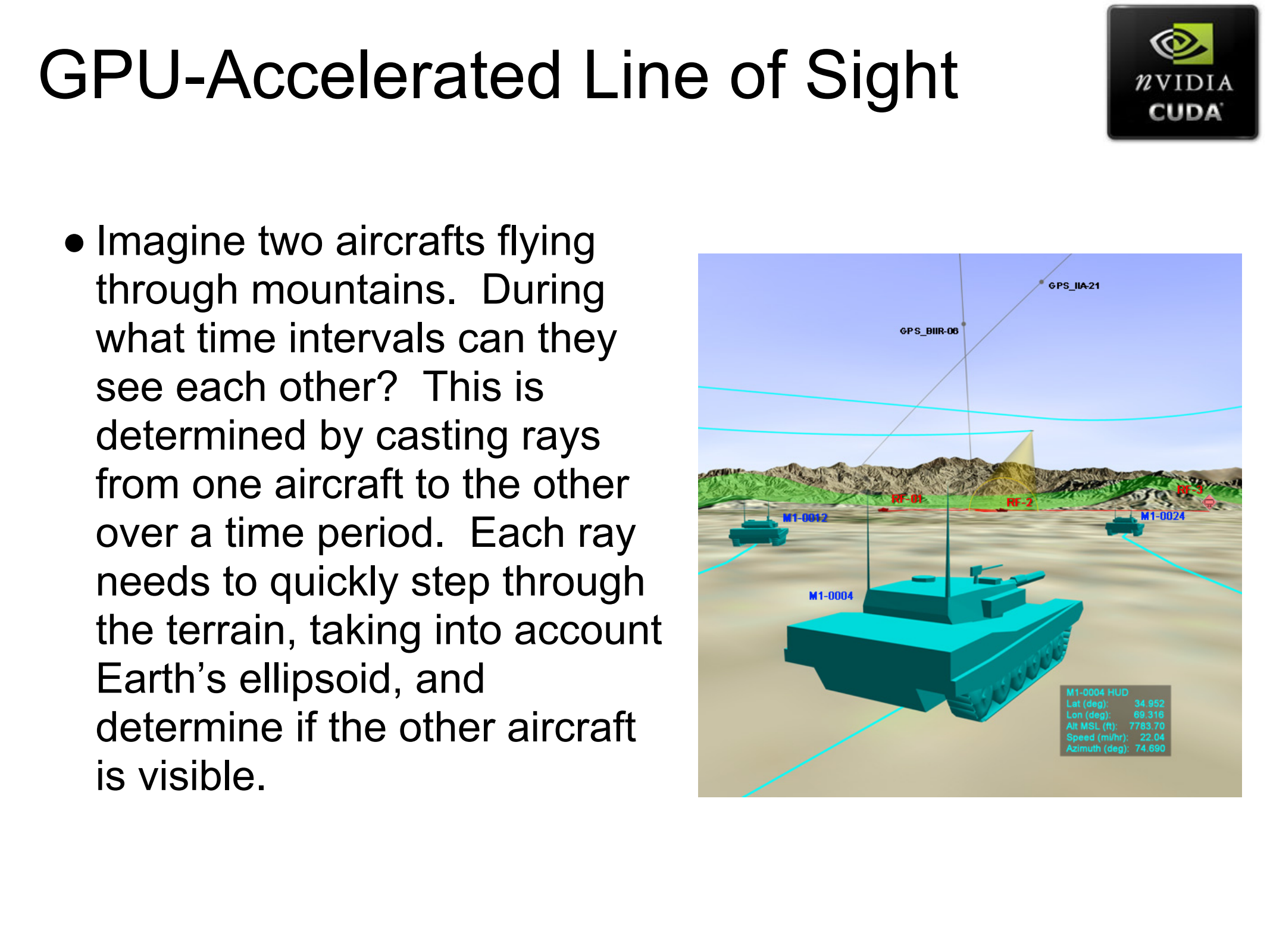


Image from http://www.khronos.org/assets/uploads/developers/library/2011-devcon5-santa-clara/DevCon5-WebGL-WebCL-and-

GPU-Accelerated Crowd Simulation

- From Mubbasir: "we can greatly exploit synchronization slack in multi-agent simulations by having agents read stale data of other agents and still have plausible agent behavior."
- Experience from student who tried it last
  - Abstract: http://dl.dropbox.com/u/1851229/Temp/565/ExploitingSynchronizationSlackinMulti-agentSimulations.pdf
  - Report: http://dl.dropbox.com/u/1851229/Temp/565/mark-project-report.doc

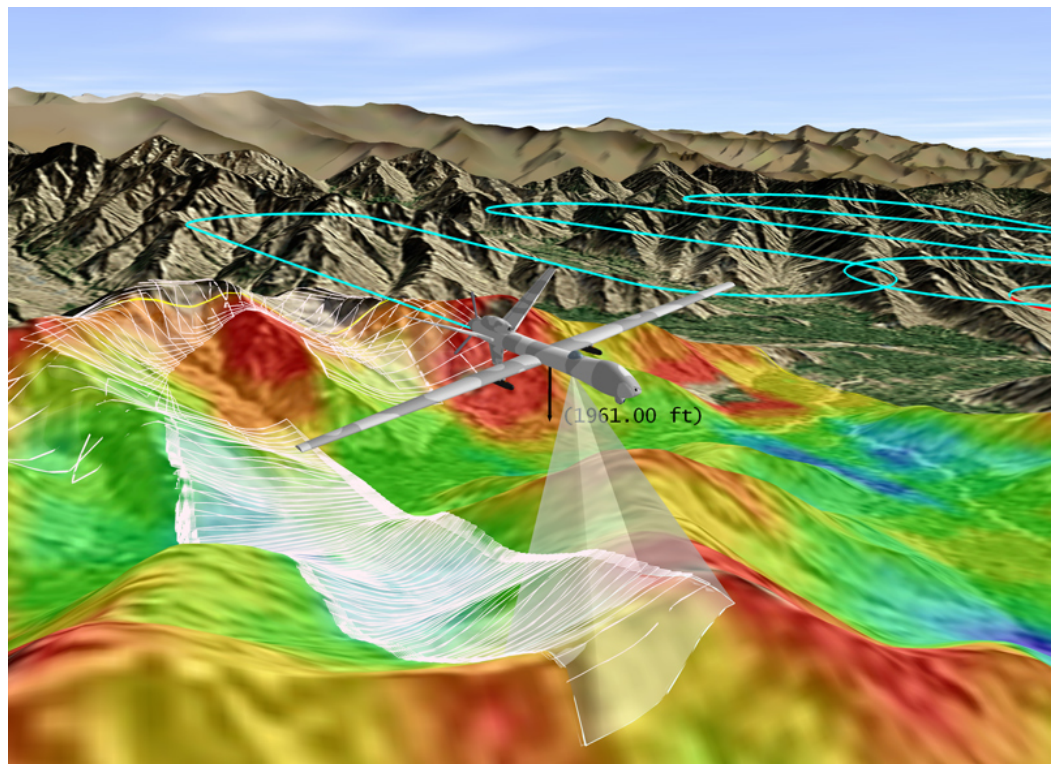
GPU-Accelerated Line of Sight



- Imagine two aircrafts flying through mountains. During what time intervals can they see each other? This is determined by casting rays from one aircraft to the other over a time period. Each ray needs to quickly step through the terrain, taking into account Earth's ellipsoid, and determine if the other aircraft is visible.

GPU-Accelerated Access

- Sensor volumes, such as those formed by a UAV's camera's field of view, can be modeled using CSG. Determining when a point moving over time is inside the volume is called access. Given many sensors and many points, use the GPU to exploit the parallelism in access.



GPU-Accelerated Label Declutter

- Displaying thousands of potentially overlapping labels creates clutter. *Declutter* automatically moves labels so they do not overlap. In the general case, this is an NP-hard problem. Can a reasonable solution be obtained with GPU-accelerated k-means clustering? What parallelism can we exploit? Also consider simulating this as a mass-spring system.

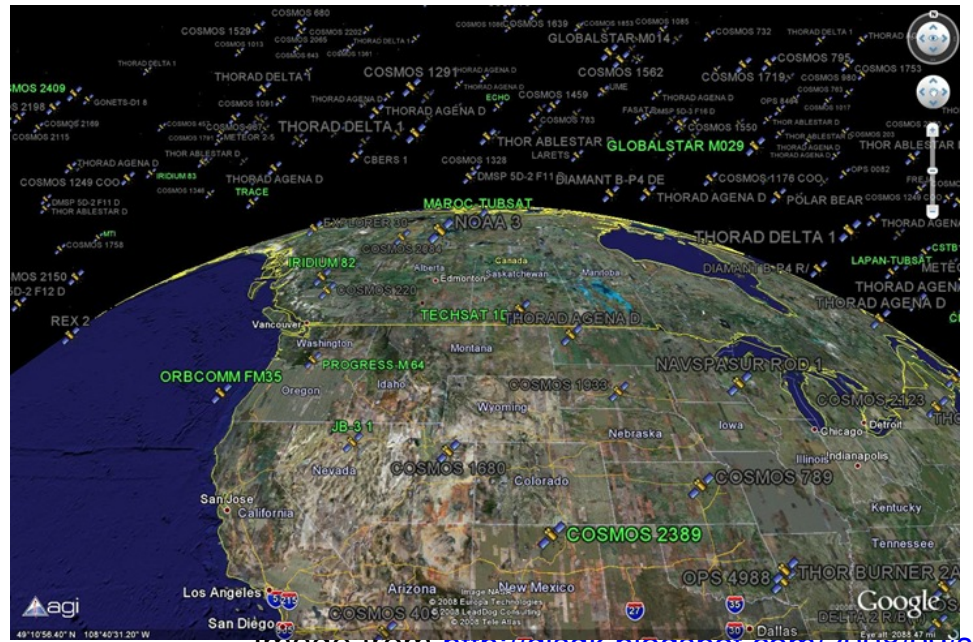
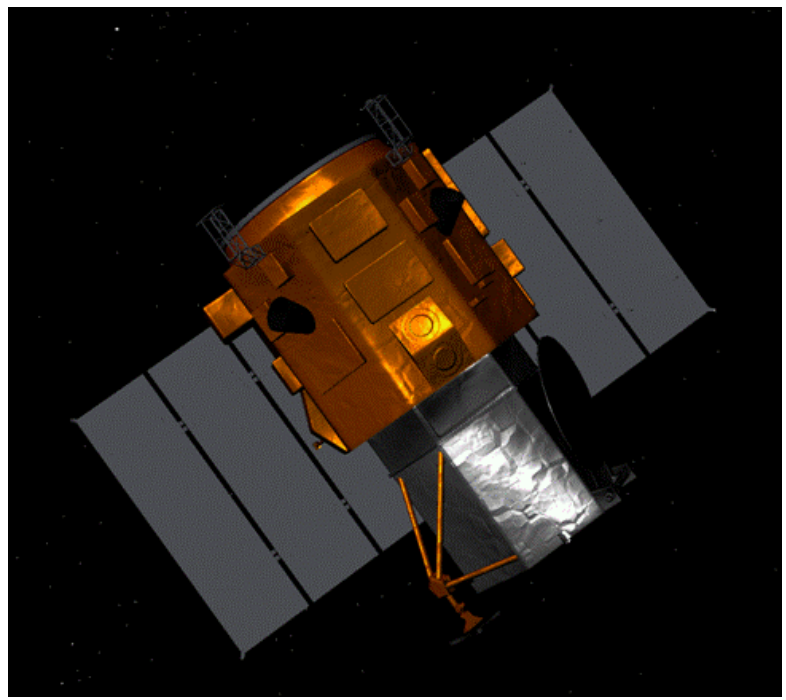


Image from http://gisdk.blogspot.com/2008/09/satellites-tracked-in-google-earth.html

GPU-Accelerated Solar Panel Temperature

- As a satellite orbits Earth, its solar panels change temperature based on their initial temperature, exposure to the sun, and material properties. When the panels are obscured by Earth or other parts of the satellite, their temperature decreases. Use the GPU to parallelize it at each time step.



GPU Particle Trimming

- Particle trimming can improve performance by reducing fragment load and only slightly increasing vertex load.
- Is a geometry-shader implementation worthwhile? What are the use cases other than dynamic particles? How common are they?
- See http://www.humus.name/index.php?page=Comments&ID=266

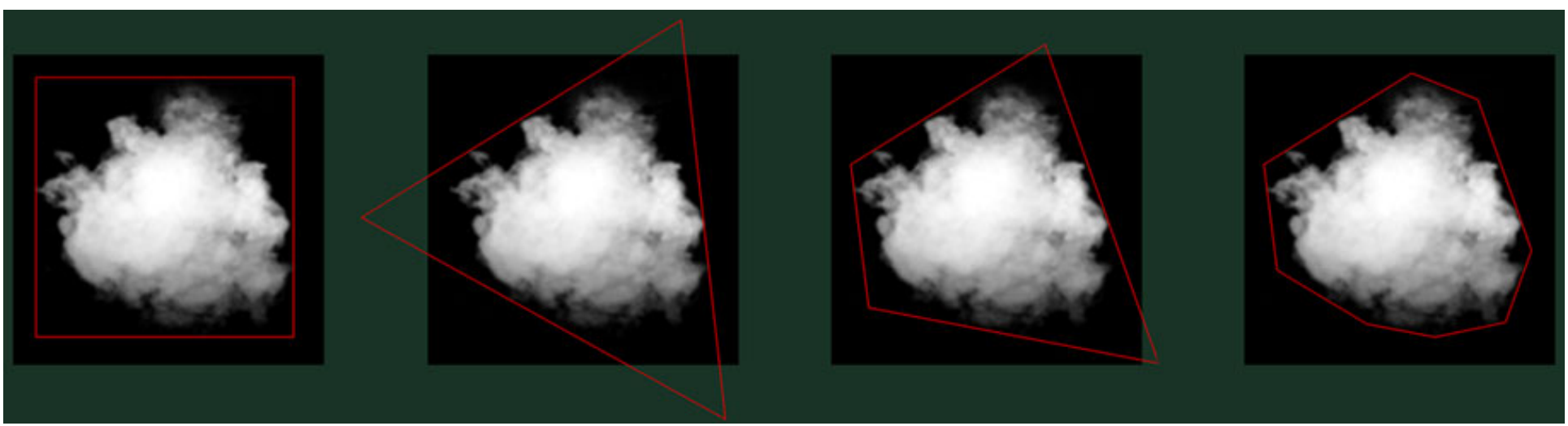


Image from http://www.humus.name/index.php?page=Comments&ID=266

Multi-GPU Deferred Shading

- How do we scale rendering across multiple GPUs? How do we load balance?
- For deferred shading, perhaps render g-buffers with one GPU. Apply post-processing effects with the other. Are there really any benefits to this?
- See OpenGL Insights, *Programming Multi-GPU's For Scalable Rendering*

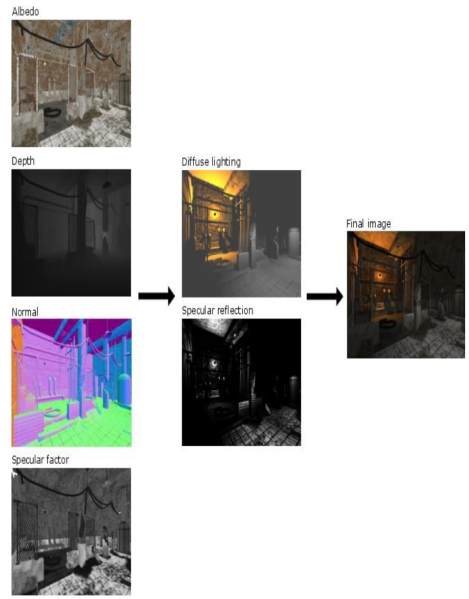


Image from http://www.leadwerks.com/files/Deferred_Rendering_in_Leadwerks_Engine.pdf

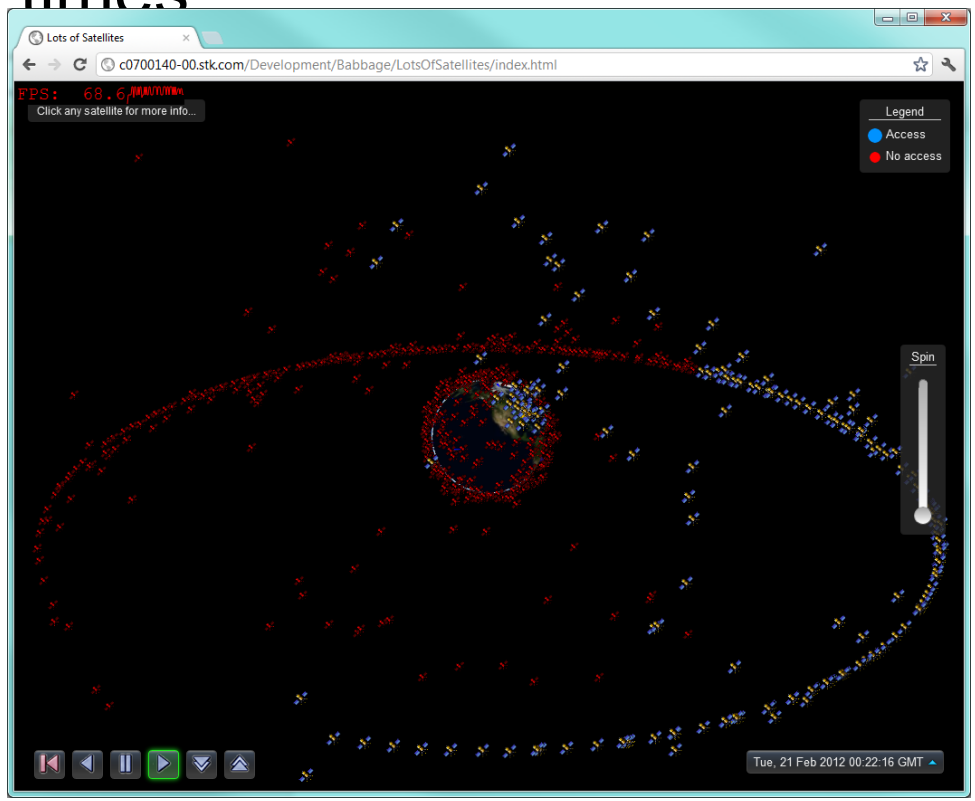
Ocean Simulation and Rendering

- There is plenty of literature on simulating and rendering ocean surfaces, and rendering massive amounts of terrain, but how do we simulate and render oceans at a global-scale - from world-view to person-view? How do waves interact with the coastline and other objects.

Image from http://outerra.blogspot.com/2011/02/ocean-rendering.html

GPU-Accelerated Satellite Orbits

- Given thousands of satellites each with an array of time/position pairs (think keyframes), animate the satellites overtime using lagrange interpolation in a vertex shader. Render using WebGL. Each array may not have the same number of pairs or times



Volumetric Clouds

- Render volumetric clouds with light scattering via GPU ray casting. Clouds can come from real-world data, can be modeled by hand, or simulated in code.
- Great chapter in Game Engine Gems 2
- Also see http://vterrain.org/Atmosphere/Clouds/

Image from http://vterrain.org/Atmosphere/Clouds/

WebGL Deferred Shading

- Without multiple render targets (MRT), deferred shading is considered too expensive; however, GPUs have come along way since this thought.
- Even though WebGL lacks MRT, and, therefore requires multiple geometry rendering passes, can we still implement an efficient deferred shader in WebGL? How does it compare to a forward shader with the same effects?

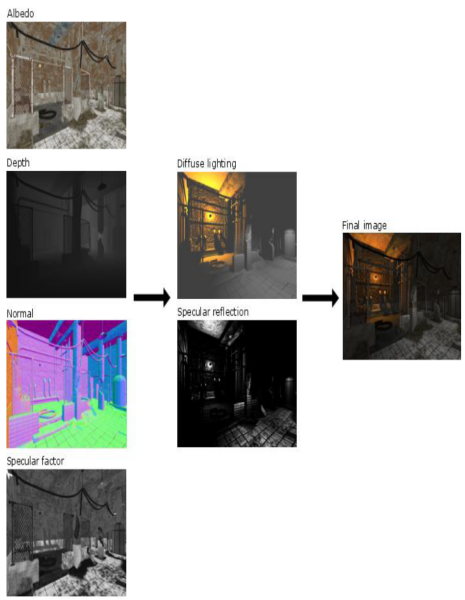


Image from http://www.leadwerks.com/files/Deferred_Rendering_in_Leadwerks_Engine.pdf

WebGL Profiler

- For widespread adoption, WebGL needs world-class developer tools. WebGL Inspection is a good start: http://benvanik.github.com/WebGL-Inspector/.
- However, how do we profile our shaders? We want to mouse over a pixel and see the shader hotspots.
- glsl-unit may be useful: http://code.google.com/p/glsl-unit/w/list

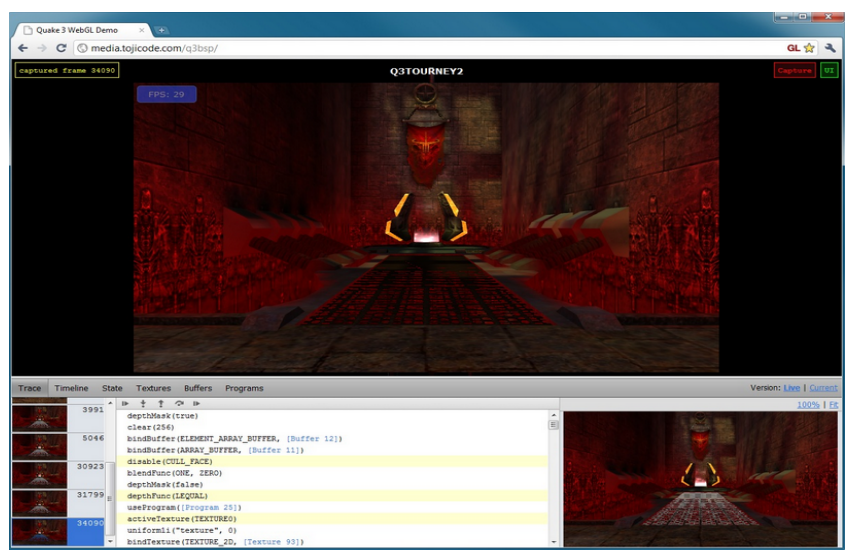


Image from http://benvanik.github.com/WebGL-Inspector/

WebGL Performance

- For widespread adoption of WebGL by C++/OpenGL developers, we need a strong set of benchmarks showing the performance of WebGL compared to OpenGL
- This includes with/without ANGLE, vertex buffers, textures, framebuffers, and compositing
- See OpenGL Insights, *WebGL for OpenGL Developers*

WebGL Secuirty

- Security holes have be found and fixed in WebGL. This includes both Denial Of Service (DoS) attacks due to long running draw calls, and cross-origin content leaking via timed fragment shaders.
- Are there any other vulnerabilities we need to exploit and fix? Are there better ways to fix the original vulnerabilities?
- http://www.khronos.org/webgl/security/

Hybrid Client/Server Rendering

- Where is the future of thin-client visualization? Is it client-side rendering with WebGL? Is it server-side rendering? Is it a hybrid? If so, what rendering is done on the server? What is done on the client? What is passed between?