CSE 167:
Introduction to Computer Graphics
Lecture #18: SSAO, Glow

Jürgen P. Schulze, Ph.D.
University of California, San Diego
Fall Quarter 2012

Announcements

- Thursday, Dec 13: Final project presentations in EBU-3B room 1202, 3-6pm
- Move CSE 190 (3D User Interfaces)
  - Currently scheduled for Mon/Wed 11-12:20
  - Conflicts with CSE 105 and CSE 141
  - Alternatives:
    - Mon/Wed 12:30-1:50pm
    - Mon/Wed 2-3:20pm
    - Tue/Thu 11-12:20pm
    - Tue/Thu 12:30-1:50pm
    - Tue/Thu 2-3:20pm

Lecture Overview

- Screen Space Ambient Occlusion
- Bloom
- Glow

Screen Space Ambient Occlusion

- Screen Space Ambient Occlusion is abbreviated as SSAO
- Rendering technique for approximating ambient occlusion in real time
- Developed by Vladimir Kajalin while working at Crytek
- First use in 2007 PC game Crysis

SSAO component

Ambient Occlusion

- Attempts to approximate global illumination
  - Very crude approximation
- Unlike local methods like Phong shading, ambient occlusion is a global method
  - Illumination at each point is a function of other geometry in the scene
- Appearance achieved by ambient occlusion is similar to the way an object appears on an overcast day
  - Example: arm pit is hit by a lot less light than top of head
- In the industry, ambient occlusion is often referred to as "sky light"

SSAO Demo

- Screen Space Ambient Occlusion (SSAO) in Crysis
  - http://www.youtube.com/watch?v=ifdAILHTcZk

Basic SSAO Algorithm

- Copy frame buffer to texture
- Pixel shader samples depth values around current pixel and tries to compute amount of occlusion
- Occlusion depends on depth difference between sampled point and current point
- Algorithm based on Deferred Shading approach

Ambient occlusion values in red color channel
Source: www.gamerendering.com

Deferred Shading

- Postpones shading calculations for a fragment until its visibility is completely determined
  - Only fragments that really contribute to the image are shaded
- Algorithm:
  - Fill a set of buffers with common data, such as diffuse texture, normals, material properties
  - For the lighting just render the light extents and fetch data from these buffers for the lighting computation
- Advantages:
  - Decouples lighting from geometry
  - Several lights can be applied with a single draw call: more than 1000 light sources can be rendered at 60 fps
- Disadvantages:
  - Consumes more memory, bandwidth and shader instructions than traditional rendering

Particle system with glowing particles. Source: Humus 3D

SSAO With Normals

- First pass: render depth information in a texture's alpha channel and scene normals in the RGB channels
- Use this information to render SSAO in a render target
- It uses the normals and pixel depth to compute the occlusion between current pixel and several samples around that pixel, chosen by sampling texels from depth map around it.

No SSAO

With SSAO

SSAO Discussion

- Advantages:
  - Screen-space algorithm: independent of scene complexity
  - No pre-processing, no memory allocation in RAM
  - Works with dynamic scenes
  - Works in the same way for every pixel
  - No CPU usage: executed completely on GPU
- Disadvantages:
  - Local and view-dependent (dependent on adjacent texel depths)
  - Hard to correctly smooth/blur out noise without interfering with depth discontinuities, such as object edges

References

- Nvidia's documentation:
  - http://developer.download.nvidia.com/SDK/10.5/direct3d/Source/ScreenSpaceAO/doc/ScreenSpaceAO.pdf
- SSAO shader code from Crysis:
  - http://69.163.227.177/forum.php?mod=viewthread&tid=772
- Another implementation:
  - http://www.gamerendering.com/2009/01/14/ssao/
- Deferred Shading Tutorial:
  - http://bat710.univ-lyon1.fr/~jciehl/Public/educ/GAMA/2007/Deferred_Shading_Tutorial_SBGAMES2005.pdf

Lecture Overview

- Screen Space Ambient Occlusion
- Bloom
- Glow

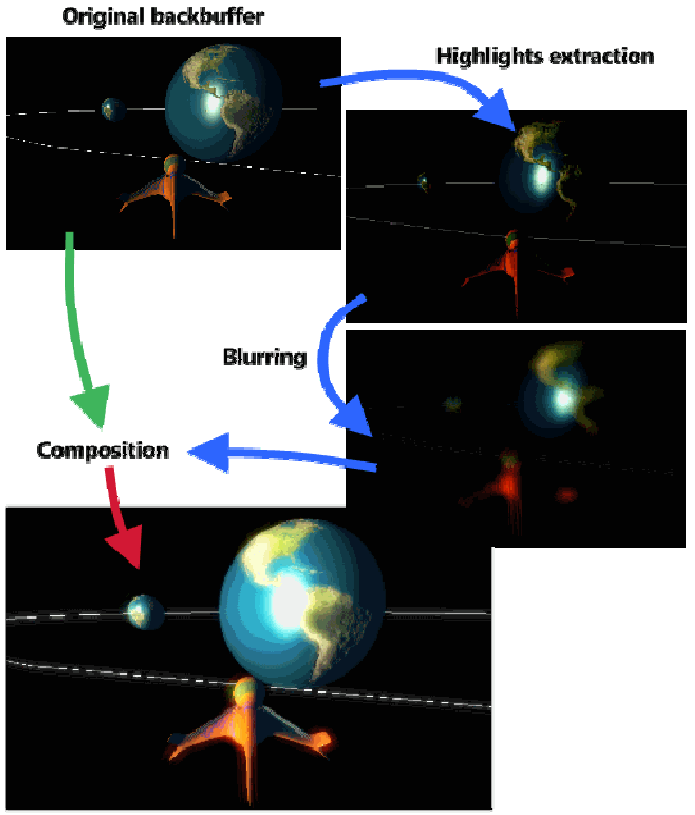
Bloom Effect

Left: no bloom, right: bloom.
Source: http://jmonkeyengine.org

- Bloom gives a scene a look of bright lighting and overexposure

Bloom Shader

- Post-processing filter: applied after scene is rendered normally
- Step 1: Extract all highlights of the rendered scene, superimpose them and make them more intense
  - Operates on back buffer
  - Often done with off-screen buffer smaller than frame buffer
  - Highlights found by thresholding
- Step 2: Blur off-screen buffer, e.g., with Gaussian blurring
- Step 3: Composite off-screen buffer with back buffer



Bloom shader render steps.
Source: http://www.klopfenstein.net

References

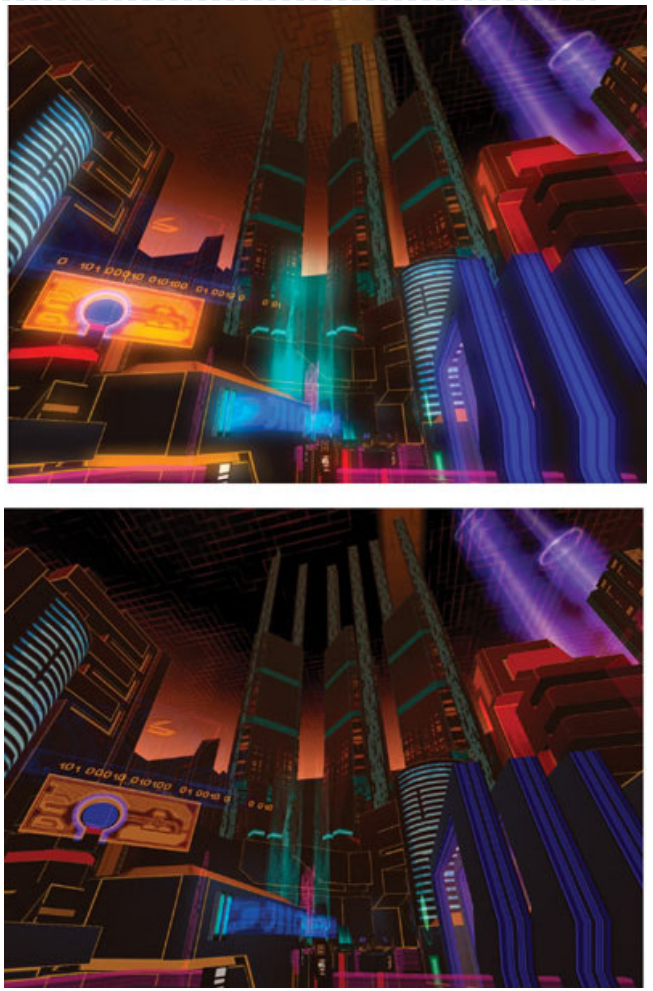
- Bloom Shader
  - http://www.klopfenstein.net/lorenz.aspx/gamecomponents-the-bloom-post-processing-filter
- Bloom and Glow
  - http://jmonkeyengine.org/wiki/doku.php/jme3:advanced:bloom_and_glow

Lecture Overview

- Screen Space Ambient Occlusion
- Bloom
- Glow

Glow Effects

- Glows and halos of light appear everywhere in the world
- They provide powerful visual cues about brightness and atmosphere
- In computer graphics, the intensity of light reaching the eye is limited, so the only way to distinguish intense sources of light is by their surrounding glow and halos
- In everyday life, glows and halos are caused by light scattering in the atmosphere or within our eyes



A cityscape with and without glow.
Source: GPU Gems

Glow and Bloom

- Bloom filter looks for highlights automatically, based on a threshold value
- If you want to have more control over what glows and does not glow, a glow filter is needed
- Glow filter adds an additional step to Bloom filter: instead of thresholding, only the glowing objects are rendered
- Render passes:
  - Render entire scene back buffer
  - Render only glowing objects to a smaller off-screen glow buffer
  - Apply a bloom pixel shader to glow buffer
  - Compose back buffer and glow buffer together

References

- GPU Gems Chapter on Glow
  - http://http.developer.nvidia.com/GPUGems/gpugems_ch21.html
- GLSL Shader for Gaussian Blur
  - http://www.ozone3d.net/tutorials/image_filtering_p2.php

Videos

- ACM Siggraph Asia, 28.11.-1.12.2012 in Singapore (3:18)
  - http://www.youtube.com/watch?v=l81MqEWmR-g
- ACM Siggraph, July 21-25, 2013, Anaheim
  - Student volunteer application deadline: Feb 5, 2013

- Crytek Shows Off the Future of Game Graphics (2:54)
  - http://www.youtube.com/watch?v=dEBuJK-7L5o
- Corning – A Day Made of Glass (5:59)
  - http://www.youtube.com/watch?v=jZkHpNnXLB0