Advanced Computer Graphics
CS 563: *Screen Space Real-time GI Algorithm*
Screen-Space Directional Occlusion

Xin Wang

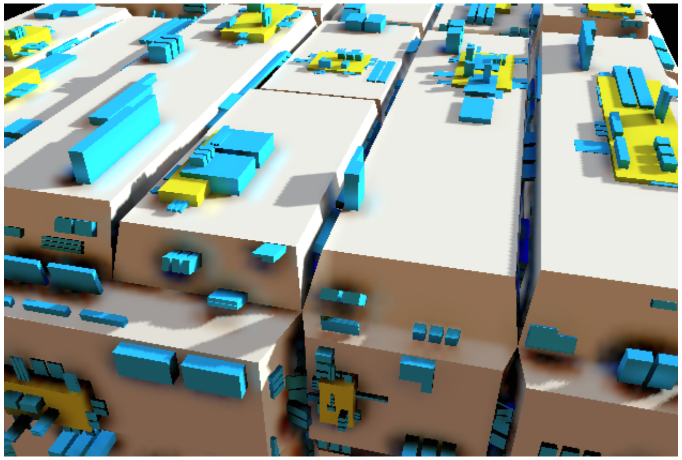
Feb,21,2012

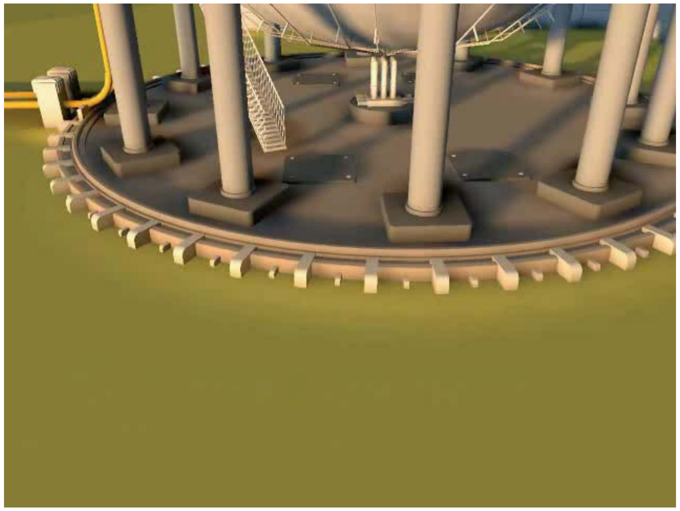
Computer Science Dept.

Worcester Polytechnic Institute (WPI)

Overview

- Real-time Global Illumination
  - Approximation
  - Computed in image space
- Extension of Ambient Occlusion
  - Small-scale geometry
  - Independent of scene complexity
  - Directional Occlusion
  - Indirect Bounces
- Complements classic illumination





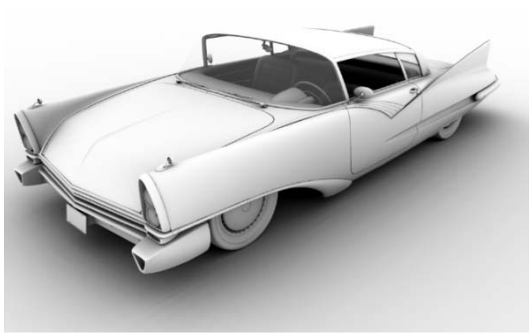
Ambient Occlusion (AO)

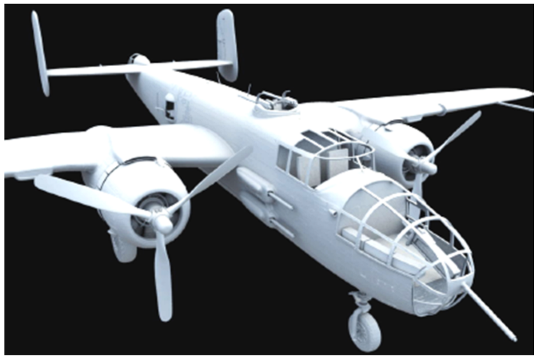
- Global illumination
  - Coarse approximation
  - Efficient (e.g. real-time)
  - Plausible results
  - Perceptually important
    [Langer and Bülthoff 1999]

- Idea:
  - Compute average visibility
  - Multiply with unoccluded illumination at run-time
  - Can be precomputed (vertex / texel)

Previous Work

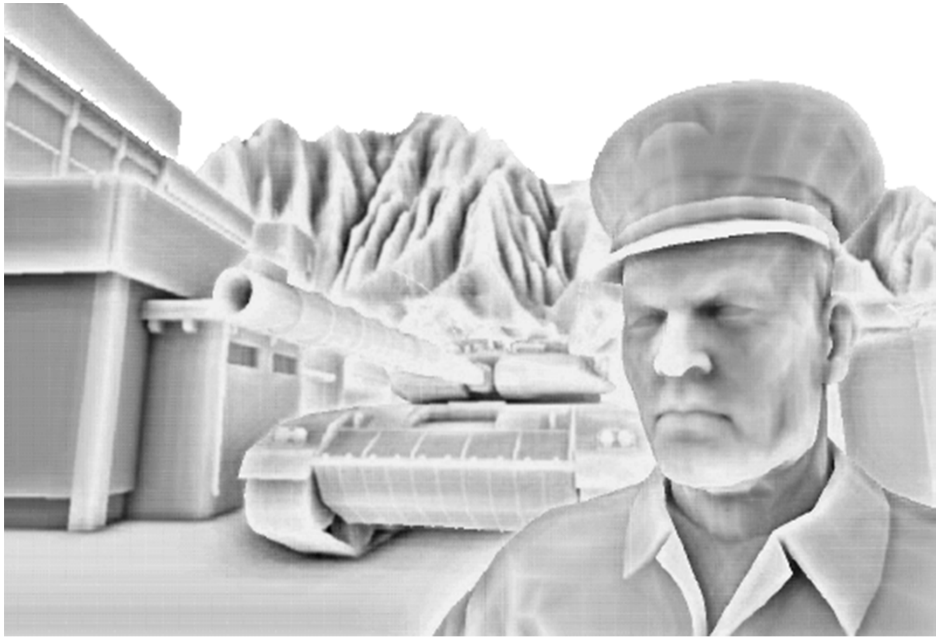
- Ambient illumination model
  [Zhukov et al. 1998]
- Production rendering
  [Landis 2002]
- Moving rigid objects
  [Kontkanen et al. 2005, Malmer et al. 2007]
- Approximate color bleeding
  [Mendez et al. 2006]
- Point-based occlusion/indirections
  [Christensen 2008]
- Screen-Space
  [Mittring 2007, Shanmugam and Arikan 2007, Bavoil et al. 2008, Filion et al. 2008]





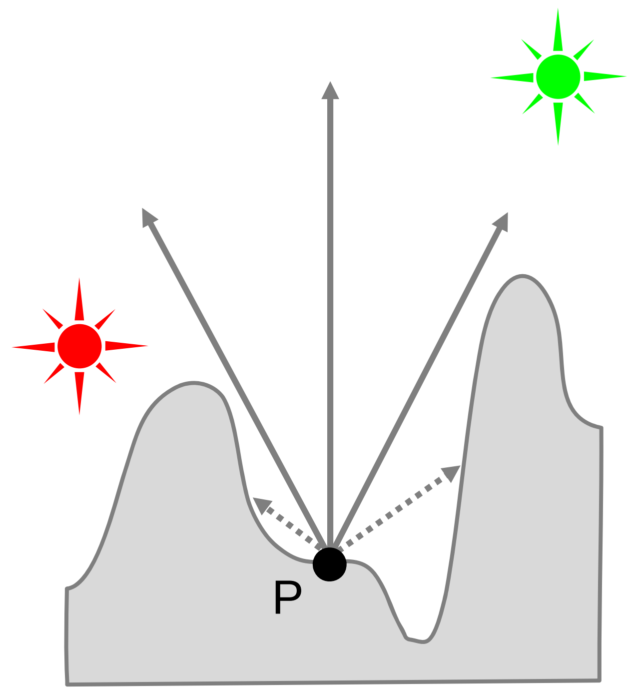
Screen-Space Ambient Occlusion

- SSAO
- Used in current games
- Ambient Occlusion for each framebuffer pixel
  - No precomputation → Dynamic scenes
  - Simple implementation
  - Independent of scene complexity, real-time on recent GPUs



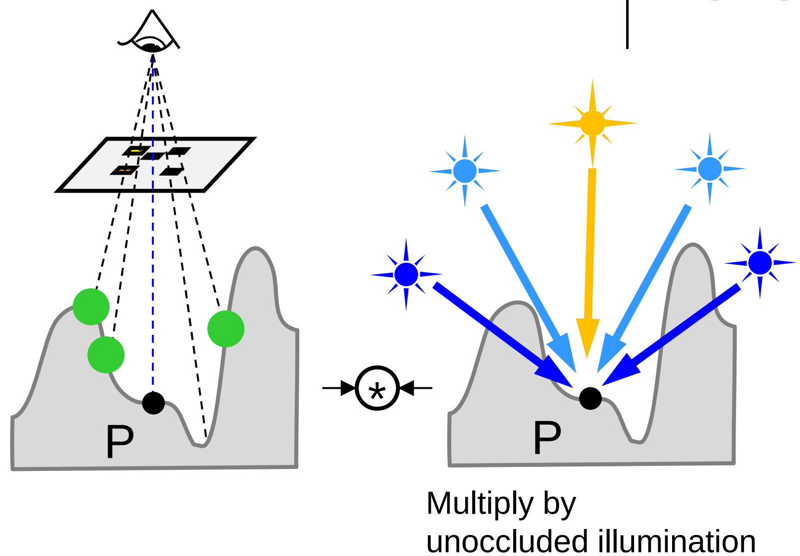
Ambient Occlusion Review

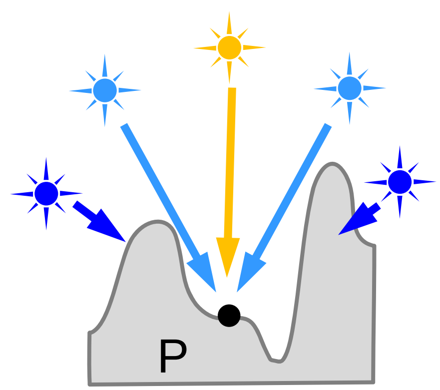
- Directionally-varying light is ignored
- Example:
  Shadow at point P should be green because only the red light is occluded
- AO computes unoccluded illumination first (=yellow) and scales by a single occlusion value (=0.4)
- → shadow at P is brown



Ideal Approach

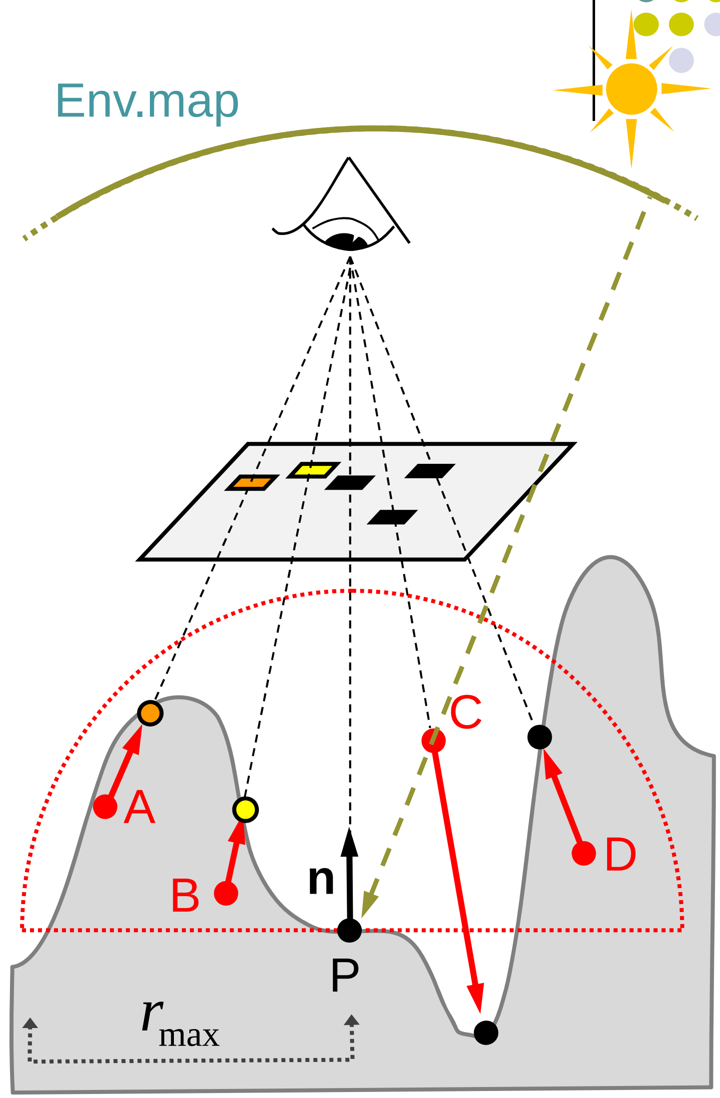
- Traditional SSAO
  [Shanmugam and Arikan 2007]
- Observation:
  - If loop though a number of samples for each pixel, why not just integrating the incoming radiance for each direction ?
- Screen-Space Directional Occlusion (SSDO)
  - Approximate visibility for each sample
  - Accumulate only visible illumination





SSDO Visibility

- For each pixel in framebuffer
  - Compute N samples (A – D) in the upper hemisphere of P with user-defined radius rmax
  - For each sample
    - Backproject sample to image
    - Compute position on surface from z-Buffer
    - If the sample point goes up, it is treated as occluded (A,B,D)
    - If the sample goes down, P is illuminated from this direction (C) (blurred envmap, filter $\approx 2\pi / N$)



Equations

- SSDO is a improvement of SSAO
- Irradiance of each fragment

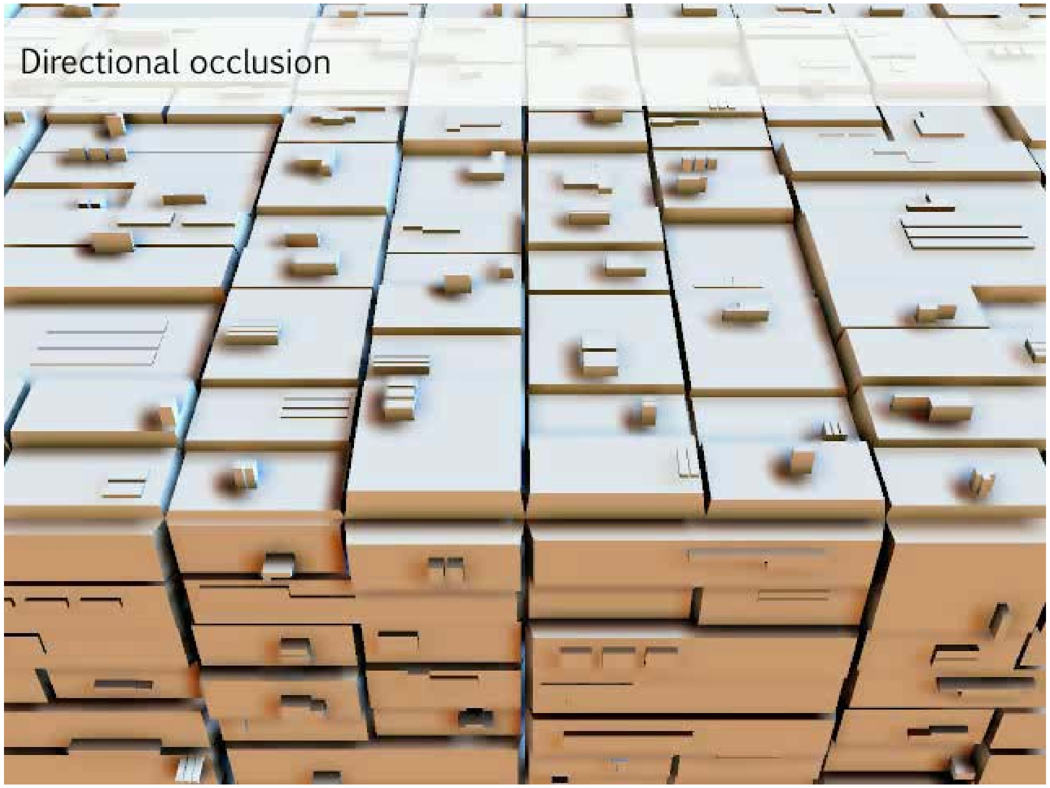
$$L_{dir}(P) = \sum_{i=1}^{N} \frac{\rho}{\pi} L_{in}(\omega_i) V(\omega_i) \cos\theta_i \Delta\omega$$

- Indirect Bounces

$$L_{ind}(P) = \sum_{i=1}^{N} \frac{\rho}{\pi} L_{pixel}(1 - V(\omega_i)) \frac{A_s \cos\theta_{si} \cos\theta_{ri}}{d_i^2}$$

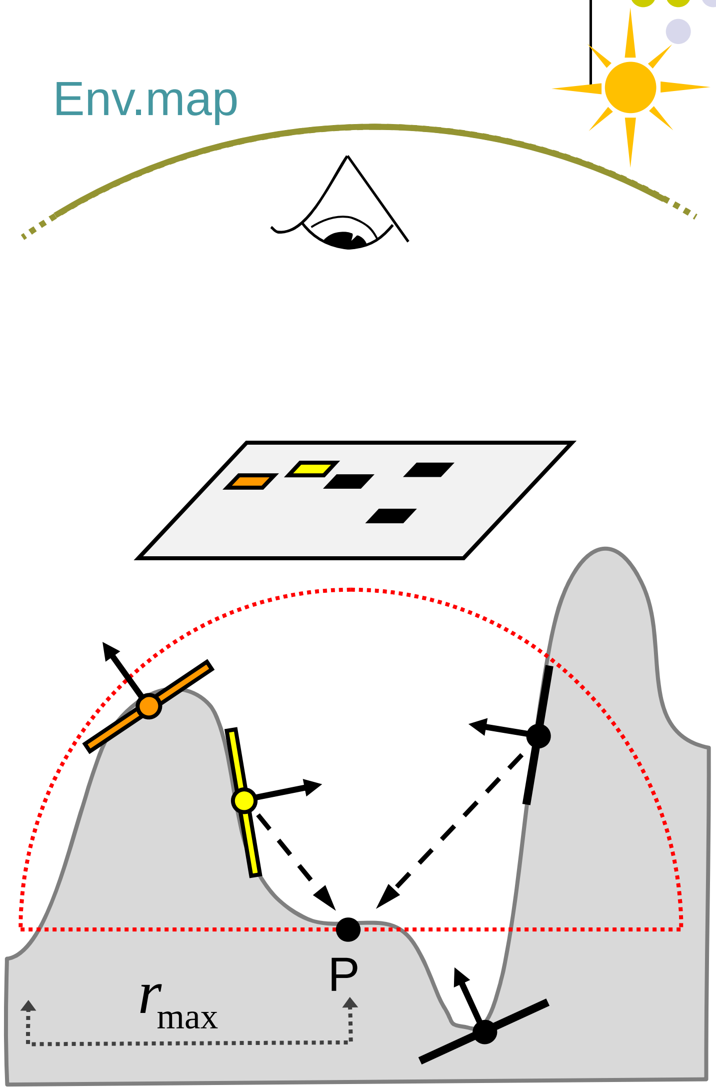
Directional Occlusion

- SSDO shadows
  - Oriented
  - Colored
- SSAO shadows
  - Not oriented
  - Grey
- Similar computations
  - SSDO overhead: 3.6%



Indirect Illumination

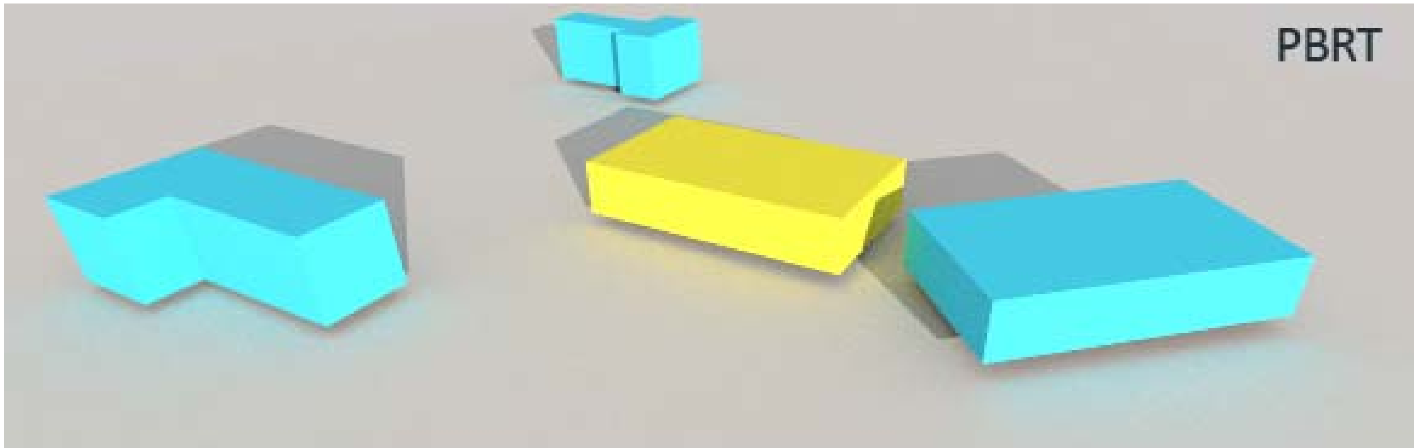
- Each sample is a small area light
  - Oriented around pixel normal
  - Radiance = direct light
- For each sample
  - Compute form factor to P and accumulate contributions
- Results in one indirect bounce of light for nearby geometry

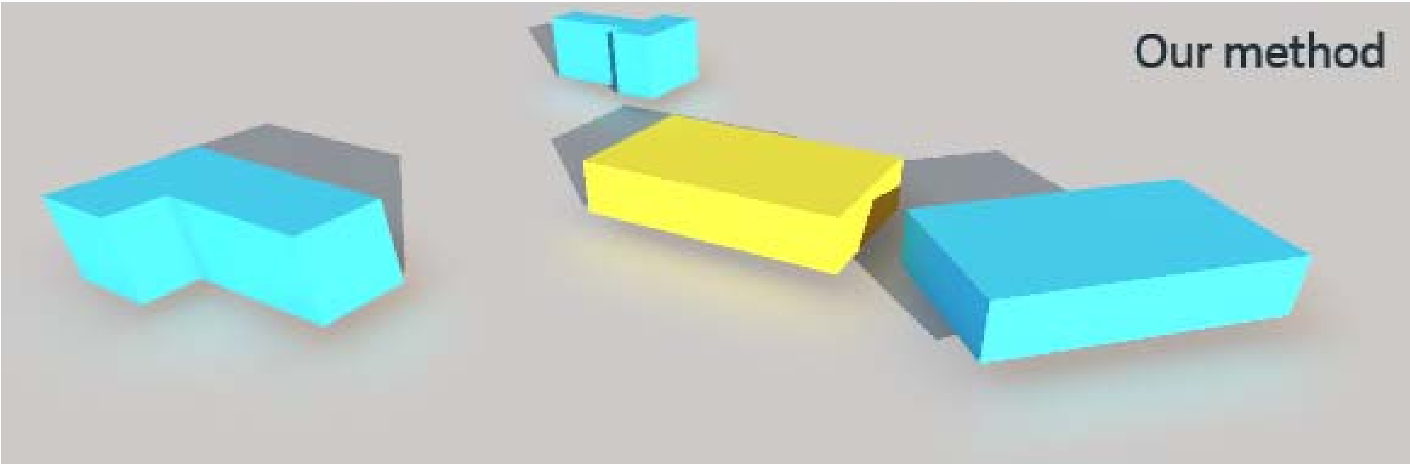


Indirect Illumination

- Direct light using e.g.
  - Environment map (Sky)
  - Point light with shadow map (Sun)
- Temporal coherence
- Overhead: ~30%

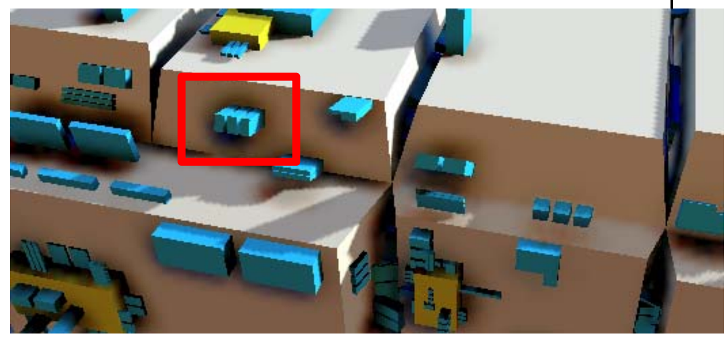
Comparison with PBRT

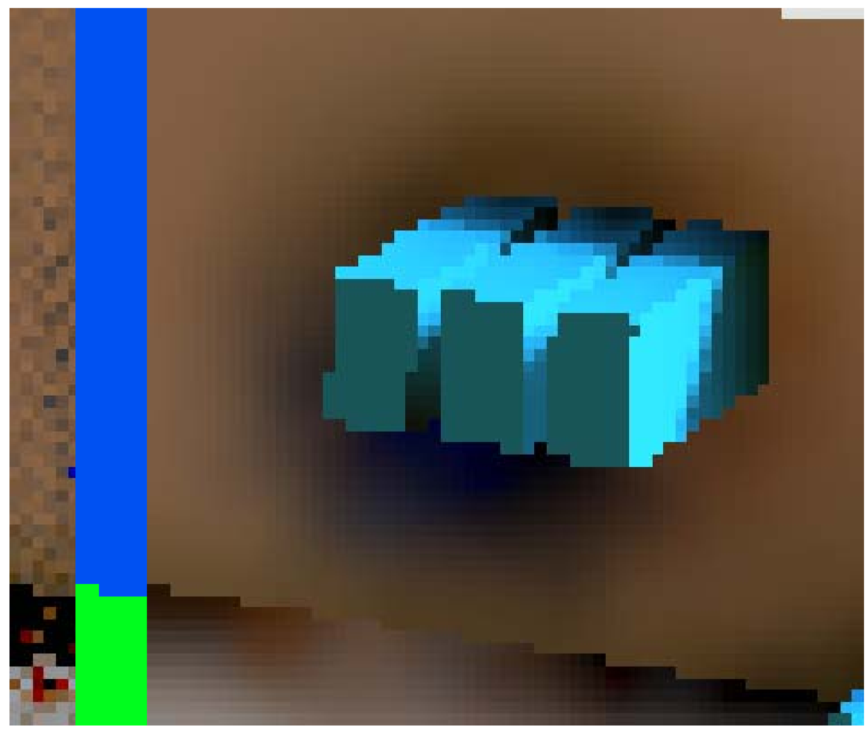




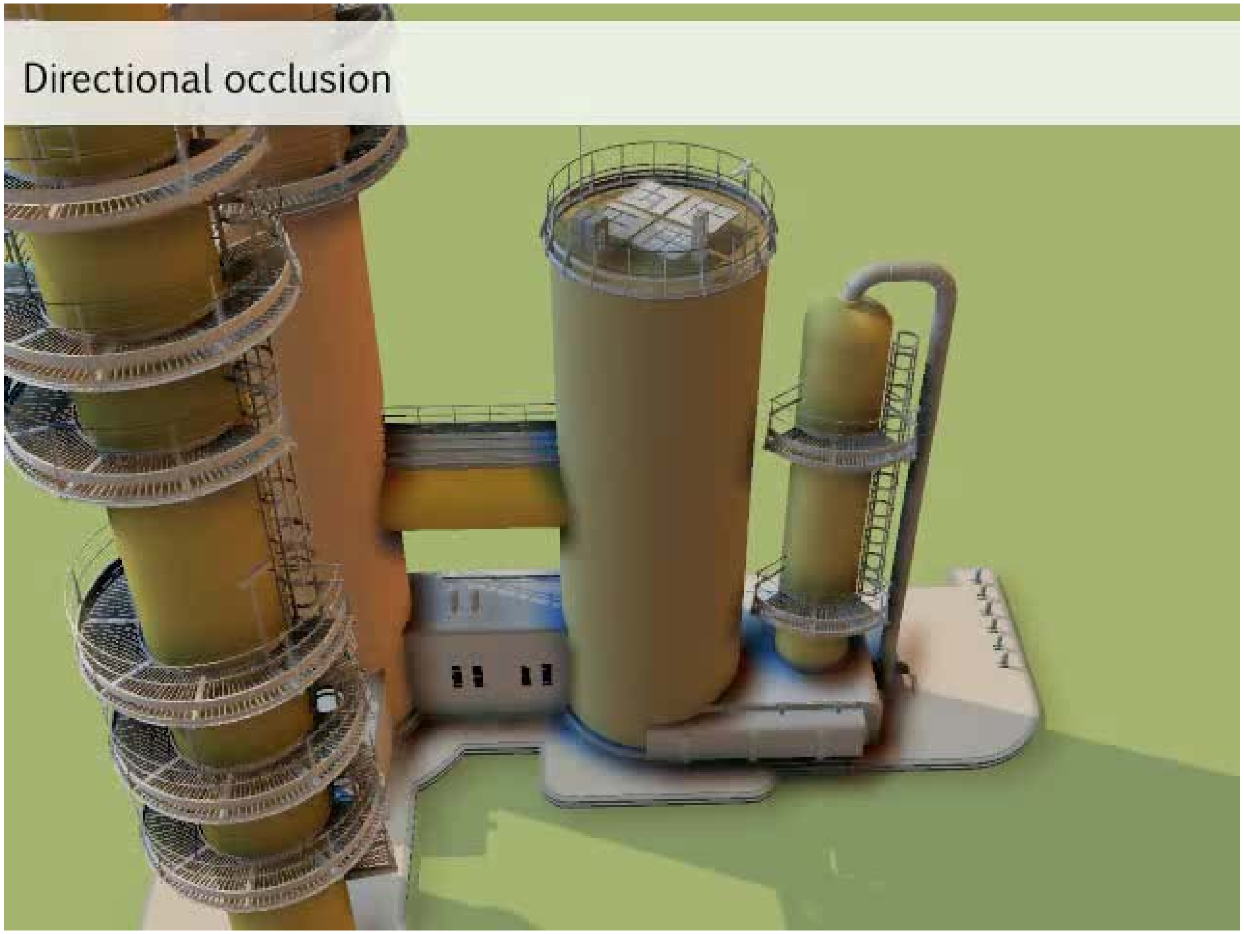
Speed-up: Interleaved Sampling

- Reduce the computation for each pixel
  [Keller and Heidrich 2001, Segovia et al. 2006]
  - Not all N samples are used
- Instead:
  - Use blocks of MxM pixels
  - Each pixel uses only N/(MxM) samples
  - Apply geometry-aware blur
    [Laine et al. 2007]





Exmple Scene



GeForce 8800 GTX

Resolution:	1024 x 768
Polys:	~300k
SSDO:	55 – 80 fps
Bounce:	40 – 65 fps

3D Model from
Dosch Design

Non-polygonal Scenes and Animations

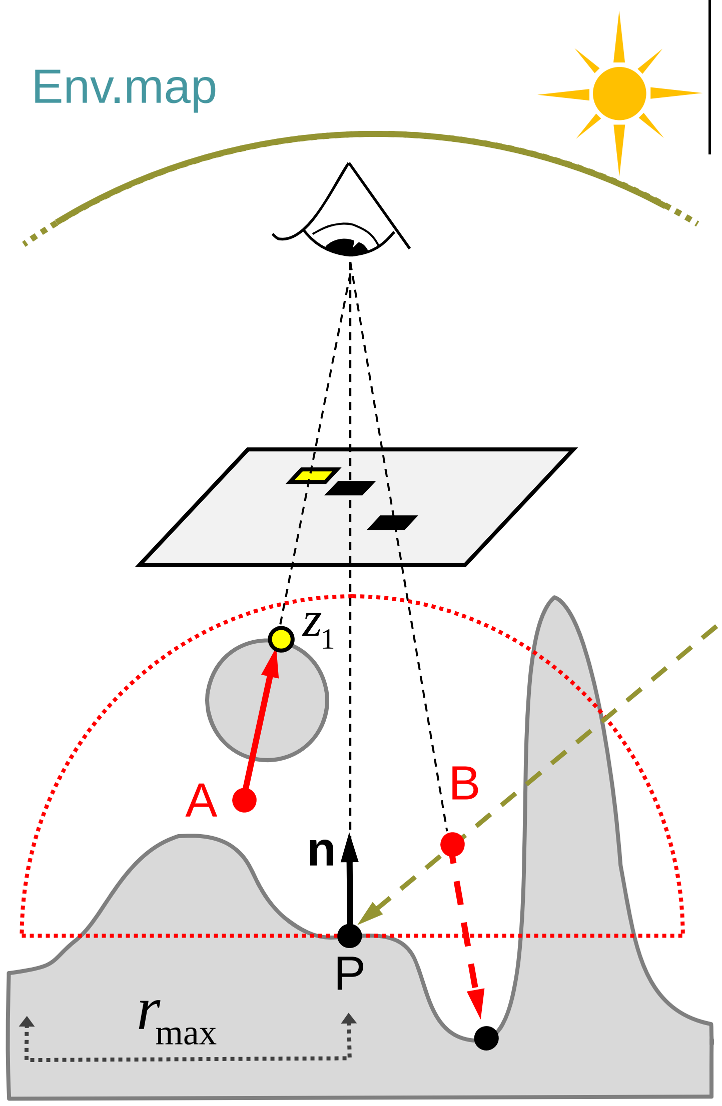
GeForce 8800 GTX

Resolution: 1024 x 768
Volume: 256^3

Volume Data from UTCT

Limitation I

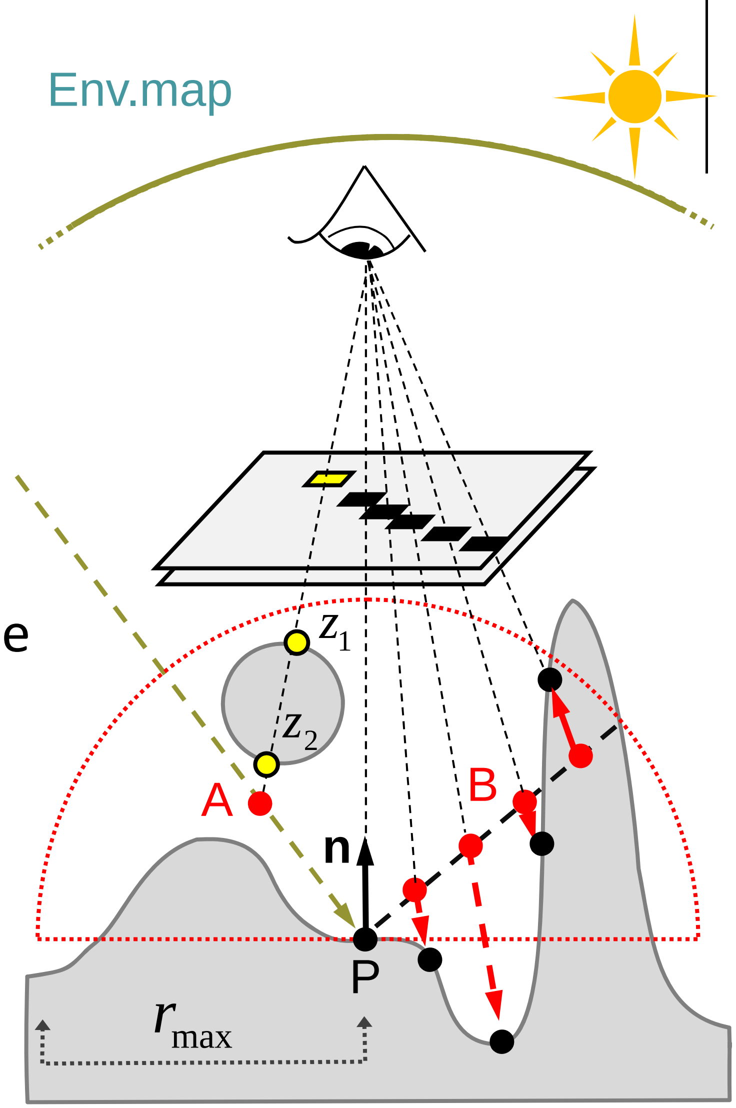
- Visibility is an approximation
- Only nearby geometry:
  No occluders outside sphere
- Wrong classification
  - Sample A:
    treated as an occluder
  - Sample B:
    treated as visible



Solutions

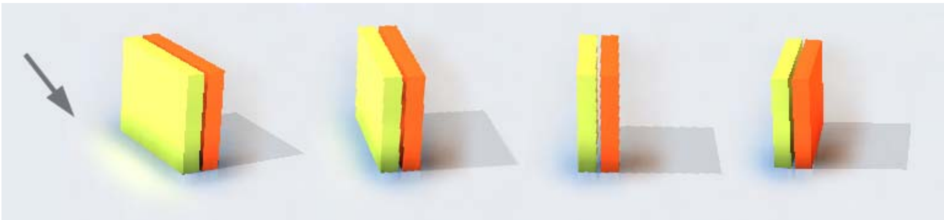
- Sample A:
  - Depth Peeling
    [Lischinski et al. 1998, Everitt 2001]

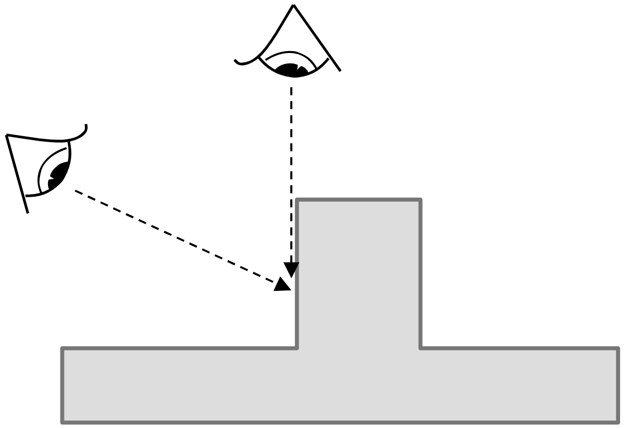
- Sample B:
  - Use multiple samples on a line



Limitations II

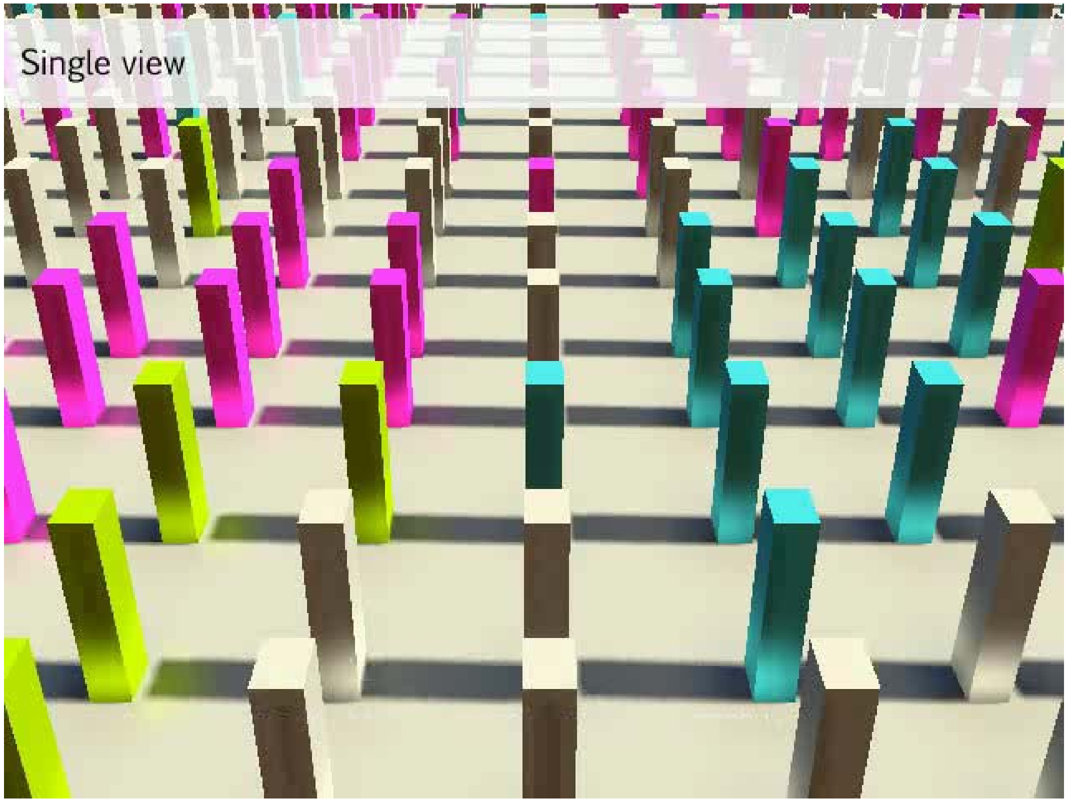
- Only visible senders can contribute to indirect illumination
  - Especially grazing angles
  - Fade in/out of indirect light
  - Fortunately no abrupt changes visible
- Solution:
  Use multiple cameras





Solution – Multiple Cameras

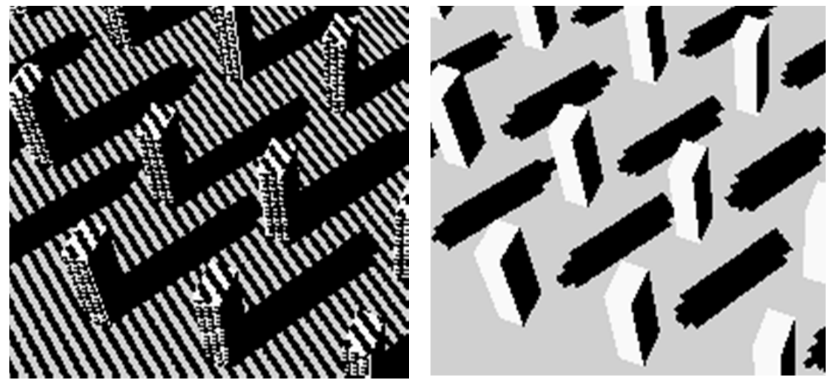
- Color Bleeding from senders viewed from a grazing angle
- Four cameras
  - Placed manually
  - Lower resolution
- Overhead:
  - 50%-160%
  - 1x fill-rate
  - 4x transform-rate



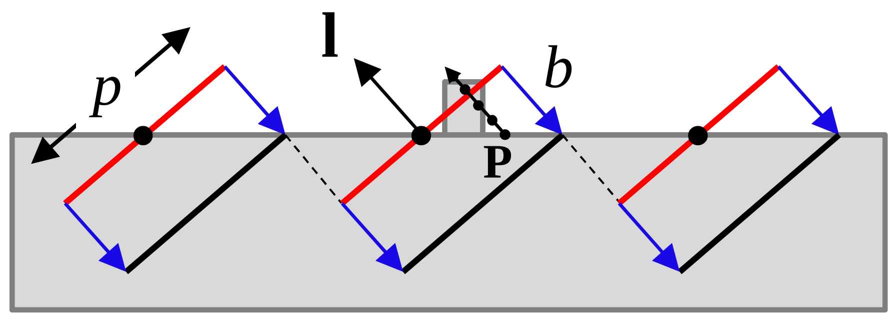
Extend Object-based Techniques

- Idea:
  Add SSDO on top of existing Global Illumination techniques

- Example:
  Shadow Mapping
  - Typical problem:
    Details and contact shadows are lost due to the depth bias
  - Can be corrected in image space ($r_{\max} \approx b$)



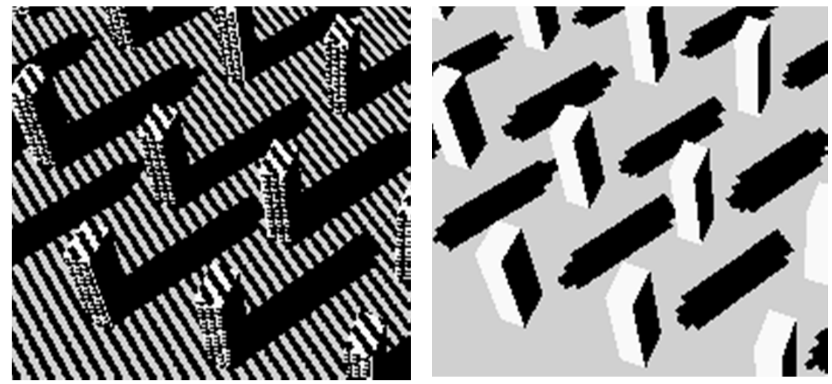
Shadow Mapping, 218 fps



Extend Object-based Techniques

- Idea:
  Add SSDO on top of existing Global Illumination techniques

- Example:
  Shadow Mapping
  - Typical problem:
    Details and contact shadows are lost due to the depth bias
  - Can be corrected in image space

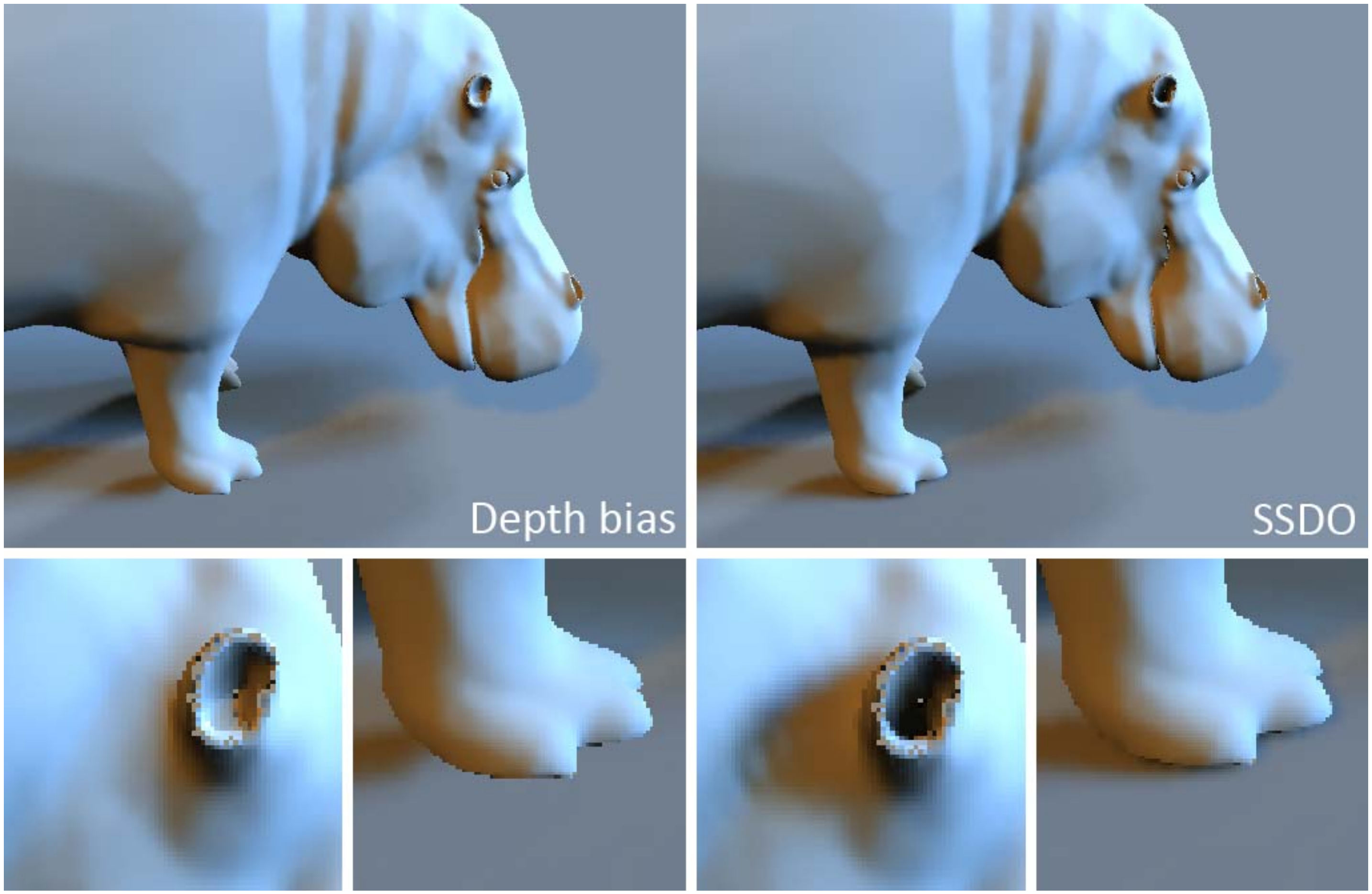


Shadow Mapping, 218 fps

r_{max} correction,
103 fps

Peeled correction,
two depth layers,
73 fps

Application: Natural Illumination



Overall Summary

- SSDO: Screen-Space Directional Occlusion
- Approximation of Global Illumination in image space
- Correctly oriented, colored shadows
- One bounce of indirect light possible
- Interactive to real-time frame rates, independent of scene complexity

References

- *Approximating Dynamic Global Illumination in Image Space. Thorsten Grosch, ACM I3D '09, Boston, MA*
- *http://www.cnblogs.com/atyuwen/archive/2010/03/31/ssdo.html*