Fancy Lighting

Robert Leslie and Max Lisaius

Project

Screen space ambient occlusion (SSAO)

Volumetric Lighting

Bounding Volume Hierarchy (BVH)

SSAO

What is It?

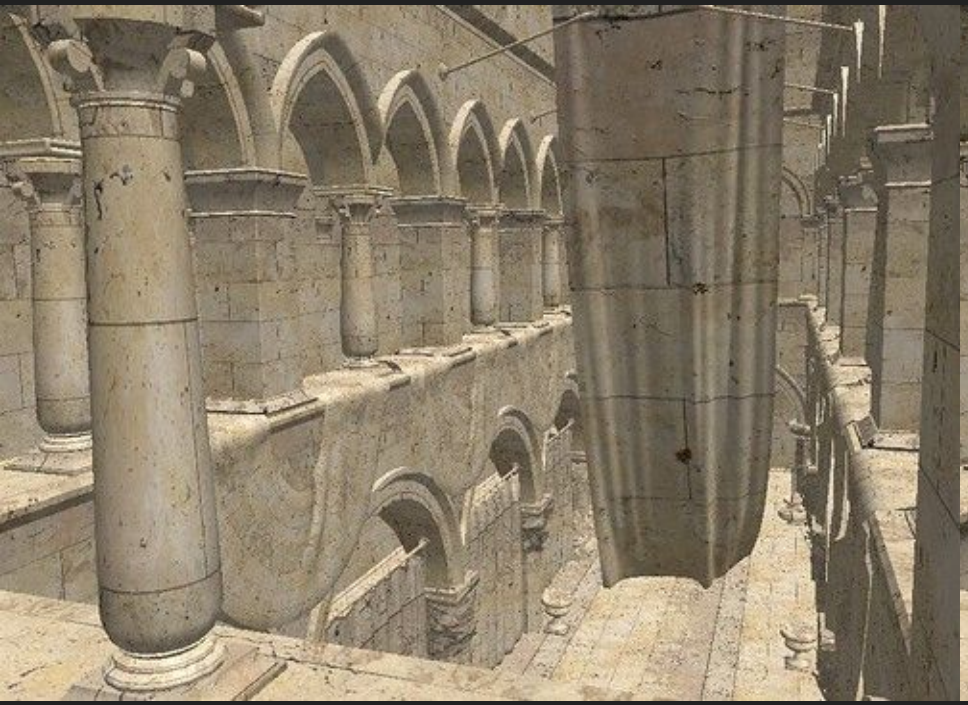
- When light travels through a scene, rebounding off surfaces, there are some places that have a smaller chance of getting hit with light: corners, tight gaps between objects, creases, etc. This results in those areas being darker than their surroundings.

- The light that we are referring to here is the Ambient light, which provides some light to everything in the scene.

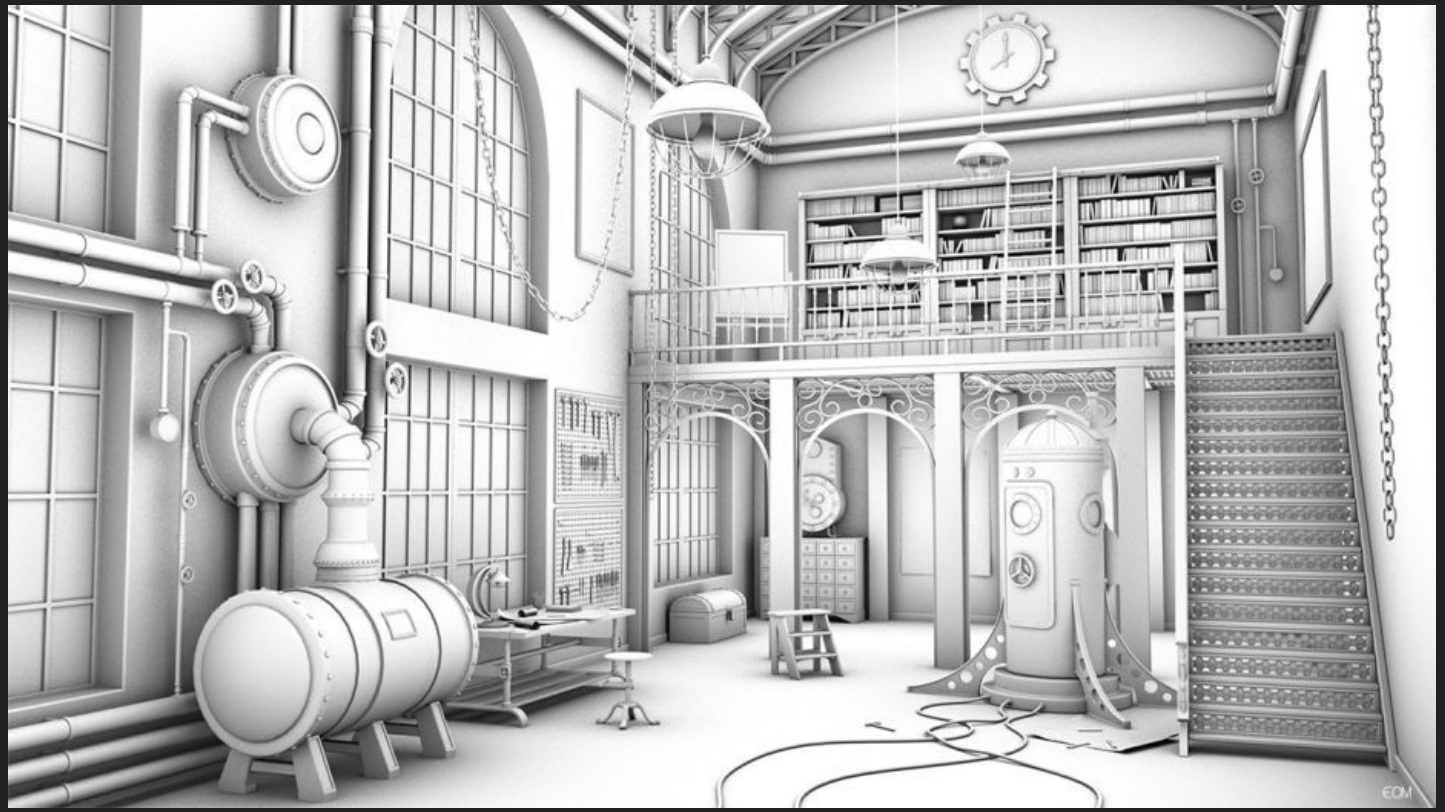
- These darker areas are from ambient occlusion (AO), and the usual method to simulate this darkening of certain areas of the scene involves testing, for each surface, how much it is "occluded" or "blocked from light" by other surfaces.

- SSAO is the technique of approximating AO in real time. It was used for the first time in the 2007 video game, Crysis by Crytek. SSAO is different from normal AO because it utilizes a depth buffer at each pixel instead of determining the light from other surfaces.
- https://www.youtube.com/watch?v=VWiSex0NhlI&t=14

SSAO



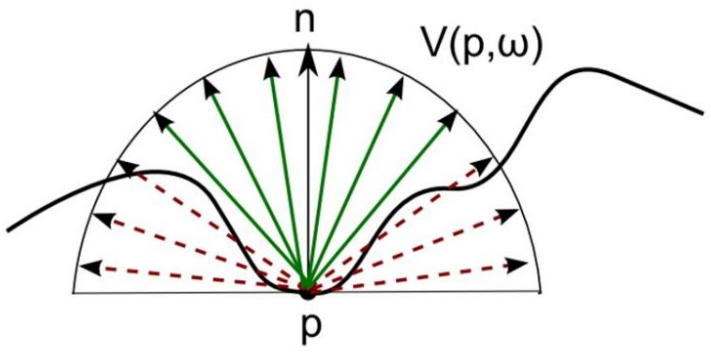
SSAO



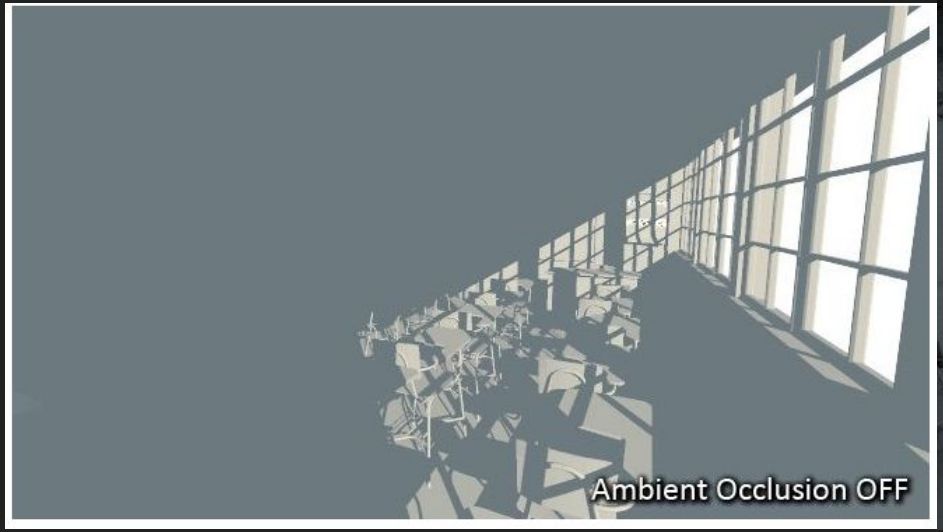
AO Implementation

The general idea is to construct a hemisphere of points around a point p on a surface, and send out rays in those directions and see if they intersect with another object. Rays that reach the background increase the brightness at that point.

This is a lot of intersections, so generally a random or patterned subset is generated



$$AO(\mathbf{p},\mathbf{n}) = \frac{1}{\pi}\int_{\Omega} V(\mathbf{p},\omega)\,\mathbf{n}\cdot\omega\,d\omega,$$


Ambient Occlusion OFF


Ambient Occlusion ON

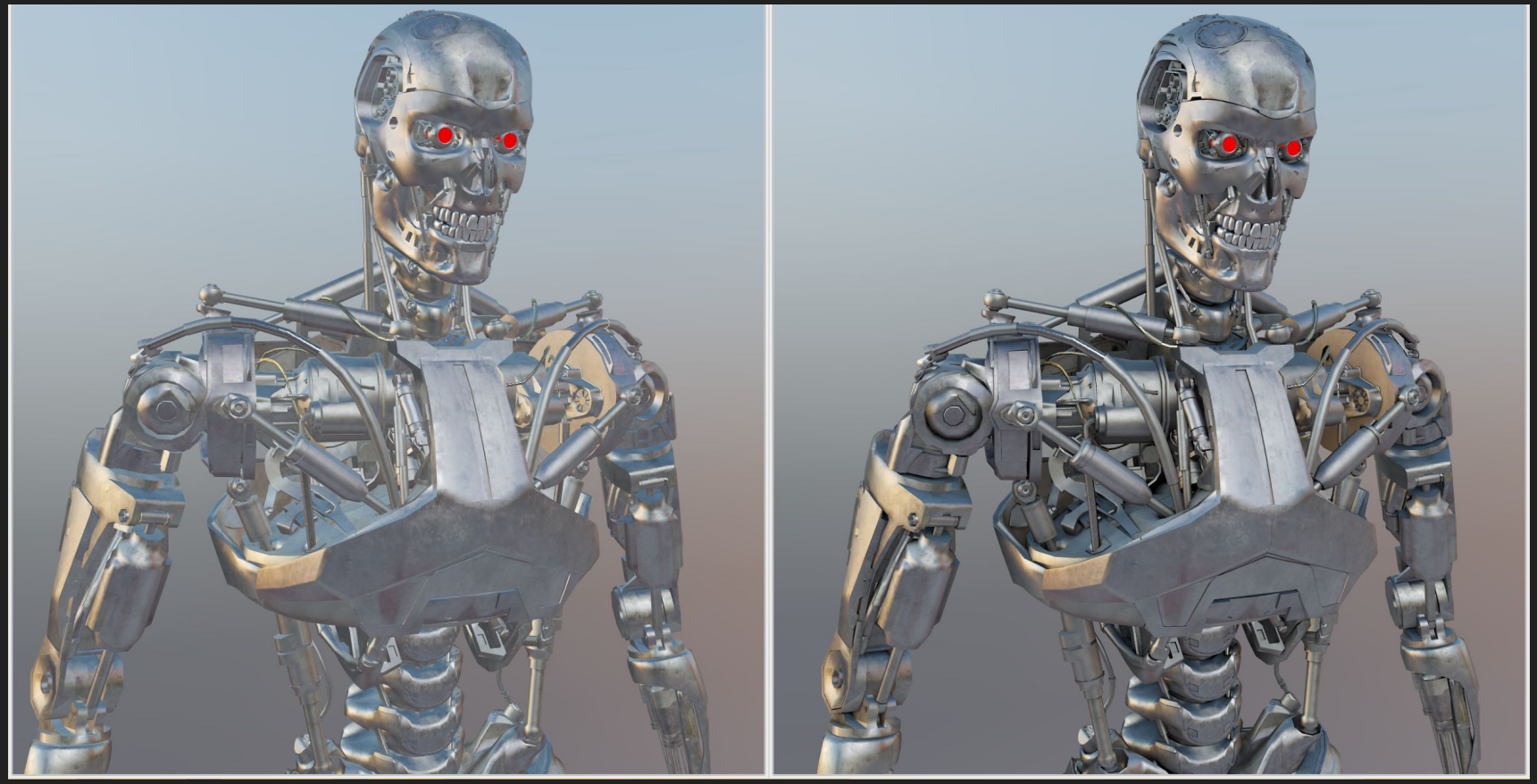
SSAO Implementation

How is it done?

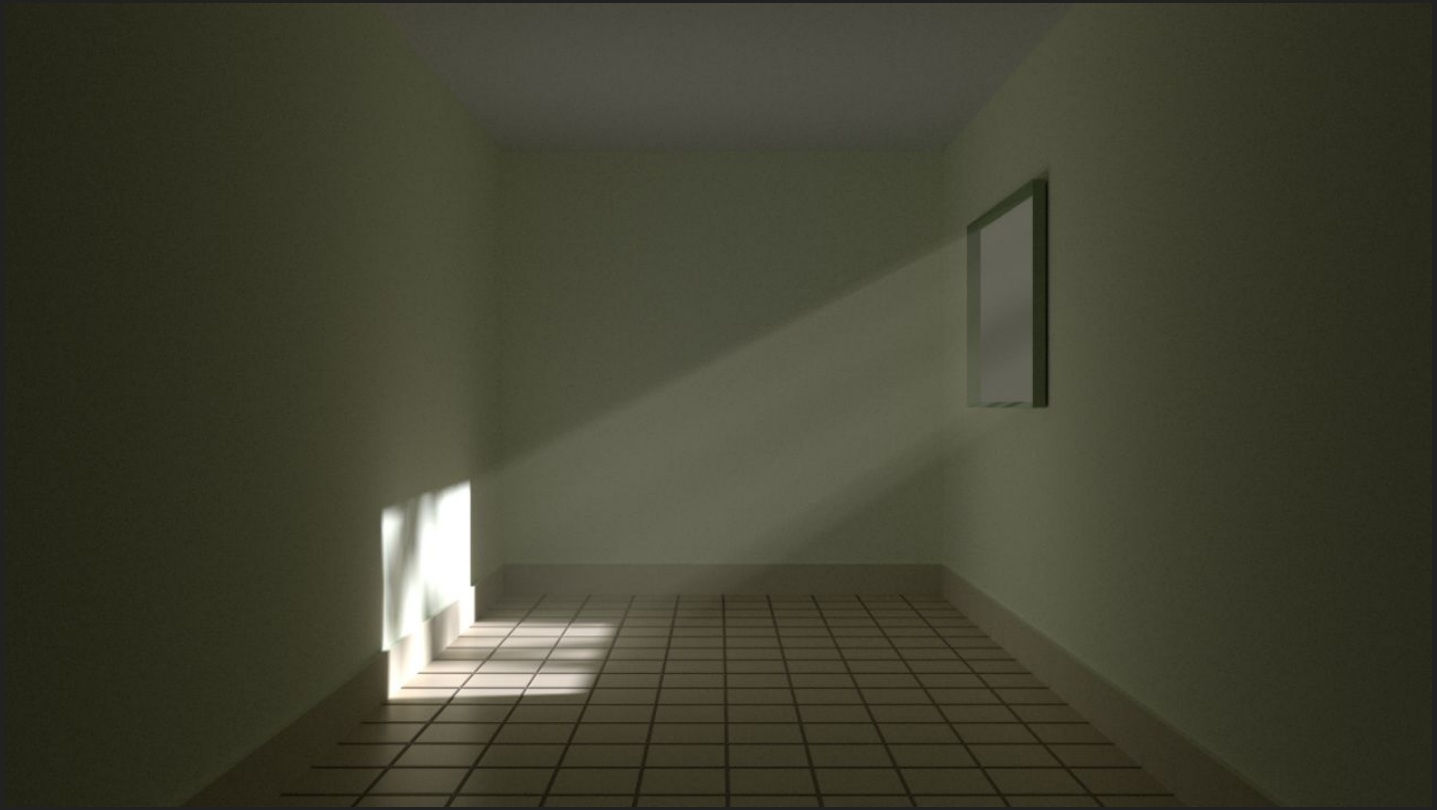
- The original implementation by Crytek had a depth buffer as input and worked roughly like this:
  - for each pixel in the depth buffer,
    - sample a few points in 3D around it
    - project them back to screen space and compare the depth of the sample and the depth at that position in the depth buffer to determine if the sample is in front (no occlusion) or behind a surface (it hits an occluding object)

- This can also be done by storing position and normals as a buffer rather than later retrieving the positions from the depth map.

SSAO



Volumetric Lighting



Volumetric Lighting

What is It?

- Also known as Crepuscular rays or "God rays" that are sunbeams that can form when the sun is below the horizon, during twilight hours
- Allows light to be visible in beams of light shining through the rendered scene
- Allows for lighting through smoke, fog, or other haziness/air particles

Volumetric Lighting

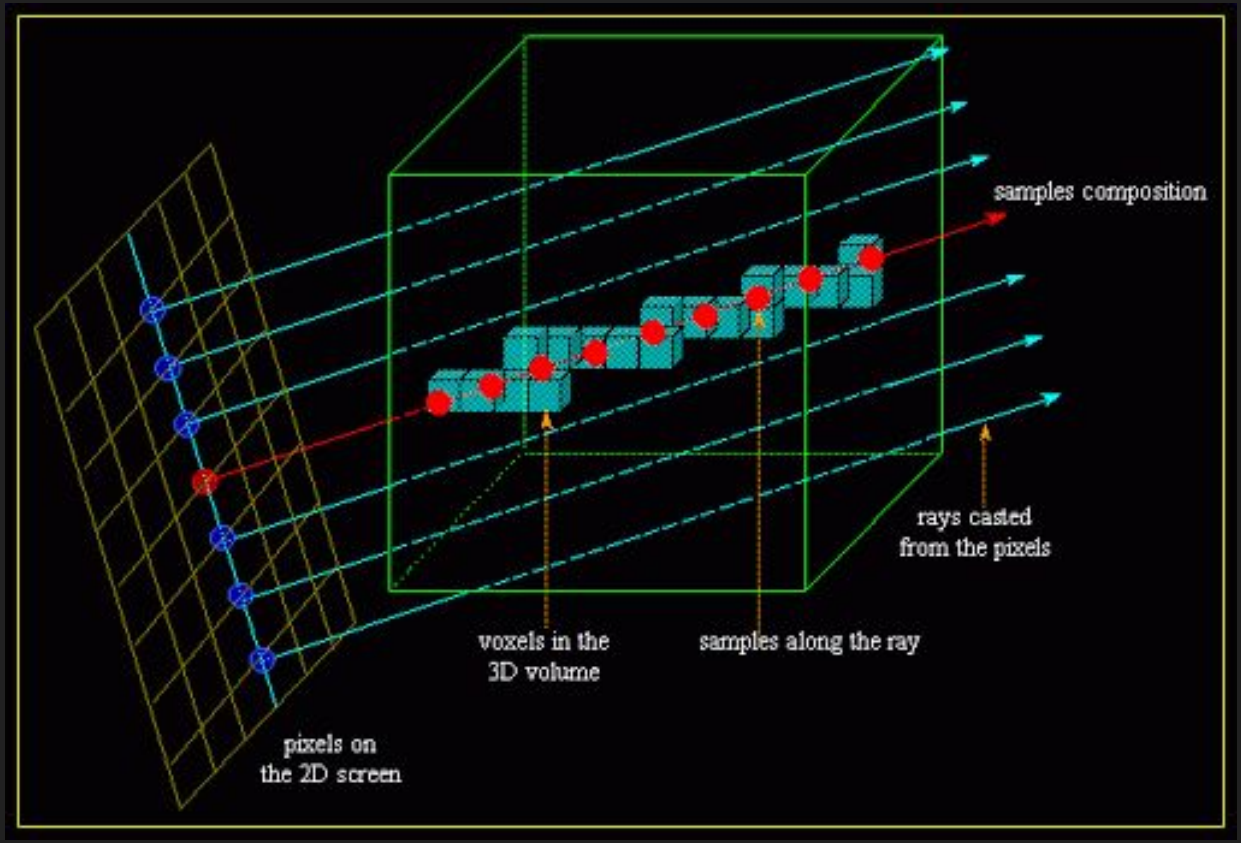
How is it done?

- The idea is to “ray march” from the world position of the current pixel to the camera position, and for each step check if the current position can be seen by the light or not, using the information from the shadow map.

- For each step the light scattered in the camera direction is accumulated using the Henyey-Greenstein phase function.

Volumetric Lighting

Ray Marching



Volumetric Lighting

1. The Henyey-Greenstein phase function.

Henyey and Greenstein (1941) introduced a function which, by the variation of one parameter, $-1 \leq g \leq 1$, ranges from backscattering through isotropic scattering to forward scattering. The function is

$$p(\theta) = \frac{1}{4\pi} \frac{1 - g^2}{[1 + g^2 - 2g \cos(\theta)]^{3/2}} , \tag{1}$$

Volumetric Lighting

Ray marching can be expensive. We can decrease its accuracy and instead add blur and noise.

Expensive version:

Volumetric Lighting

Ray marching can be expensive. We can decrease its accuracy and instead add blur and noise.

Cheap version:

Volumetric Lighting

Ray marching can be expensive. We can decrease its accuracy and instead add blur and noise.

Cheap + Noise:

Volumetric Lighting

Ray marching can be expensive. We can decrease its accuracy and instead add blur and noise.

Cheap + Noise + Blur:

Questions?