Stochastic-Depth Ambient Occlusion

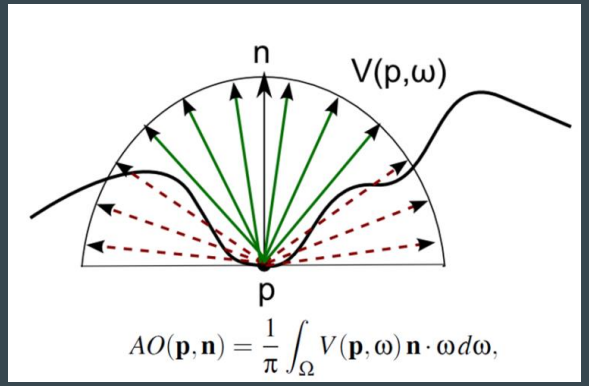
Authors: Jop Vermeer, Leonardo Scandolo, Elmar Eisemann

Presenter: 20160249 박승건

Review

Ambient Occlusion (AO)

- Used to calculate how exposed each point is to ambient lighting
- With AO, we can have a sense of the depths of the surfaces
  - Exposed, outer surface: brighter
  - Deeper, inner surface: darker



https://nbertoa.files.wordpress.com/2017/02/ao_integral.png?w=756

https://docs.unity3d.com/kr/2018.3/Manual/PostProcessing-AmbientOcclusion.html

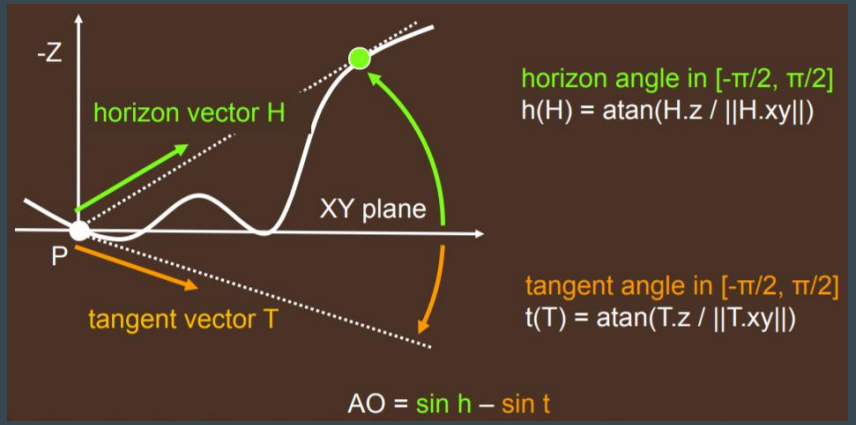
Previous Works

Previous Works

- Image-space Horizon-based Ambient Occlusion
  - Bavoil et al. [2008]
  - Keyword: **AO pop-in**
- Screen-space Ambient Occlusion using A-buffer Techniques
  - Bauer et al. [2013]
  - Keyword: **Multiple fragment layers**
- Stochastic Transparency
  - Enderton et al. [2010]

Previous Work #1: Image-space Horizon-based Ambient Occlusion

- HBAO is one of the most widely used real-time ambient occlusion technique
- Raymarch in several directions and keep updating the maximum elevation (of depth value)
- Then compare the elevation angle with an angle between tangent and horizon



https://developer.download.nvidia.com/presentations/2008/SIGGRAPH/HBAO_SIG08b.pdf

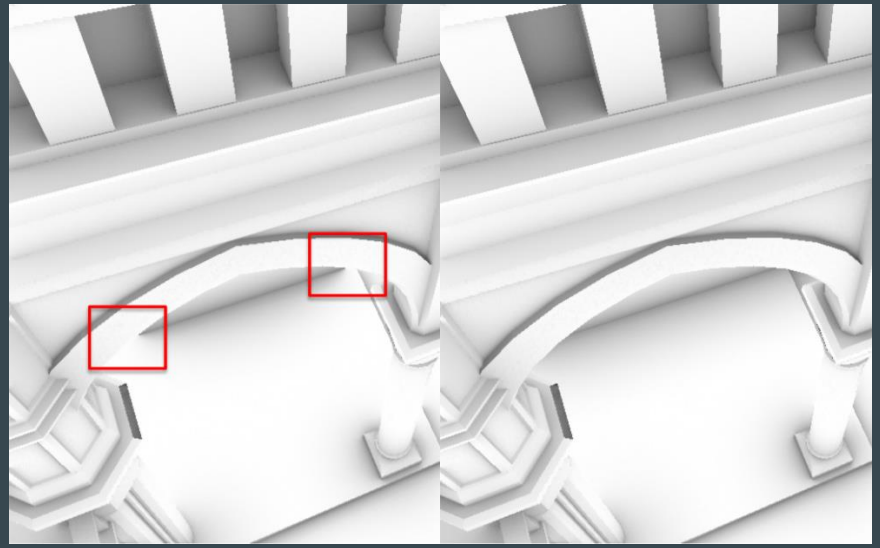
Previous Work #1: Image-space Horizon-based Ambient Occlusion

- Still, HBAO relies on depth buffer, which means AO values depend on the viewpoint
- AO pop-in: as the camera moves, occluded objects come into view, inducing sudden changes in occlusion shading

https://www.youtube.com/watch?v=vvnG3-JwLl4&t=285s

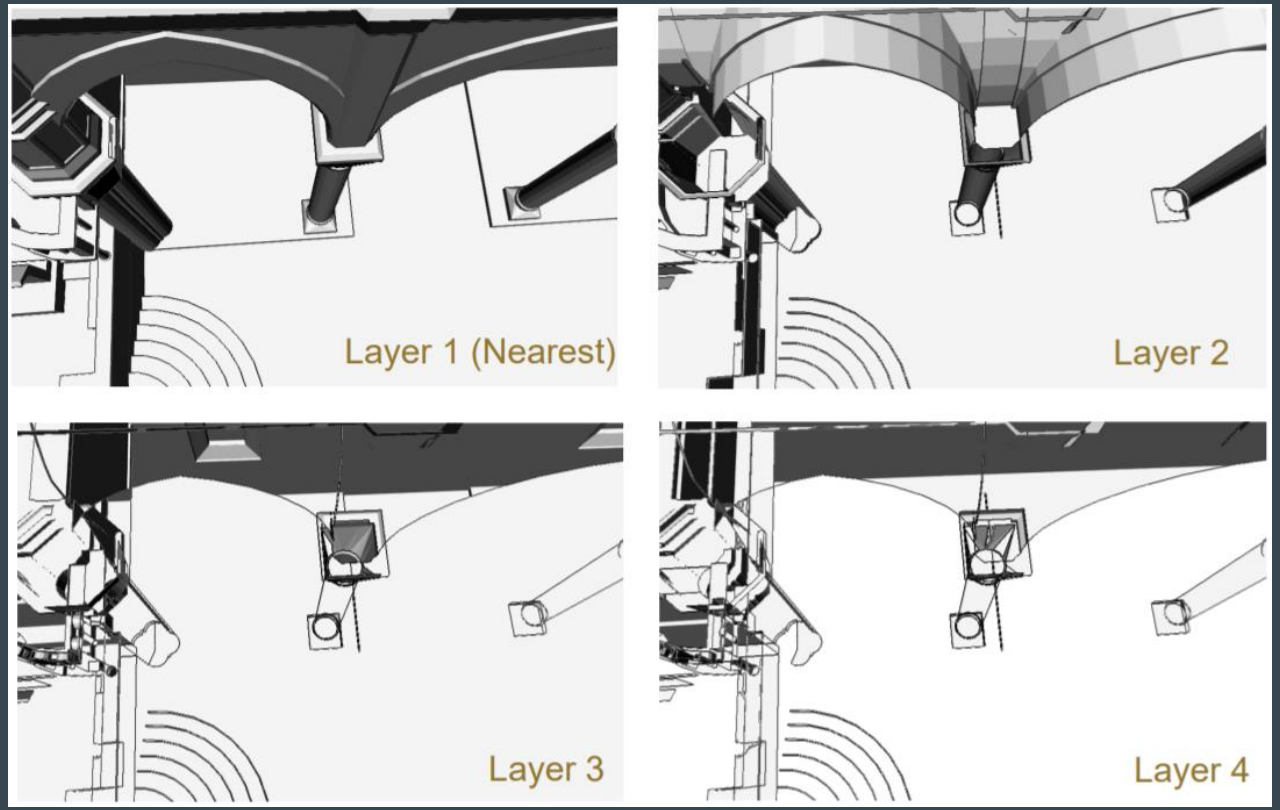
Previous Work #2: Screen-space Ambient Occlusion using A-buffer Techniques

- Screen-space AO using depth buffer is only be applied to the first visible layer of geometry, yielding artifacts
- A-buffer
  - Introduced in 1984
  - Stores per-pixel lists of fragment data
  - Not only screen-space-revealed fragments' data, but also occluded fragments' data
- A-buffer allows accessing multiple geometry layers to produce more plausible AO for both of opaque & translucent geometry



https://developer.download.nvidia.com/presentations/2009/SIGGRAPH/Bavoil_MultiLayerDualResolutionSSAO.pdf

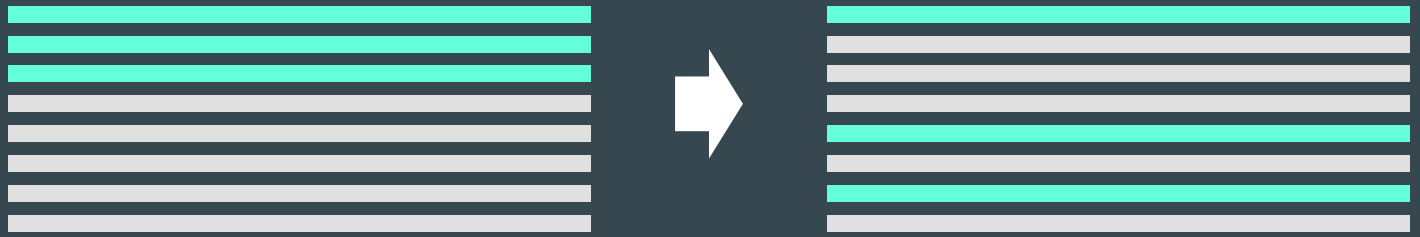
Previous Work #2: Screen-space Ambient Occlusion using A-buffer Techniques



Paper Introduction

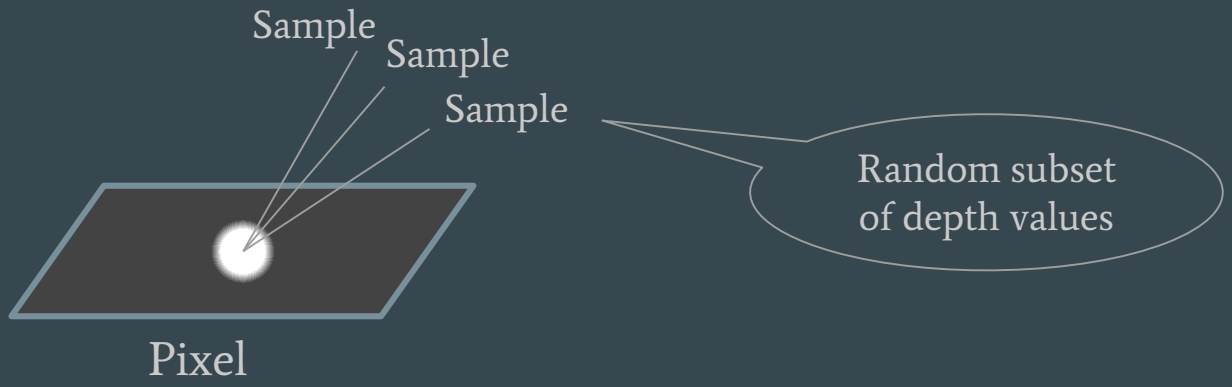
Main Idea

- In previous work, using the first *n* visible layers either requires multiple render passes, or has large memory requirements
- Inspiration from ‘Stochastic Transparency’
  - Enderton et al. [2010]
  - Instead of an ordered list of the first visible layers, use a random subset of the scene layers
  - Generate ‘**stochastic depth map**’ to do so



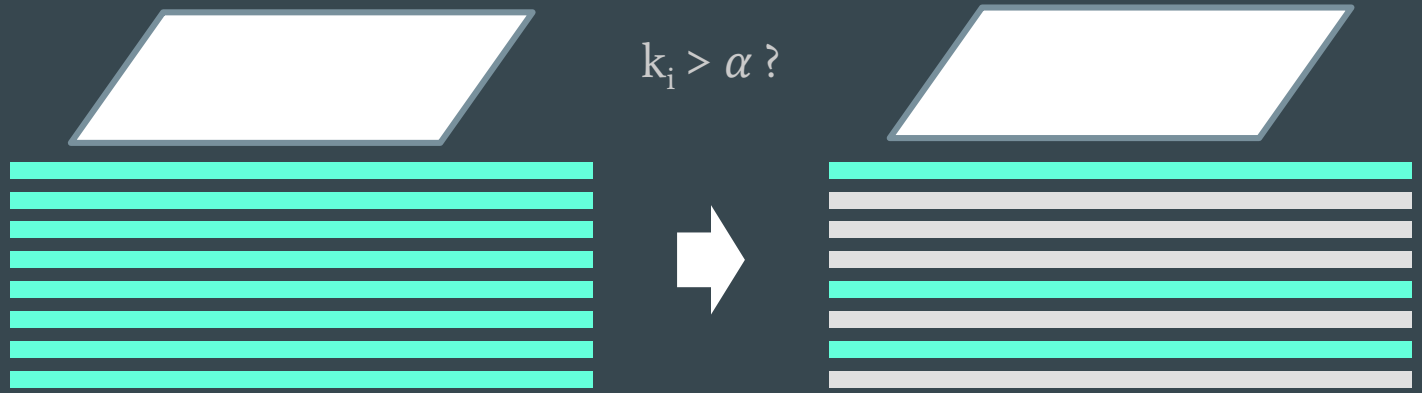
Stochastic Depth Map

- A multisample texture, containing one or more depth values per pixel, corresponding to random scene surfaces mapping to the pixel
  - Multisample texture: Texture with multiple samples per pixel (SPP)
- Stochastic depth-map pixel contains a random subset of the scene layers
- All sample locations are set to the pixel center, so that we can store all samples in a single render pass

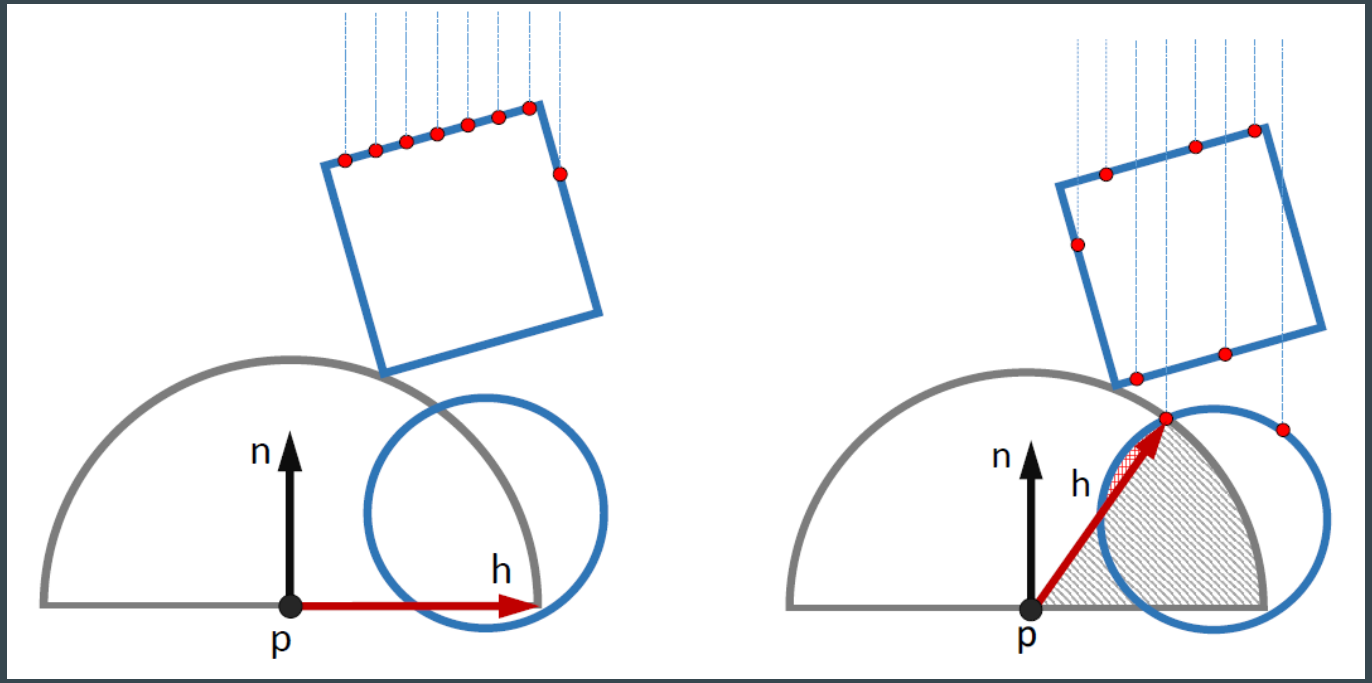


Generating Stochastic Depth Map (simple solution)

- Generate a uniformly distributed random number between 0 and 1 for every fragment using hash function
- The value is then compared to a global transparency value α in order to decide on storing or discarding it



Extending HBAO

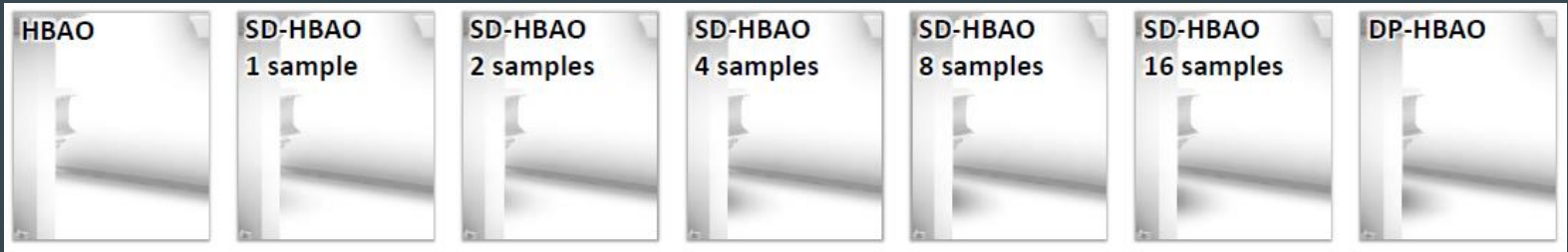


Heuristic & Improvement

- Sampling efficiency
  - Retrieving all depth samples can be costly
  - In simple geometric configurations (e.g. a flat surface), using only the regular depth buffer will be enough
- Multiple viewpoints
  - To avoid AO pop-in, use secondary stochastic depth maps at different viewpoint

Result

	SSAO	HBAO	HBAO+
Regular	0.926	0.734	0.694
1 stochastic sample	1.192	1.124	1.081
2 stochastic samples	1.460	1.302	1.372
4 stochastic samples	2.161	1.762	1.978
8 stochastic samples	3.557	2.528	3.212
16 stochastic samples	5.967	4.823	5.527



Limitation

- The values used to decide whether to discard the fragment or not, are uniformly distributed random numbers
- Therefore, when many layers exist between an occluder and the camera, AO values can be underestimated since fewer samples will be captured in the stochastic depth map
- Adding a compensation multiplier for AO from stochastic layers can restore some of the missing occlusion