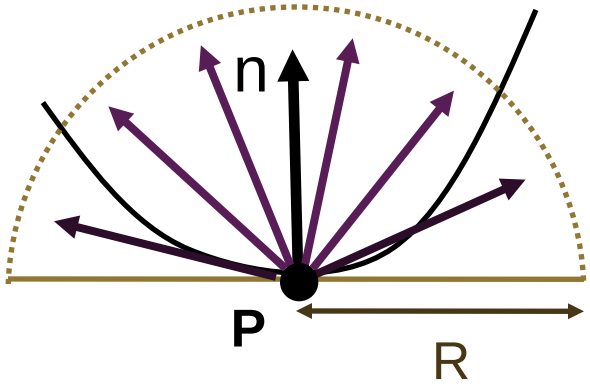
Multi-Layer Dual-Resolution Screen-Space Ambient Occlusion

Louis Bavoil, Miguel Sainz

NVIDIA

Ambient Occlusion

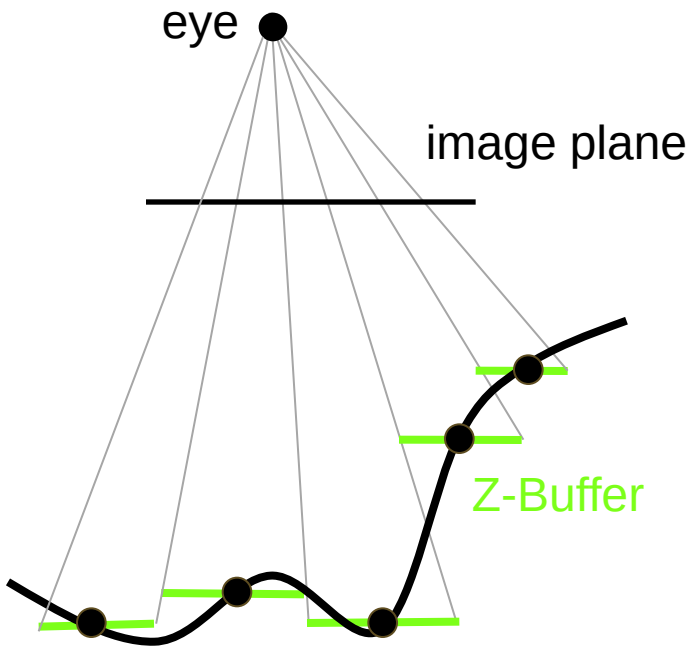
- Simplest form of AO
  - Soft shadows from sky dome
- General formulation
  - AO = % of rays that intersect local geometry inside radius R
  - Occlusion falls off with distance to soften inside-radius test



[Zhukov et al. 98]
[Landis 02]
[Christensen 03]

Screen-Space Ambient Occlusion

- Approach introduced by
  - [Shanmugam and Orikan 07] [Mittring 07] [Kajalin 09]
- Input = Z-Buffer
  - With or without normals, depending on the algorithm
  - Render approximate AO for dynamic scenes with no precomputations
- Done in post-processing

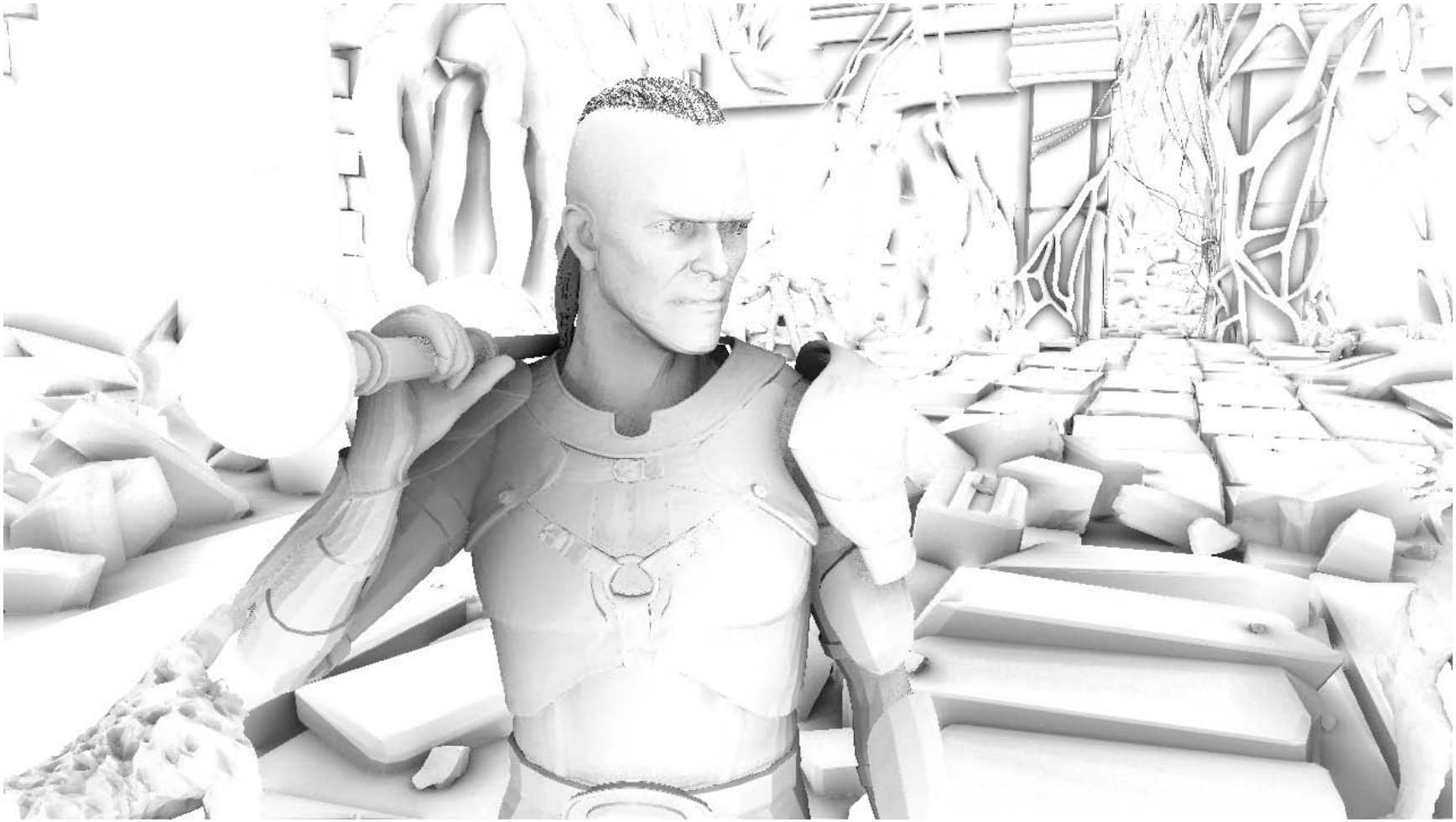


Screen-Space Ambient Occlusion

Without SSAO

With SSAO

SSAO Image



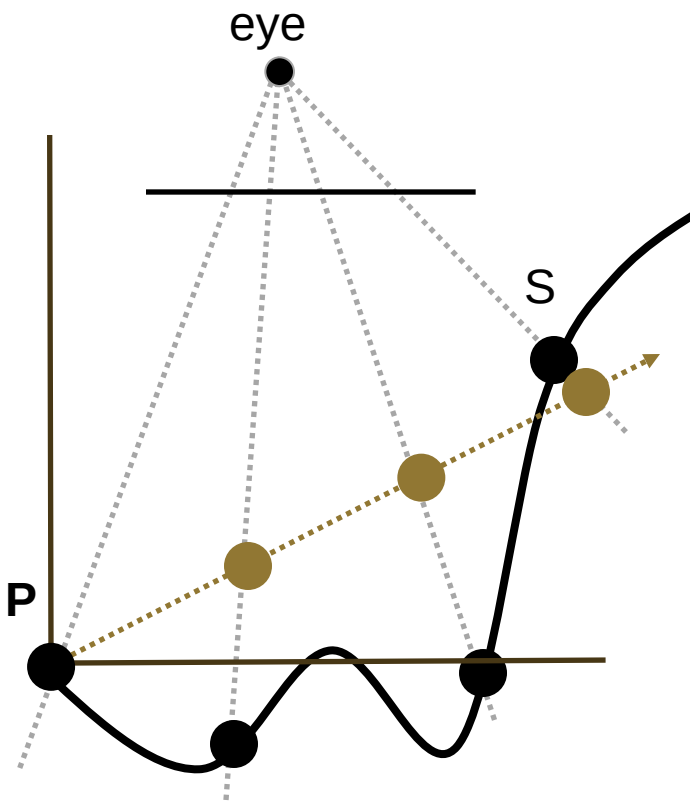
Horizon-Based Screen-Space AO [Bavoil and Sainz 08]

Agenda

- Part 1: Multi-Layer SSAO
  - Goal: Improving quality
  - Use cases: game cinematics and interactive film preview
- Part 2: Dual-Resolution SSAO
  - Goal: Improving performance

Ray-Marched SSAO

- Scene representation = Z-Buffer
- Cast rays from surface point P
  - Ray-march along each ray
  - Ray hits if behind a Z-Buffer sample S and ||S - P|| < R
  - Assuming locally continuous depth layer inside radius R

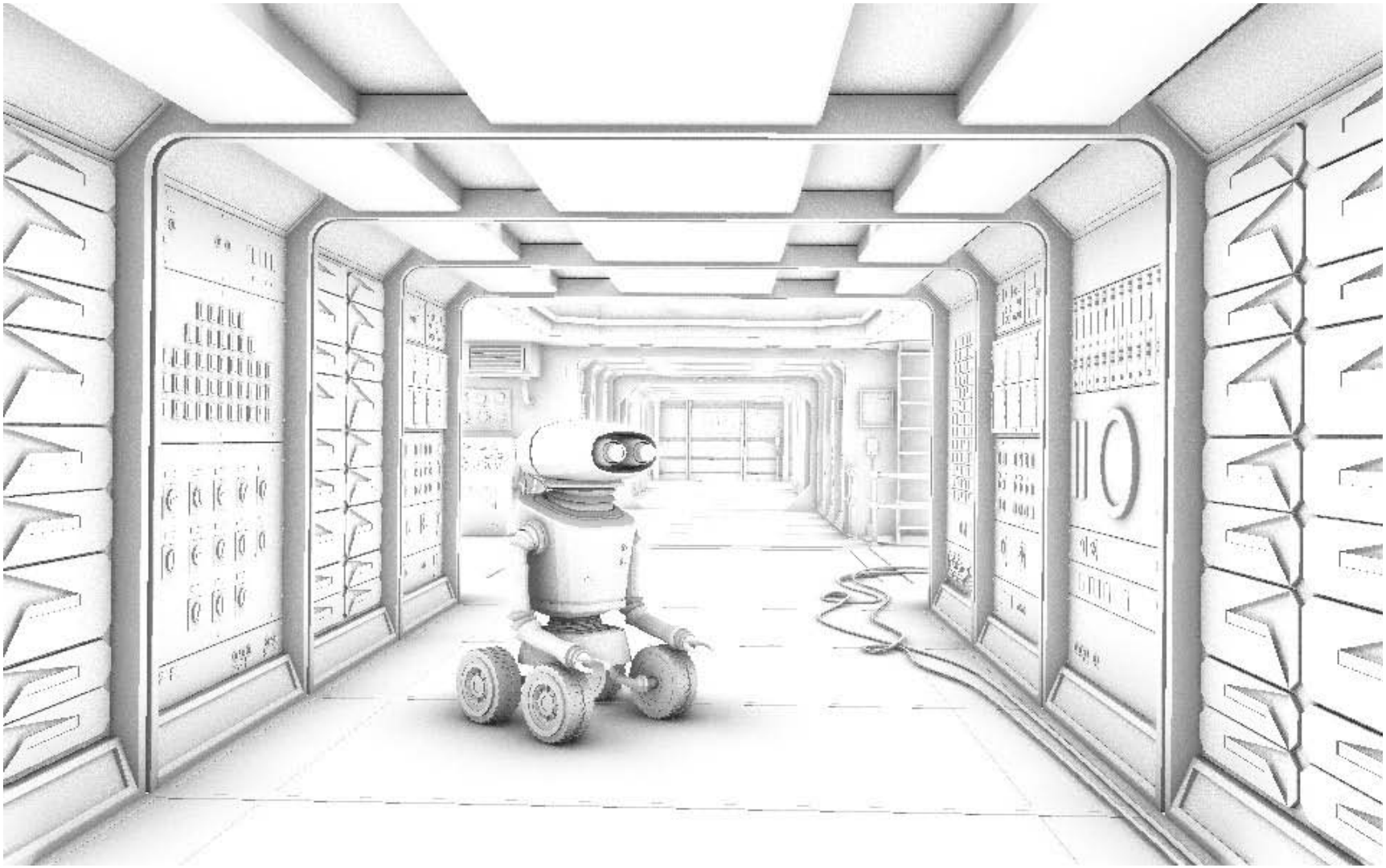


Horizon-Based SSAO

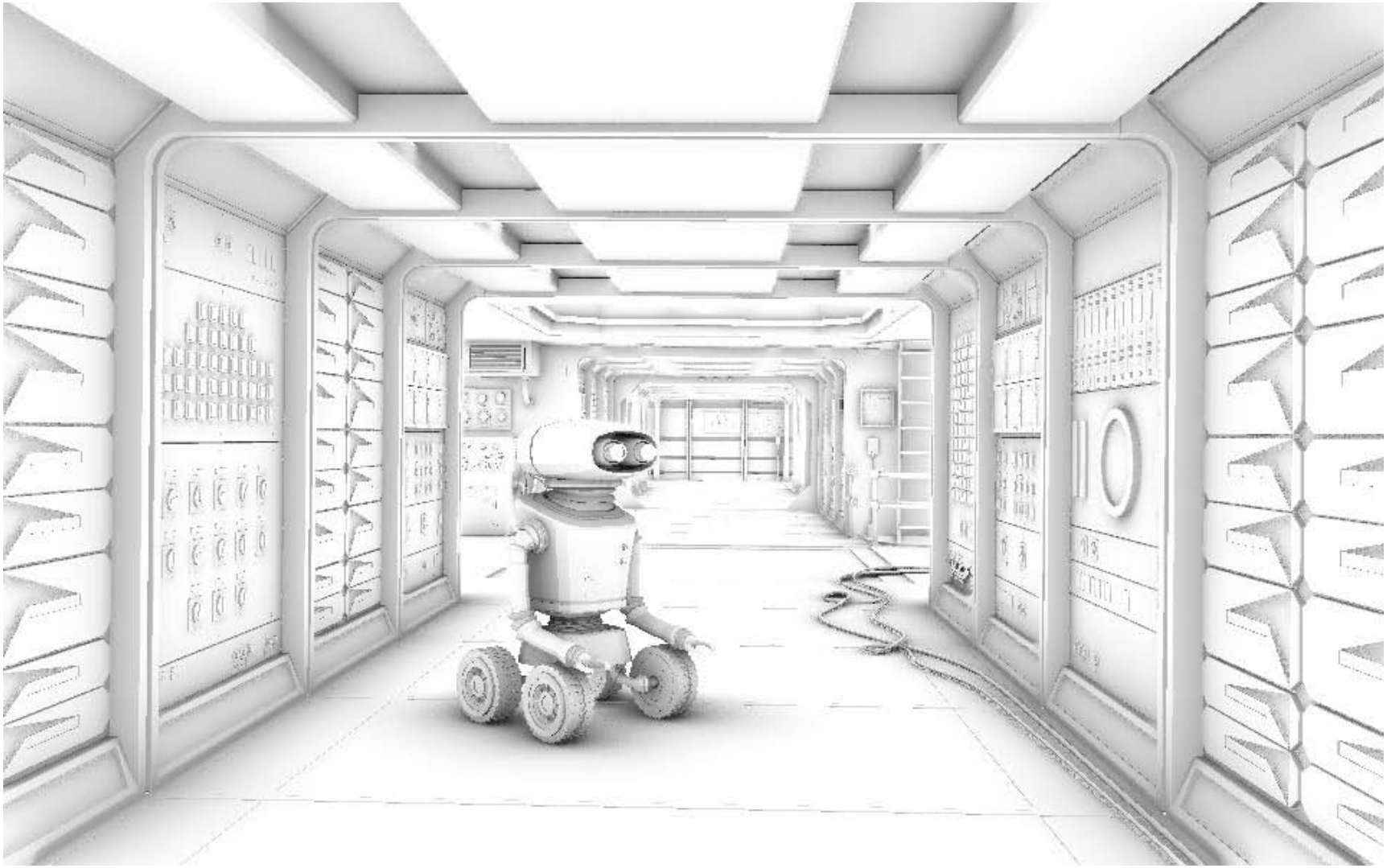
- We use the horizon-based SSAO algorithm
  - Approximation of ray-marched SSAO
  - [Bavoil and Sainz 08]

- Our quality-improvement changes can be applied to any SSAO algorithm

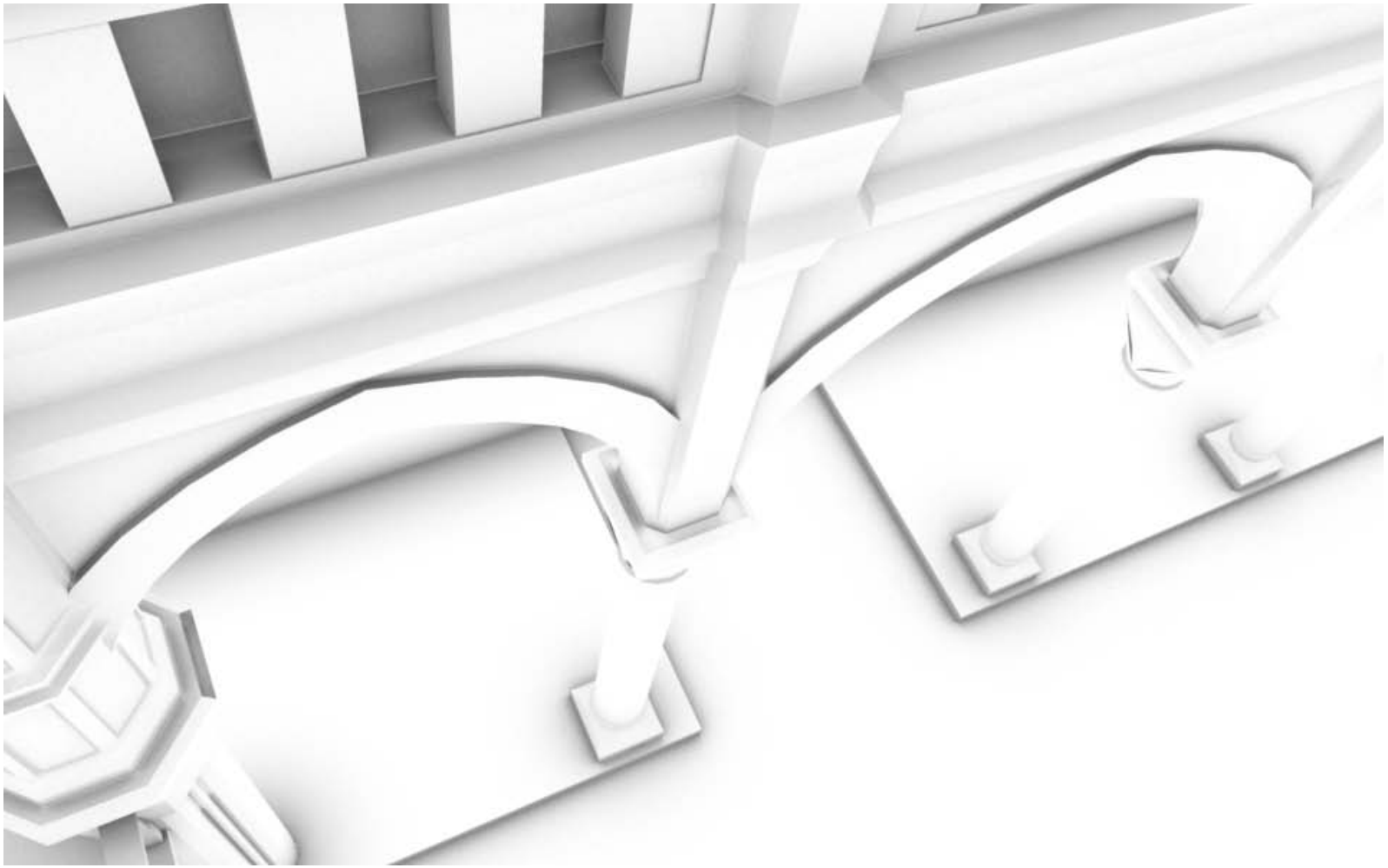
Horizon-Based SSAO



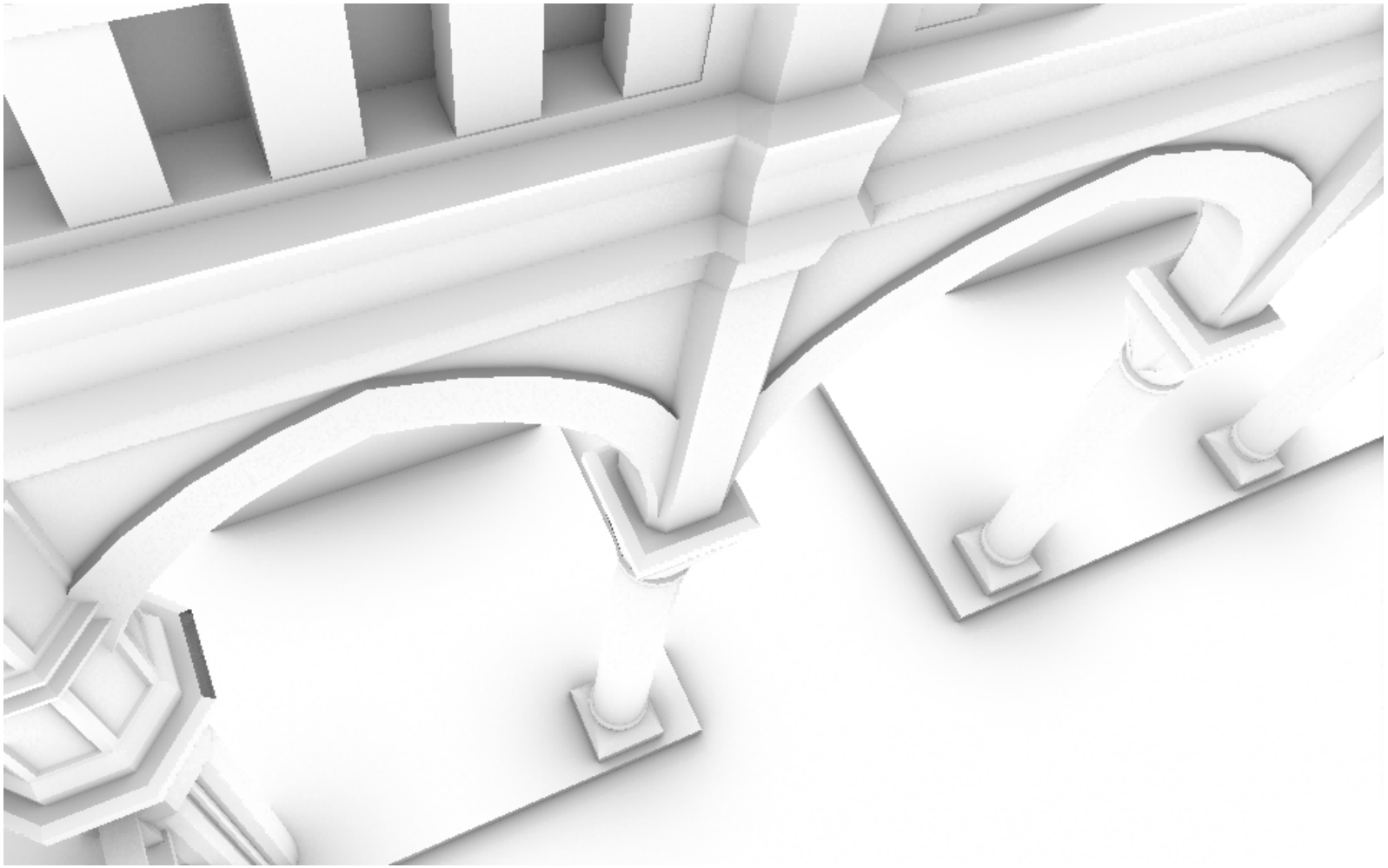
Ray-Marched SSAO



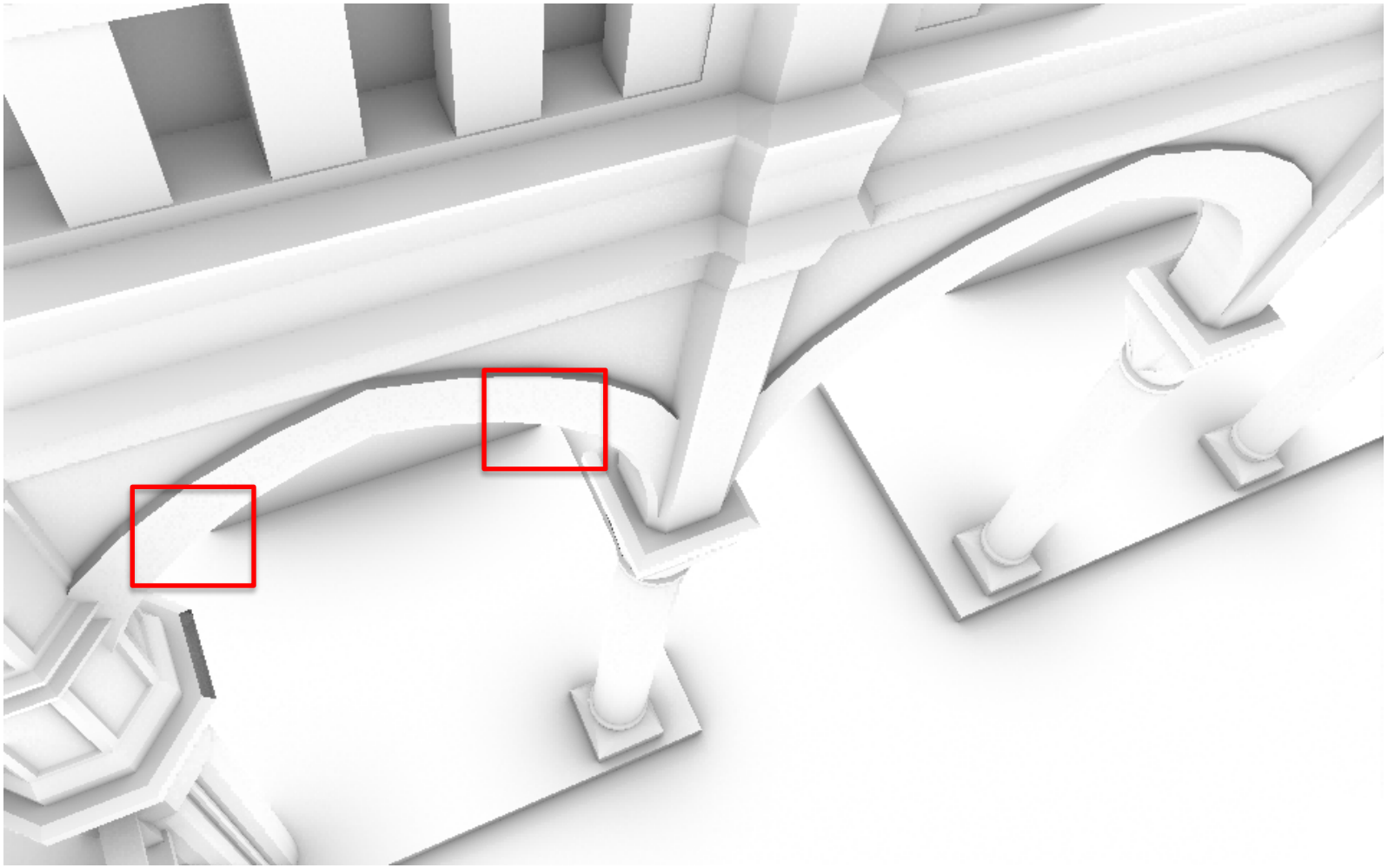
Ray-Traced AO using Gelato



Problem #1: Single Depth Layer

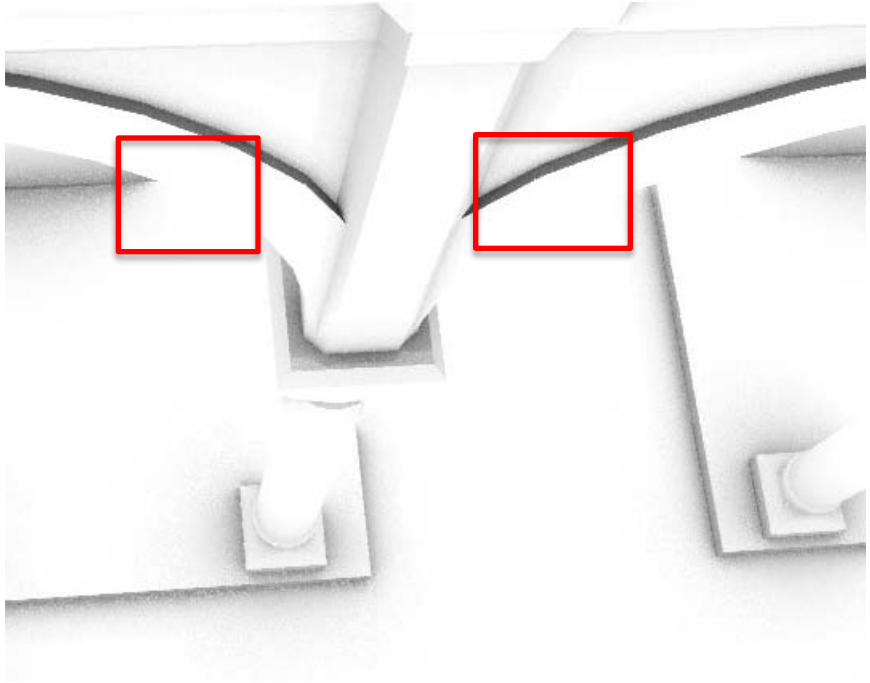


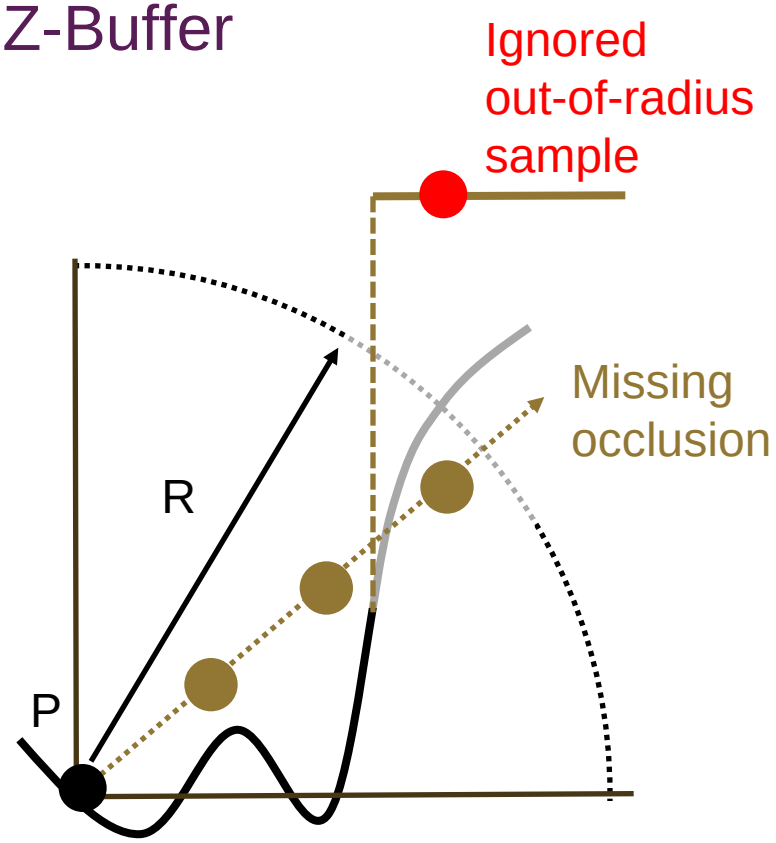
Problem #1: Single Depth Layer



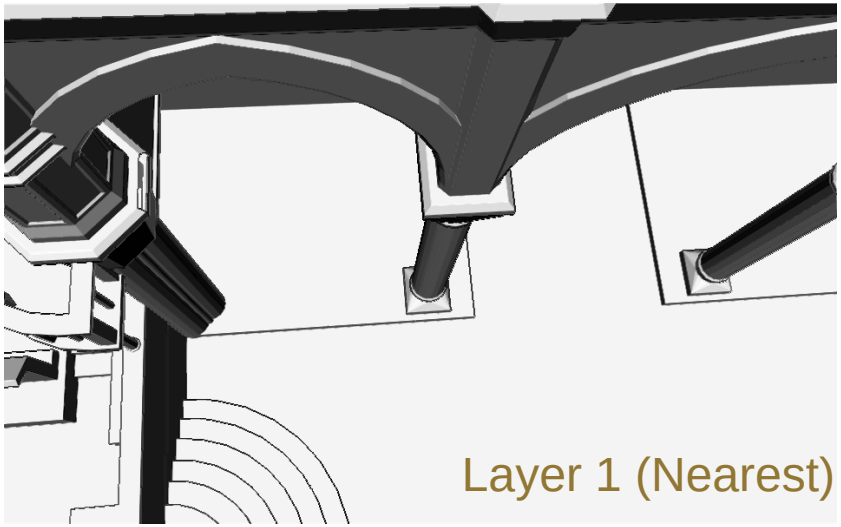
Problem #1: Single Depth Layer

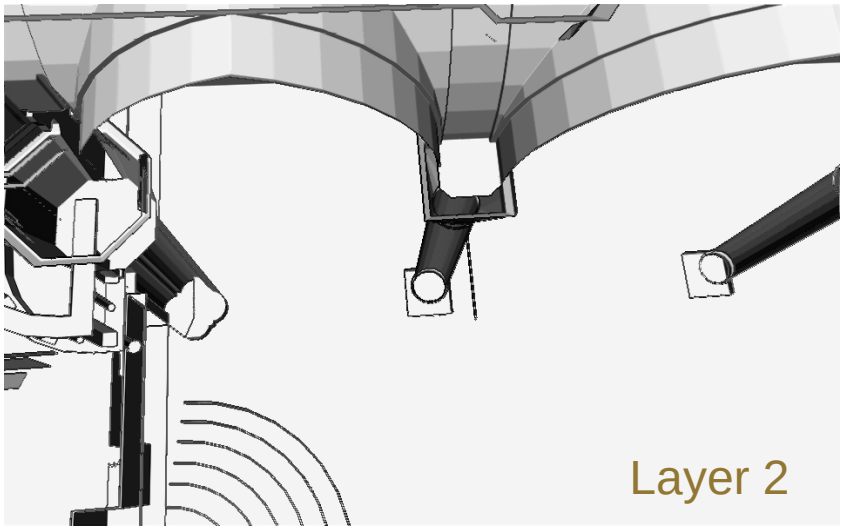
- Missing information behind the Z-Buffer

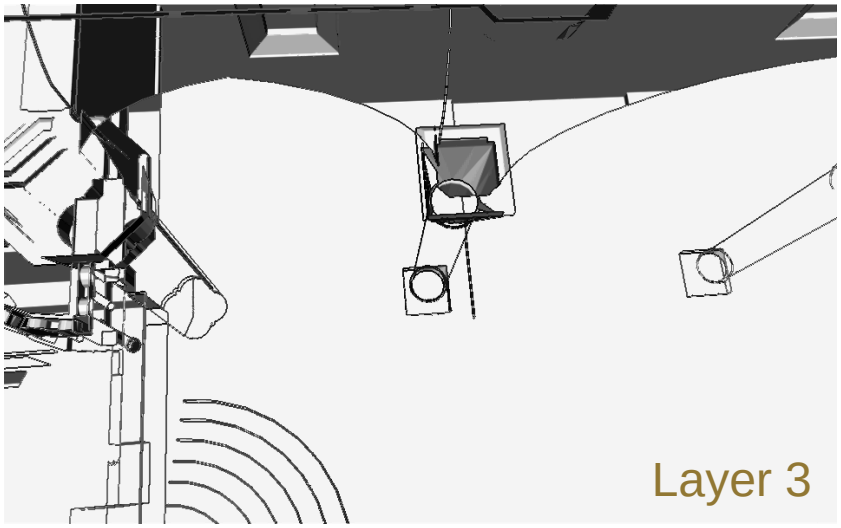


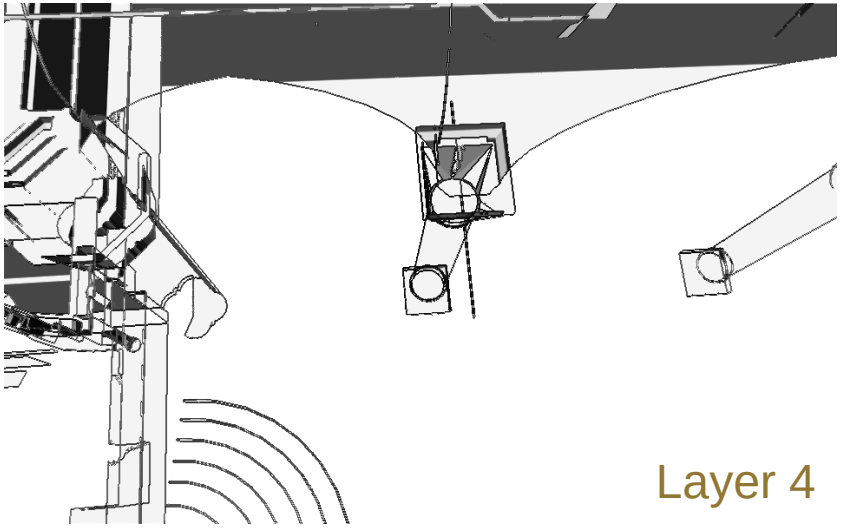


Depth Peeling [Mannen 84] [Everitt 01]



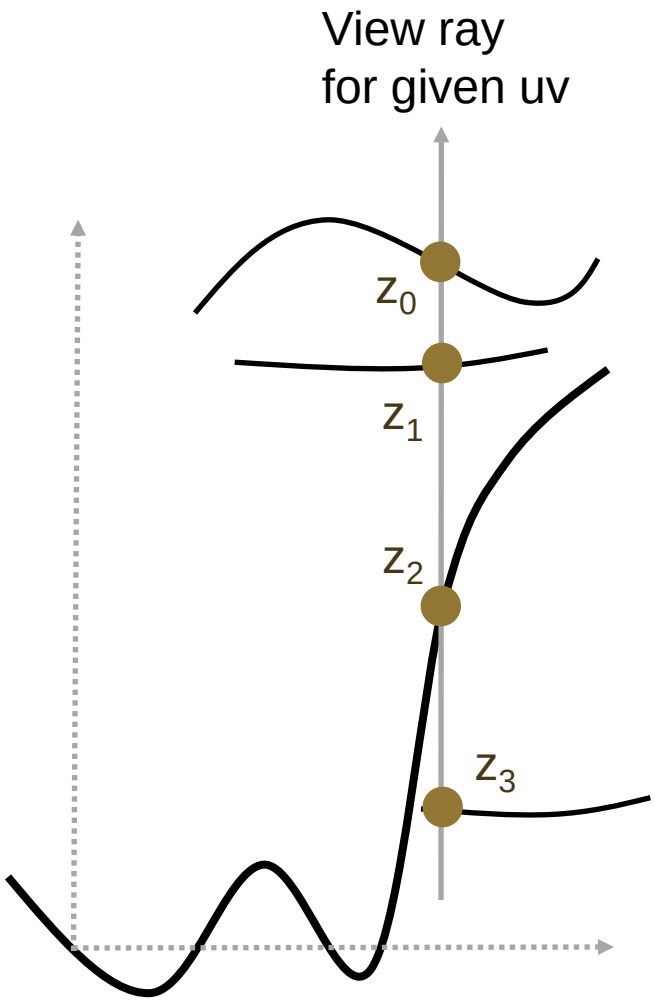




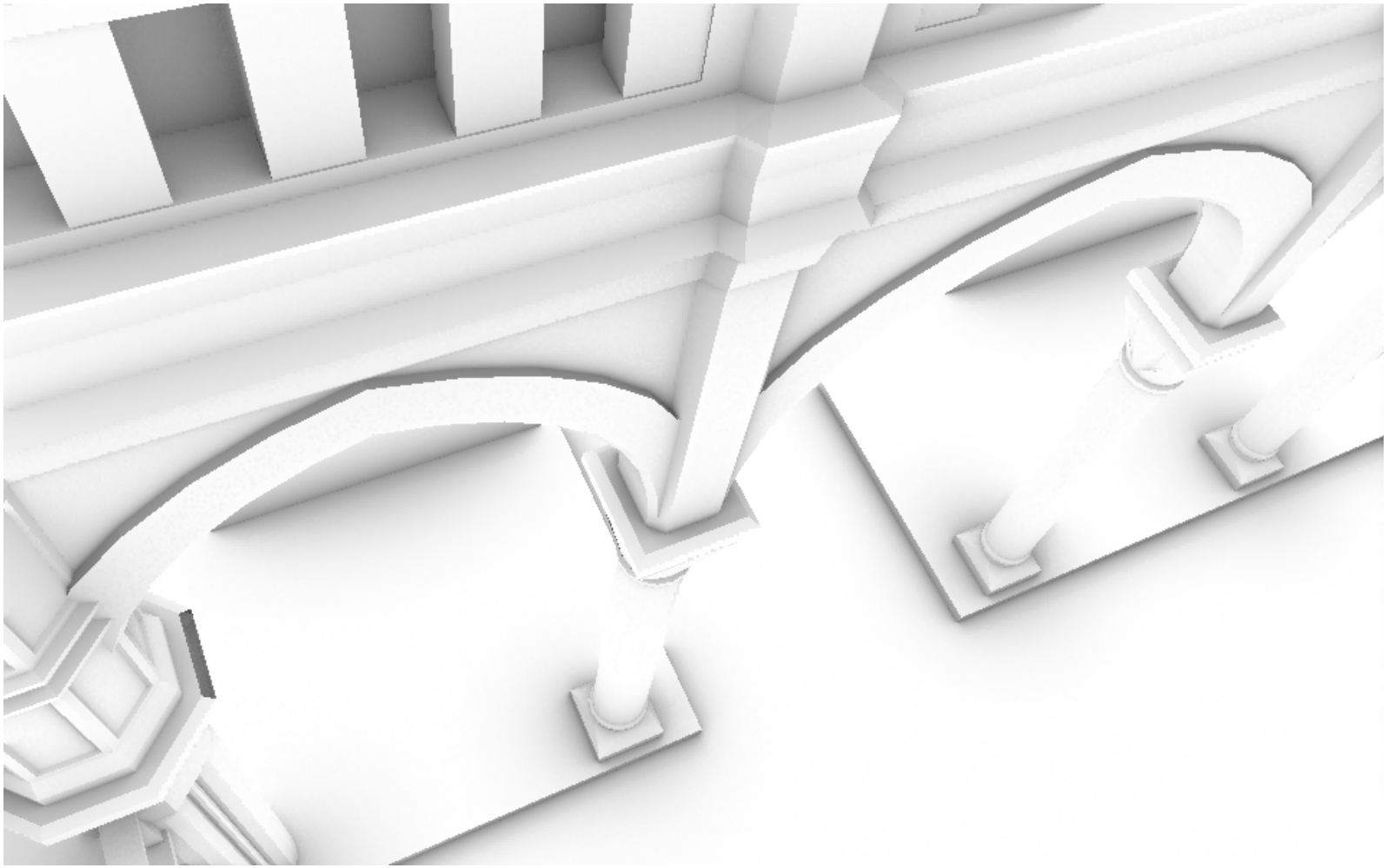


Multi-Layer SSAO

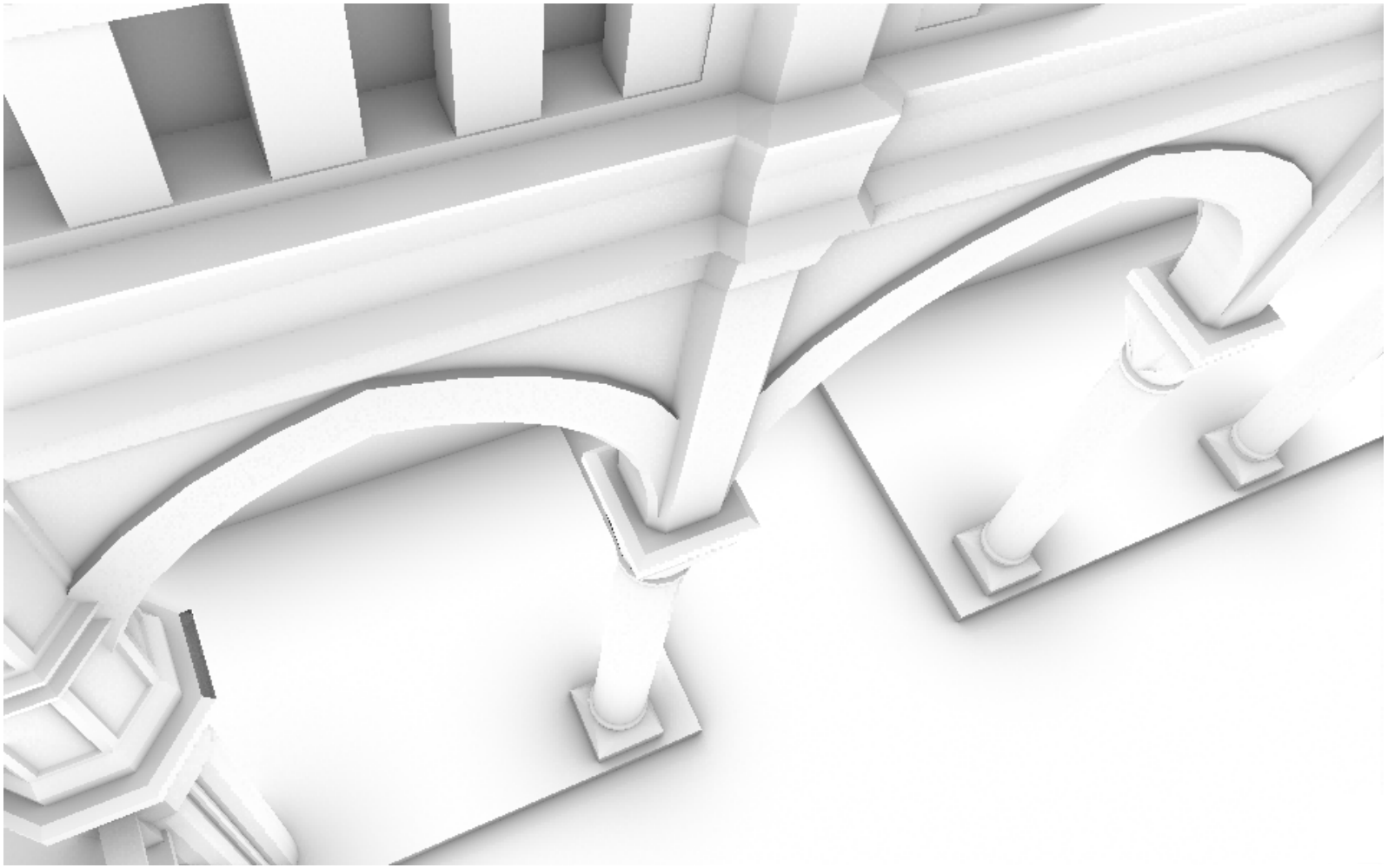
- Generic Solution
  - Lookup all the depths z_0, z_1, …, z_n for given texture coordinate uv
  - Use AO(uv) = max{ AO(uv,z_i) }
- Related Work
  - [Xie et al. 07] [Agrawala et al. 00] Intersect ray with multi-layer shadow map
  - [Ritschel et al. 09] Assumes all objects are solid objects



1-Layer SSAO

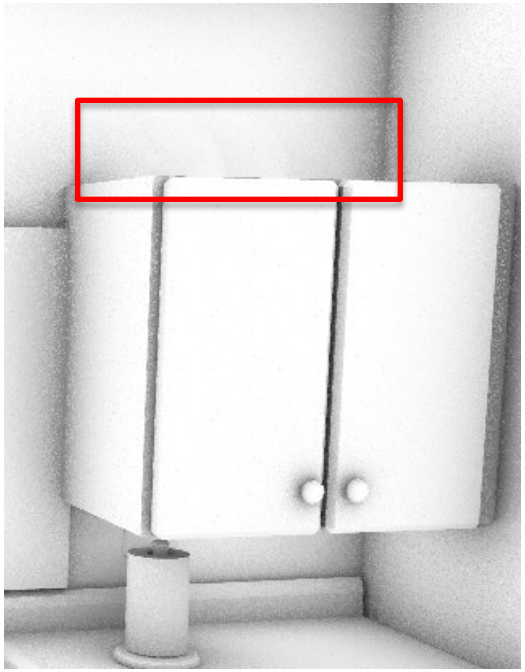


3-Layer SSAO



Limitations of depth peeling

- Not enough layers
  - Surfaces at grazing angle with view rays
  - Occluders with large depth complexity
- Can use multiple viewpoints
  - [Ritschel et al. 09]



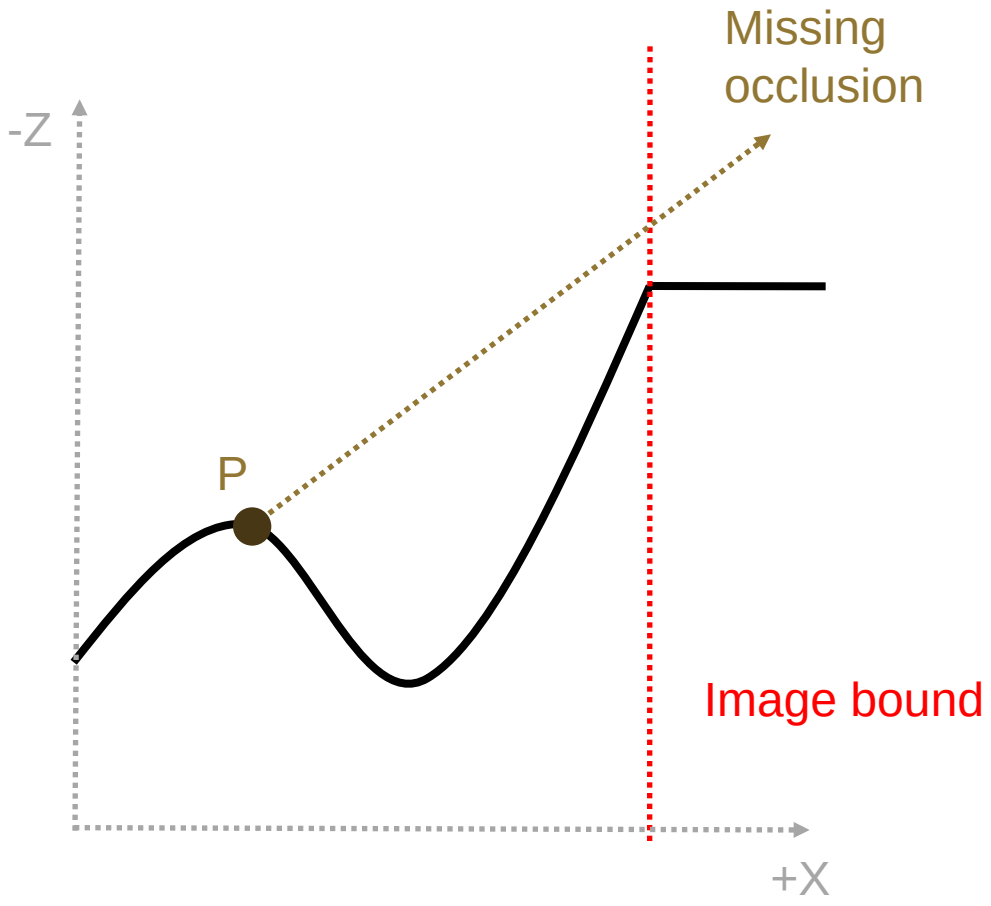
SSAO with 8 layers

Problem #2: View frustum

No depth information outside view frustum

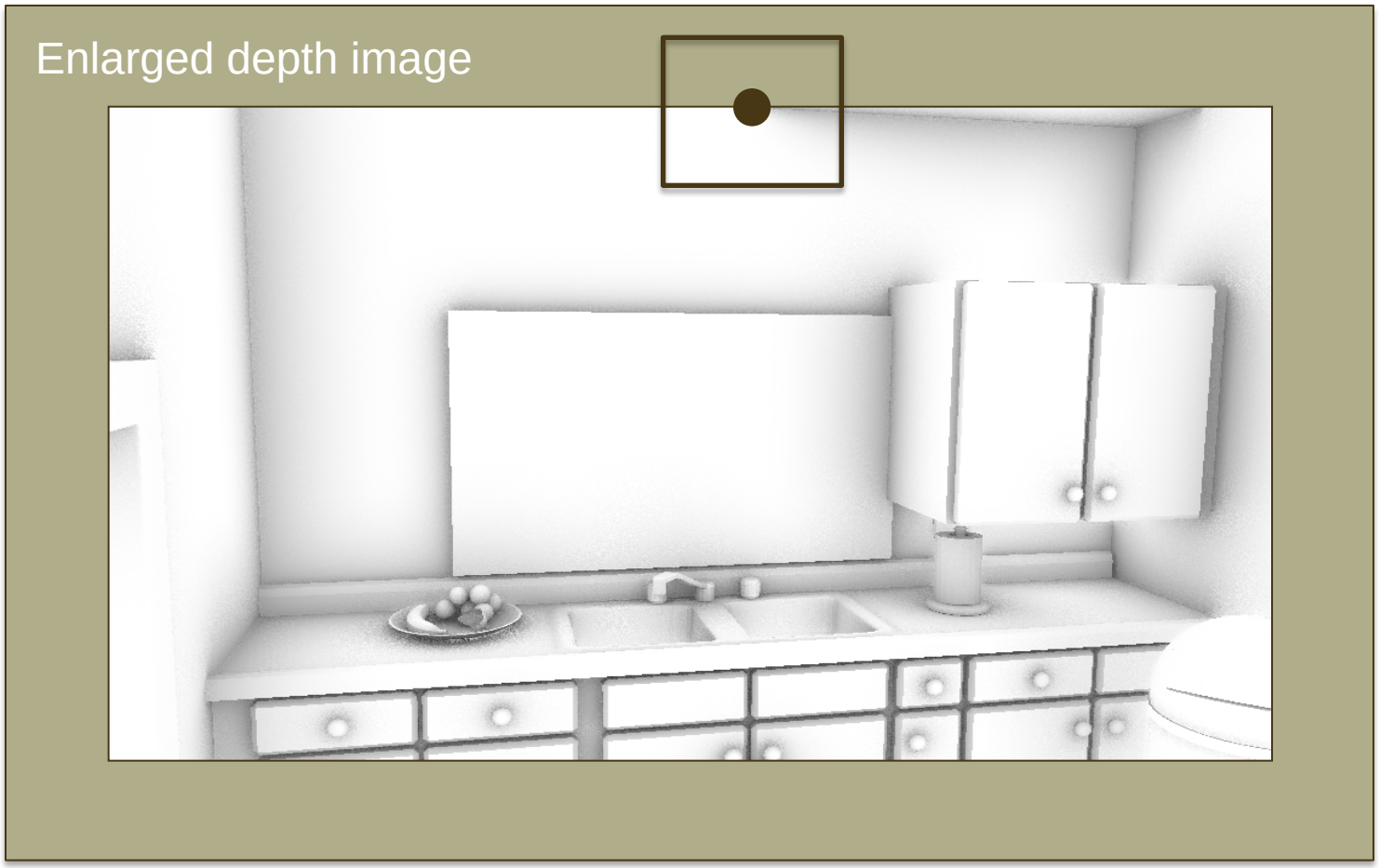
Clamping depths outside image bounds

- We use the CLAMP_TO_EDGE texture warping mode
- Has the effect of flattening the depths outside the view frustum

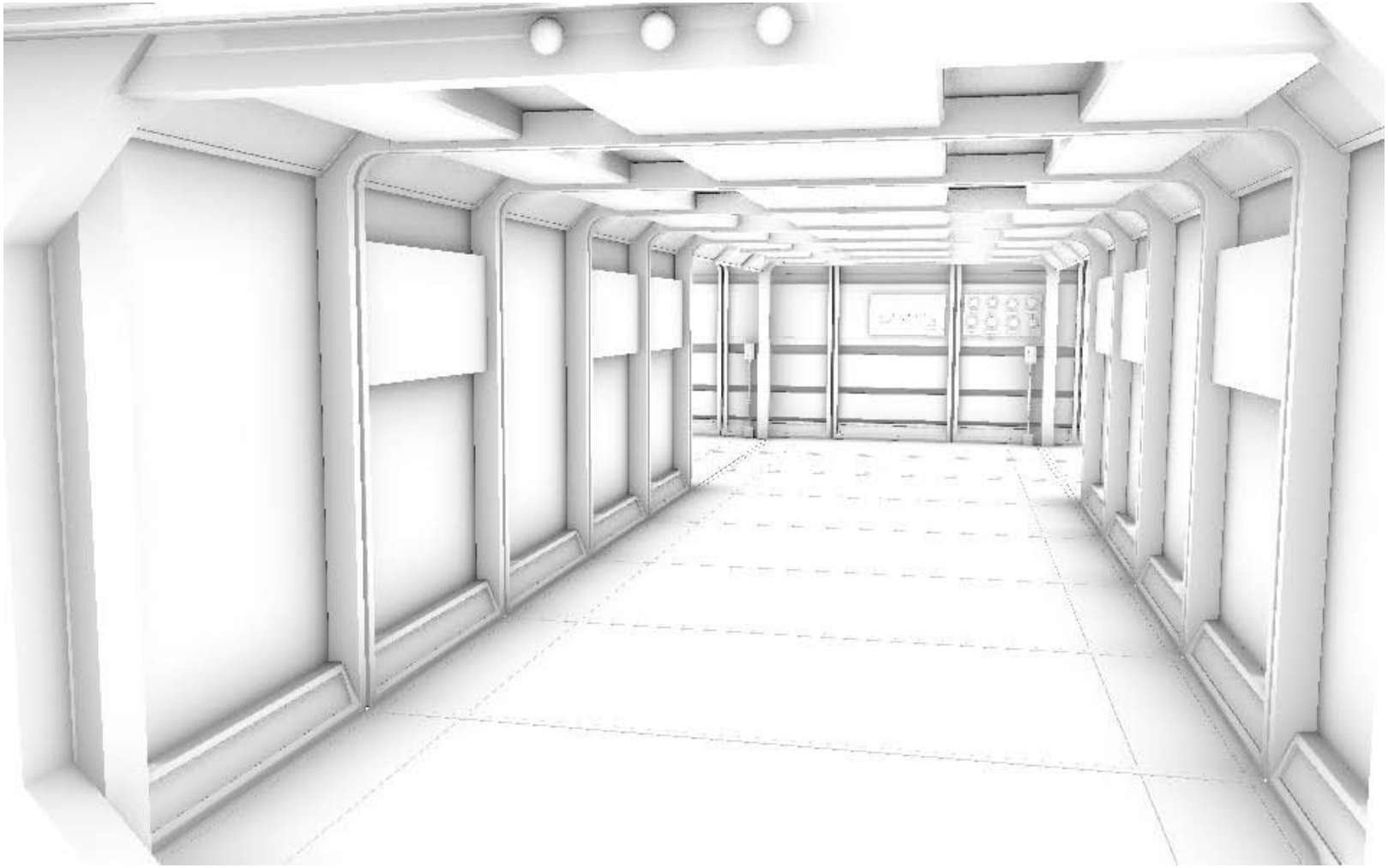


Solution: Enlarged frustum

We use enlarged depth images and enlarged field of view

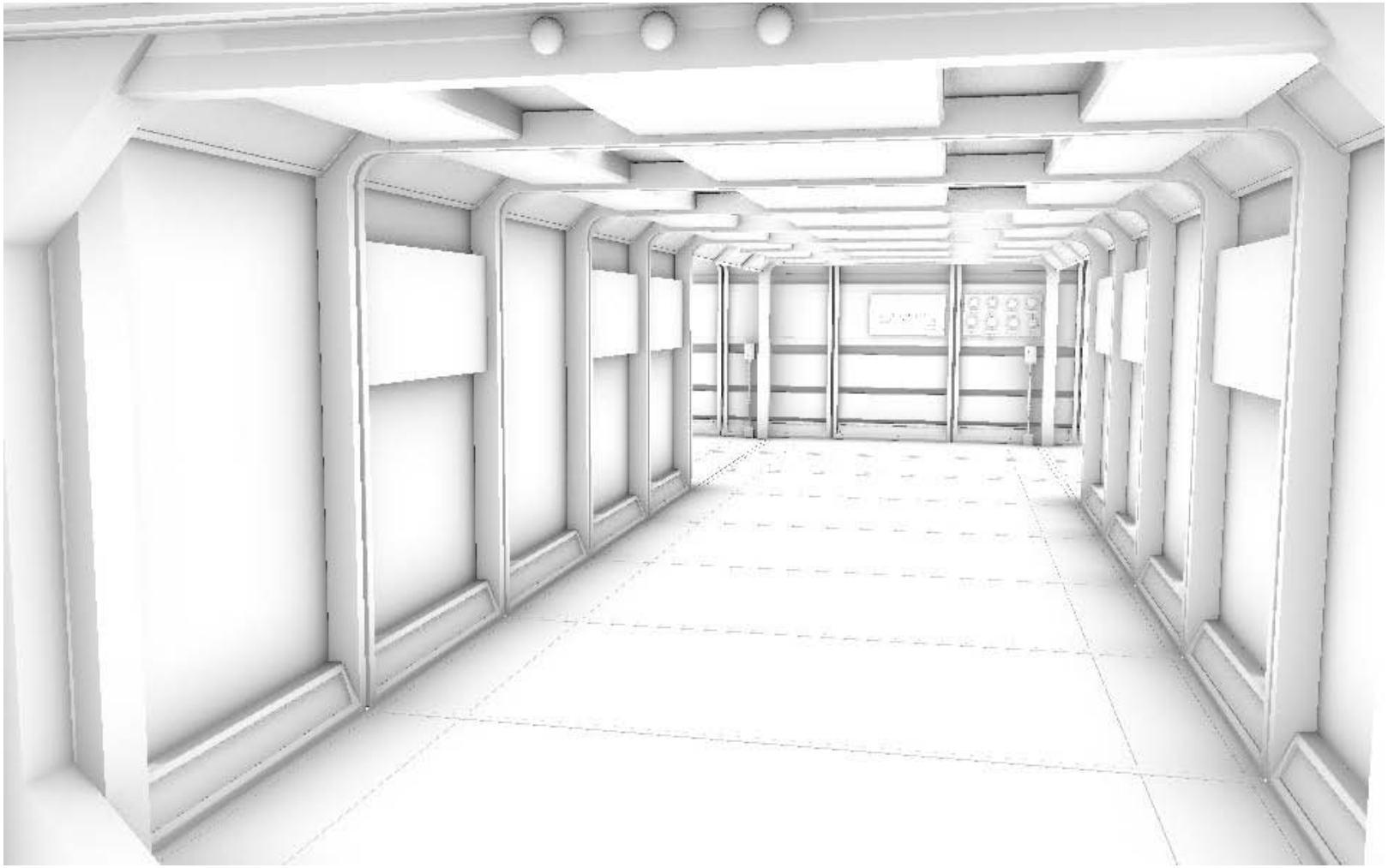


Without Enlarged Frustum



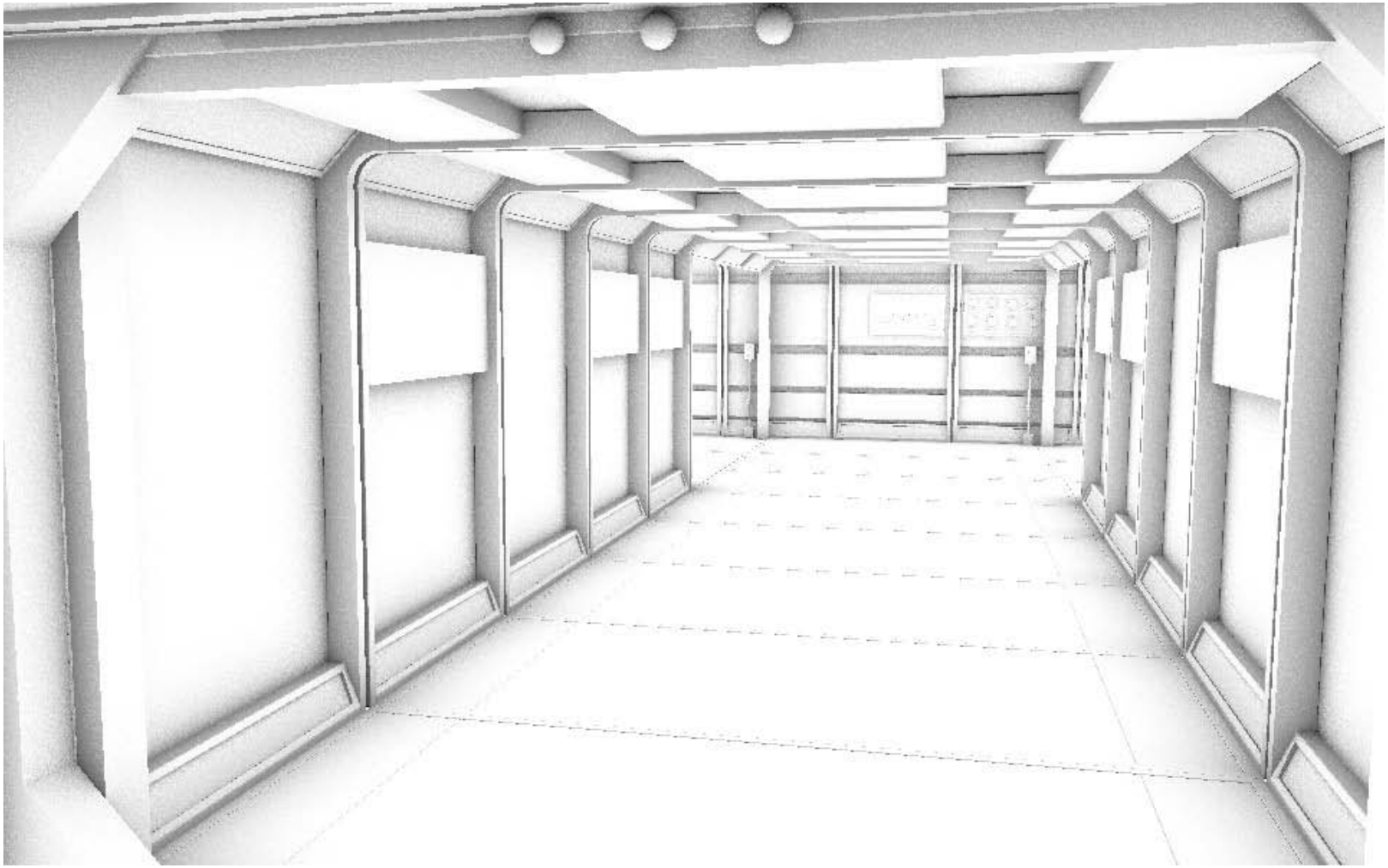
Horizon Based, 16x32 samples, 3 layers

With Enlarged Frustum

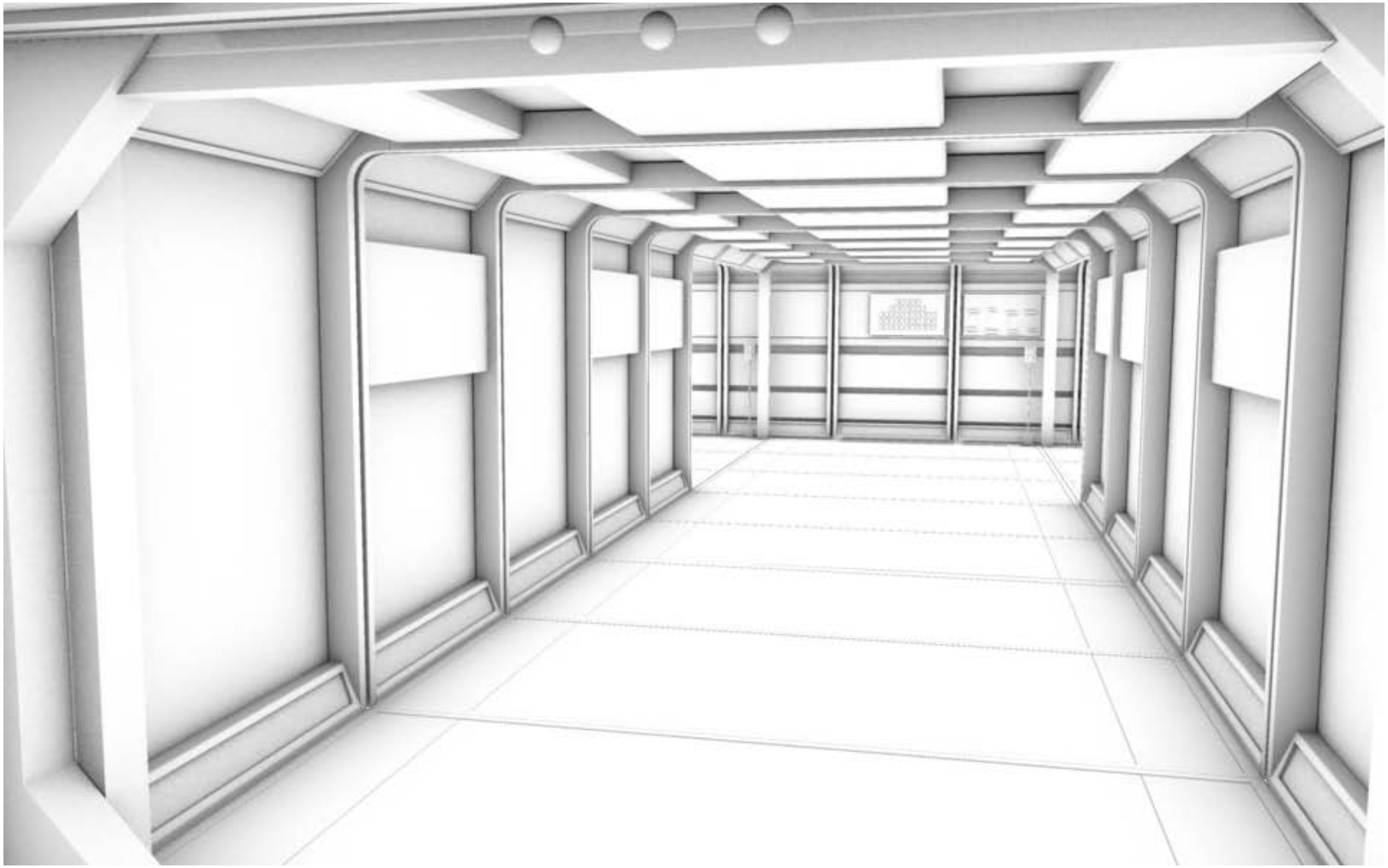


Horizon Based, 16x32 samples, 3 layers

8-Layer Ray-Marched SSAO

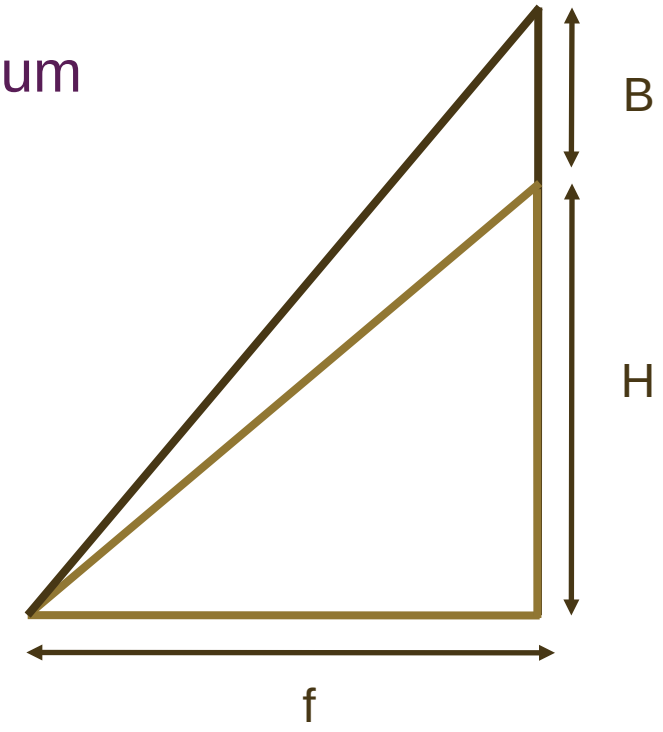


Ray-Traced AO (Gelato)



How much to enlarge?

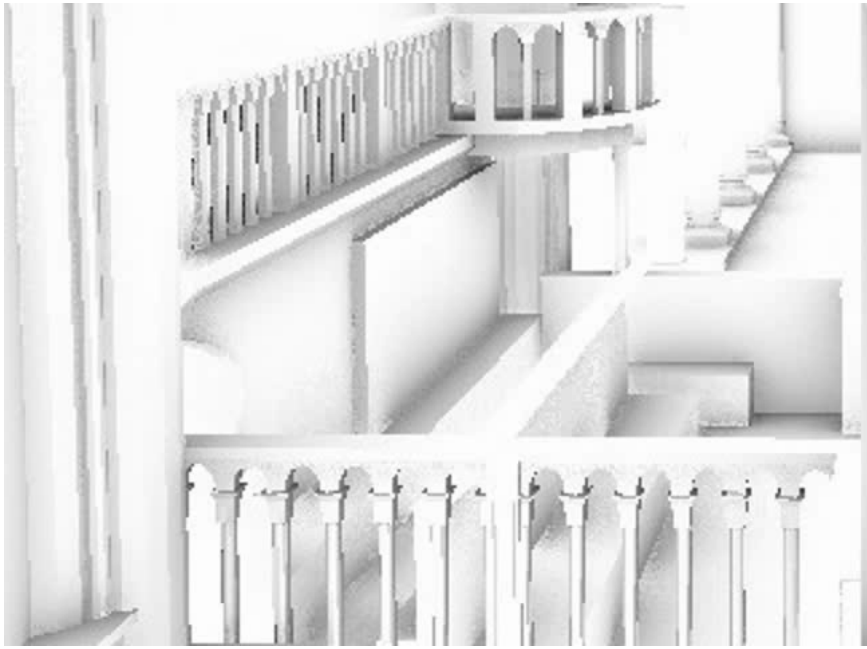
- Need upper bound of SSAO footprint radius (in pixels)
  - Could use the minimum depth in the current frame to compute the maximum kernel footprint
  - We just use a constant parameter (border size B)
- Enlarged field of view fovy' such as tan(fovy/2) = H/f and tan(fovy'/2) = (H+B)/f



Single-Layer SSAO

- Missing information outside frustum and behind z-buffer

Single-Layer vs Multi-Layer SSAO



1 depth layer, original frustum

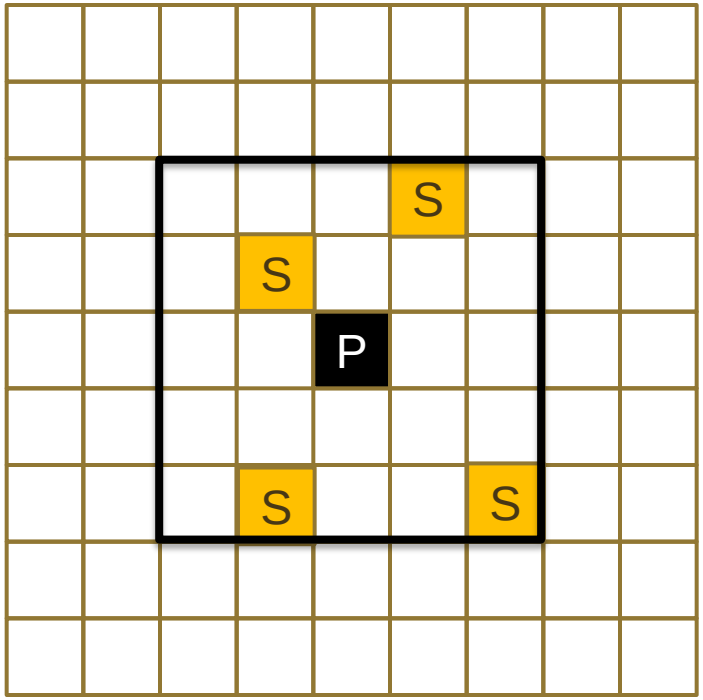
5 depth layers, enlarged frustum

Agenda

- Part 1: Multi-Layer SSAO
  - Goal: Improving quality
- **Part 2: Dual-Resolution SSAO**
  - Goal: Improving performance

Footprint Clamping

- Performance of SSAO algorithms depends heavily on kernel footprint in image space
- Max footprint parameter
  - Upper bound of SSAO kernel radius, in pixels
  - We discard the samples that are outside the max footprint
- Speedup due to less samples and more local access



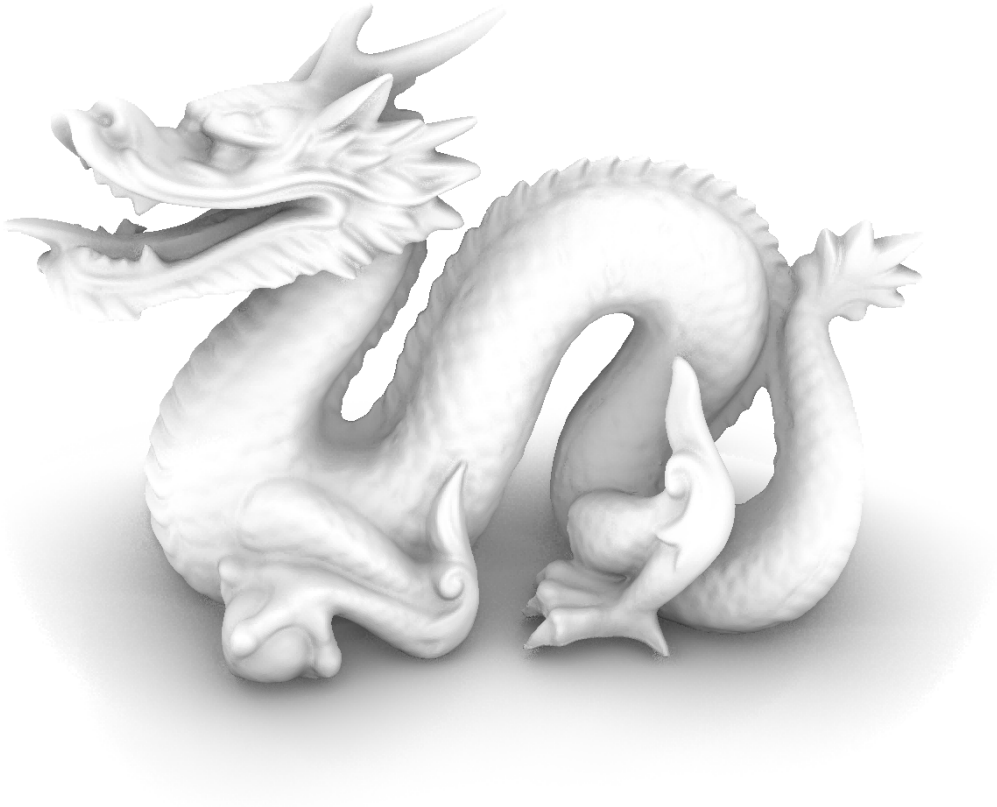
Example kernel footprint

Without Footprint Clamping

1600x1200, 32x32 samples, 3 layers
No max footprint
4 fps on GeForce GTX280

With Footprint Clamping

1600x1200, 32x32 samples, 3 layers
Max footprint = 10% of screen width
8 fps on GeForce GTX280



With Footprint Clamping

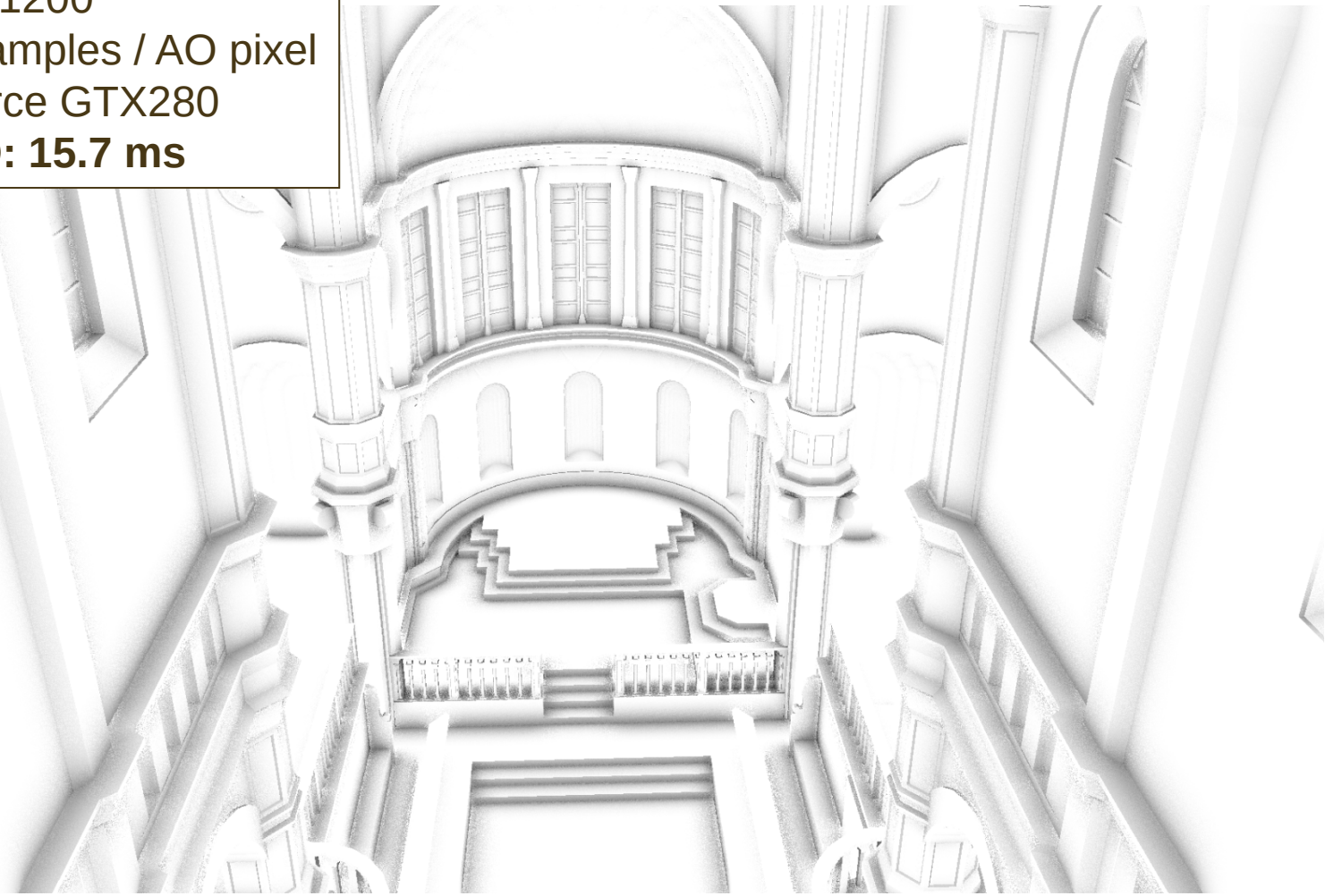
1600x1200, 32x32 samples, 3 layers
Max footprint = 5% of screen width
13 fps on GeForce GTX280

Using Half-Resolution SSAO

- AO is low frequency
- Render SSAO in half-resolution
  - Using half-resolution depth and normal input textures to improve texture cache efficiency
- Upsample the half-resolution AO using an edge-preserving filter
  - Sourcing full-resolution depths
  - [Kopf et al. 07]
    [Petschnigg et al. 04]
    [Eisemann and Durand 04]

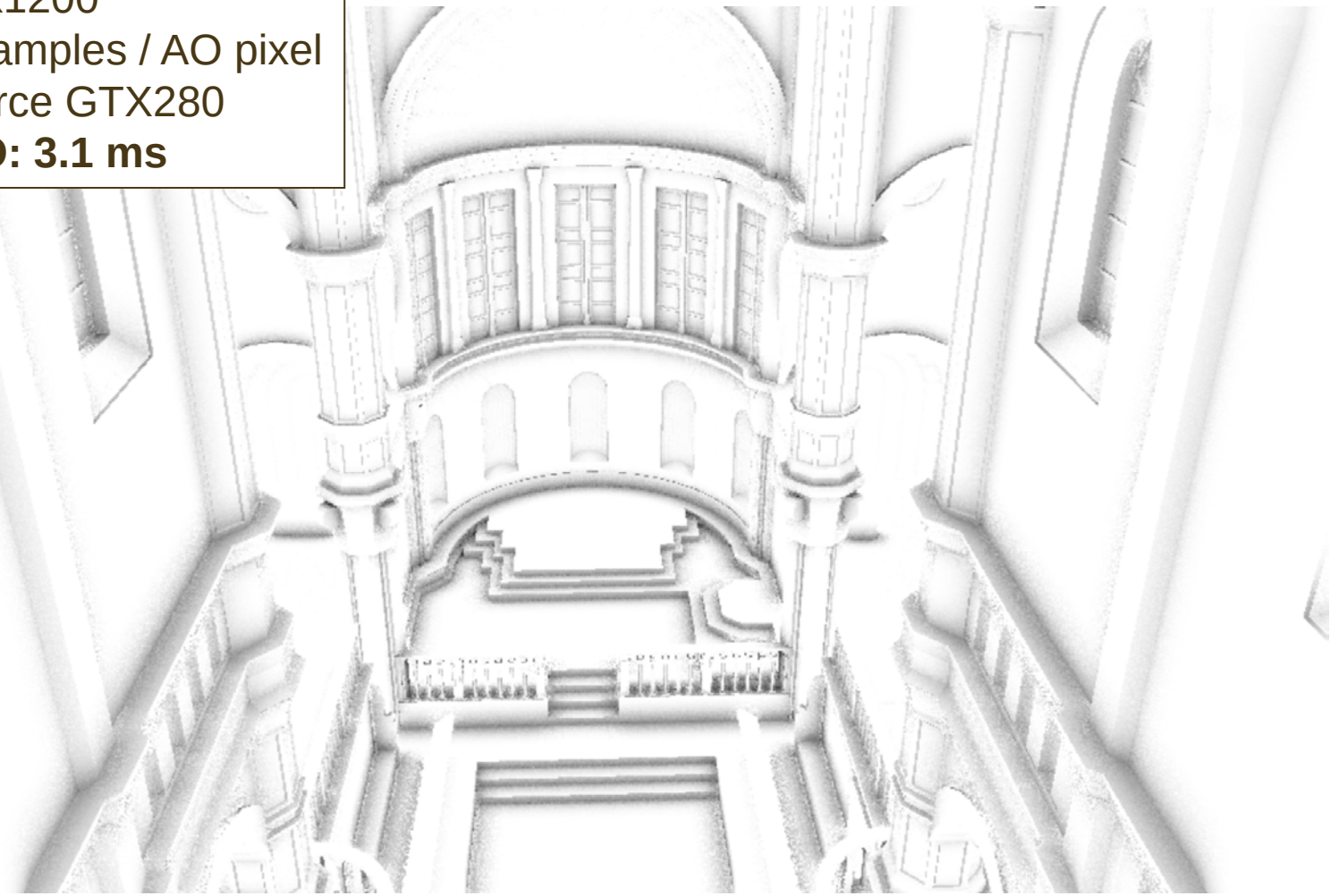
Full-Resolution SSAO

1920x1200
6x6 samples / AO pixel
GeForce GTX280
SSAO: 15.7 ms



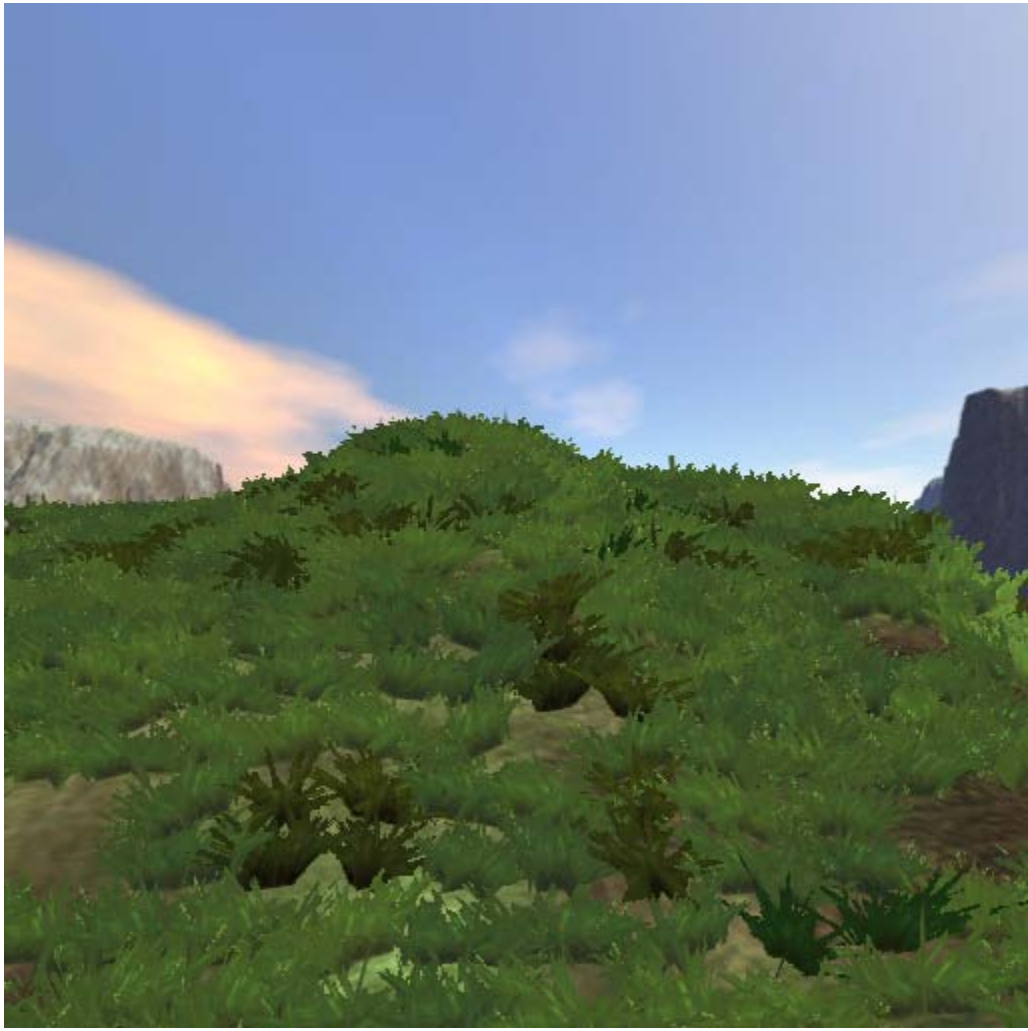
Half-Resolution SSAO

1920x1200
6x6 samples / AO pixel
GeForce GTX280
SSAO: 3.1 ms



High-Frequency Geometry

- For geometry smaller than 2 pixels generates shimmering
- Ideally, would render grass full-resolution with fullscreen antialiasing
- Shimmering due to half-res rendering



Dual-Resolution SSAO

- Do a first half-resolution pass
  - Cost ~3 ms in 1920x1200 on GTX280 with 6x6 samples
- Do a full-resolution refinement pass
  - Compute half-resolution AO variance in a small kernel
  - If (variance > threshold) compute full-resolution AO
  - Else, return half-resolution AO
- Similar refinement approaches used in
  - [Cantlay 07] [Nichols and Wyman 09] [Shopf 09]

Variance Estimation

- We use the range (max(AO) – min(AO)) as an upper bound for the variance
- We compute the local min and max half-resolution AO in a kernel (typically 3x3 or 5x5)
  - Approximate solution that works well in practice

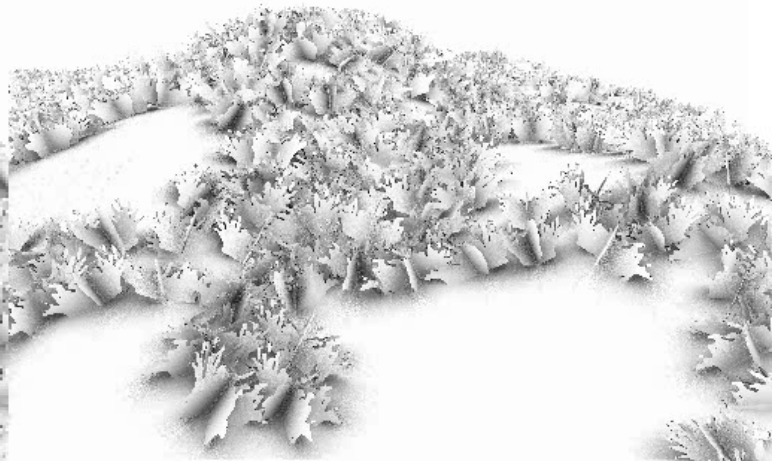
Refinement Threshold

- Refine pixel in full resolution if (max – min > threshold)
  - Threshold = 0 => all pixels are refined in full-resolution
  - Threshold = 1 => all pixels will be in half-resolution
  - Typical value: 0.1

Dual-Resolution SSAO

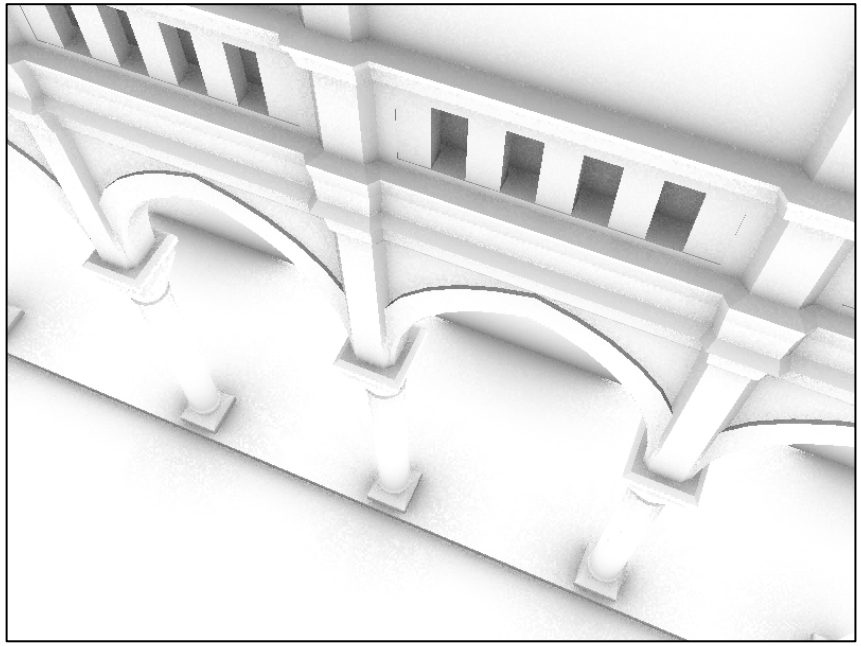
- Refining the grass pixels in full resolution

Half-Resolution

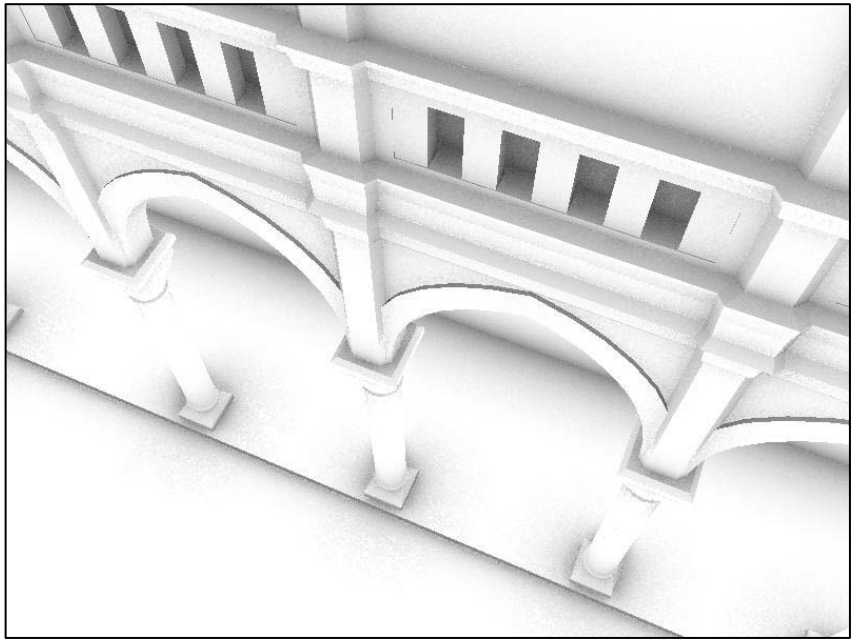


Dual-Resolution

Performance with Dual Resolution



1-Layer SSAO, 90 fps



3-Layer SSAO, 35 fps

- (800+100)x(600+100) resolution, GeForce 9800 GT
- Horizon Based SSAO, 8x16 samples per pixel
- AO range threshold = 0.1

Conclusion

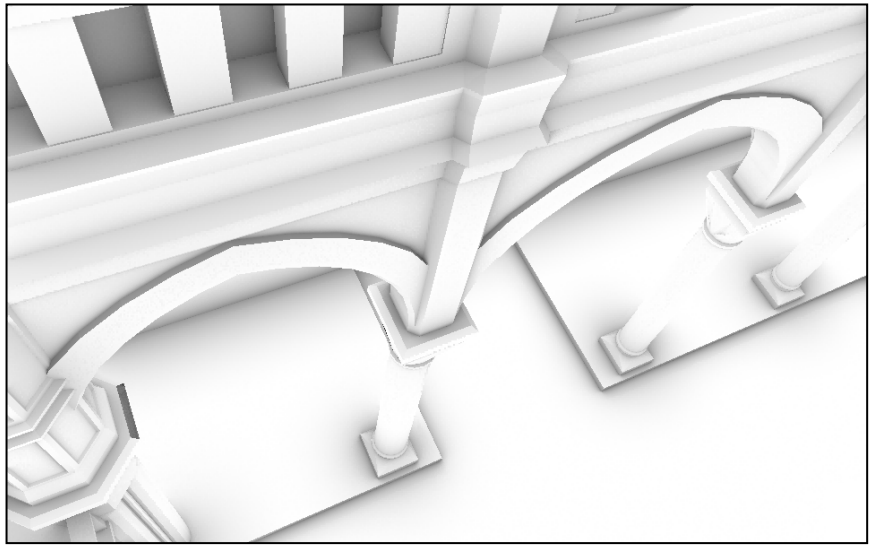
- SSAO is a scalable approach
  - Has room for improving quality while remaining interactive
- For higher-quality interactive AO
  - Input data = Z-Buffer + optional depth-peeled layers + optional borders
- For improving performance
  - Footprint clamping
  - Dual-resolution refinement based on low-resolution AO

Acknowledgments

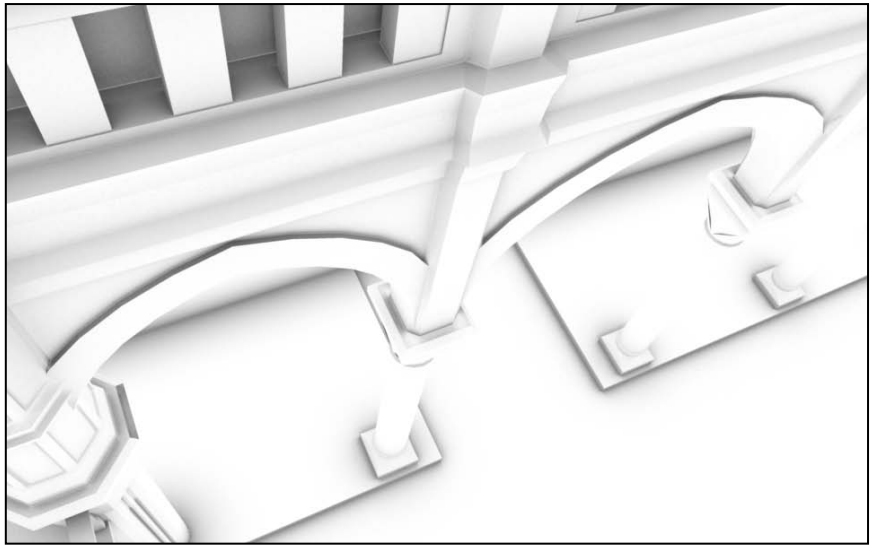
- Ideas
  - NVIDIA, Developer Technology Group, Byungmoon Kim
  - Manchor Ko, Chris Wyman
- Models
  - Warrior scene: NVIDIA demo team
  - Sci-Fi scene: Juan Carlos Silva
  - Kitchen scene: Jeremy Birn
  - Sibenik cathedral: Marko Dabrovic
  - Dragon: Stanford mesh repository

Questions?

- lbavoil@nvidia.com
- msainz@nvidia.com



3-Layer SSAO



Ray-Traced AO

References

- [Mammen 84] A. Mammen, “Transparency and antialiasing algorithms implemented with the virtual pixel maps technique”, IEEE Computer Graphics and Applications, 9:43–55, July 1984.
- [Zhukov et al. 98] Zhukov S., Inoes A., Kronin G., “An ambient light illumination model”, In Rendering Techniques ’98, Eurographics.
- [Agrawala et al. 00] Agrawala, Ramamoorthi, Heirich and Moll, “Efficient Image-Based Methods for Rendering Soft Shadows”, ACM SIGGRAPH 2000.
- [Everitt 01] Everitt C., “Interactive order-independent transparency”, Technical report, NVIDIA Corporation, 2001.
- [Landis 02] Landis, “Production-Ready Global Illumination”, ACM SIGGRAPH 2002, Course 16.
- [Christensen 03] Christensen, P. H., “Global illumination and all that”, ACM SIGGRAPH 2003, Course 9.

References

- [Eisemann and Durand 04] Eisemann E., Durand F., “Flash Photography Enhancement via Intrinsic Relighting”, ACM SIGGRAPH 2004.
- [Petschnigg et al. 04] Petschnigg, Szeliski, Agrawala, Cohen, Hoppe, Toyama, “Digital photography with flash and no-flash image pairs”, ACM SIGGRAPH 2004.
- [Mittring 07] Mittring, M. 2007. “Finding next gen: Cry Engine 2”. In SIGGRAPH ’07: ACM SIGGRAPH 2007 courses.
- [Shanmugam and Orikan 07] Shanmugam, P., Orikan, O. 2007. “Hardware accelerated ambient occlusion techniques on GPUs”. In I3D ’07: Proceedings of the 2007 symposium on Interactive 3D graphics and games.
- [Kopf et al. 07] Johannes Kopf, Michael F. Cohen, Dani Lischinski, Matt Uyttendaele. “Joint Bilateral Upsampling”. ACM SIGGRAPH 2007 papers.
- [Cantlay 07] Cantlay, I., 2007. “High-Speed, Off-Screen Particles”. GPU Gems 3.

References

- [Xie et al. 07] Xie, Tabellion, Pearce, “Soft Shadows by Ray Tracing Multilayer Transparent Shadow Maps”. Eurographics Symposium on Rendering 2007.
- [Bavoil and Sainz 08] Bavoil, L., Sainz, M., Dimitrov, R. 2008. "Image-Space Horizon-Based Ambient Occlusion", ACM SIGGRAPH 2008 talks.
- [Ritschel et al. 09] Ritschel, T., Grosch, T., Seidel, H.-P., “Approximating Dynamic Global Illumination in Image Space”. Proceedings ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (i3D) 2009.
- [Kajalin 09] Kajalin, V. 2009. “Screen Space Ambient Occlusion”, ShaderX7 - Advanced Rendering Techniques.
- [Nichols and Wyman 09] Nichols, Wyman, "Multiresolution Splatting for Indirect Illumination." ACM Symposium on Interactive 3D Graphics and Games 2009.
- [Shopf 09] Shopf, J. “Mixed Resolution Rendering”, Game Developers Conference (GDC) 2009.