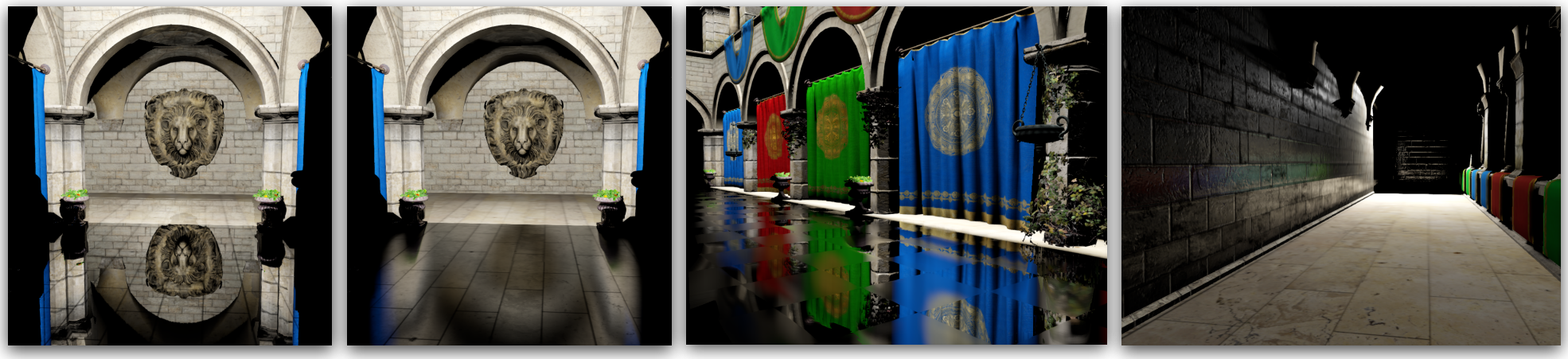
Screen Space Cone Tracing for Glossy Reflections

Lukas Hermanns · **TU** Darmstadt · **Fraunhofer** IGD · lukas.hermanns@igd.fraunhofer.de
Tobias Alexander Franke · **TU** Darmstadt · **Fraunhofer** IGD · tobias.franke@igd.fraunhofer.de



Introduction

An insight in [1] is that screen space algorithms can sample indirect bounces from nearby pixels. With Screen Space Local Reflections (SSLR) ray-tracing is employed to create the effect of reflections of the near-environment for each pixel. Performing this operation in screen space allows the effect to be used as a post-process. We extend SSLR by tracing a cone instead of a ray to support glossy indirect reflections on surfaces of varying roughness. We call this approach Screen Space Cone Tracing (SSCT).

Technical Approach

Our goal is to simulate reflections on surfaces of varying roughness in a post-process. For each pixel of a surface in an image we want to find the reflected geometry of the near-environment. However, depending on the surface's roughness, we may need to sample incident radiance from a cone of many rays with potentially incoherent texture lookups [2]. Instead, we rely on the idea of cone-tracing, which replaces rays with cones in a ray-tracer. By pre-filtering the sample space (in our case the screen-space information of the scene), each cone can approximate incident radiance with very few or just one sample. Figure **1** illustrates the general idea.

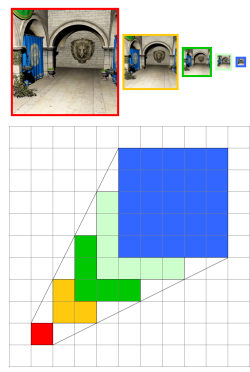


Figure 1: Each quad represents a single pixel sample of a pre-filtered image. At any point on the cone its distance and solid angle can be used to determine a filter level.

We start out with a regular render pass from the camera view into a Geometry Buffer (GBuffer) which includes textures for linear-depth, normals and colors. To perform cone-tracing later on we pre-filter all three textures in the GBuffer with hardware MIP-mapping, similar to [3].

In the SSCT post-processing step we reconstruct the position P of each pixel in world space and its normal N from the GBuffer and ray-march through the image from P in the reflected viewing direction V_R with continuously increasing step size by a factor of 1.5 for a maximum of 100 steps. If we step through the depth buffer (i.e. the current cone positions Z is behind the sampled depth), a hit-point is detected. We go back to the midpoint of the current position Q_N and the previous position Q_{N-1} and further refine the hit position Q with a binary search of 30 steps, decreasing the step size by a factor of 0.5 in each iteration. If Q is within a defined ε we found our final hit-point.

For each texture access to the depth buffer, we first compute the current path length and solid angle of the cone to compute the MIP-level where one sample represents the approximate average depth of geometry. After we found a hit point Q, we also access the filtered normal and color texture to compute the final incident radiance of the cone, which we add as secondary contribution to the pixel's radiance. As shown above surfaces with varying glossiness can be rendered with this approach.

Discussion

By relying on MIP maps of the camera view space certain integration errors are unavoidable (see Figure **2a**). The most apparent is that a solid angle can subtend either flat spaces or multiple pieces of geometry visible only at grazing angles, which can manifest as alias or temporal inconsistency when moving the camera view. As the cone angle size increases to simulate less glossy appearances, this error will increase notably, both for integrated colors and visibility of geometry.

For SSCT using a proper texture filter is crucial. Next to naive MIP mapping, we also tested manual filtering by using slices of a 3D texture, where each slice is a Gaussian blurred version of the original texture. This results in significantly better quality and smoother transitions at geometric discontinuities (see Figure **2b**). Note that we also generate a 3D texture for the depth map to produce more precise results in finding a ray hit for glossy reflections. This approach however has the disadvantage of losing fast texture access since we sample from a high-resolution 3D texture.

As with most post-processes, SSCT suffers from unreconstructed or hidden geometry. When for instance the camera looks down onto the floor reflections will disappear, since they lie outside of the available screen space and cannot be sampled. To avoid sharp boundaries at these instances, we fade out reflections when the hit-point is close to the screen space edge (see Figure **2c**).

(a) (b) (c)

Figure 2: (a) A simple texture filtering for diffuse, normal- and linear-depth maps may cause artifacts. (b) The result with a Gaussian filter. (c) Fading hides hard edges at unreconstructed geometry.

Conclusion

SSCT is a real-time method for glossy reflections with varying surface roughness in a post-process. The reflection works well in the near environment but some limitations are unavoidable. The quality depends highly on the chosen texture filters.

We are currently exploring other ways to solve the problem of finding the correct hit-point. In our current implementation we do not use edge-aware filtering, which causes artifacts when filtering geometry depth. We want to extend SSCT by using several layers in the spirit of [4].

We also want to investigate using a cube map GBuffer from the current view position to partially circumvent the problem of unreconstructed geometry not present in just the camera view. This approach however is not a screen space effect and thus cannot be used as a pure post-process.

References

[1] Ritschel, T., Grosch, T., and Seidel, H.-P. 2009. ***Approximating dynamic global illumination in image space.*** Symposium on Interactive 3D Graphics and Games.

[2] Soler, C., Hoel, O., and Rochet, F. 2010. ***A deferred shading pipeline for real-time indirect illumination.*** ACM SIGGRAPH 2010 Talks.

[3] Crassin, C., Neyret, F., Sainz, M., Green, S., and Eisemann, E. 2011. ***Interactive indirect illumination using voxel cone tracing.*** Computer Graphics Forum.

[4] Sugihara, M., Rauwendaal, R. and Salvi, M. 2014. ***Layered Reflective Shadow Maps for Voxel-based Indirect Illumination.*** High Performance Graphics 2014.

SIGGRAPH2014 · TECHNISCHE UNIVERSITÄT DARMSTADT · Fraunhofer IGD