

Real-time path tracing using a hybrid deferred approach

Thomas Schander
 @ThomasSchander
Clemens Musterle



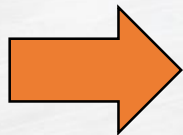
GTC EUR 2017, Talk #23026

What stops us from real-time PT?

High performance



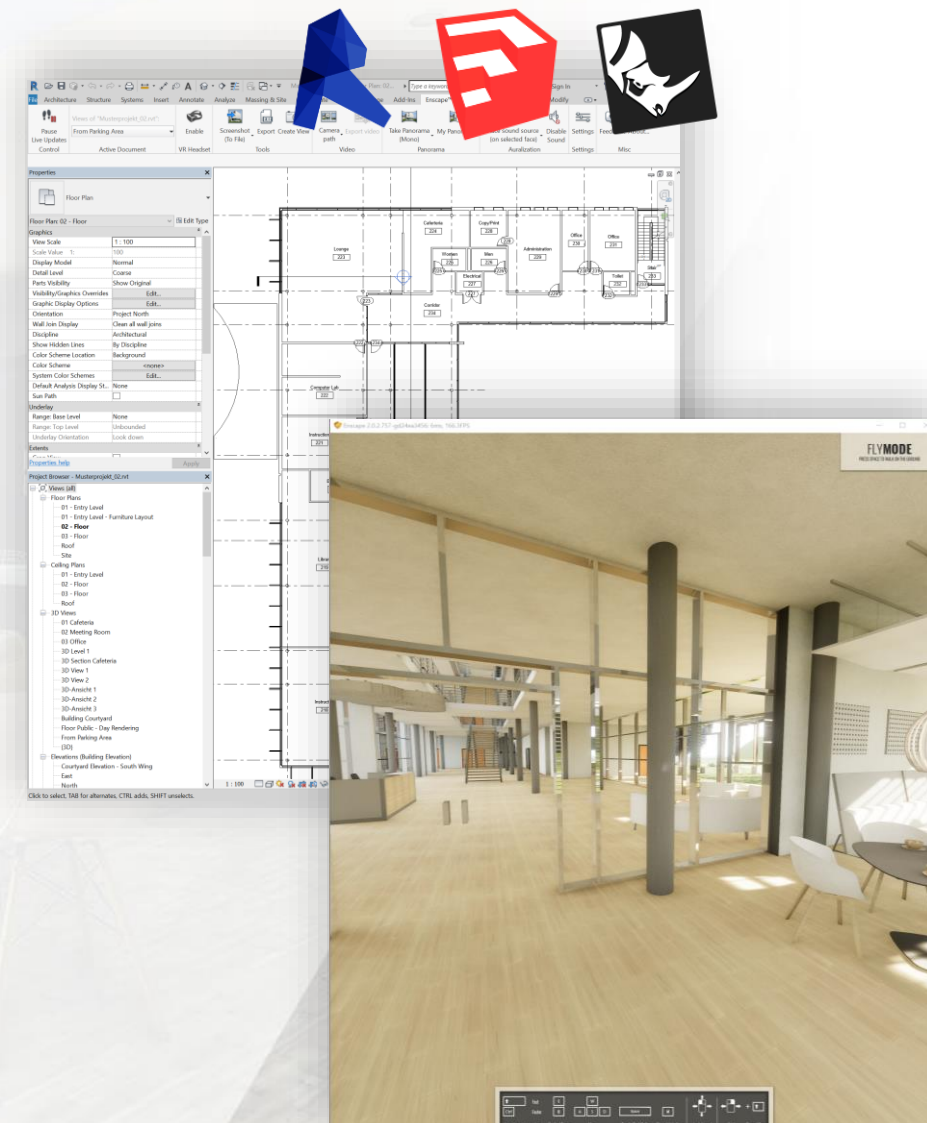
Unbiased quality



How can we create an intermediate, scalable solution?

What is Enscape?

- Real-time rendering plugin for architectural construction programs
 - Used by 78 of the Top 100 architecture companies world wide
- The construction tool serves as the editor
- WYSIWYG for CAD model changes
- Huge projects, massive polycount, unprepared for rendering
- Offline quality without precalculation









Problems of real-time GI

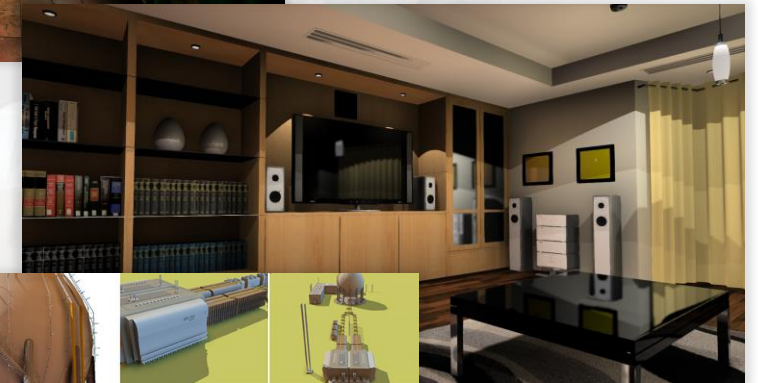
- ❖ Light leaking
- ❖ Offscreen reflections
- ❖ Incompatible to complex BRDFs
- ❖ Indirect shadows
- ❖ High frequency indirect lighting
- ❖ Precomputation necessary
- ❖ Do not scale to ground truth
- ❖ Scene size assumptions



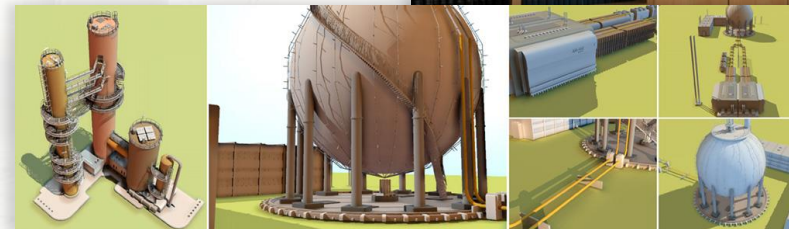
Voxel Cone Tracing, Crassin



LPVs, Kaplanyan



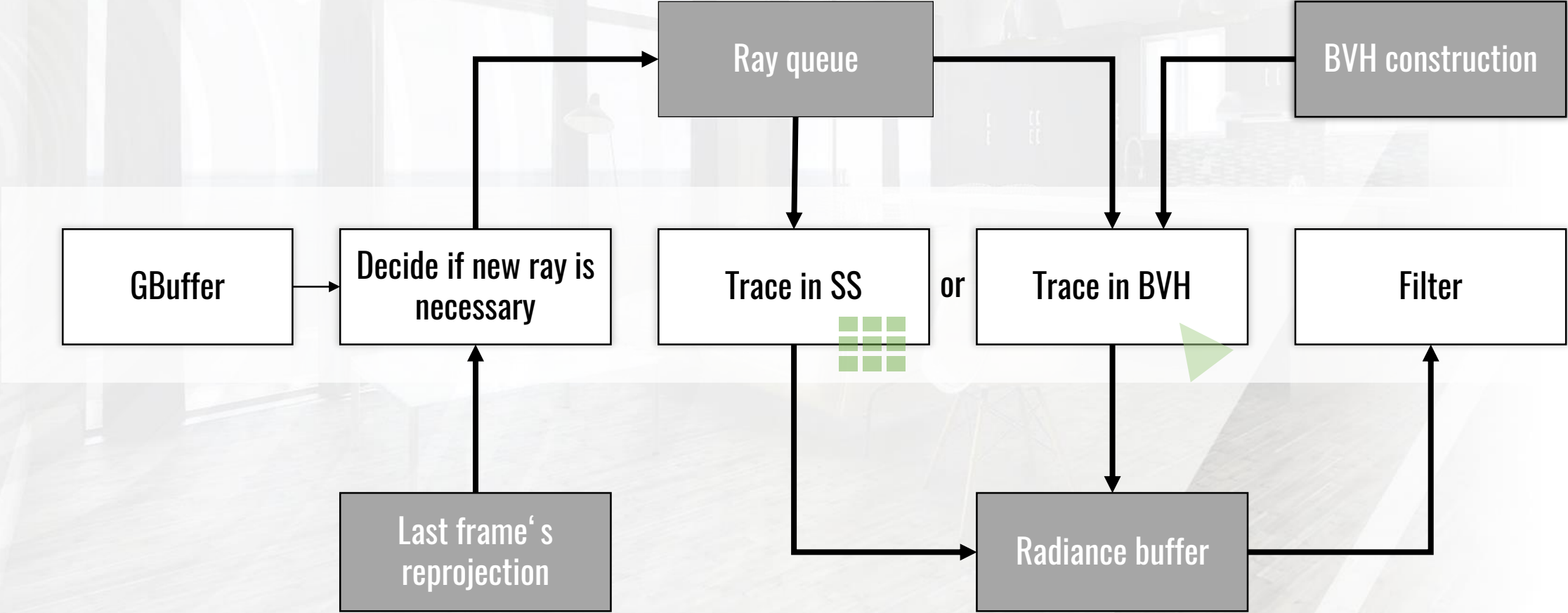
Light Field Probes, McGuire



SSDO, Ritschel

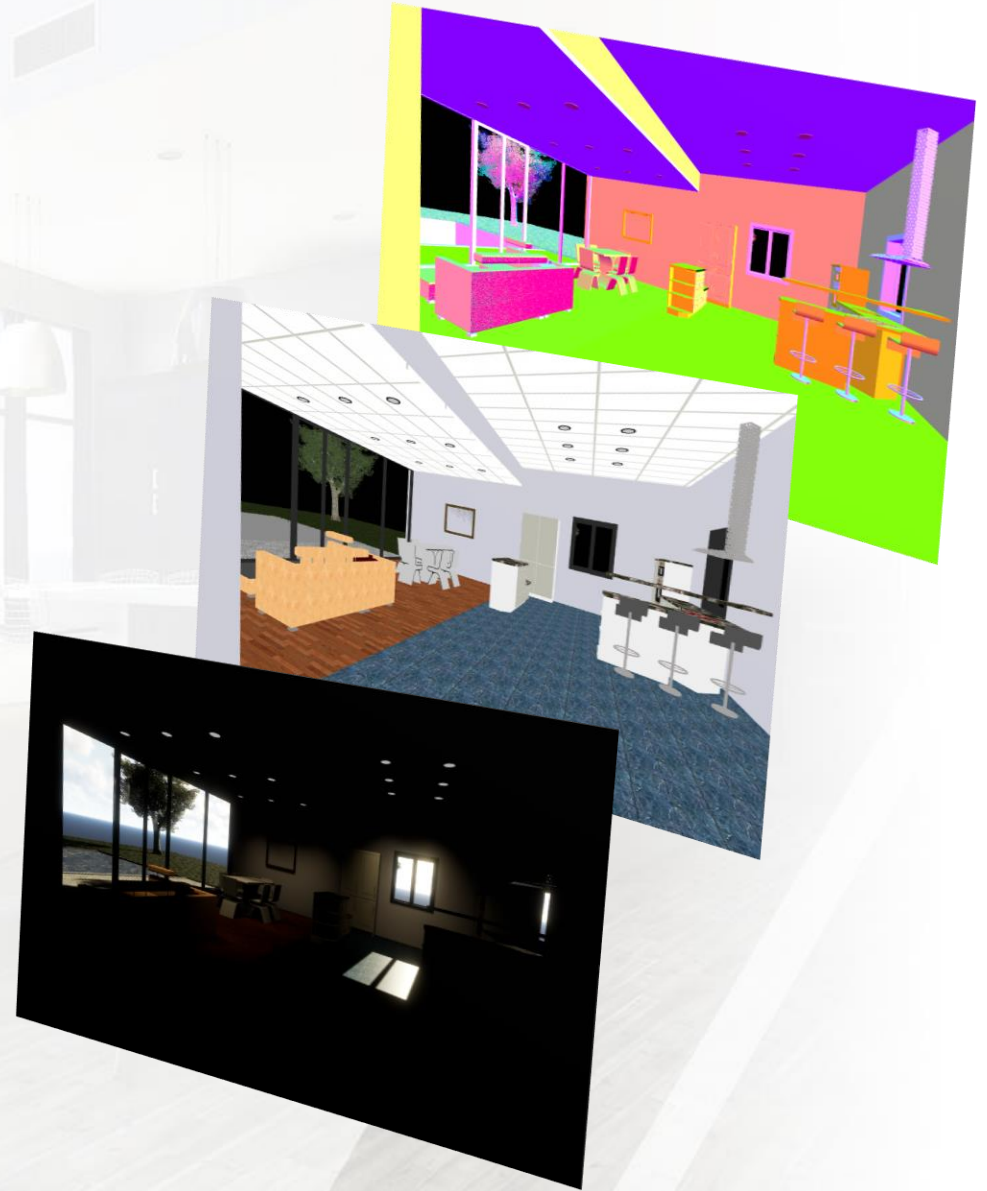
Deferred Path Tracing

Overview for diffuse and specular



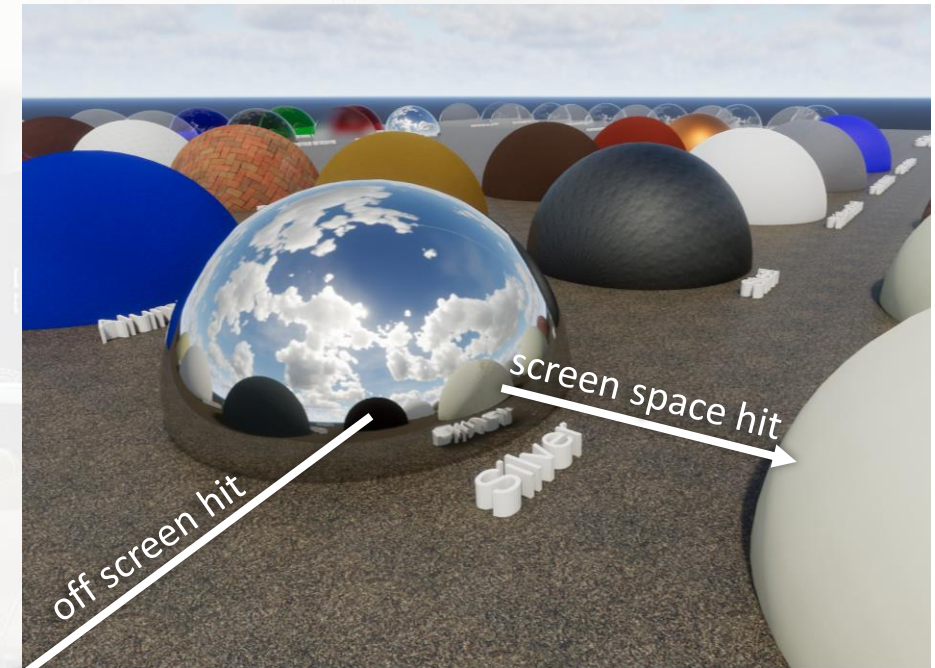
Hybrid Approach

- ♦ Rasterization techniques where possible
 - ♦ Gbuffer
 - ♦ Shadow Maps
 - ♦ Post processing techniques
- ♦ Complements rasterization rendering
- ♦ Eliminates the need for primary rays



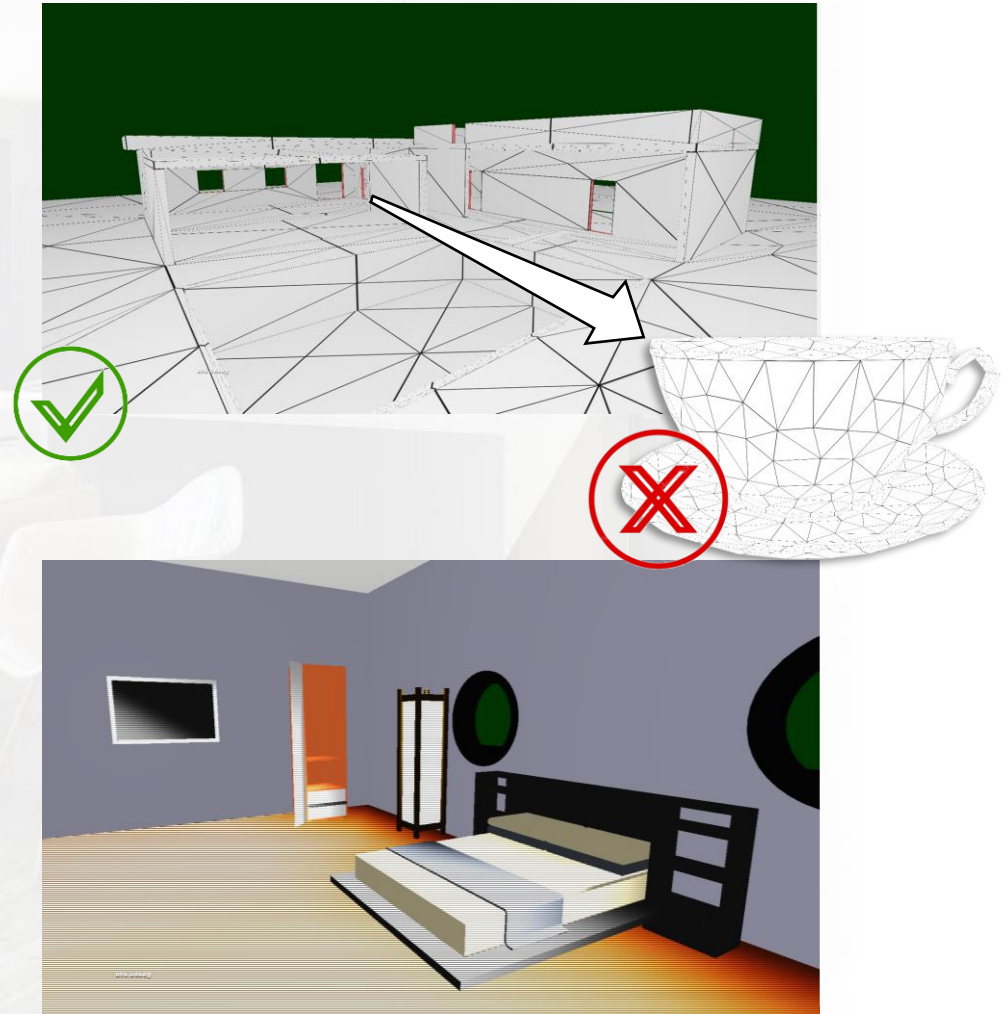
Ray traversal scheme

- ♦ Sample arbitrary *BRDF* with N number of rays per fragment
- ♦ Check for ray in screen space
 - ♦ If not, append to queue and traverse *BVH* later
- ♦ Temporally accumulate per pixel radiance and recycle until fragment can't be reused
- ♦ Lazy request of new rays only when sampling density becomes too low



BVH Construction

- ♦ Async *BVH* construction and streaming with *LOD* based on scene location
- ♦ Calculate a score for each object based on its occlusion relevance vs traversal cost
- ♦ Include the best objects until maximum *BVH* size is reached
- ♦ Bake direct artificial light
 - ♦ Evaluate sun shading at traversal time to allow quick time of day changes



Sample Accumulation

- ✦ Calculate desired *spp* based on albedo, material parameters and direct light amount
 - ✦ Skip ray if *spp* is sufficient
- ✦ Use variable filter kernel depending on number of existing samples in sample accumulation buffer
- ✦ Keep accumulating for a while when camera stops before idling
 - ✦ Add more *spp* for powerful machines

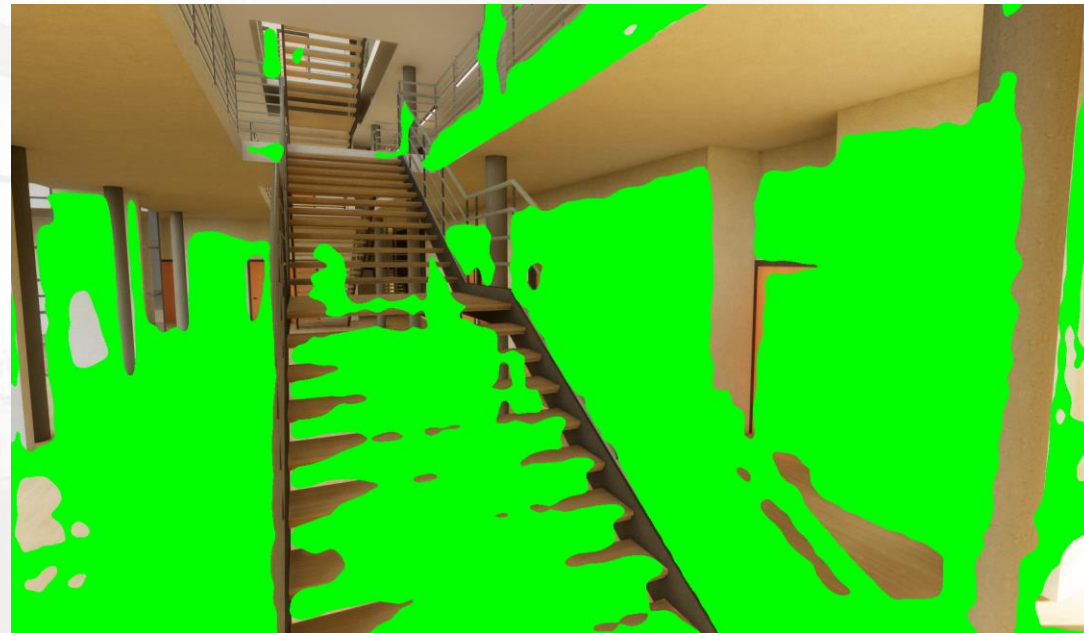


Sample Accumulation

- ♦ Variance tile-based over the whole image to stop accumulating where we reached a sufficient noise level

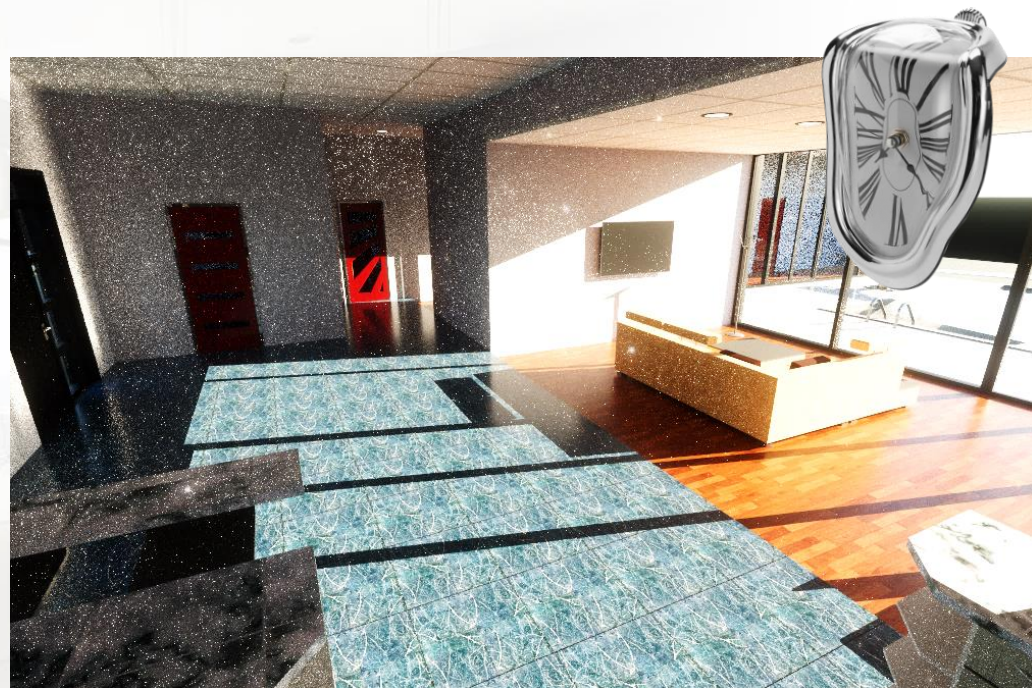
$$\sigma = \sqrt{avg^2 - avgSqr}$$

- ♦ Bicubic tile filtering for stenciling and erosion mask for passes that are the source of a non-local filter kernel



Ground Truth Comparison

- ♦ Reliable in-app debug validation of indirect lighting results
- ♦ *CPU* based classic path tracer
- ♦ Same shading models as real-time renderer
- ♦ Outputs an *HDR* target which is then injected into the normal post processing chain
 - ♦ Allows to toggle results immediately



Ground Truth Comparison



Ground Truth



Enscape

Ground Truth Comparison



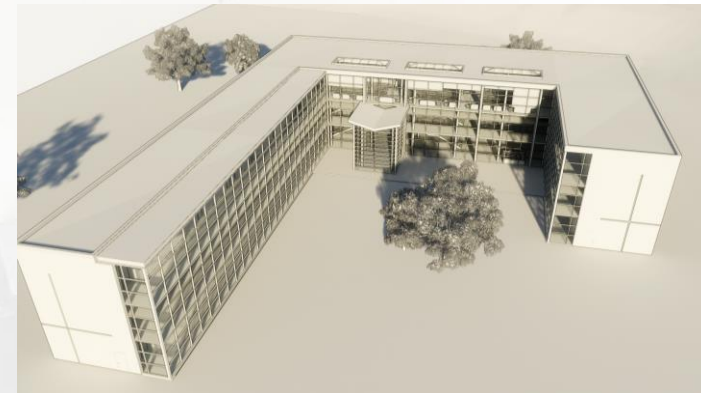
Ground Truth



Enscape

Results

- Reaches stable real-time framerates
- Scalable for low spec machines & *VR (spp, range)* up to ground truth
- Async *BVH* Update every 2 seconds
- Scales well even to huge architectural projects
- Complete diffuse & specular indirect light paths except for caustics and indirect specular
- Almost no light leaking. Only cause are missing relevant objects in *BVH*, which is rare.



Pass	ms
GBuffer	0.7
Direct Light	0.6
Transparency	1.8
Diffuse Rays	1.4
Diffuse Filtering	2.5
Specular Rays	1.6
Specular Filtering	0.3
Post Processing	1.6
Total	10.5

Average Timings @1080p on GTX1080

Thank you!

- ◆ Contact us for licensing or API
- ◆ We're hiring for our office in Karlsruhe, Germany



www.enscape3d.com