LUND UNIVERSITY

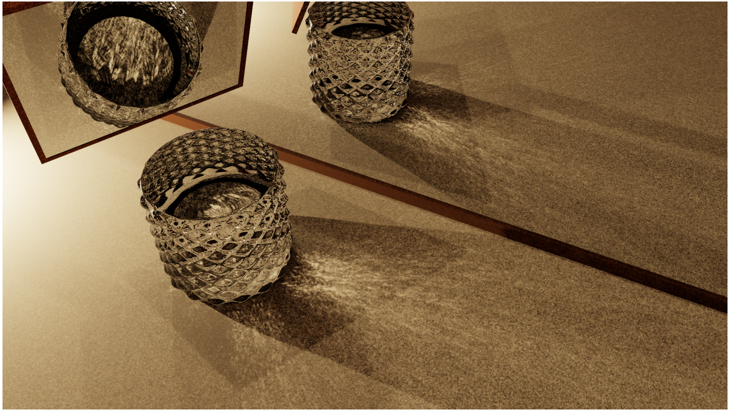
LTH
FACULTY OF ENGINEERING

Real-time Rendering of Indirectly Visible Caustics

Pierre Moreau and Michael Doggett
Lund University, Sweden

Problem: Rendering indirectly-visible caustics

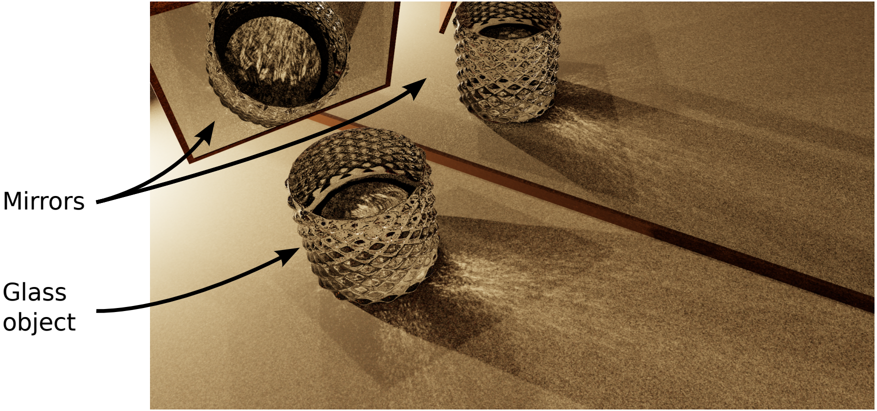
Indirectly-visible caustics: caustics visible to the user via specular or glossy bounces.



Reference

Problem: Rendering indirectly-visible caustics

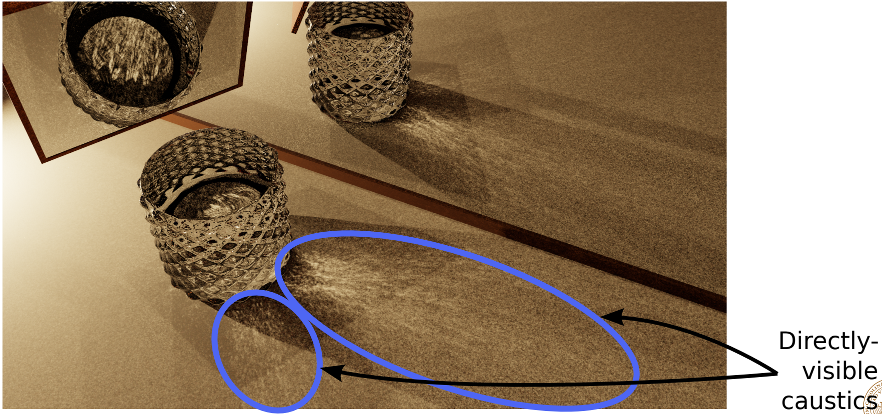
Indirectly-visible caustics: caustics visible to the user via specular or glossy bounces.



Reference

Problem: Rendering indirectly-visible caustics

Indirectly-visible caustics: caustics visible to the user via specular or glossy bounces.

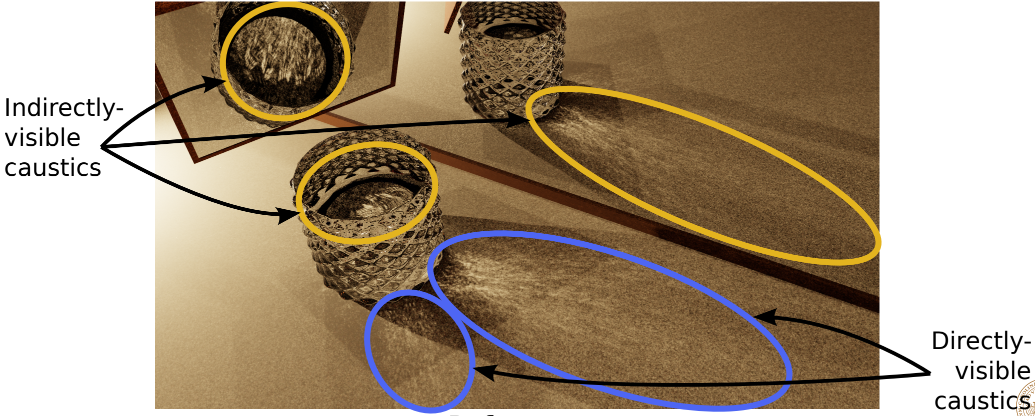


Reference

LUND UNIVERSITY

Problem: Rendering indirectly-visible caustics

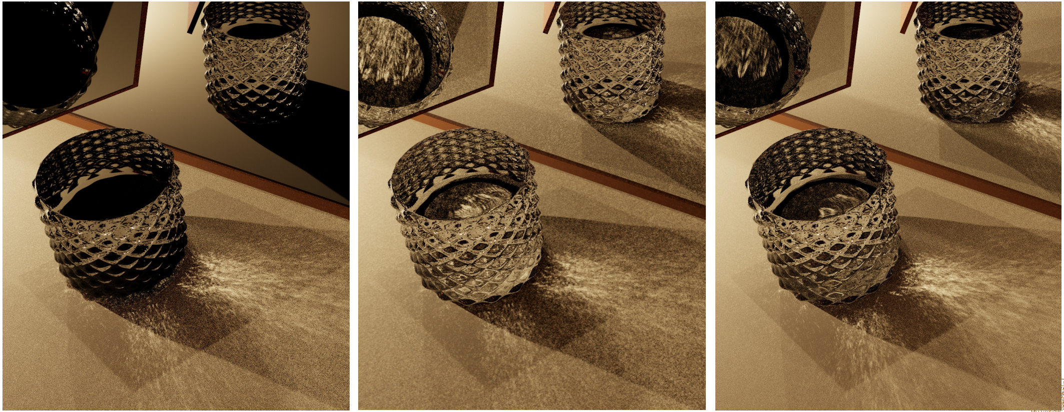
Indirectly-visible caustics: caustics visible to the user via specular or glossy bounces.



Reference

Problem: Rendering indirectly-visible caustics

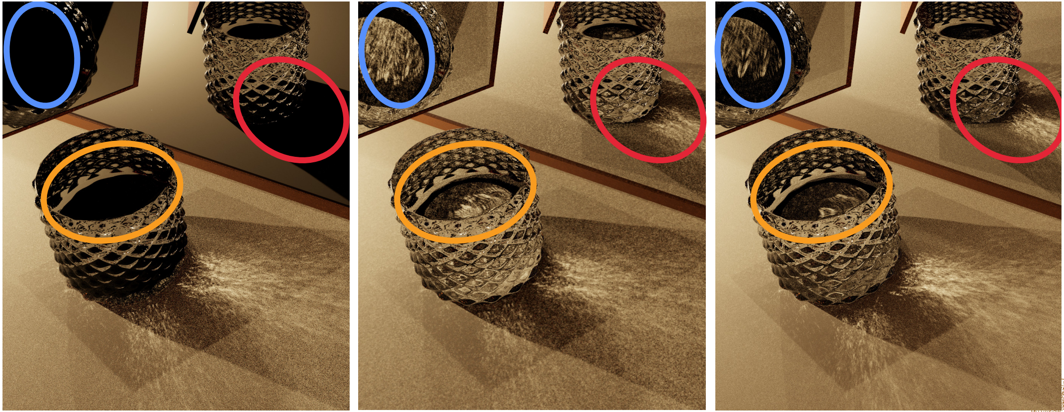
Indirectly-visible caustics: caustics visible to the user via specular or glossy bounces.



Kim, 2019: 1.00 s **Our, temporal: 1.02 s** **Reference: 100k spp**

Problem: Rendering indirectly-visible caustics

Indirectly-visible caustics: caustics visible to the user via specular or glossy bounces.

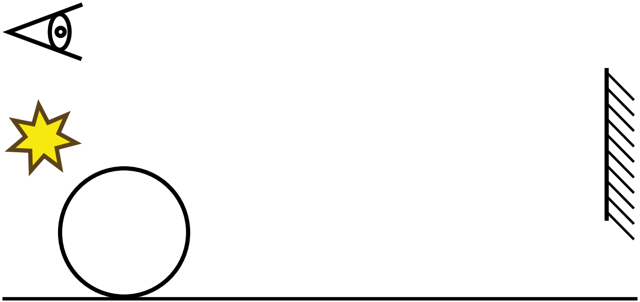


Kim, 2019: 1.00 s **Our, temporal: 1.02 s** **Reference: 100k spp**

Previous work: Screen-based accumulation approaches

A few recent previous work:

- Kim, 2019
- Yang and Ouyang, 2021



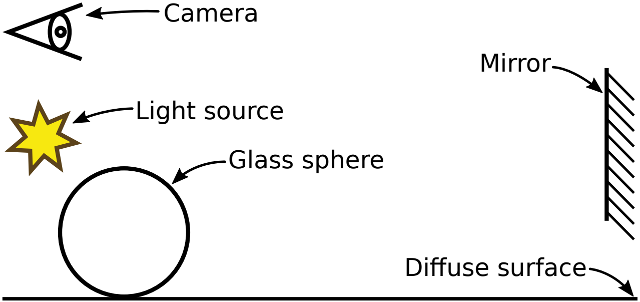
Previous work: Screen-based accumulation approaches

A few recent previous work:

- Kim, 2019
- Yang and Ouyang, 2021

Challenges to overcome:

1. Mirror can not be perfectly specular;



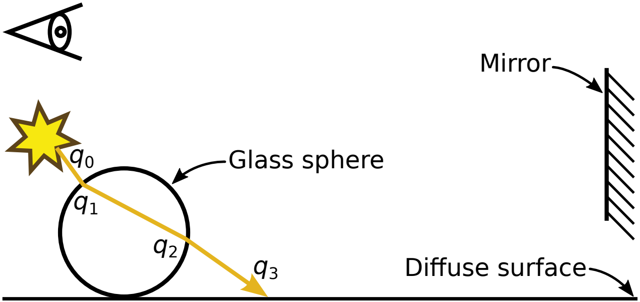
Previous work: Screen-based accumulation approaches

A few recent previous work:

- Kim, 2019
- Yang and Ouyang, 2021

Challenges to overcome:

1. Mirror can not be perfectly specular;



LUND UNIVERSITY

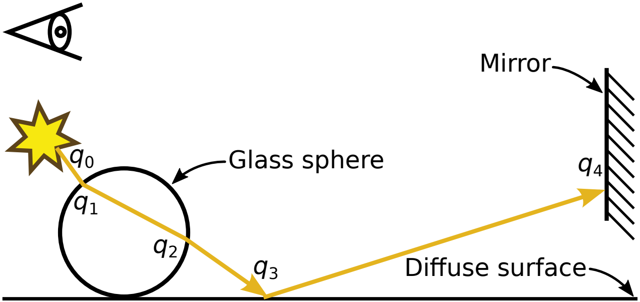
Previous work: Screen-based accumulation approaches

A few recent previous work:

- Kim, 2019
- Yang and Ouyang, 2021

Challenges to overcome:

1. Mirror can not be perfectly specular;
2. Hit mirror when sampling diffuse surface at q_3;



LUND UNIVERSITY

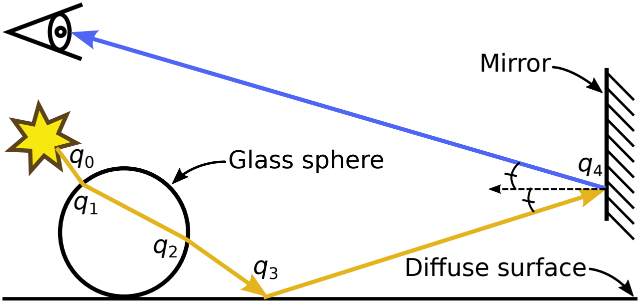
Previous work: Screen-based accumulation approaches

A few recent previous work:

- Kim, 2019
- Yang and Ouyang, 2021

Challenges to overcome:

1. Mirror can not be perfectly specular;
2. Hit mirror when sampling diffuse surface at q_3;
3. Unlikely of being reflected from q_4 towards the eye.



LUND UNIVERSITY

Previous work: Evangelou et al. 2021

Goal: Accelerate radius searches using ray tracing hardware-accelerated GPUs.

Algorithm (applied to photon mapping):

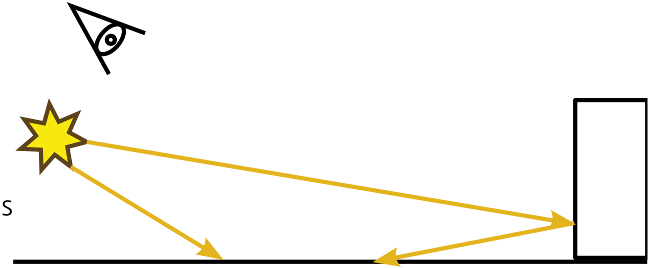
- Build BVH around photons.
- Gather photons by tracing ray, from point of interest, against photon BVH.
- Accumulate the contributions of intersected photons.

Previous work: Evangelou et al. 2021

Goal: Accelerate radius searches using ray tracing hardware-accelerated GPUs.

Algorithm (applied to photon mapping):

- Build BVH around photons.
- Gather photons by tracing ray, from point of interest, against photon BVH.
- Accumulate the contributions of intersected photons.

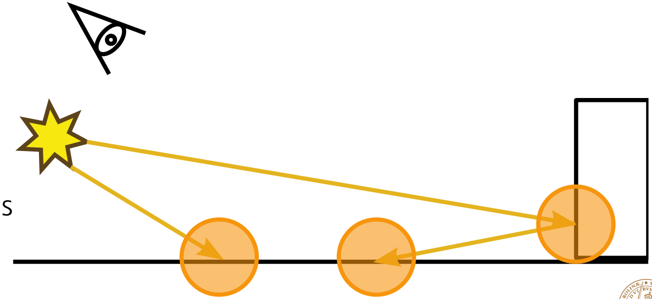


Previous work: Evangelou et al. 2021

Goal: Accelerate radius searches using ray tracing hardware-accelerated GPUs.

Algorithm (applied to photon mapping):

- Build BVH around photons.
- Gather photons by tracing ray, from point of interest, against photon BVH.
- Accumulate the contributions of intersected photons.

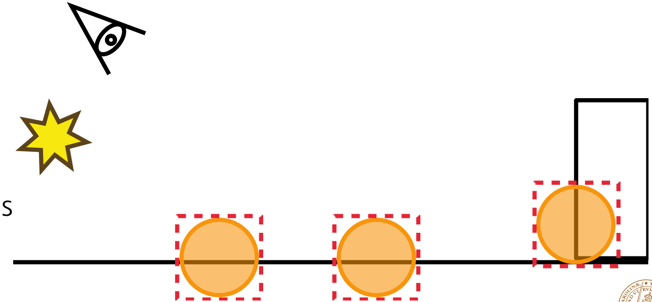


Previous work: Evangelou et al. 2021

Goal: Accelerate radius searches using ray tracing hardware-accelerated GPUs.

Algorithm (applied to photon mapping):

- Build BVH around photons.
- Gather photons by tracing ray, from point of interest, against photon BVH.
- Accumulate the contributions of intersected photons.



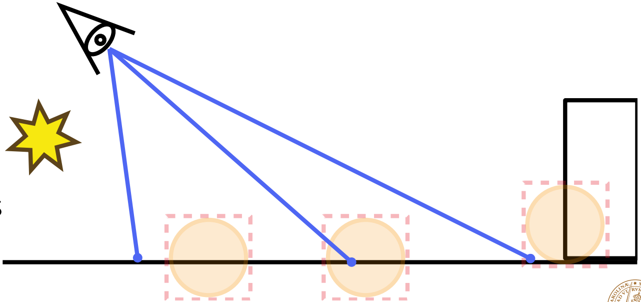
LUND UNIVERSITY

Previous work: Evangelou et al. 2021

Goal: Accelerate radius searches using ray tracing hardware-accelerated GPUs.

Algorithm (applied to photon mapping):

- Build BVH around photons.
- Gather photons by tracing ray, from point of interest, against photon BVH.
- Accumulate the contributions of intersected photons.

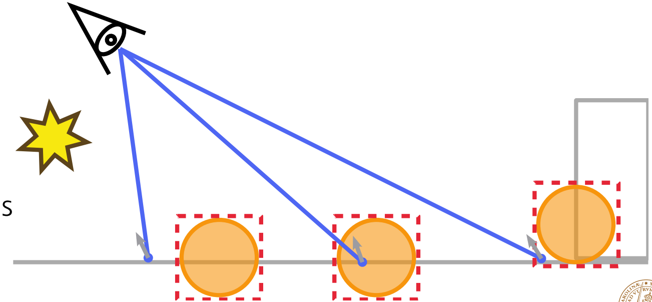


Previous work: Evangelou et al. 2021

Goal: Accelerate radius searches using ray tracing hardware-accelerated GPUs.

Algorithm (applied to photon mapping):

- Build BVH around photons.
- Gather photons by tracing ray, from point of interest, against photon BVH.
- Accumulate the contributions of intersected photons.



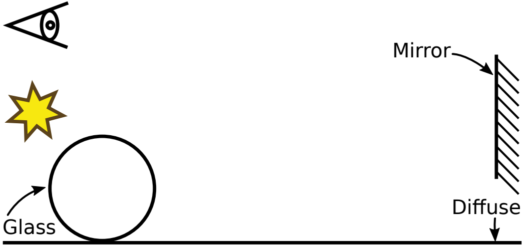
Our Approach

Our approach: Basic algorithm

LUND UNIVERSITY

Our approach: Basic algorithm

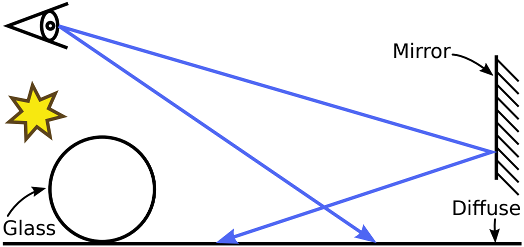
1. *Identify collection points*
   - Trace one camera sub-path per pixel.
   - Create one collection point at first diffuse hit.
   - Collection point sized using ray cones (Akenine-Möller et al., 2021).
   - Save throughput leading to collection point.



LUND UNIVERSITY

Our approach: Basic algorithm

1. *Identify collection points*
   - Trace one camera sub-path per pixel.
   - Create one collection point at first diffuse hit.
   - Collection point sized using ray cones (Akenine-Möller et al., 2021).
   - Save throughput leading to collection point.

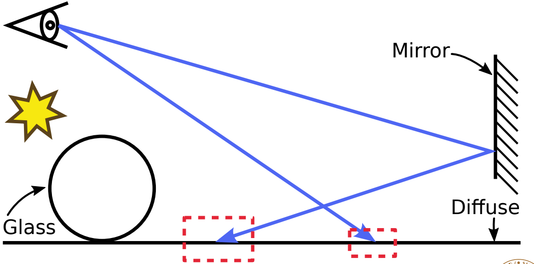


LUND UNIVERSITY

Our approach: Basic algorithm

Identify collection points → Build collection point AS → Trace light sub-paths → Resolve

1. *Identify collection points*
   - Trace one camera sub-path per pixel.
   - Create one collection point at first diffuse hit.
   - Collection point sized using ray cones (Akenine-Möller et al., 2021).
   - Save throughput leading to collection point.



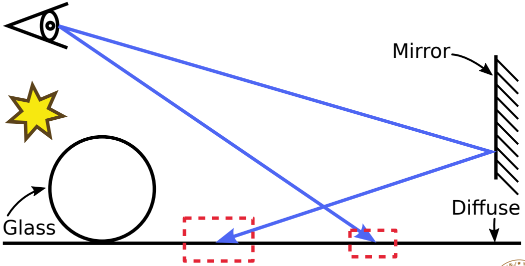
Lund University

Our approach: Basic algorithm

Identify collection points → **Build collection point AS** → Trace light sub-paths → Resolve

2. *Build collection point AS*
(modified Evangelou et al., 2021)

- One AABB per collection point.
- Single BLAS around all AABBs.
- BVH built using the DXR API.
- BVH rebuilt every frame.

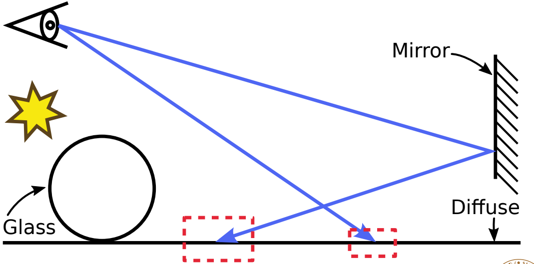


Lund University

Our approach: Basic algorithm

Identify collection points → Build collection point AS → **Trace light sub-paths** → Resolve

3. *Trace light sub-paths*
   - (Light sampling strategy orthogonal to approach)
   - Trace until maximum depth.
   - If sub-path intersected a specular material, at each following hit:
     - Locate nearby collection points.
     - Accumulate radiant intensity at each collection point.

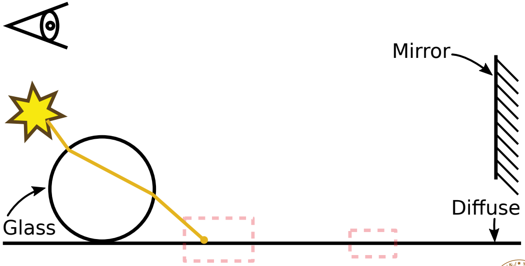


LUND UNIVERSITY

Our approach: Basic algorithm

Identify collection points → Build collection point AS → **Trace light sub-paths** → Resolve

3. *Trace light sub-paths*
   - (Light sampling strategy orthogonal to approach)
   - Trace until maximum depth.
   - If sub-path intersected a specular material, at each following hit:
     - Locate nearby collection points.
     - Accumulate radiant intensity at each collection point.

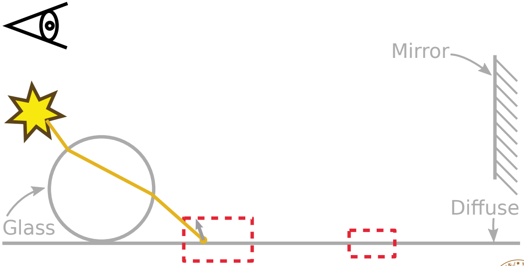


LUND UNIVERSITY

Our approach: Basic algorithm

Identify collection points → Build collection point AS → **Trace light sub-paths** → Resolve

3. *Trace light sub-paths*
   - (Light sampling strategy orthogonal to approach)
   - Trace until maximum depth.
   - If sub-path intersected a specular material, at each following hit:
     - Locate nearby collection points.
     - Accumulate radiant intensity at each collection point.



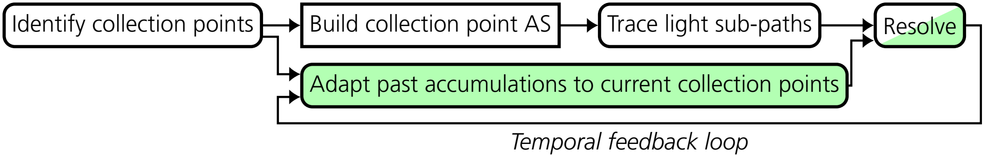
LUND UNIVERSITY

Our approach: Basic algorithm

4. *Resolve*
   - Transform accumulated radiant intensity into radiance.
   - Apply throughput of camera sub-path to radiance from light sub-paths.

Our approach: Temporal version

Our approach: Temporal version



Collection point reuse:

- Locate collection points from frame $i-1$ overlapping those from frame i.
- Store per pixel the weighted average of overlapping predecessors.

Modifications to *Resolve*:

- Exponential moving average between past weighted average and current contributions.

Results

Results: Stable performance during animations

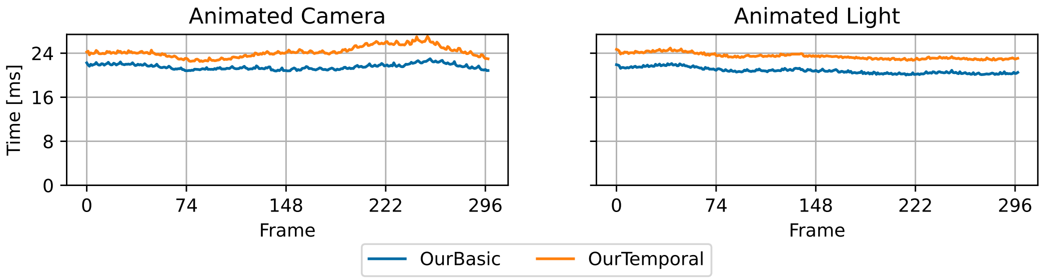


Figure: Total frame time, including G-buffer creation, path tracing and filtering.
Slightly glossy mirrors, ≤ 5 bounces with 1024^2 light paths, at 1080p on a NVIDIA RTX 3080.

LUND UNIVERSITY

Results: Varying performance overheads

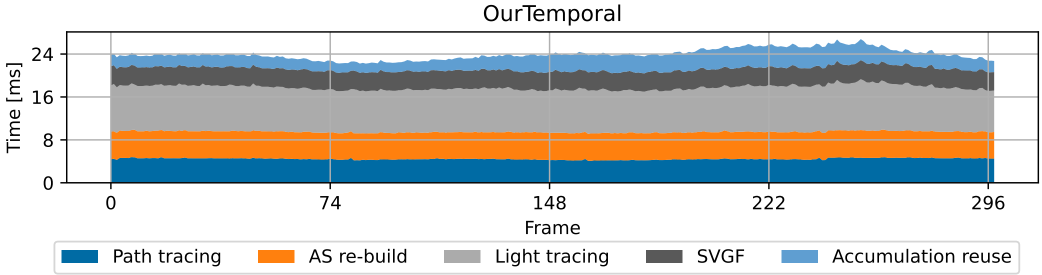
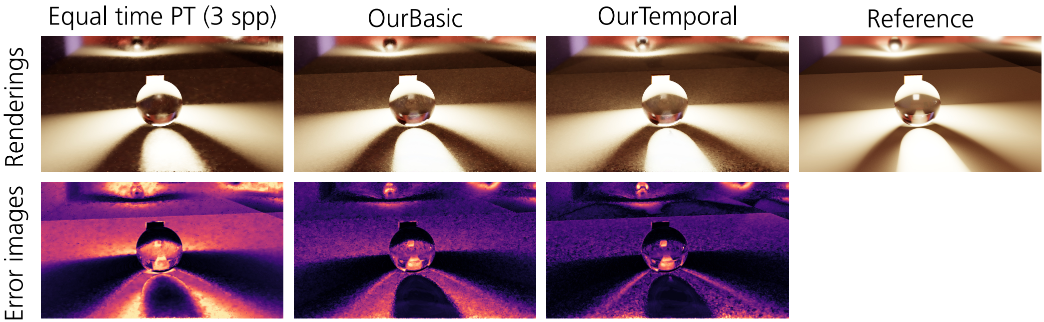


Figure: Frame-time breakdown, in the *animated camera* scene, for each main steps.
Slightly glossy mirrors, ≤ 5 bounces with 1024^2 light paths, at 1080p on a NVIDIA RTX 3080.

Results: Temporal sampling improves image quality



0.363	0.229	0.179	ꟻLIP[1] score (lower is better)

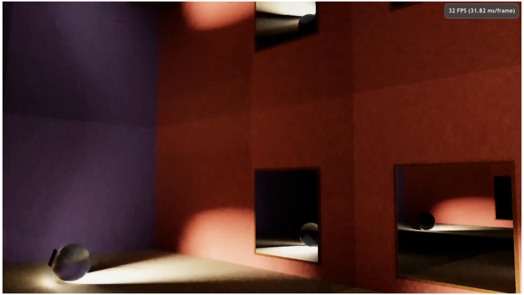
[1] Andersson et al., 2020

LUND UNIVERSITY

Results: Rendering indirectly-visible caustics in action

Results: Rendering indirectly-visible caustics in action

(a) Kim, 2019: $10 \cdot 1024^2$ light paths



(b) OurTemporal: 1024^2 light paths

Figure: Comparison against previous work in the *animated light* scene.
Specular mirrors, ≤ 5 bounces, at 1080p on a NVIDIA RTX 3080.

Conclusion

Conclusion

Contributions:

- Enables indirectly-visible caustics at real-time framerates.
- Supports perfectly-specular and glossy materials.
- Uses recent advances in GPU hardware.
- Reusable caching approach.

Source code available at
`https://github.com/LUGGPublic/Indirectly-Visible-Caustics`.

LUND UNIVERSITY

LTH

FACULTY OF ENGINEERING

References I

- Akenine-Möller, T., Crassin, C., Boksansky, J., Belcour, L., Panteleev, A., and Wright, O.
  Improved shader and texture level of detail using ray cones.
  Journal of Computer Graphics Techniques (JCGT) 10, 1 (Jan. 2021), 1–24.

- Andersson, P., Nilsson, J., Akenine-Möller, T., Oskarsson, M., Åström, K., and Fairchild, M. D.
  ꟻLIP: A Difference Evaluator for Alternating Images.
  Proceedings of the ACM on Computer Graphics and Interactive Techniques 3, 2 (2020), 15:1–15:23.

References II

Evangelou, I., Papaioannou, G., Vardis, K., and Vasilakis, A. A.
Fast radius search exploiting ray tracing frameworks.
Journal of Computer Graphics Techniques (JCGT) 10, 1 (2021), 25–48.

Kim, H.
Caustics using screen-space photon mapping.
In *Ray Tracing Gems: High-Quality and Real-Time Rendering with DXR and Other APIs*, E. Haines and T. Akenine-Möller, Eds. Apress, Berkeley, CA, USA, 2019, p. 543–555.

References III

Yang, X., and Ouyang, Y.
Real-time ray traced caustics.
In *Ray Tracing Gems II: Next Generation Real-Time Rendering with DXR, Vulkan, and OptiX*, A. Marrs, P. Shirley, and I. Wald, Eds. Apress, Berkeley, CA, USA, 2021, p. 469–497.

Appendix

Results: Bistro Interior comparison — Kim, 2019

Figure: *Bistro Interior* rendered by Kim, 2019 in 38 ms.
≤ 6 bounces with $2 \cdot 1024^2$ light paths, at 1080p on a NVIDIA RTX 3080.

LUND UNIVERSITY

Results: Bistro Interior comparison — OurBasic

Figure: *Bistro Interior* rendered by OurBasic in 40 ms.
$\leq$ 6 bounces with 1024^2 light paths, at 1080p on a NVIDIA RTX 3080.

LUND UNIVERSITY

Results: Bistro Interior comparison — OurTemporal

Figure: *Bistro Interior* rendered by OurTemporal in 42 ms.
$\leq$ 6 bounces with 1024^2 light paths, at 1080p on a NVIDIA RTX 3080.

Results: Bistro Interior comparison — Reference

Figure: Reference image for the *Bistro Interior* scene.
$\leq$ 6 bounces with 104k spp, at 1080p on a NVIDIA RTX 3080.

LUND UNIVERSITY