Real-Time Path Tracing and Denoising in *Quake II*

Christoph Schied, PhD Student, Karlsruhe Institute of Technology

Alexey Panteleev, Principal DevTech Engineer, NVIDIA

OUTLINE

Part I: Q2VKPT

- Path Tracing Overview
- Denoising with A-SVGF
- Sampling

Part II: Quake II RTX

- Improvement Process
- Final Renderer Overview

q2vkpt

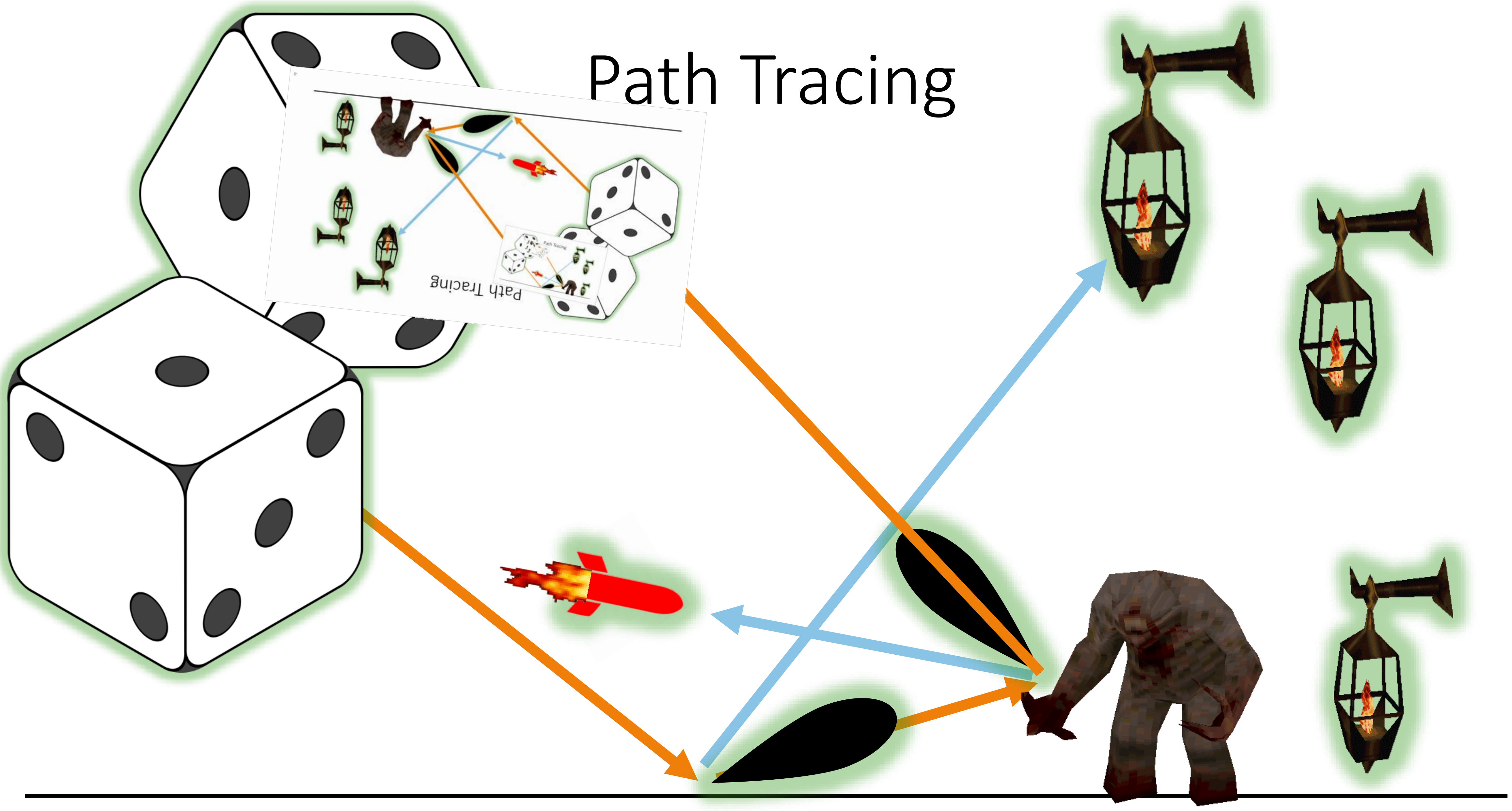
Quake 2
Vulkan
Path Tracer

Christoph Schied

@c_schied

http://brechpunkt.de/q2vkpt

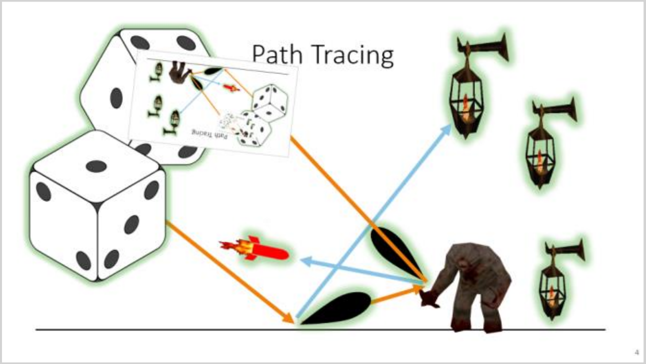
Path Tracing



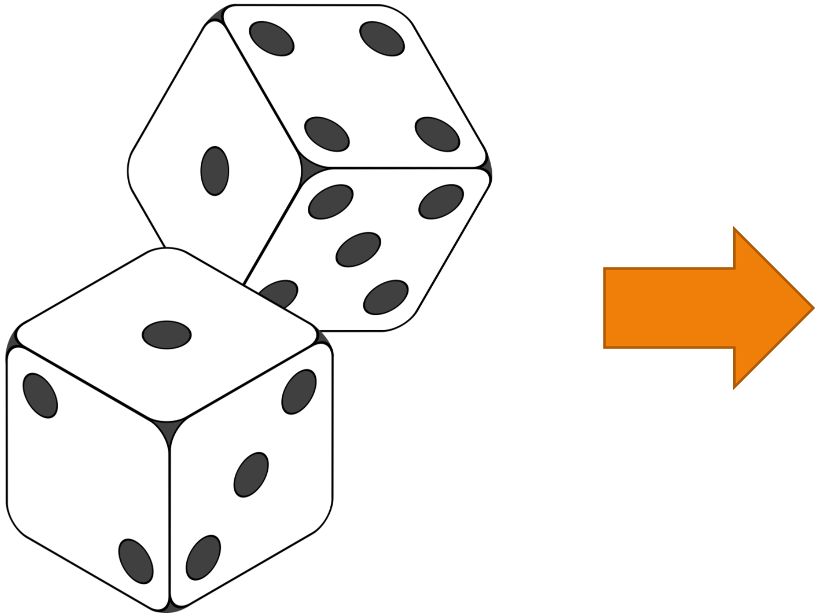
Path Tracing

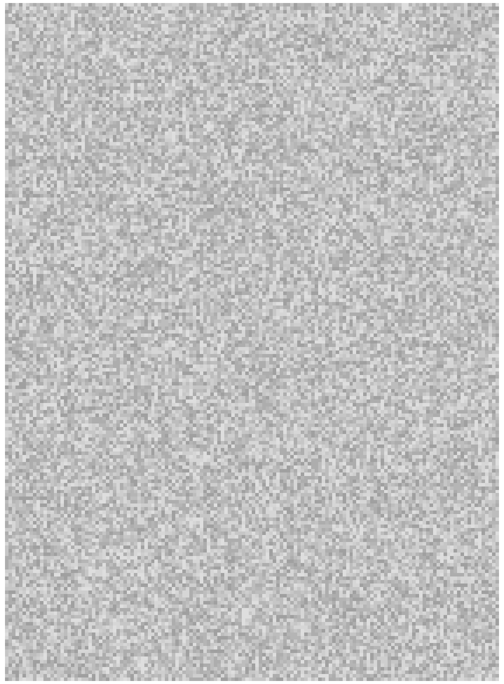
Number of samples, n=1 in q2vkpt

$$L \approx \frac{1}{n}\sum_{i=1}^{n}$$

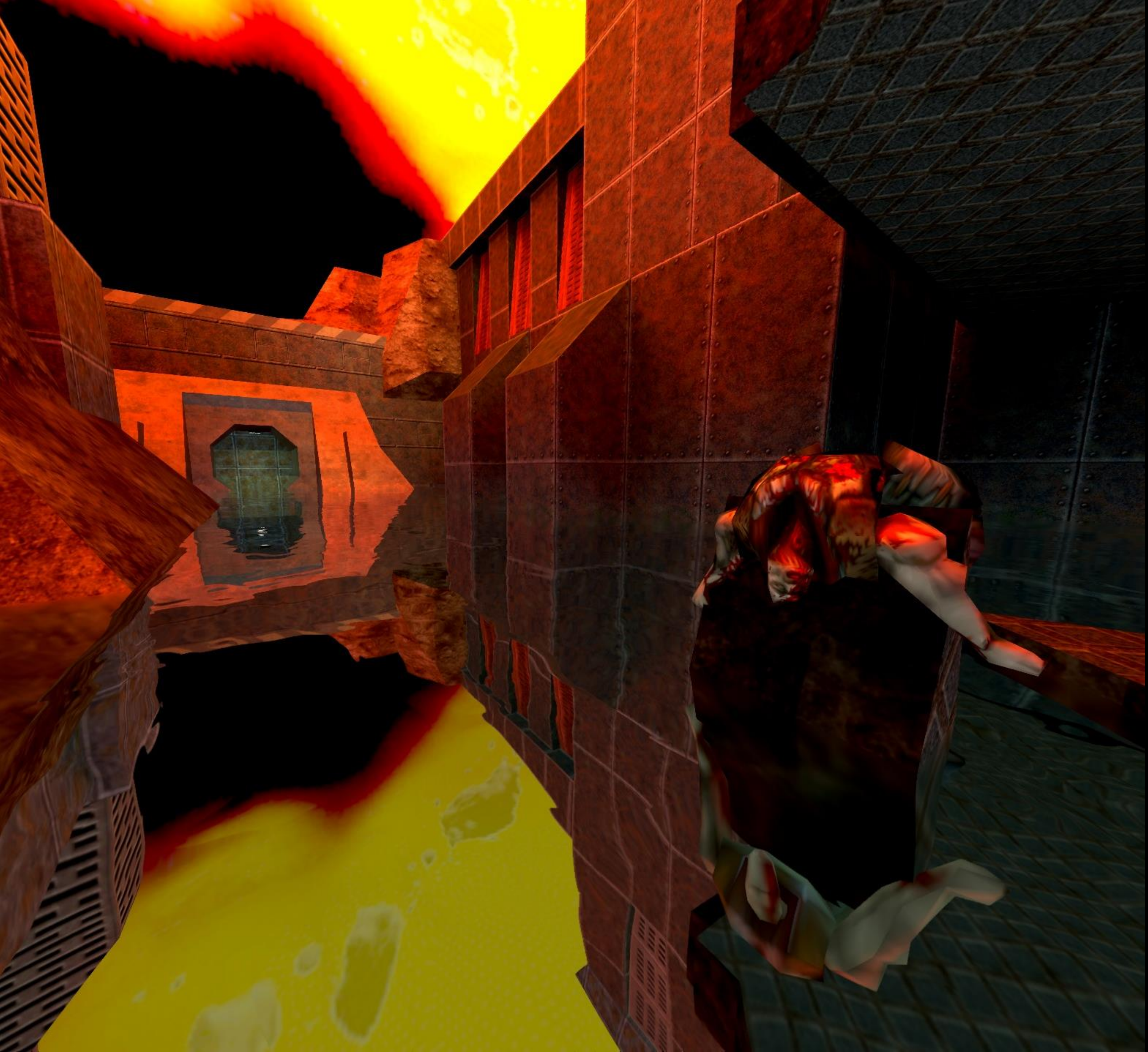


Main challenges





- Better Sampling → less noise
- Denoising



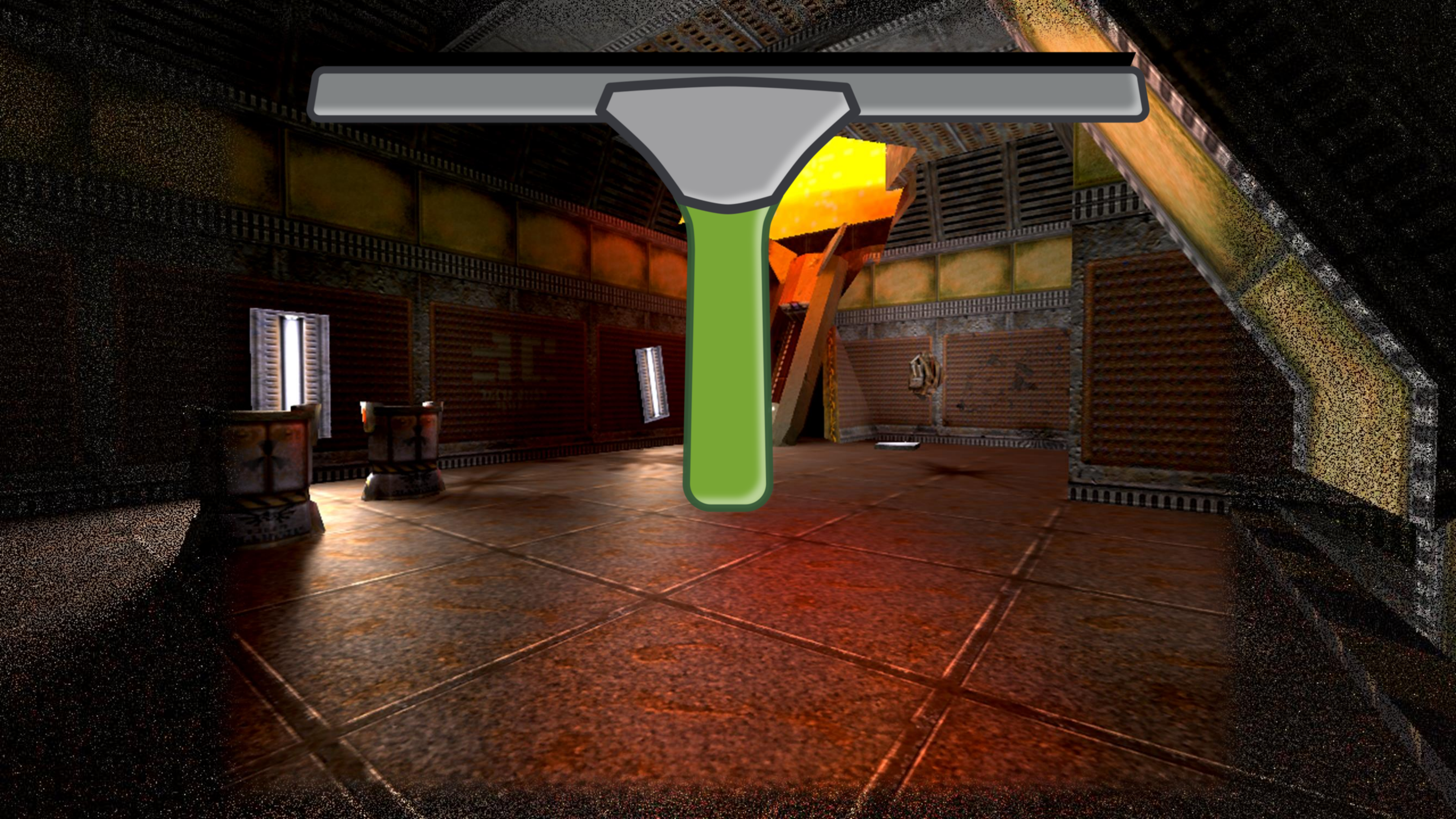
Q2VKPT

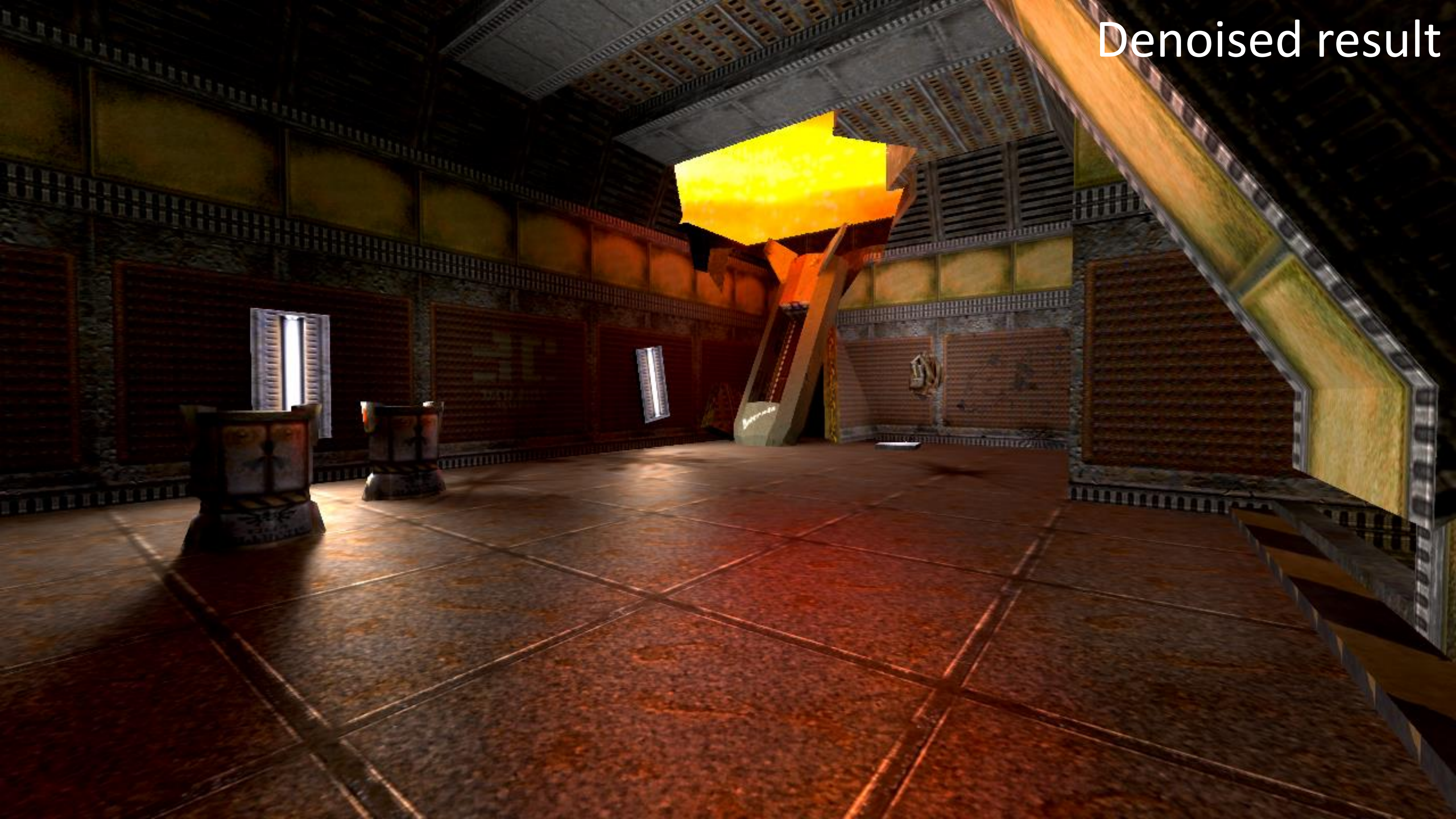
- Research prototype to evaluate current state of real-time path tracing
- Open source
  https://github.com/cschied/q2vkpt
- Entirely raytraced
- Real-time path tracing
  (one indirect bounce)
- C99, Vulkan, GLSL, RTX


93
26
22


frame time 15.24 ms
instance geometry 0.02 ms
bvh update 0.52 ms
asvgf gradient samples 0.28 ms
path tracer 10.89 ms
asvgf full 3.44 ms
asvgf reconstruct gradient 0.27 ms
asvgf temporal 0.68 ms
asvgf atrous 2.16 ms
asvgf taa 0.32 ms
2560x1440, RTX2080 Ti

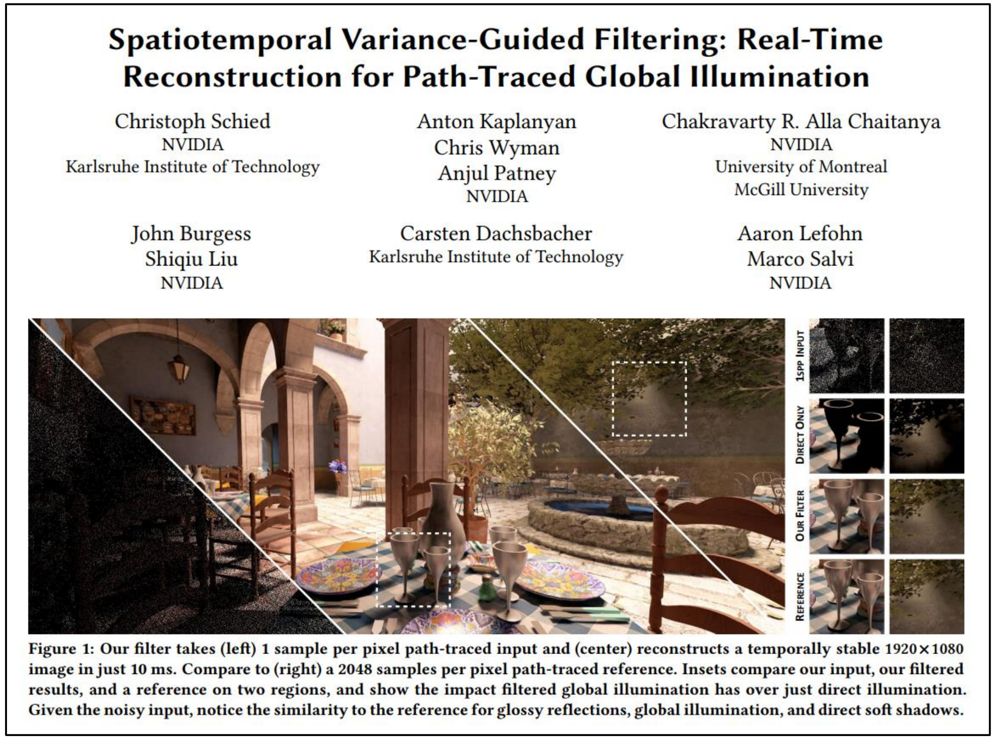
Denoising




Denoised result


Path tracer output
(1spp)

Denoising (A-SVGF)

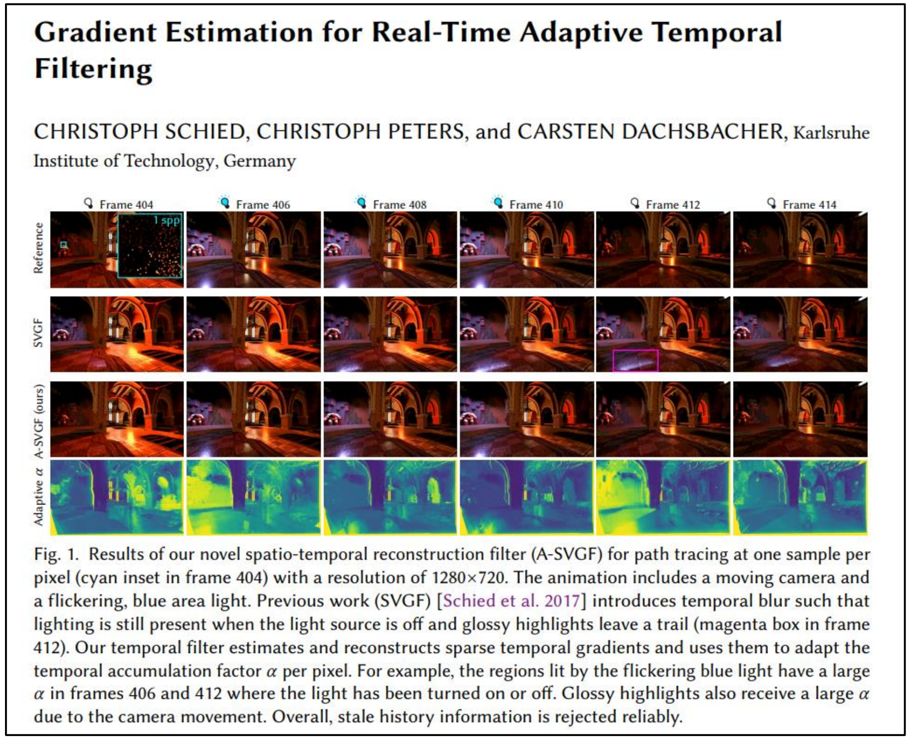


Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination

Christoph Schied, NVIDIA, Karlsruhe Institute of Technology
Anton Kaplanyan, Chris Wyman, Anjul Patney, NVIDIA
Chakravarty R. Alla Chaitanya, NVIDIA, University of Montreal, McGill University
John Burgess, Shiqiu Liu, NVIDIA
Carsten Dachsbacher, Karlsruhe Institute of Technology
Aaron Lefohn, Marco Salvi, NVIDIA

Figure 1: Our filter takes (left) 1 sample per pixel path-traced input and (center) reconstructs a temporally stable 1920×1080 image in just 10 ms. Compare to (right) a 2048 samples per pixel path-traced reference. Insets compare our input, our filtered results, and a reference on two regions, and show the impact filtered global illumination has over just direct illumination. Given the noisy input, notice the similarity to the reference for glossy reflections, global illumination, and direct soft shadows.

+



Gradient Estimation for Real-Time Adaptive Temporal Filtering

CHRISTOPH SCHIED, CHRISTOPH PETERS, and CARSTEN DACHSBACHER, Karlsruhe Institute of Technology, Germany

Fig. 1. Results of our novel spatio-temporal reconstruction filter (A-SVGF) for path tracing at one sample per pixel (cyan inset in frame 404) with a resolution of 1280×720. The animation includes a moving camera and a flickering, blue area light. Previous work (SVGF) [Schied et al. 2017] introduces temporal blur such that lighting is still present when the light source is off and glossy highlights leave a trail (magenta box in frame 412). Our temporal filter estimates and reconstructs sparse temporal gradients and uses them to adapt the temporal accumulation factor α per pixel. For example, the regions lit by the flickering blue light have a large α in frames 406 and 412 where the light has been turned on or off. Glossy highlights also receive a large α due to the camera movement. Overall, stale history information is rejected reliably.

Denoising (A-SVGF)

Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination

Christoph Schied
NVIDIA
Karlsruhe Institute of Technology

Anton Kaplanyan
Chris Wyman
Anjul Patney
NVIDIA

Chakravarty R. Alla Chaitanya
NVIDIA
University of Montreal
McGill University

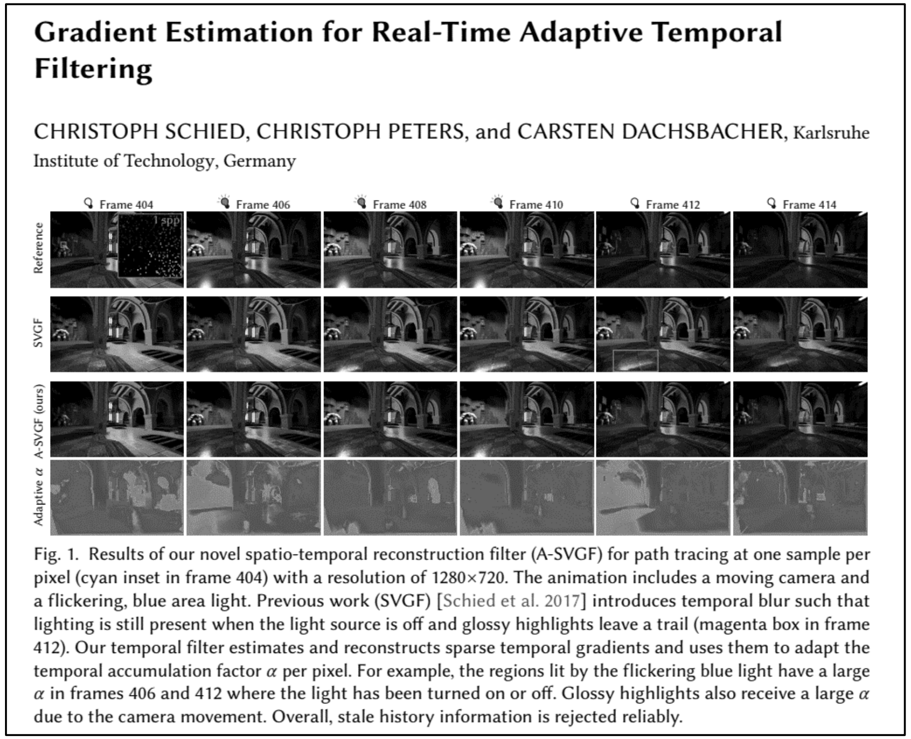
John Burgess
Shiqiu Liu
NVIDIA

Carsten Dachsbacher
Karlsruhe Institute of Technology

Aaron Lefohn
Marco Salvi
NVIDIA

Figure 1: Our filter takes (left) 1 sample per pixel path-traced input and (center) reconstructs a temporally stable 1920×1080 image in just 10 ms. Compare to (right) a 2048 samples per pixel path-traced reference. Insets compare our input, our filtered results, and a reference on two regions, and show the impact filtered global illumination has over just direct illumination. Given the noisy input, notice the similarity to the reference for glossy reflections, global illumination, and direct soft shadows.

+

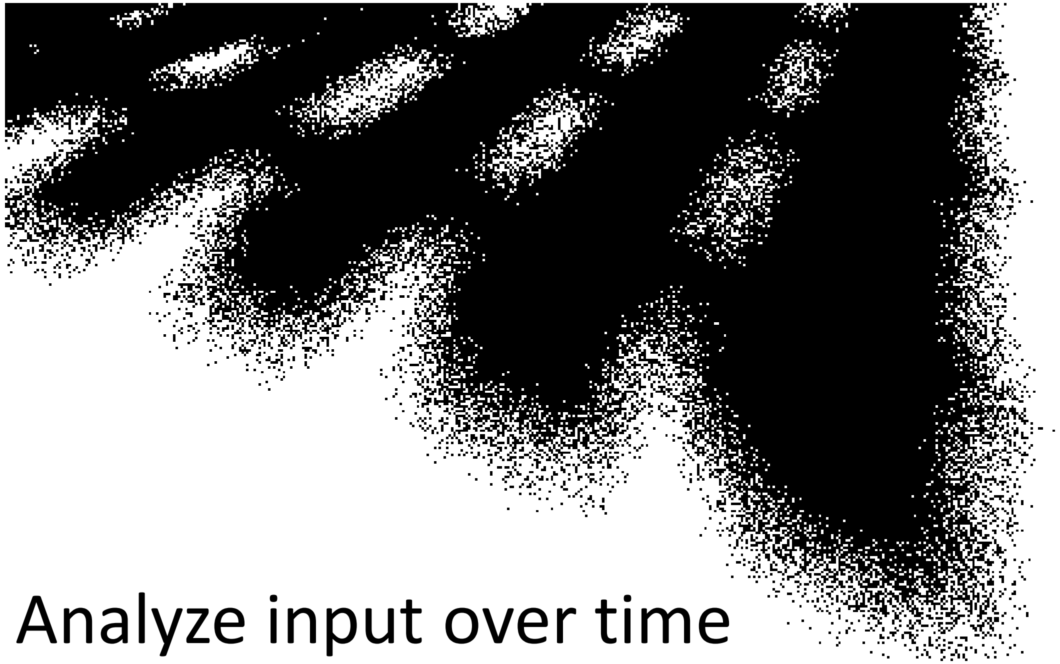


Gradient Estimation for Real-Time Adaptive Temporal Filtering

CHRISTOPH SCHIED, CHRISTOPH PETERS, and CARSTEN DACHSBACHER, Karlsruhe Institute of Technology, Germany

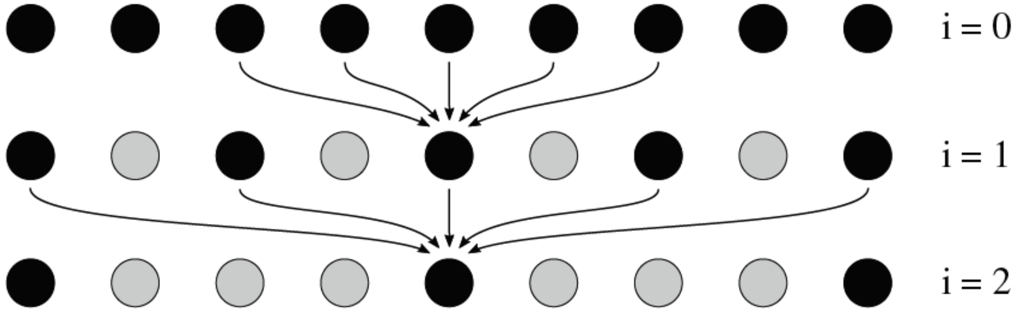
Fig. 1. Results of our novel spatio-temporal reconstruction filter (A-SVGF) for path tracing at one sample per pixel (cyan inset in frame 404) with a resolution of 1280×720. The animation includes a moving camera and a flickering, blue area light. Previous work (SVGF) [Schied et al. 2017] introduces temporal blur such that lighting is still present when the light source is off and glossy highlights leave a trail (magenta box in frame 412). Our temporal filter estimates and reconstructs sparse temporal gradients and uses them to adapt the temporal accumulation factor α per pixel. For example, the regions lit by the flickering blue light have a large α in frames 406 and 412 where the light has been turned on or off. Glossy highlights also receive a large α due to the camera movement. Overall, stale history information is rejected reliably.

Main concepts of SVGF



Analyze input over time

- Temporally unstable → blur more
- Temporally stable → blur less

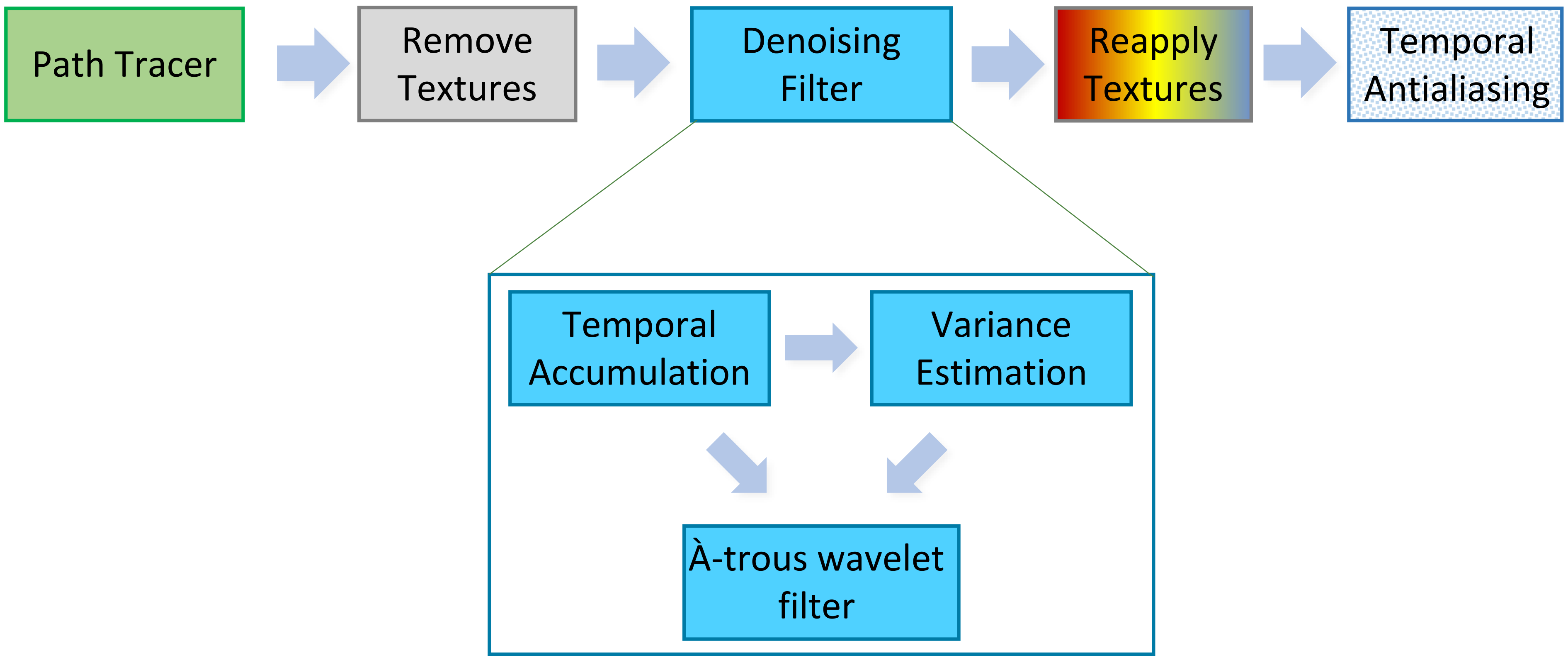


Filter hierarchically, starting small

- Estimate temporal stability after each filter iteration

→ Strong blur more likely in early iterations

SVGF

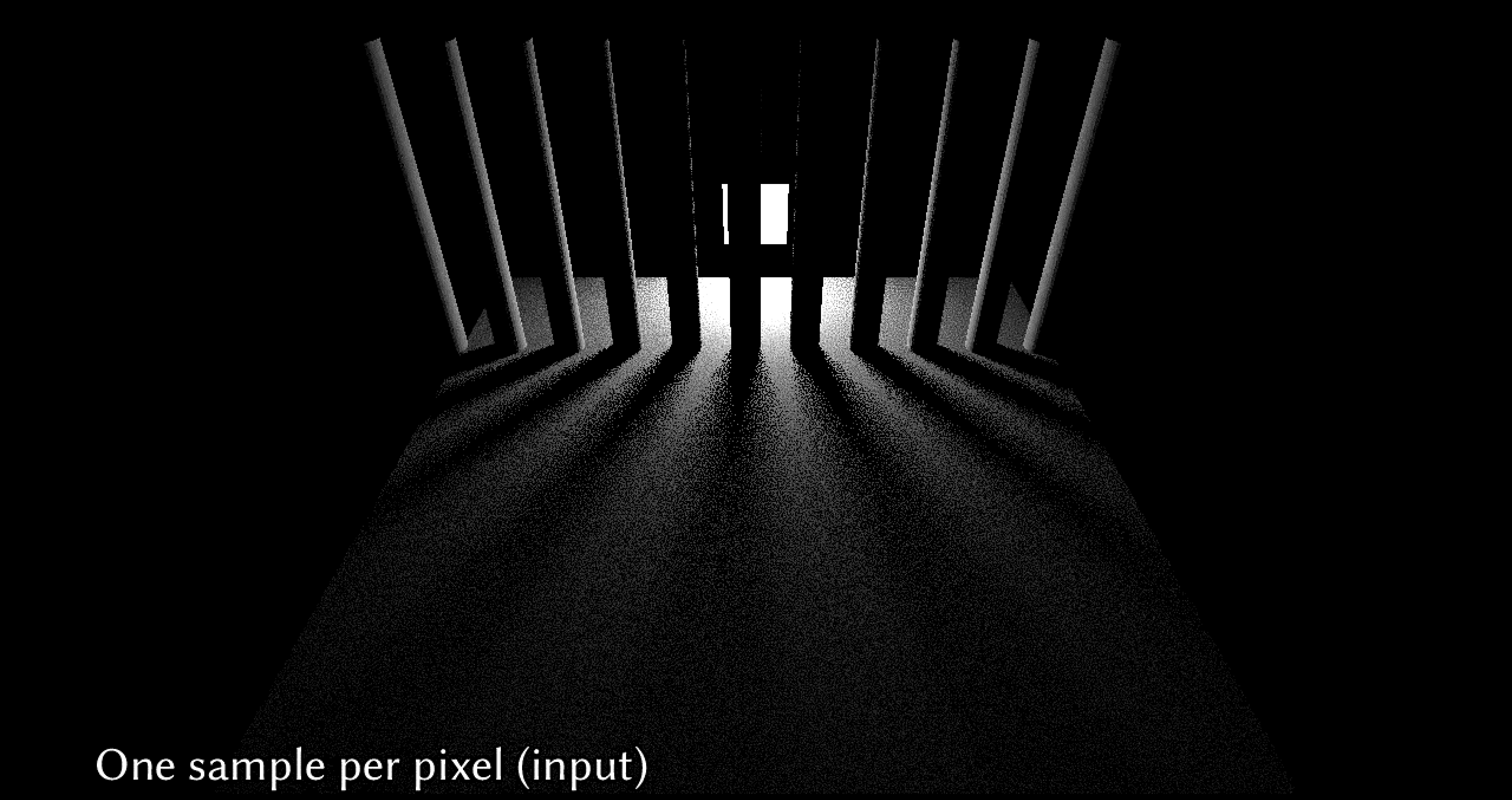

Path Tracer
Remove Textures
Denoising Filter
Reapply Textures
Temporal Antialiasing
Temporal Accumulation
Variance Estimation
À-trous wavelet filter

Edge-avoiding À-trous Wavelets

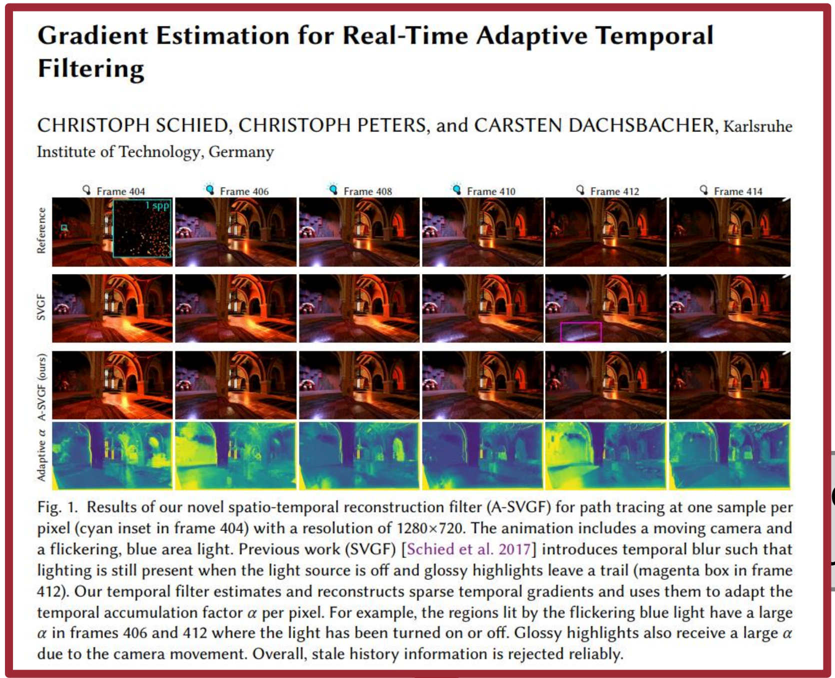
$$\hat{c}_{i+1}(p) = \frac{\sum_{q \in \Omega} h(q) \cdot w(p, q) \cdot \hat{c}_i(q)}{\sum_{q \in \Omega} h(q) \cdot w(p, q)}$$

In q2vkpt:

- 3x3 box kernel
- 5 iterations



One sample per pixel (input)


Gradient Estimation for Real-Time Adaptive Temporal Filtering

CHRISTOPH SCHIED, CHRISTOPH PETERS, and CARSTEN DACHSBACHER, Karlsruhe Institute of Technology, Germany

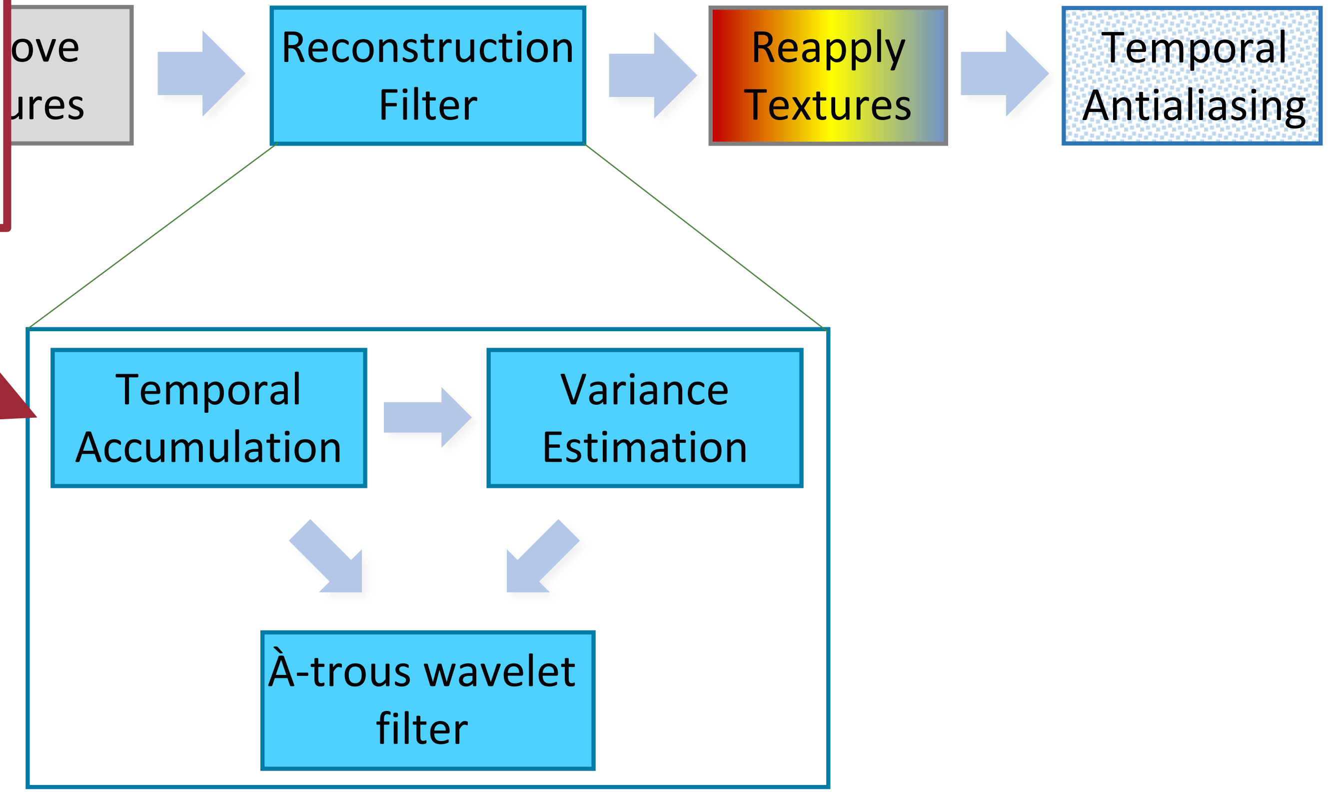
Frame 404 | Frame 406 | Frame 408 | Frame 410 | Frame 412 | Frame 414

Reference | SVGF | A-SVGF (ours) | Adaptive α

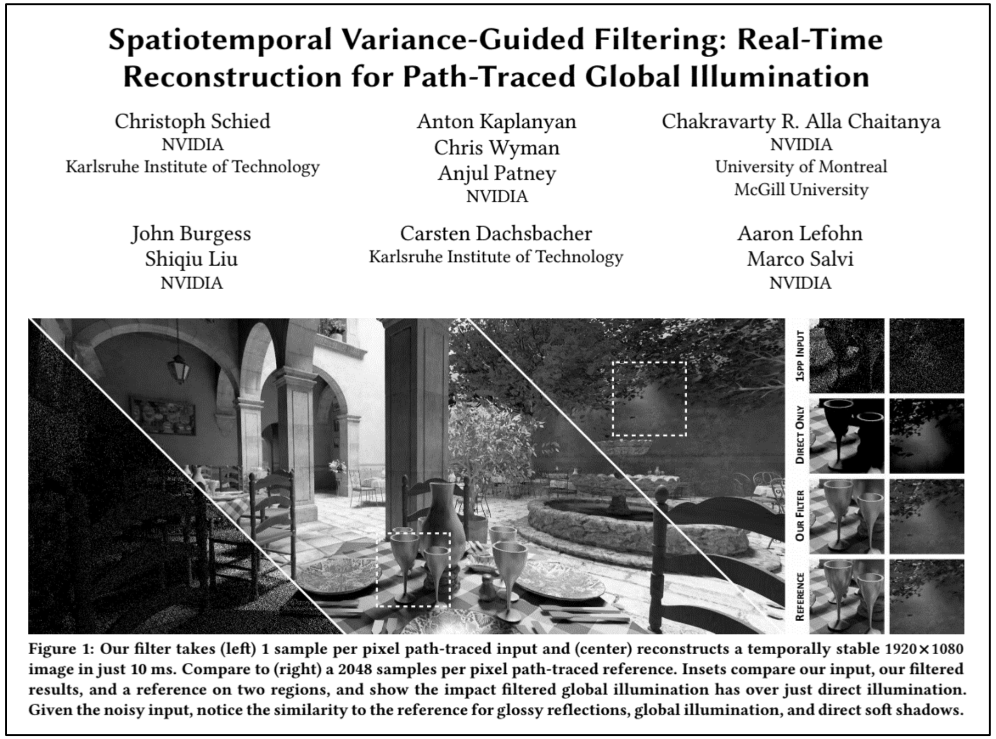
1 spp

Fig. 1. Results of our novel spatio-temporal reconstruction filter (A-SVGF) for path tracing at one sample per pixel (cyan inset in frame 404) with a resolution of 1280×720. The animation includes a moving camera and a flickering, blue area light. Previous work (SVGF) [Schied et al. 2017] introduces temporal blur such that lighting is still present when the light source is off and glossy highlights leave a trail (magenta box in frame 412). Our temporal filter estimates and reconstructs sparse temporal gradients and uses them to adapt the temporal accumulation factor α per pixel. For example, the regions lit by the flickering blue light have a large α in frames 406 and 412 where the light has been turned on or off. Glossy highlights also receive a large α due to the camera movement. Overall, stale history information is rejected reliably.

SVGF



Denoising (A-SVGF)



Spatiotemporal Variance-Guided Filtering: Real-Time Reconstruction for Path-Traced Global Illumination

Christoph Schied
NVIDIA
Karlsruhe Institute of Technology

Anton Kaplanyan
Chris Wyman
Anjul Patney
NVIDIA

Chakravarty R. Alla Chaitanya
NVIDIA
University of Montreal
McGill University

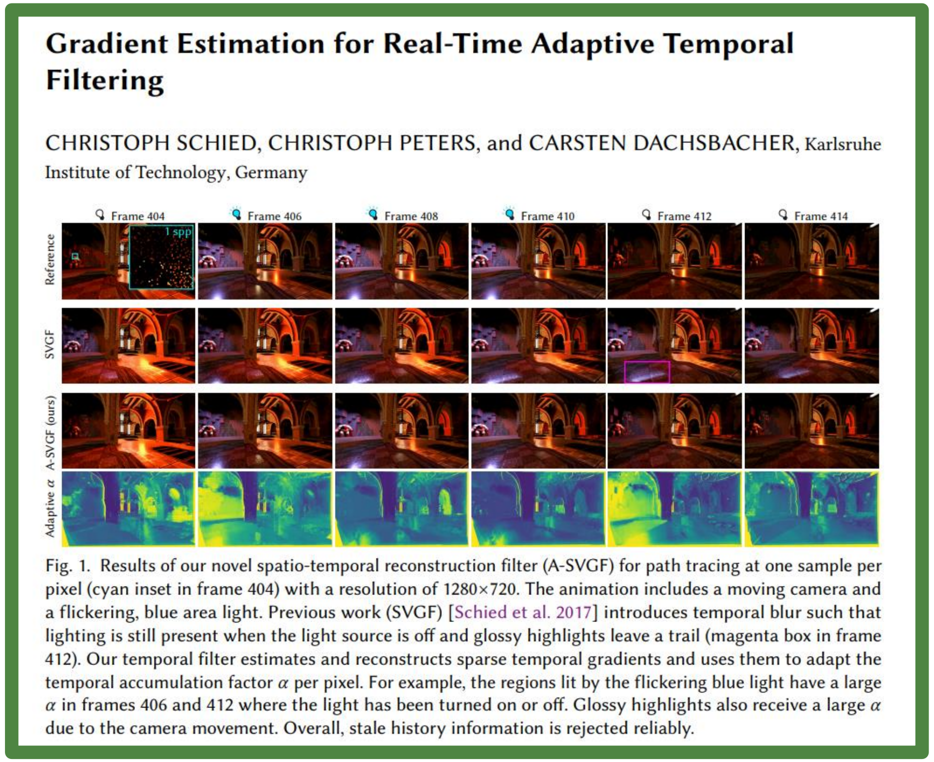
John Burgess
Shiqiu Liu
NVIDIA

Carsten Dachsbacher
Karlsruhe Institute of Technology

Aaron Lefohn
Marco Salvi
NVIDIA

Figure 1: Our filter takes (left) 1 sample per pixel path-traced input and (center) reconstructs a temporally stable 1920×1080 image in just 10 ms. Compare to (right) a 2048 samples per pixel path-traced reference. Insets compare our input, our filtered results, and a reference on two regions, and show the impact filtered global illumination has over just direct illumination. Given the noisy input, notice the similarity to the reference for glossy reflections, global illumination, and direct soft shadows.

+

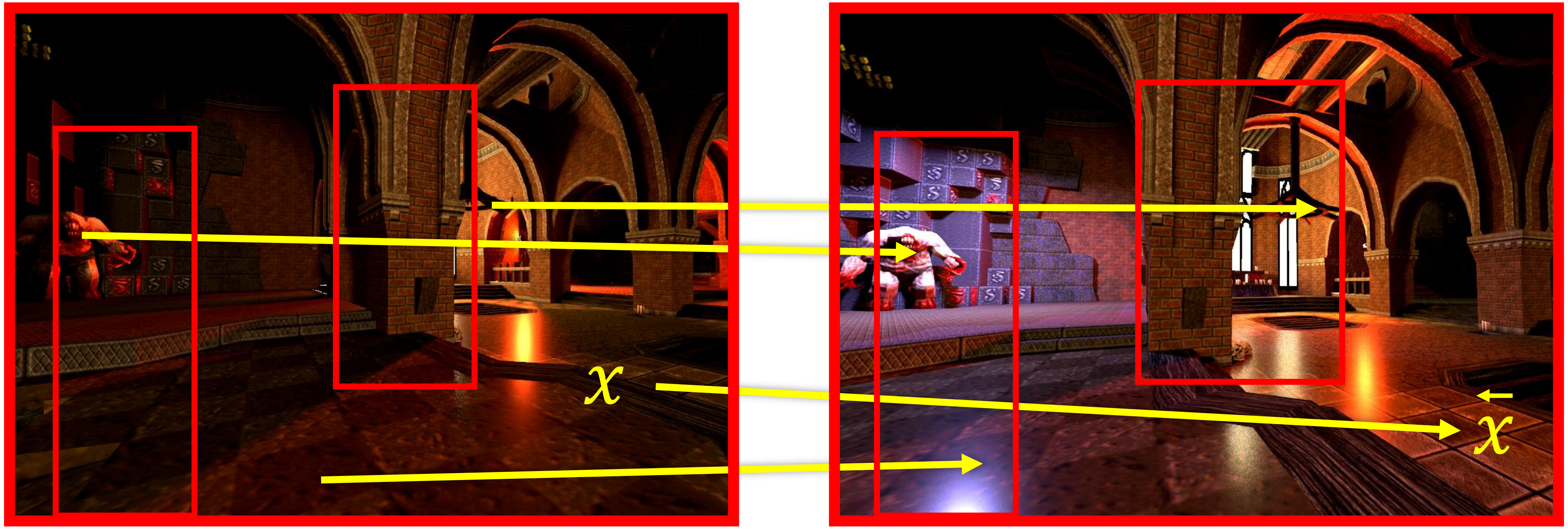


Gradient Estimation for Real-Time Adaptive Temporal Filtering

CHRISTOPH SCHIED, CHRISTOPH PETERS, and CARSTEN DACHSBACHER, Karlsruhe Institute of Technology, Germany

Fig. 1. Results of our novel spatio-temporal reconstruction filter (A-SVGF) for path tracing at one sample per pixel (cyan inset in frame 404) with a resolution of 1280×720. The animation includes a moving camera and a flickering, blue area light. Previous work (SVGF) [Schied et al. 2017] introduces temporal blur such that lighting is still present when the light source is off and glossy highlights leave a trail (magenta box in frame 412). Our temporal filter estimates and reconstructs sparse temporal gradients and uses them to adapt the temporal accumulation factor α per pixel. For example, the regions lit by the flickering blue light have a large α in frames 406 and 412 where the light has been turned on or off. Glossy highlights also receive a large α due to the camera movement. Overall, stale history information is rejected reliably.

Screen-space Reprojection



Current frame c_i

Previous filtered frame $\hat{c}_{i-1}$

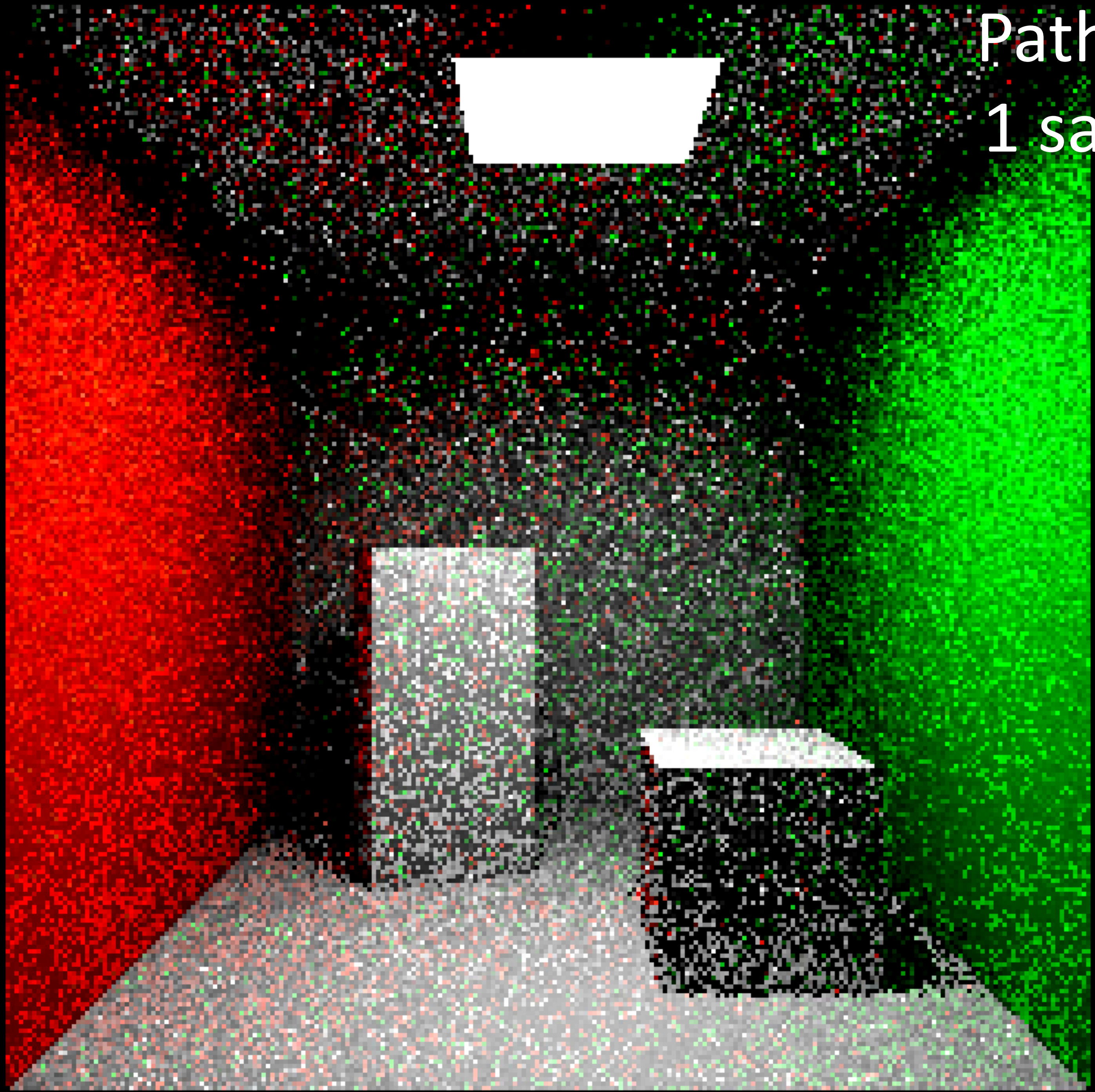
$$\hat{c}_i(x) = \alpha \cdot \boxed{c_i}\boxed{(x)} + (1-\alpha) \cdot \boxed{\hat{c}_{i-1}}\boxed{(\overleftarrow{x})}$$

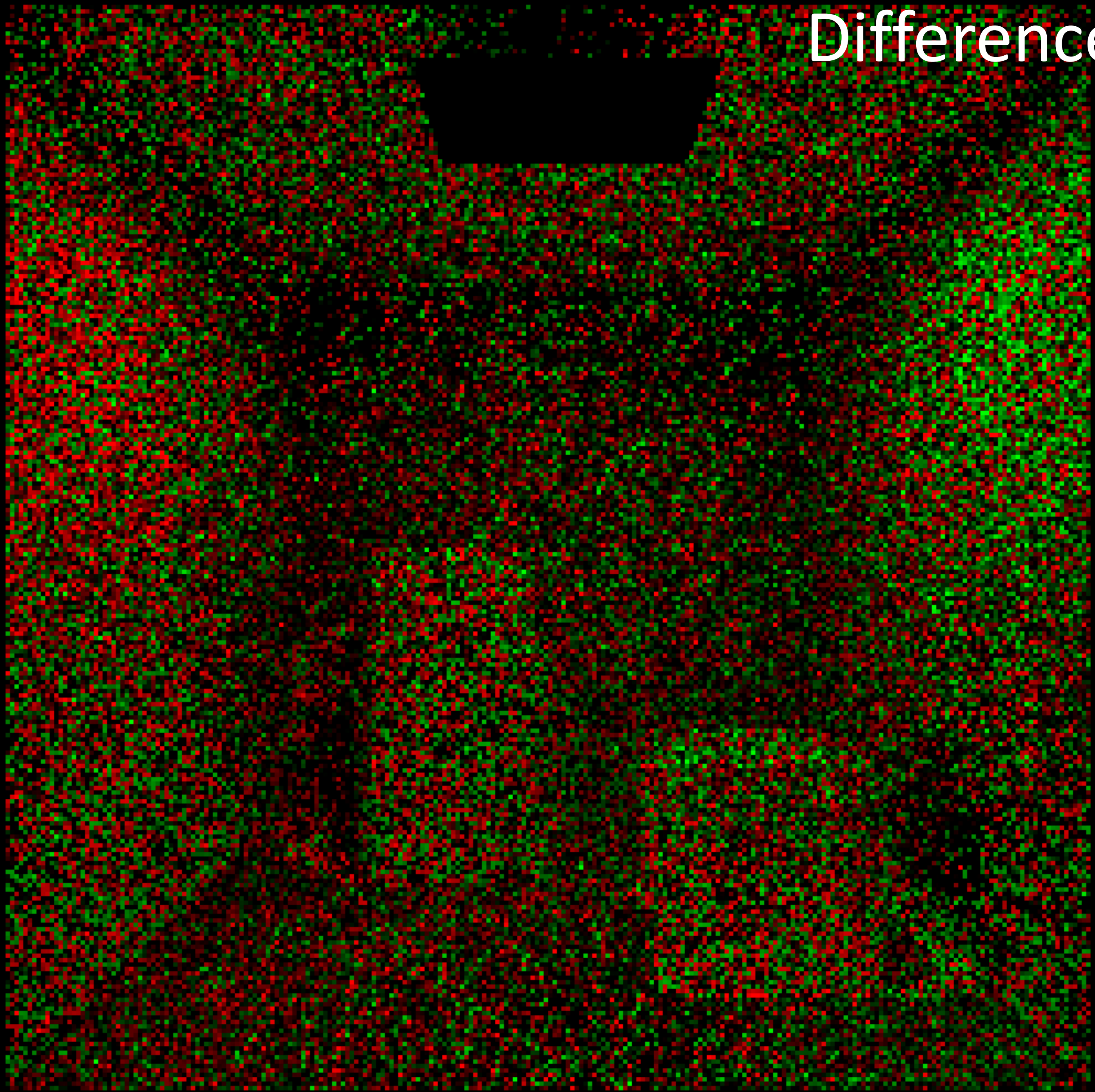
Adaptive Temporal Filtering

$$\hat{c}_i(x) = \boxed{\alpha} \cdot c_i(x) + \boxed{(1 - \alpha)} \cdot \hat{c}_{i-1}(\overleftarrow{x})$$

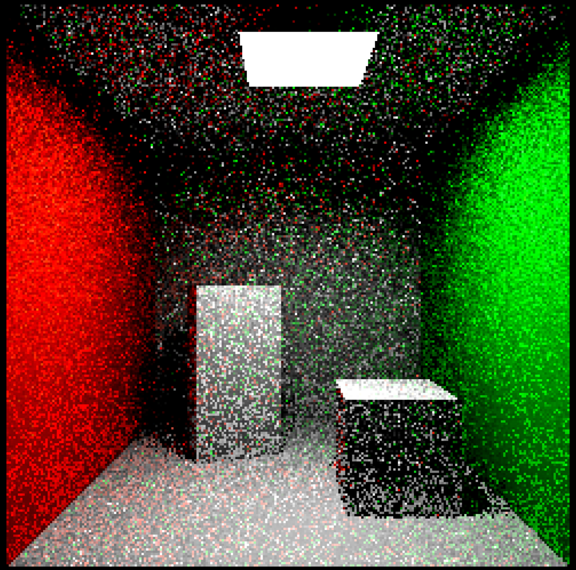
- Set α according to changes of the shading function
  - Moving shadows, glossy highlights, flickering light sources, …
- Make α per-pixel weight for local adaptivity
- **Need information about changes of shading (temporal gradient)**

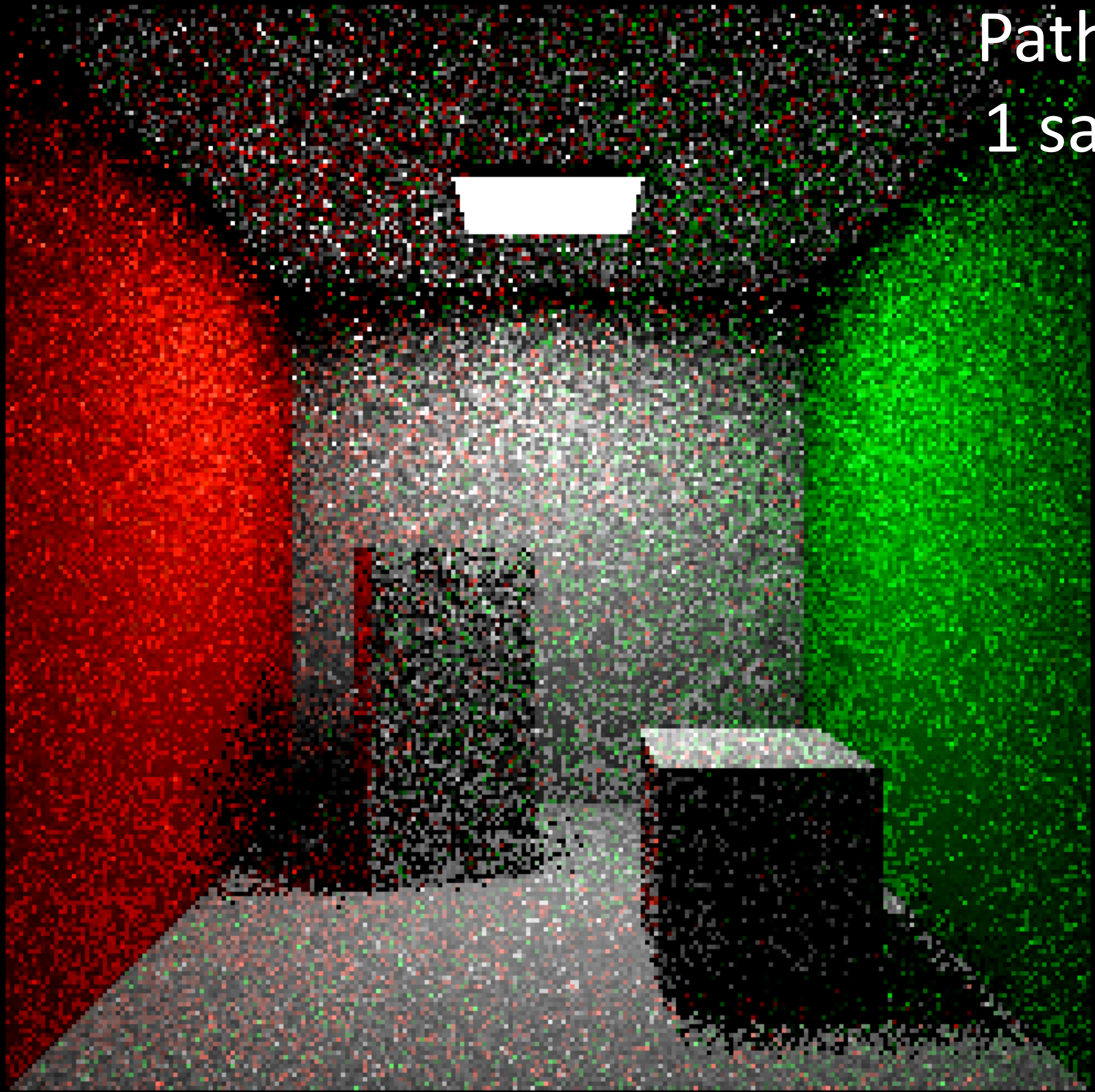
Path tracer output
1 sample per pixel





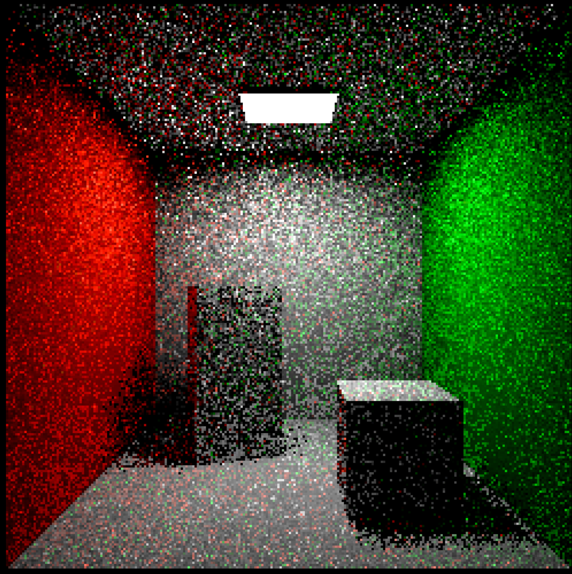
Difference of luminance
green positive
red negative

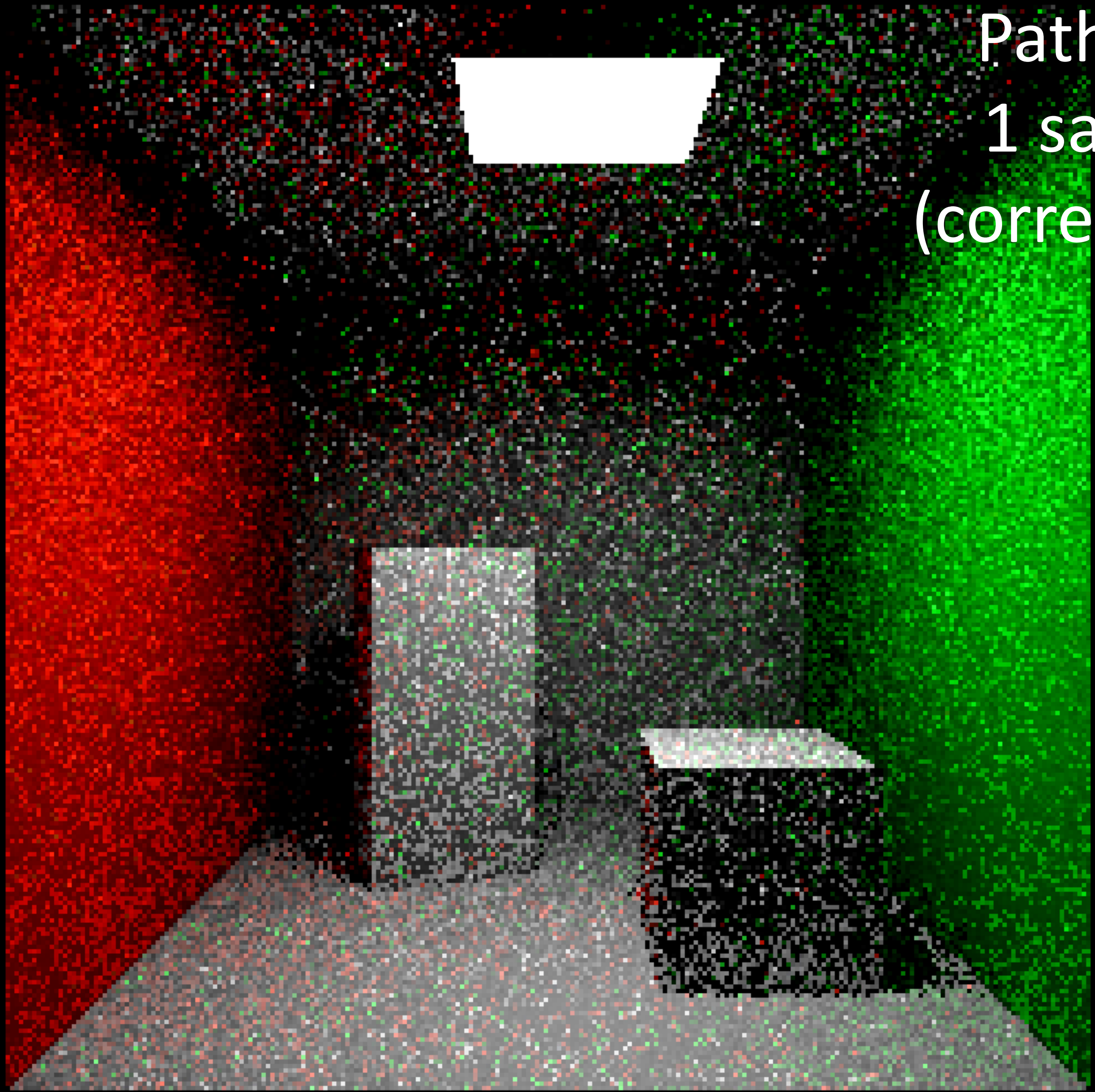




Path tracer output
1 sample per pixel

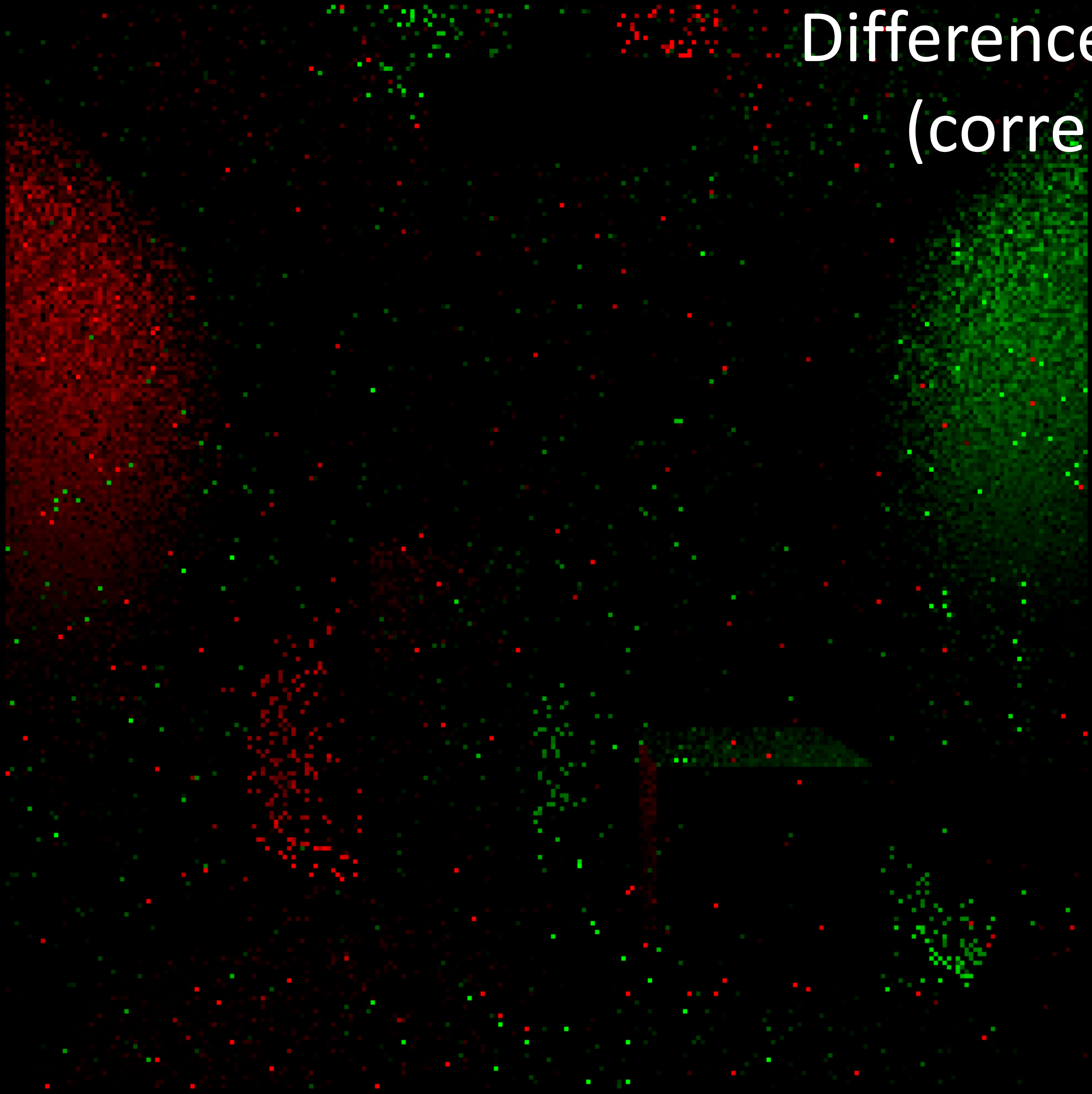
Difference of luminance
green positive
red negative

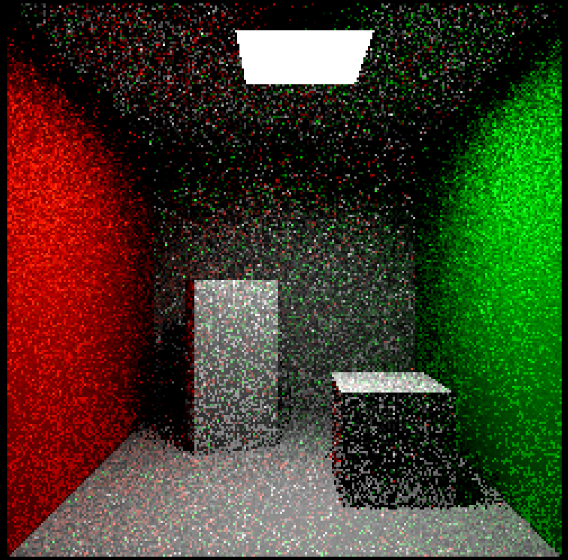




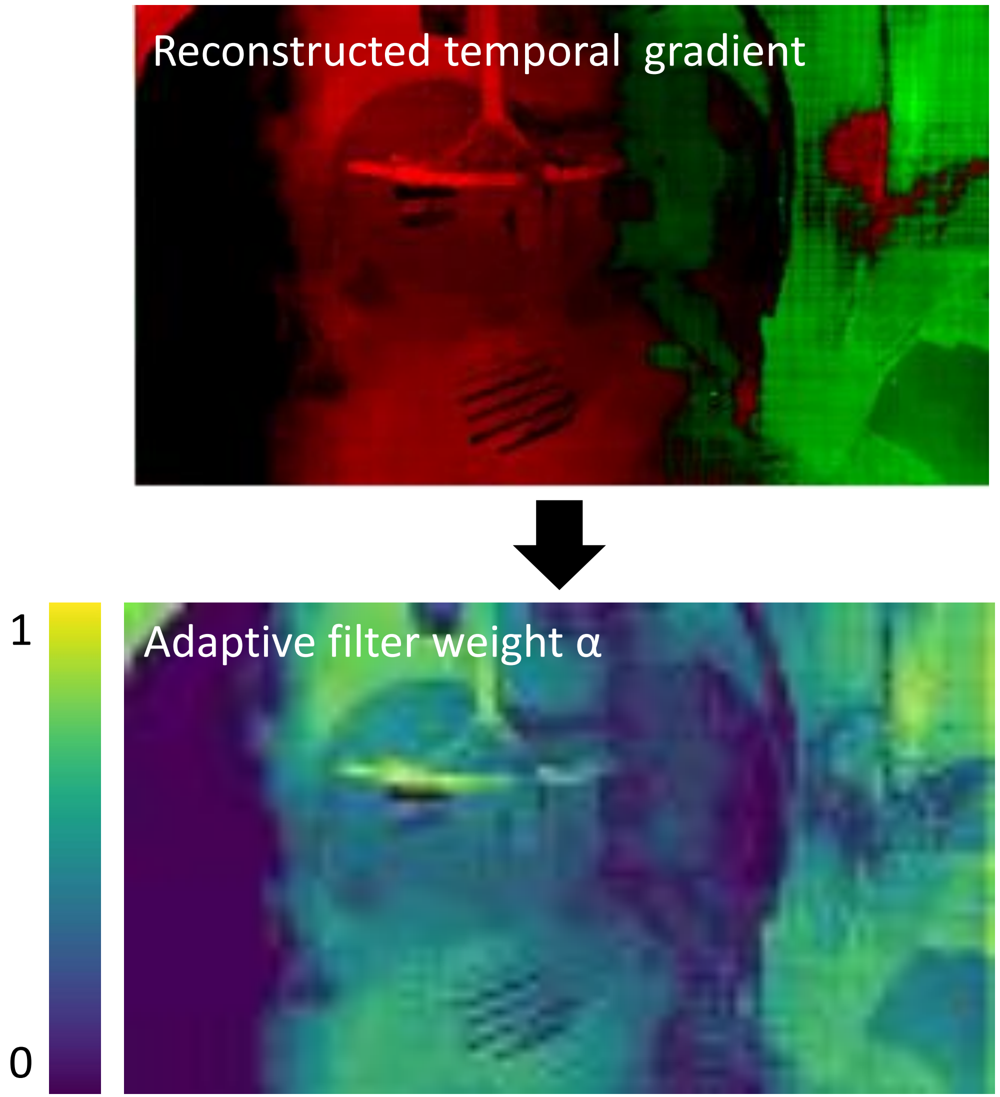
Path tracer output
1 sample per pixel
(correlated samples)

Difference of luminance
(correlated samples)
green positive
red negative





Adaptive temporal filter weight



- Sample and reconstruct temporal gradient
- Change α according to relative rate of change


100

Path Tracer


Treat each triangle of light meshes as individual area light

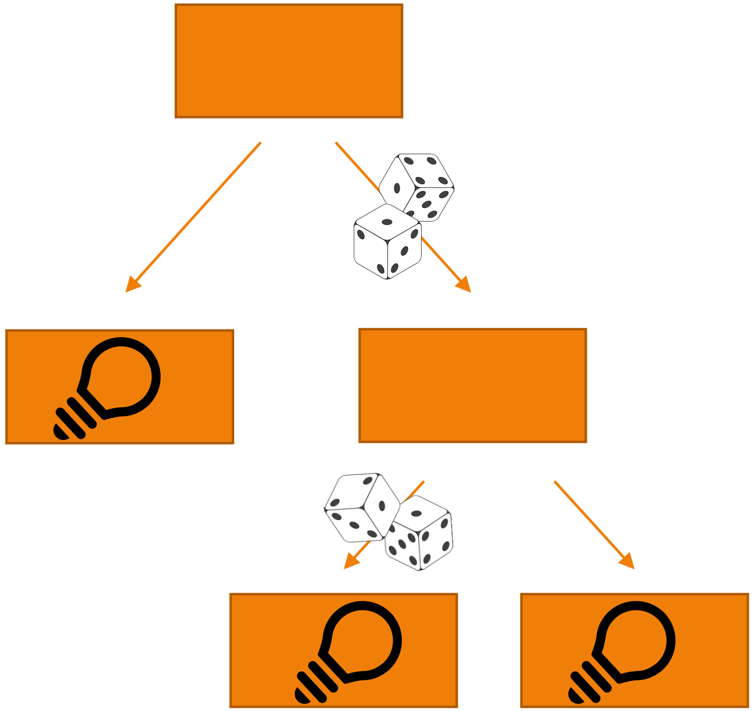
Light selection / sampling

Tried:

- Light hierarchy

Issues:

- Speed
- Inconsistent quality under animation



Importance Sampling of Many Lights with Adaptive Tree Splitting

ALEJANDRO CONTY ESTEVEZ, Sony Pictures Imageworks
CHRISTOPHER KULLA, Sony Pictures Imageworks

Fig. 1. A procedural city with 363,036 lights, one GI bounce and participating media. Rendered with 16 samples per pixel, each shading point takes an average of 7 shadow rays (45 for the volume integral). We shoot an average of 1700 rays per pixel. The image rendered in 20 minutes on a quad core Intel i7.

We present a technique to importance sample large collections of lights (including mesh lights as collections of small emitters) in the context of Monte-Carlo path tracing. A bounding volume hierarchy over all emitters is traversed at each shading point using a single random number in a way that importance samples their predicted contribution. The tree aggregates energy, spatial and orientation information from the emitters to enable accurate prediction of the effect of a cluster of lights on any given shading point. We further improve the performance of the algorithm by forcing splitting until the importance of a cluster is sufficiently representative of its contents.

CCS Concepts: • **Computing methodologies → Ray tracing**;

Additional Key Words and Phrases: illumination, ray tracing, many lights

ACM Reference Format:
Alejandro Conty Estevez and Christopher Kulla. 2018. Importance Sampling of Many Lights with Adaptive Tree Splitting. *Proc. ACM Comput. Graph. Interact. Tech.* 1, 2, Article 25 (August 2018), 17 pages. https://doi.org/10.1145/3233305

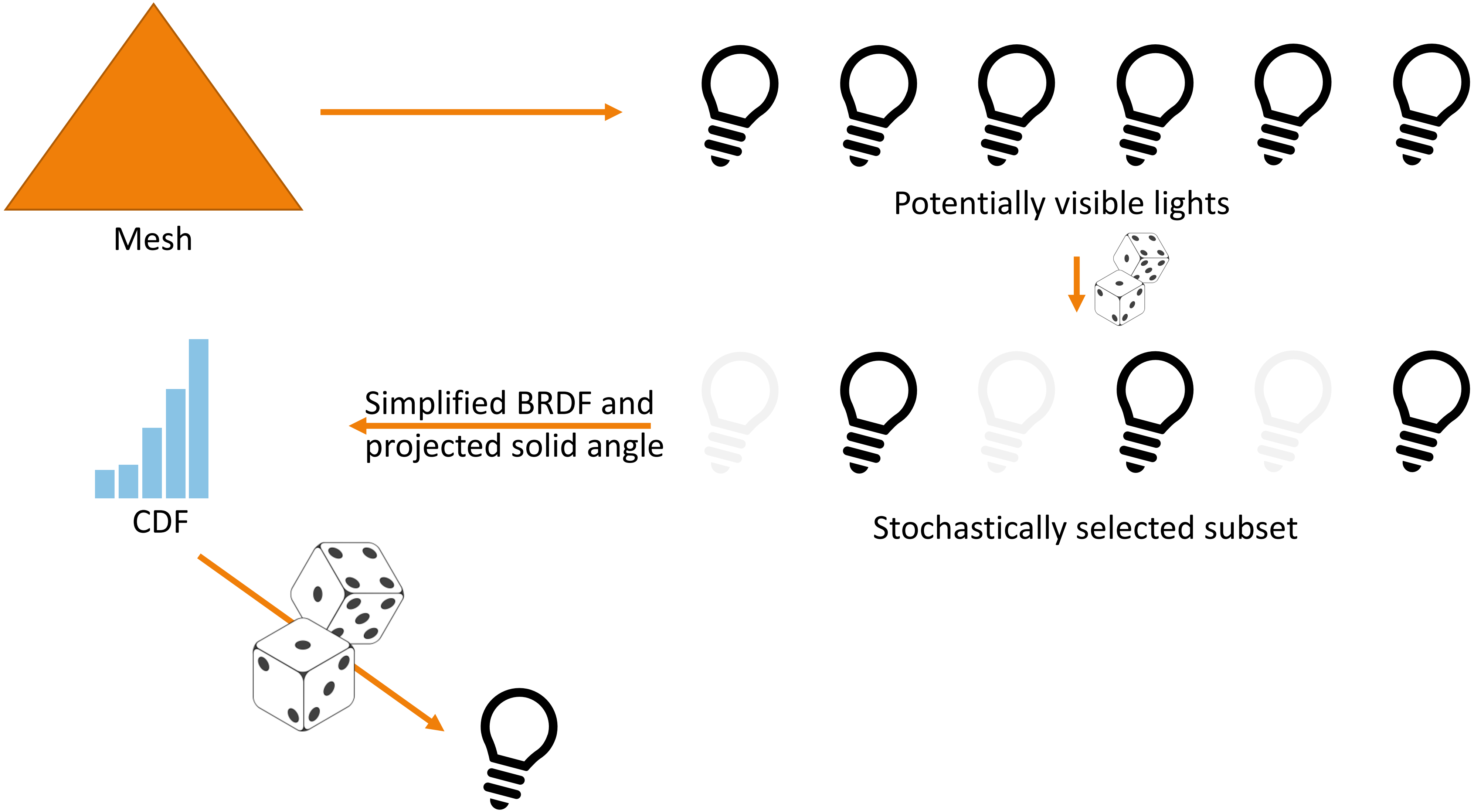
1 INTRODUCTION

Direct lighting calculations are a critical part of modern path tracing renderers with next event estimation. While sampling from simple light shapes [Shirley et al. 1996] is well understood, relatively little attention has been devoted to the problem of efficiently sampling from large collections of such shapes. In production renderers, this problem appears both in the form of scenes containing many distinct lights (Figure 1), and scenes with meshes acting as emitters (sometimes

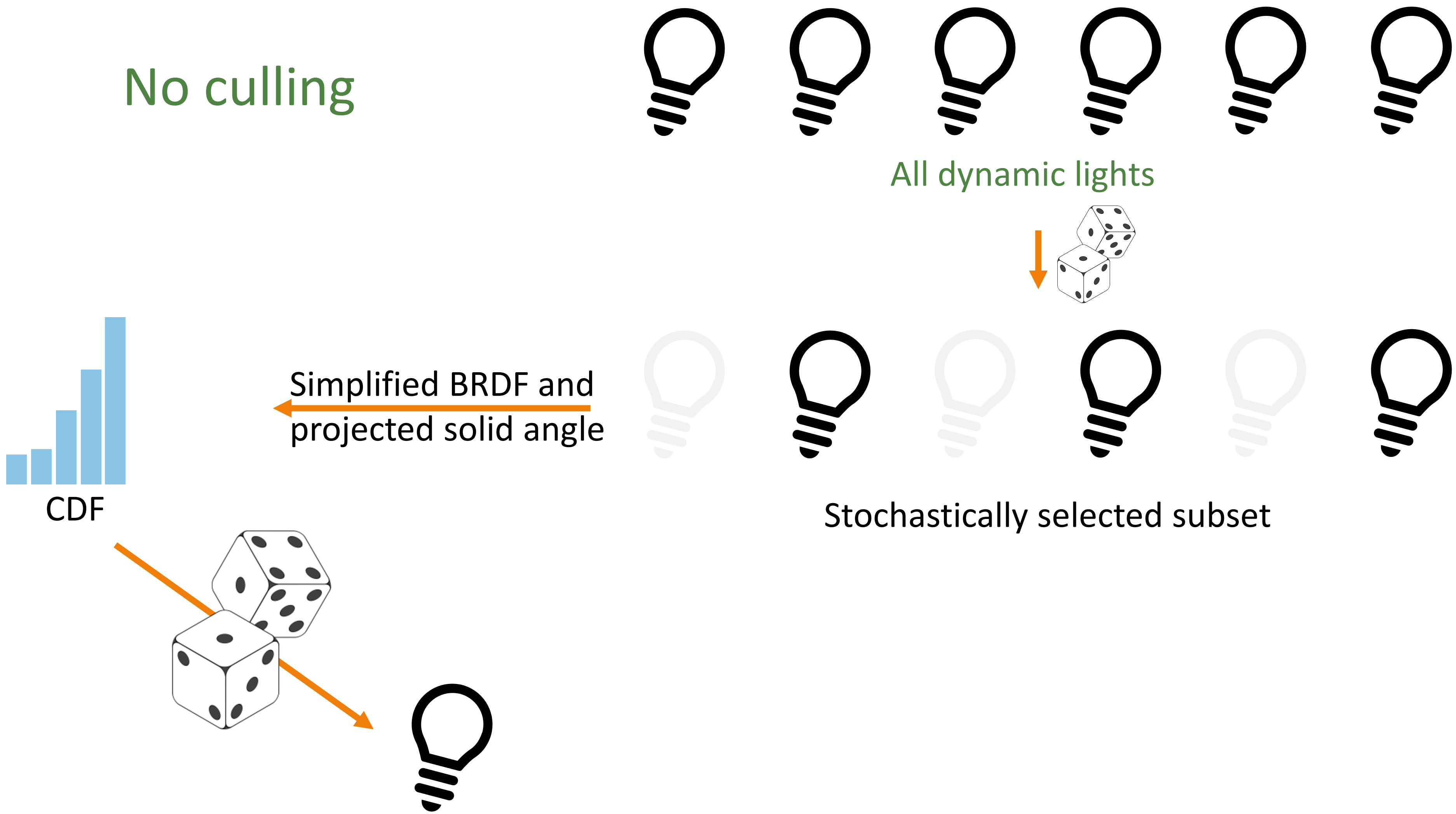
Authors' addresses: Alejandro Conty Estevez, Sony Pictures Imageworks; Christopher Kulla, Sony Pictures Imageworks.

2577-6193/2018/8-ART25 $15.00
https://doi.org/10.1145/3233305

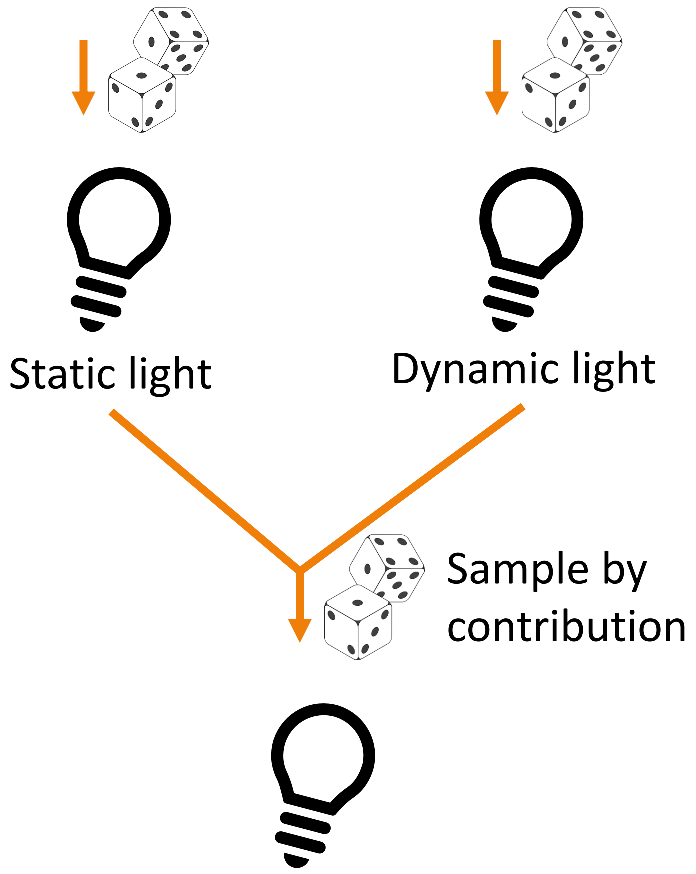
Light selection / sampling for static lights



Light selection / sampling for dynamic lights

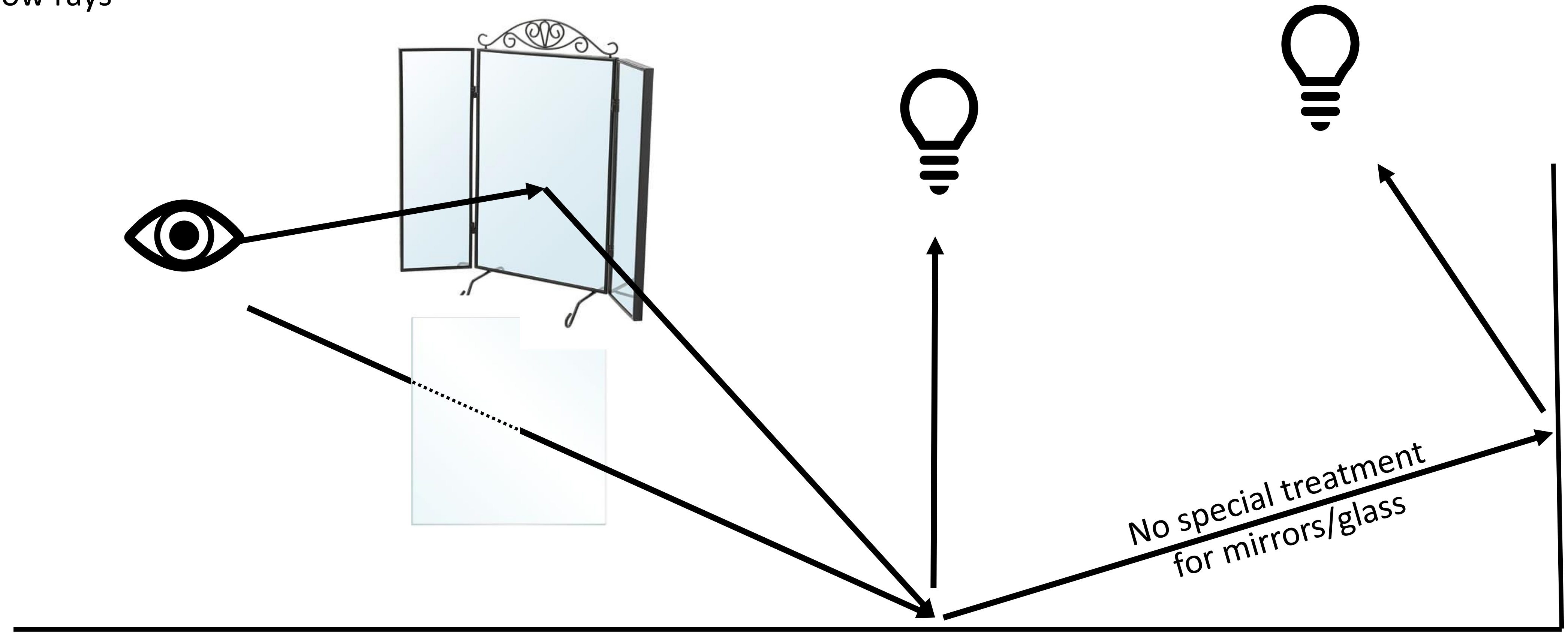


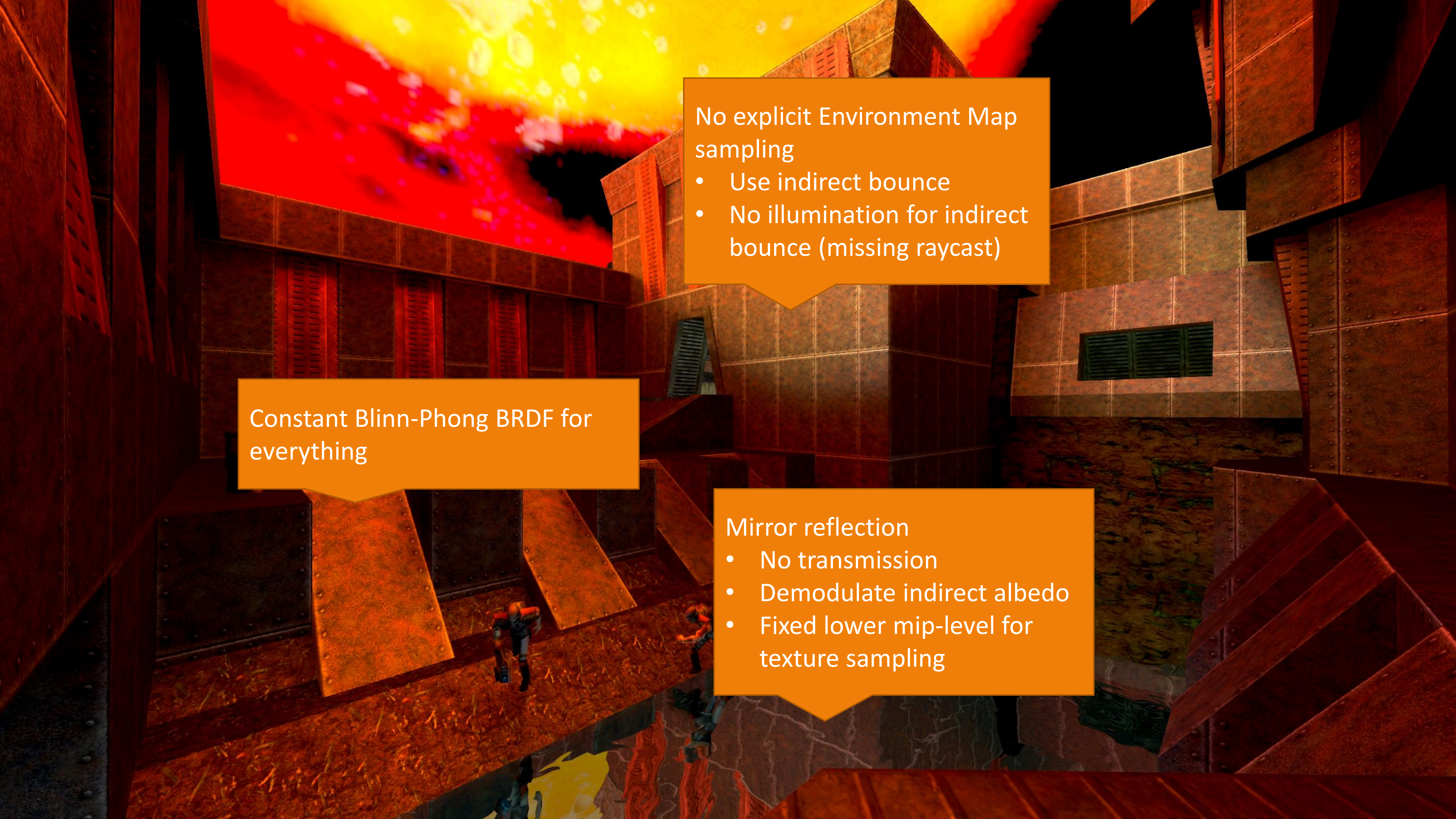
Light selection



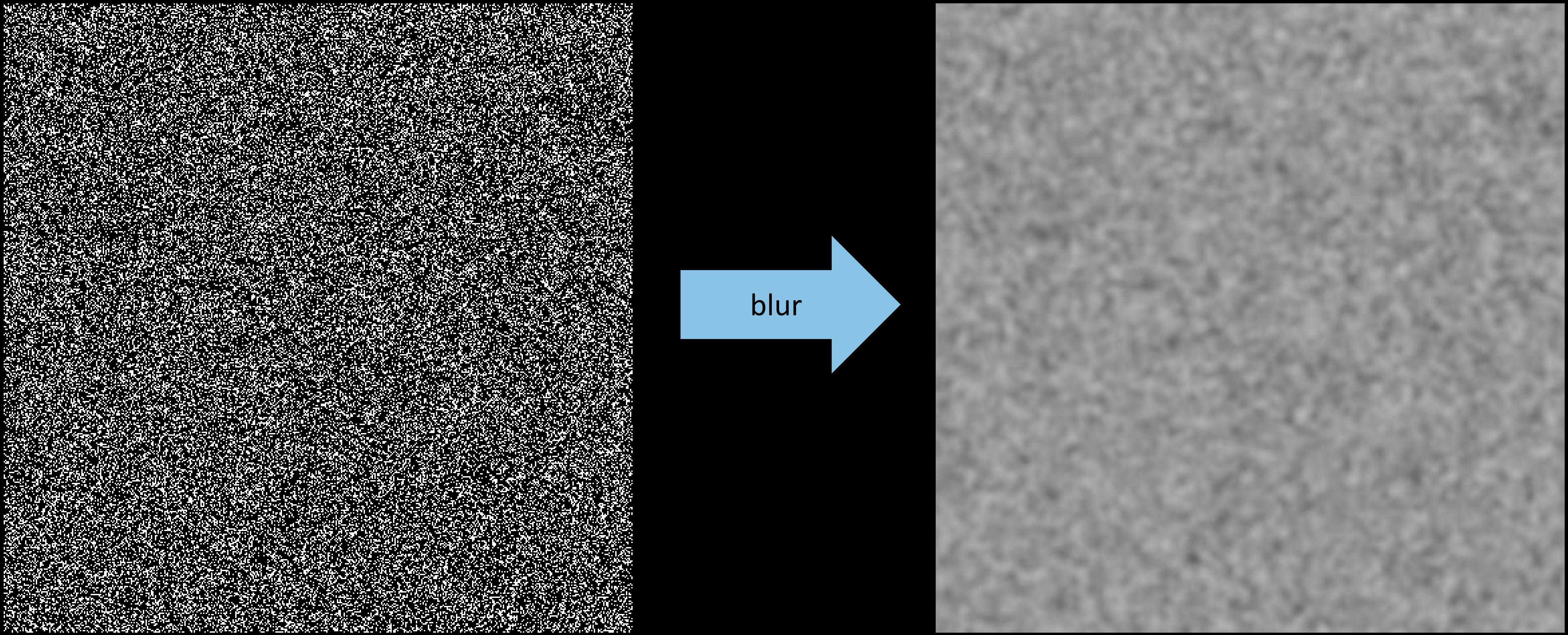
Path tracer

- One path per pixel
- One indirect bounce
- Two shadow rays



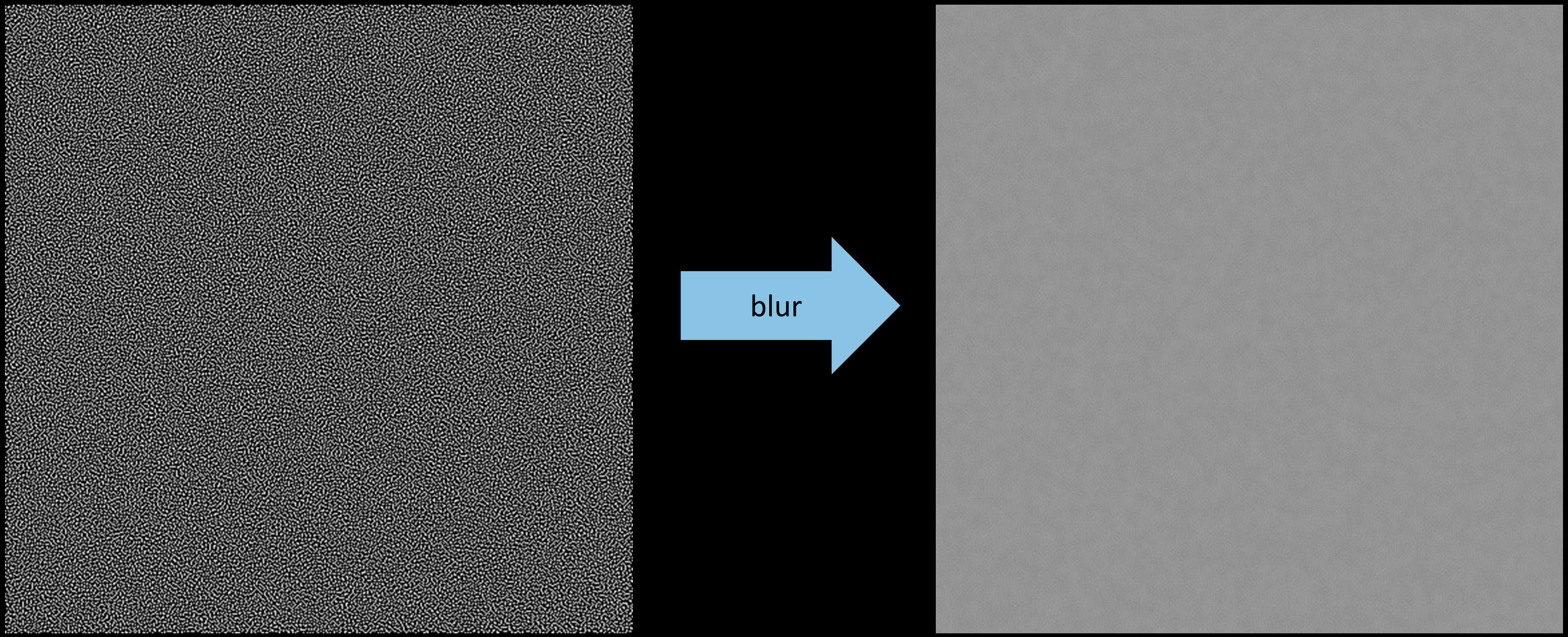

No explicit Environment Map sampling
• Use indirect bounce
• No illumination for indirect bounce (missing raycast)
Constant Blinn-Phong BRDF for everything
Mirror reflection
• No transmission
• Demodulate indirect albedo
• Fixed lower mip-level for texture sampling

White noise



Source: http://momentsingraphics.de/?p=127

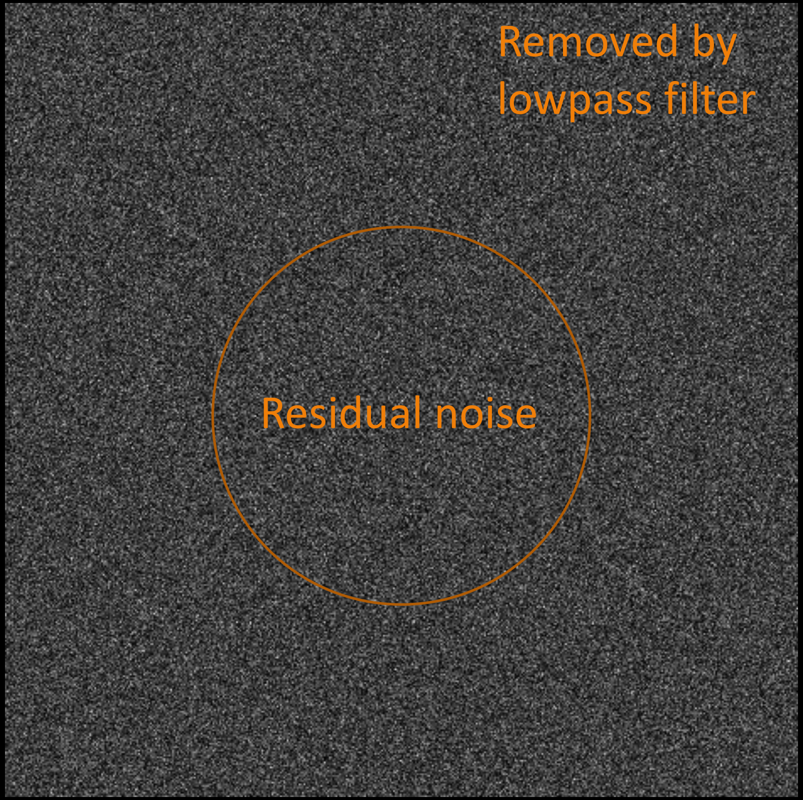
Blue noise



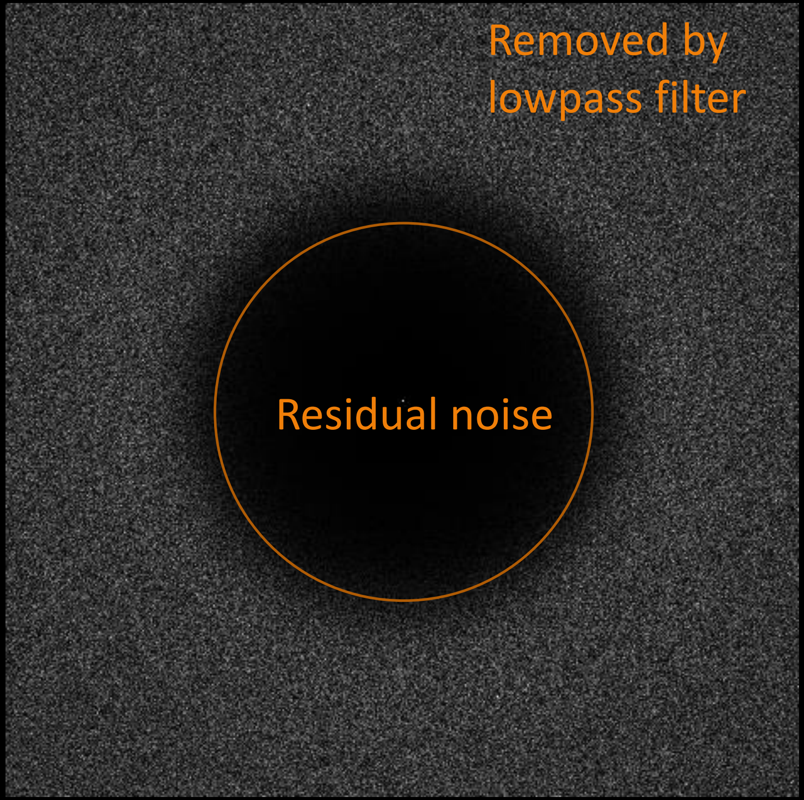
Source: http://momentsingraphics.de/?p=127

Magnitude of Fourier Transform

White noise



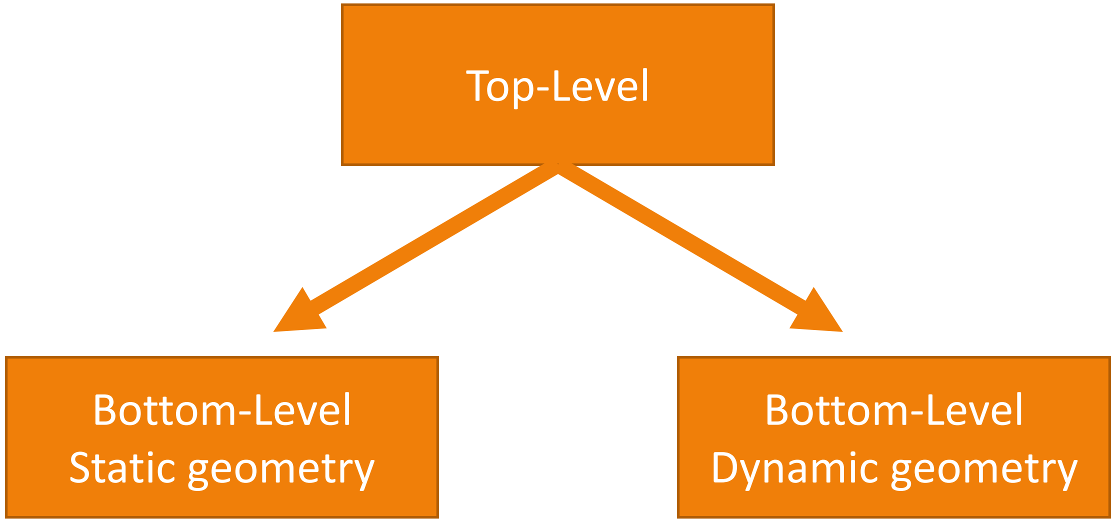
Blue noise



Source: http://momentsingraphics.de/?p=127

Implementation details

Acceleration structures

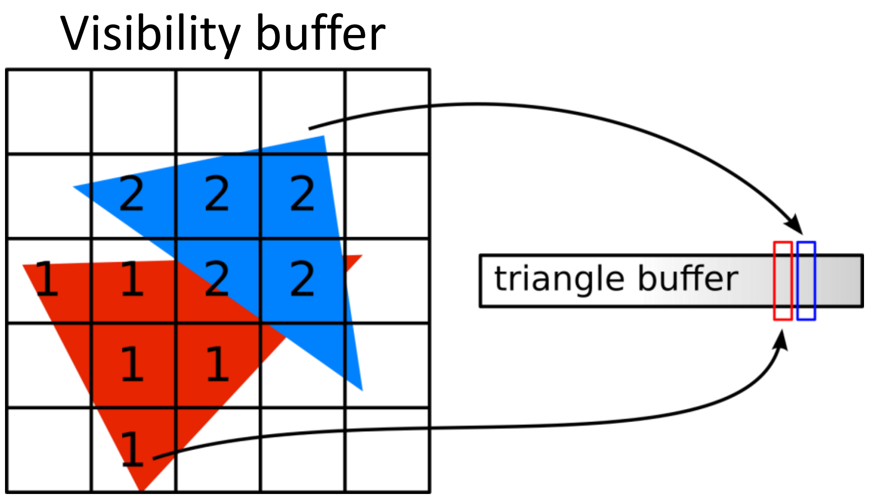


Built once on map-load

Rebuilt from scratch per frame

Forward / Backward projection

- Required for Adaptive Temporal Filtering
- Visibility buffer for forward projection
- Map instances between frames



Conclusion

- Real-time path tracing is possible (in the near future)

- Transition difficult
  - Random access to everything
  - Tweaking of assets

- Need more research specifically tailored towards real-time rendering
  - Fast and robust importance sampling
  - Denoising

Thanks!

Addis Dittebrandt
Alisa Jung
Anton Kaplanyan
Christoph Peters
Florian Reibold
Johannes Hanika
Stephan Bergmann
Tobias Zirr

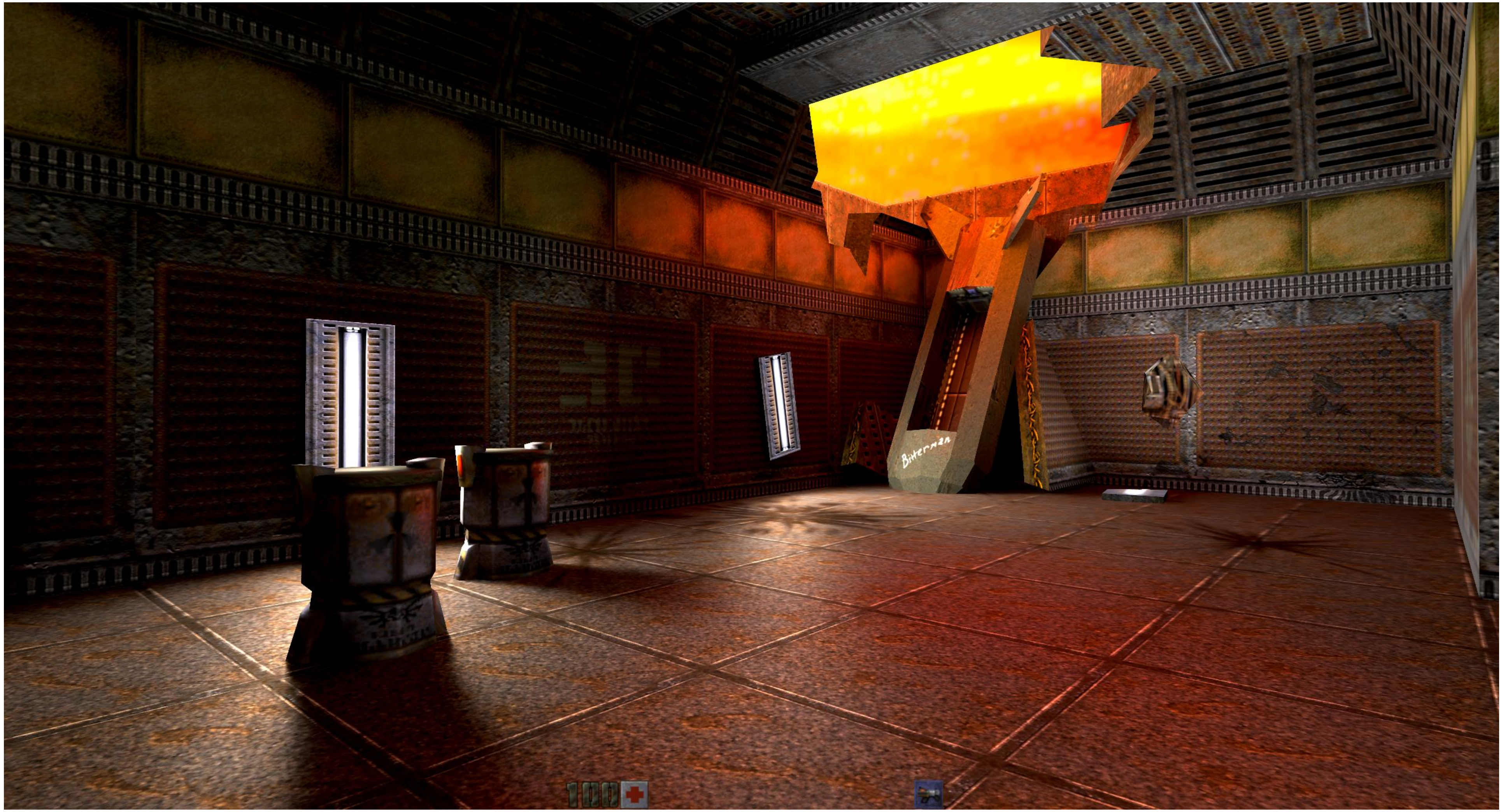
NVIDIA
id Software

http://brechpunkt.de/q2vkpt
https://github.com/cschied/q2vkpt
schied@brechpunkt.de
@c_schied

Quake II RTX

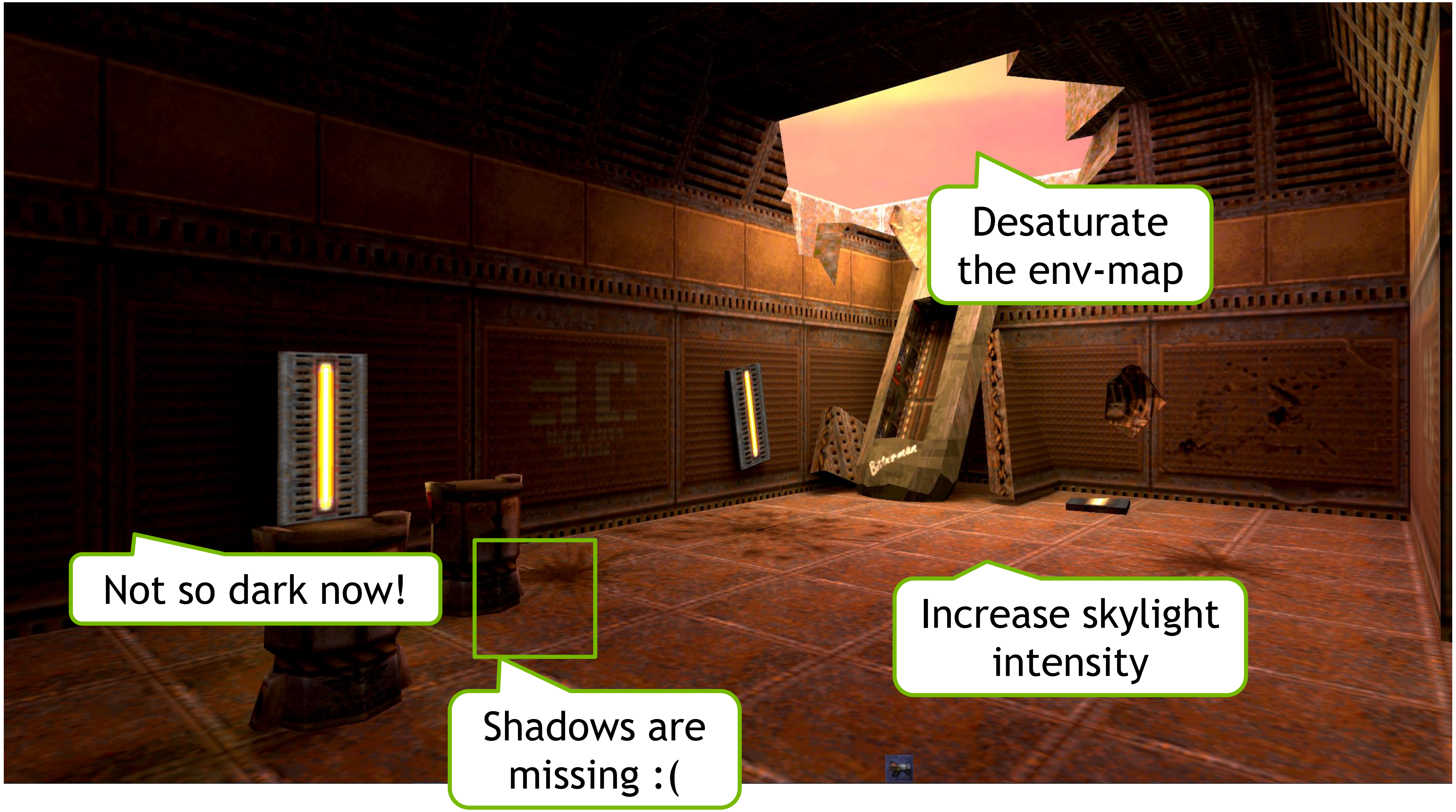
Alexey Panteleev

Original Q2VKPT Image



Textures from Quake 2

Fixing the Sky

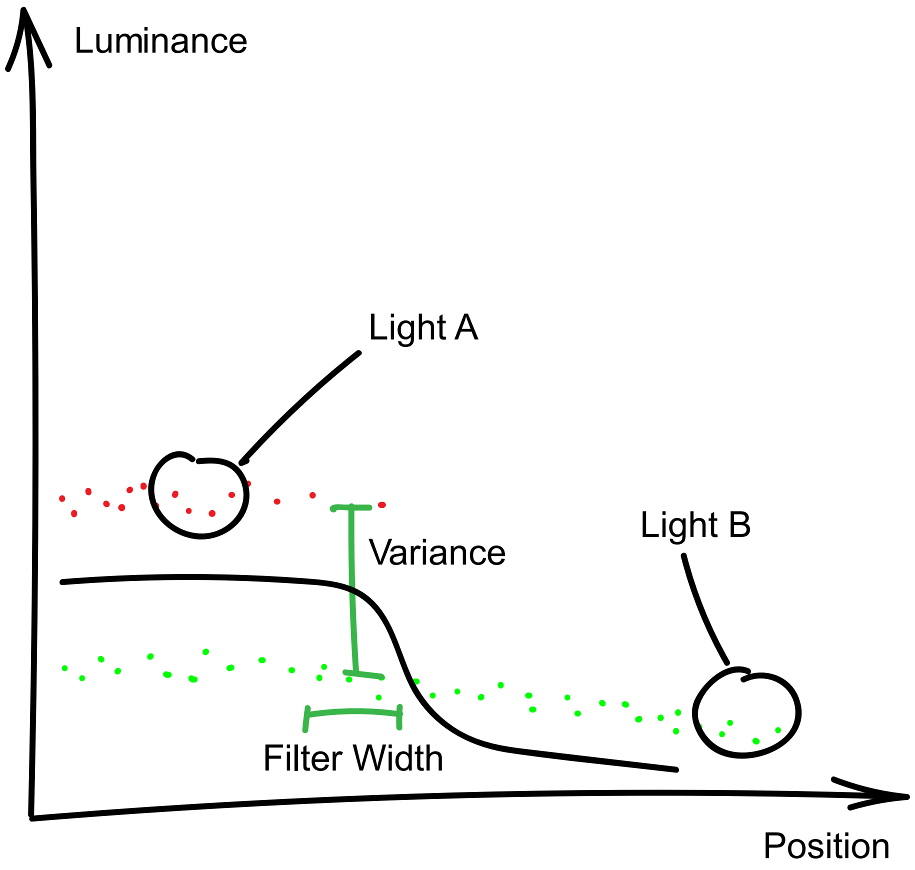


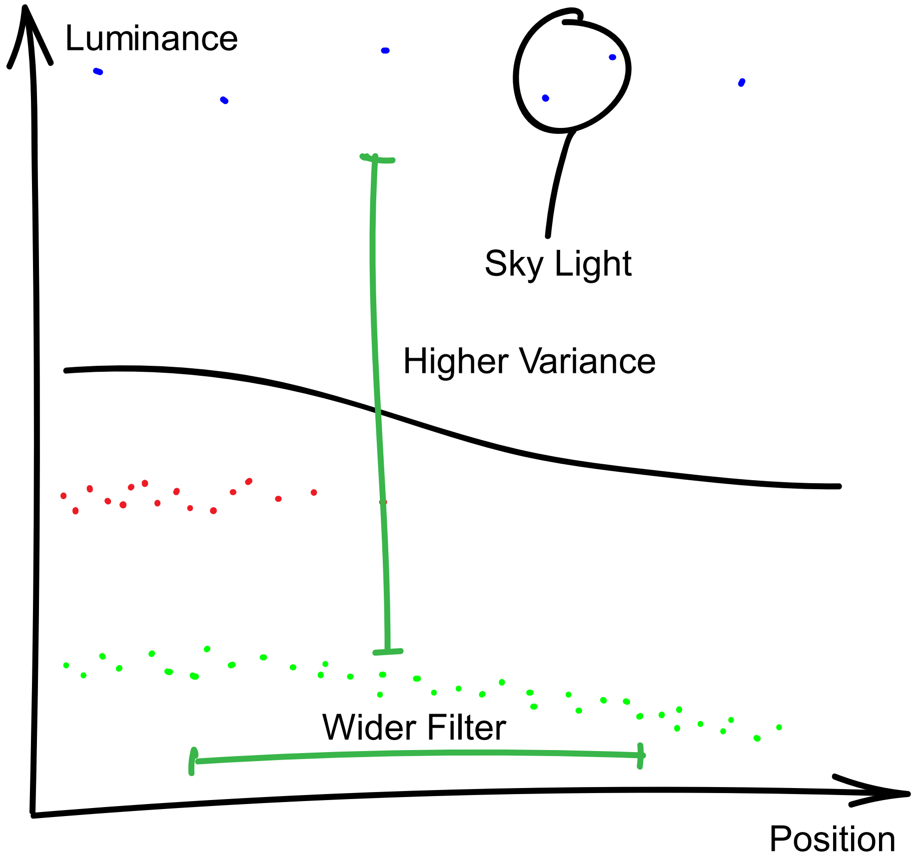
Environment Noise


Original environment map

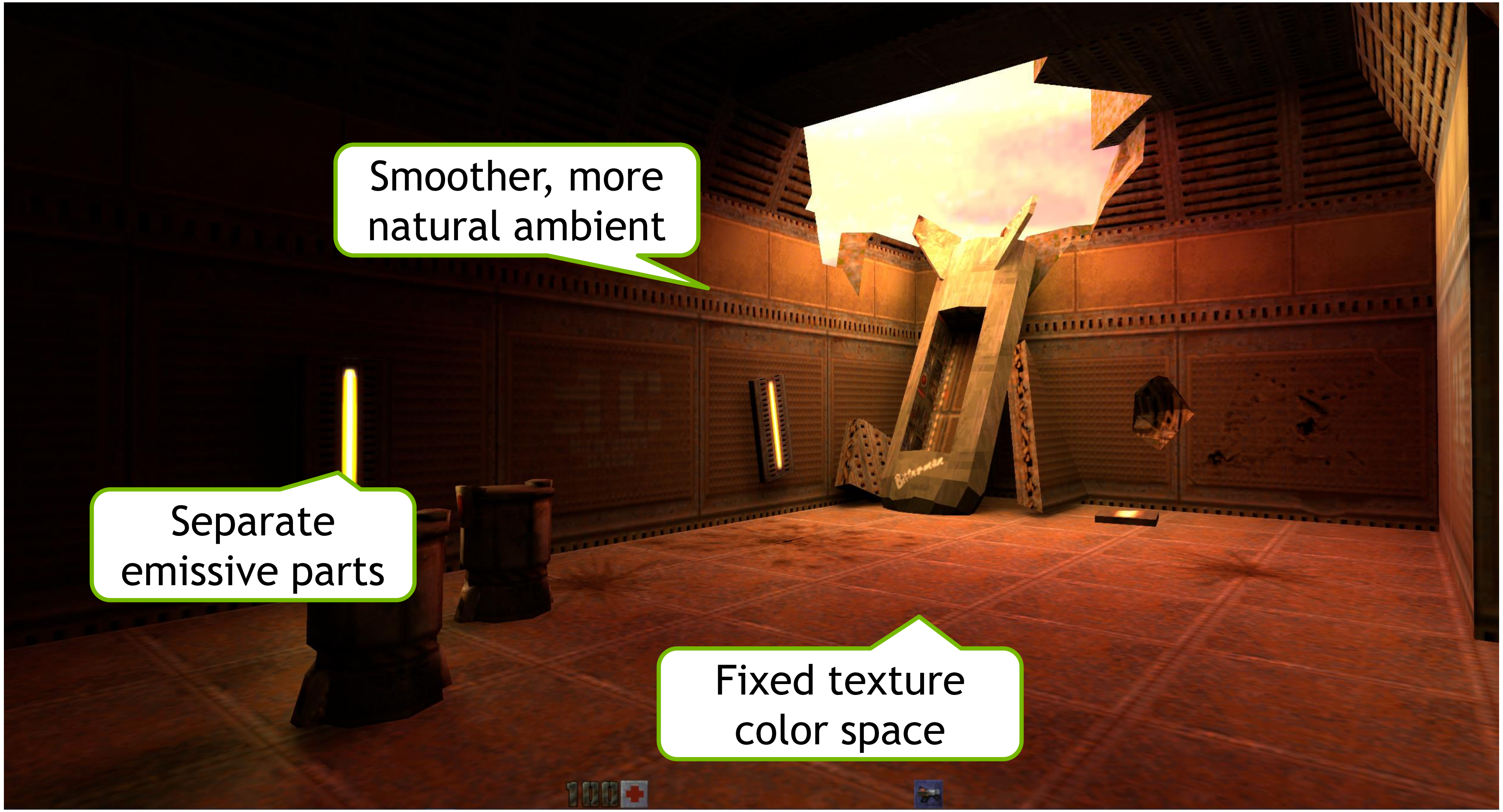

Bright environment map

Environment Noise



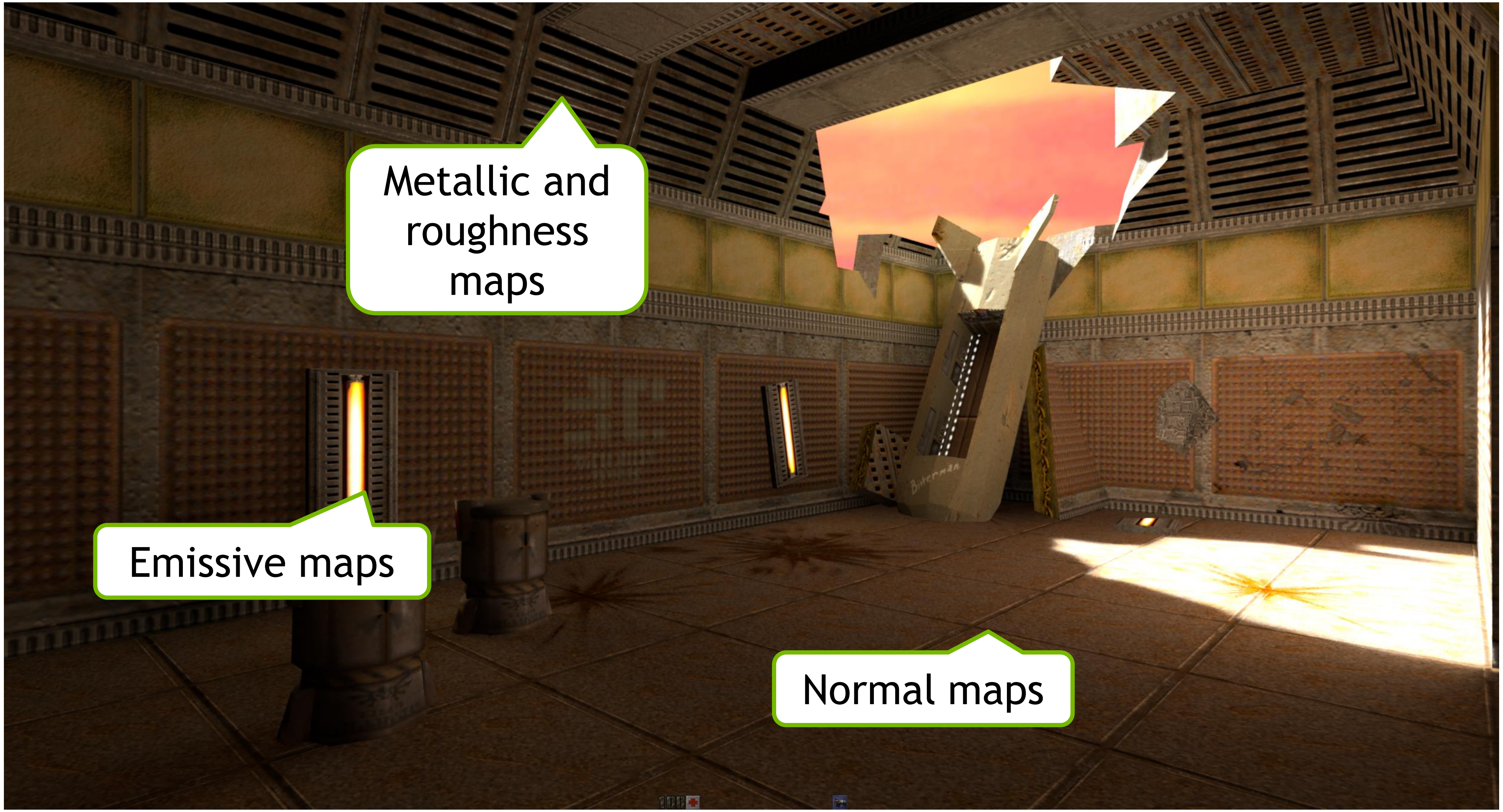


Textures and Tone Mapping

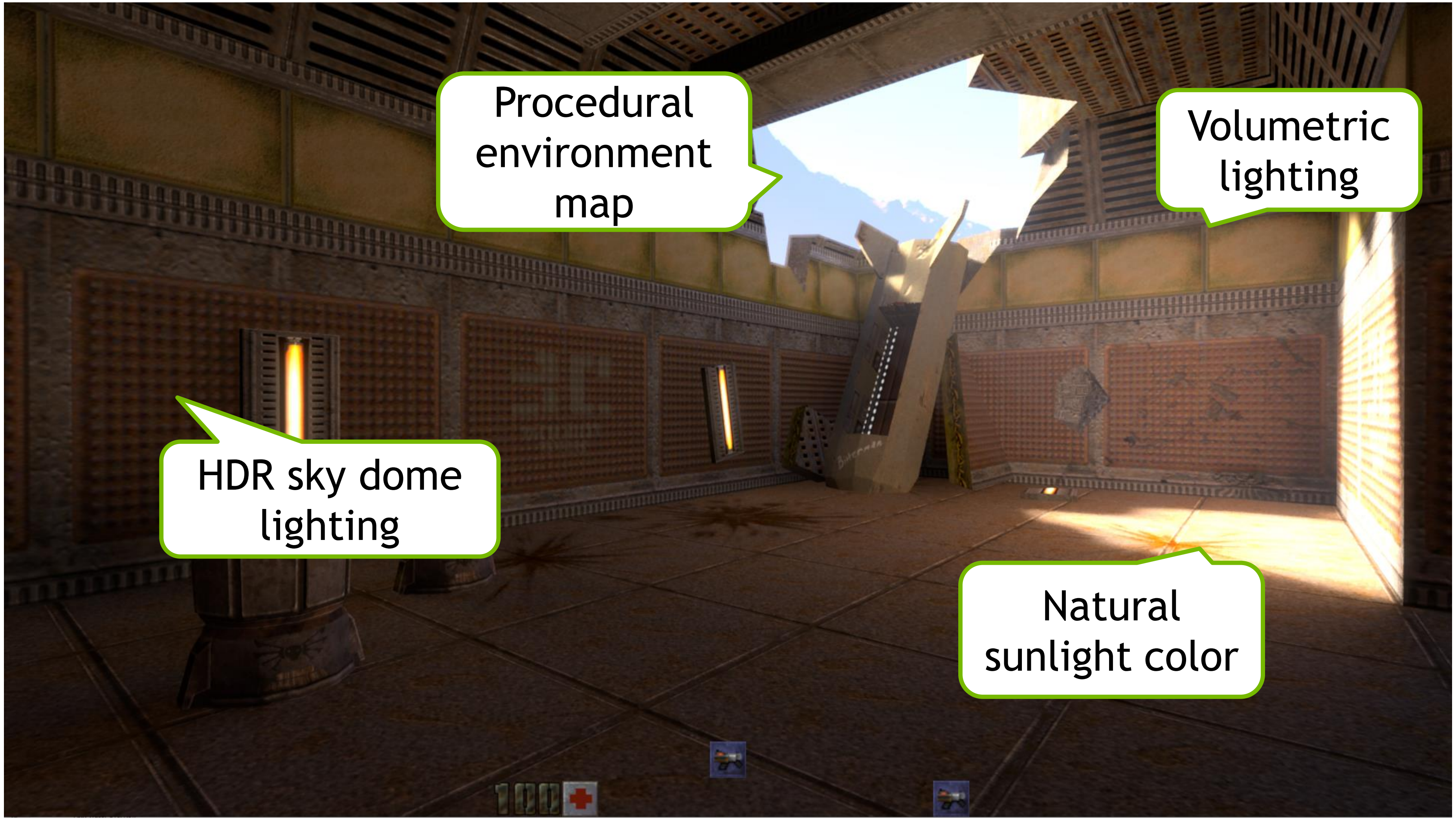


Add Sunlight and Denoiser Channels

Materials



Environment



Final Image


Path Tracer Output
100

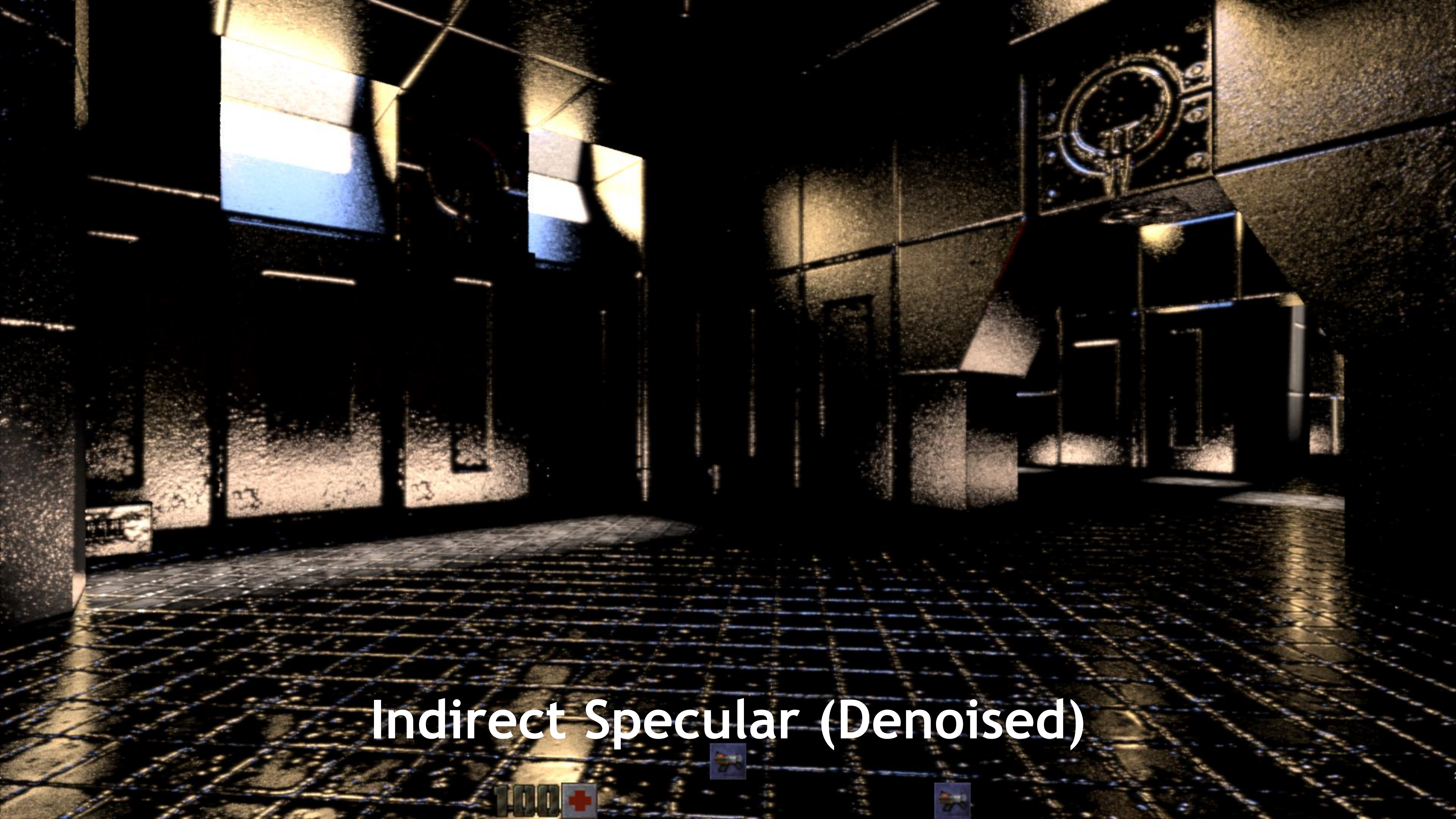
Direct Diffuse

Direct Diffuse (Denoised)

Indirect Diffuse


Indirect Diffuse (Denoised with SH)
100

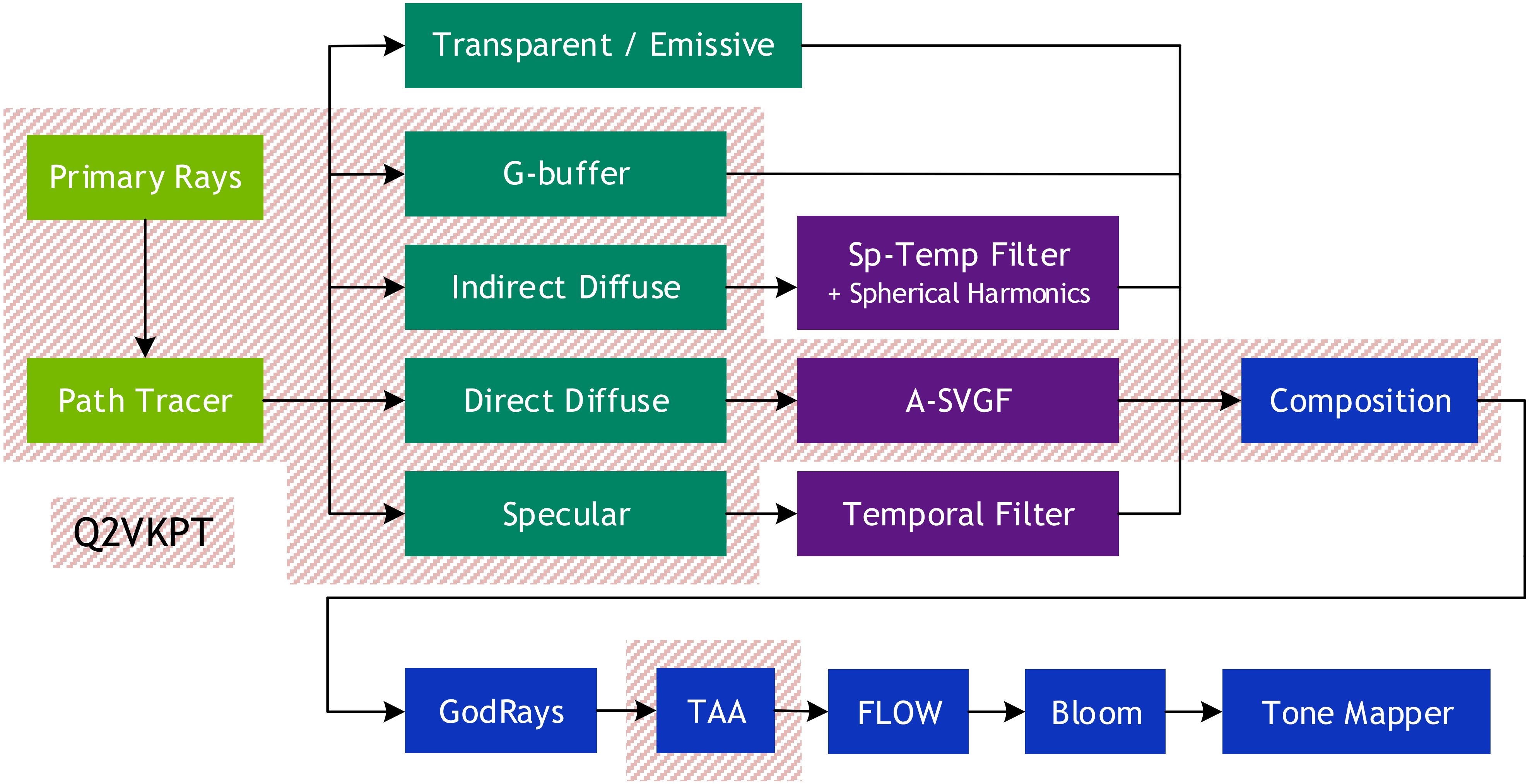

Indirect Specular
100


Indirect Specular (Denoised)
100

Irradiance Channels Combined

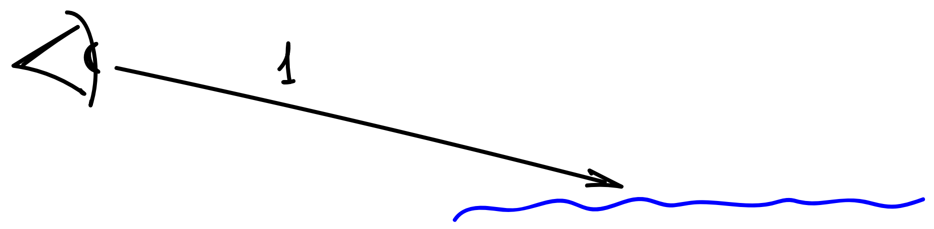
Final Image

Quake II RTX Rendering Pipeline



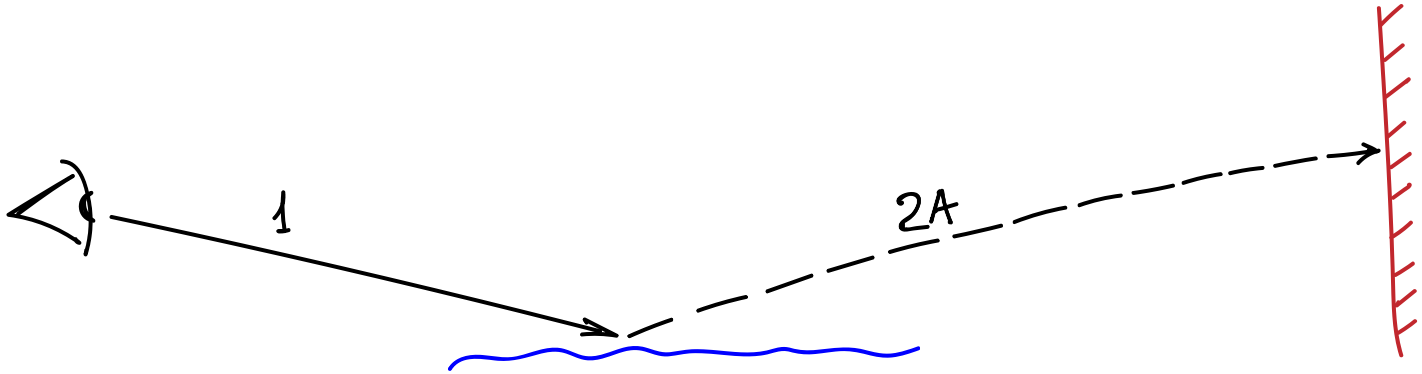
Path Tracer Overview

1. Primary



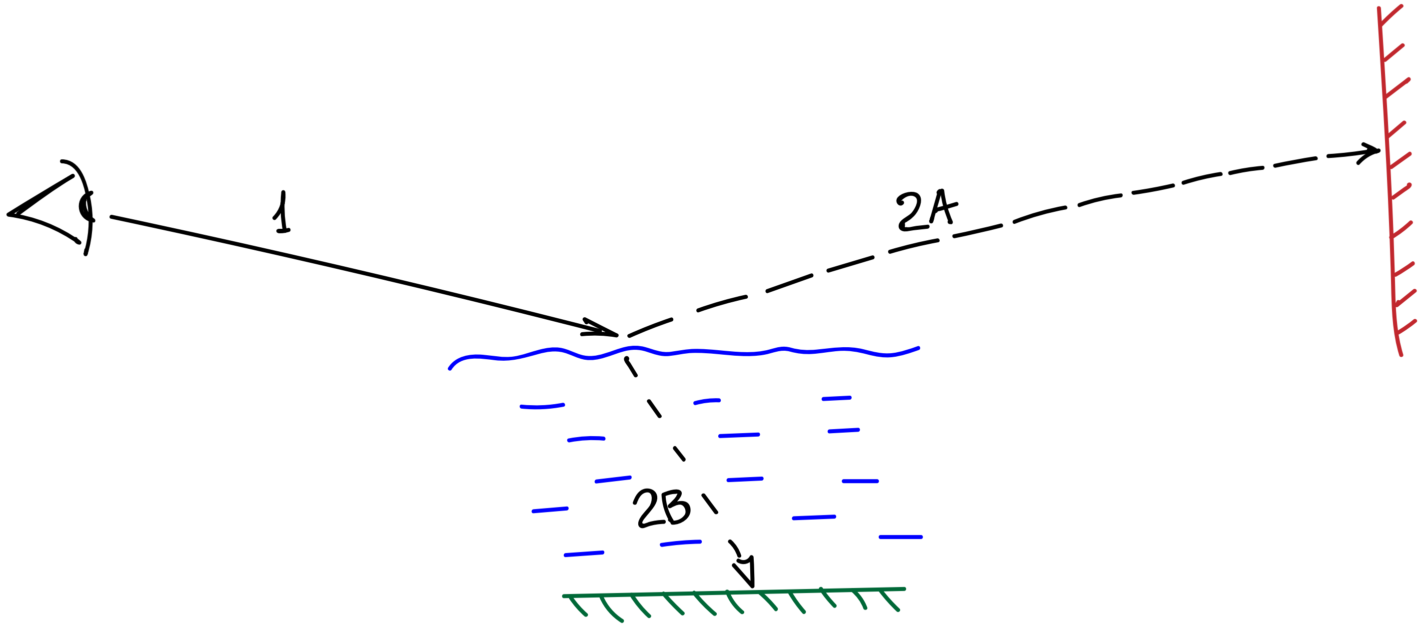
Path Tracer Overview

1. Primary
2. Reflection…



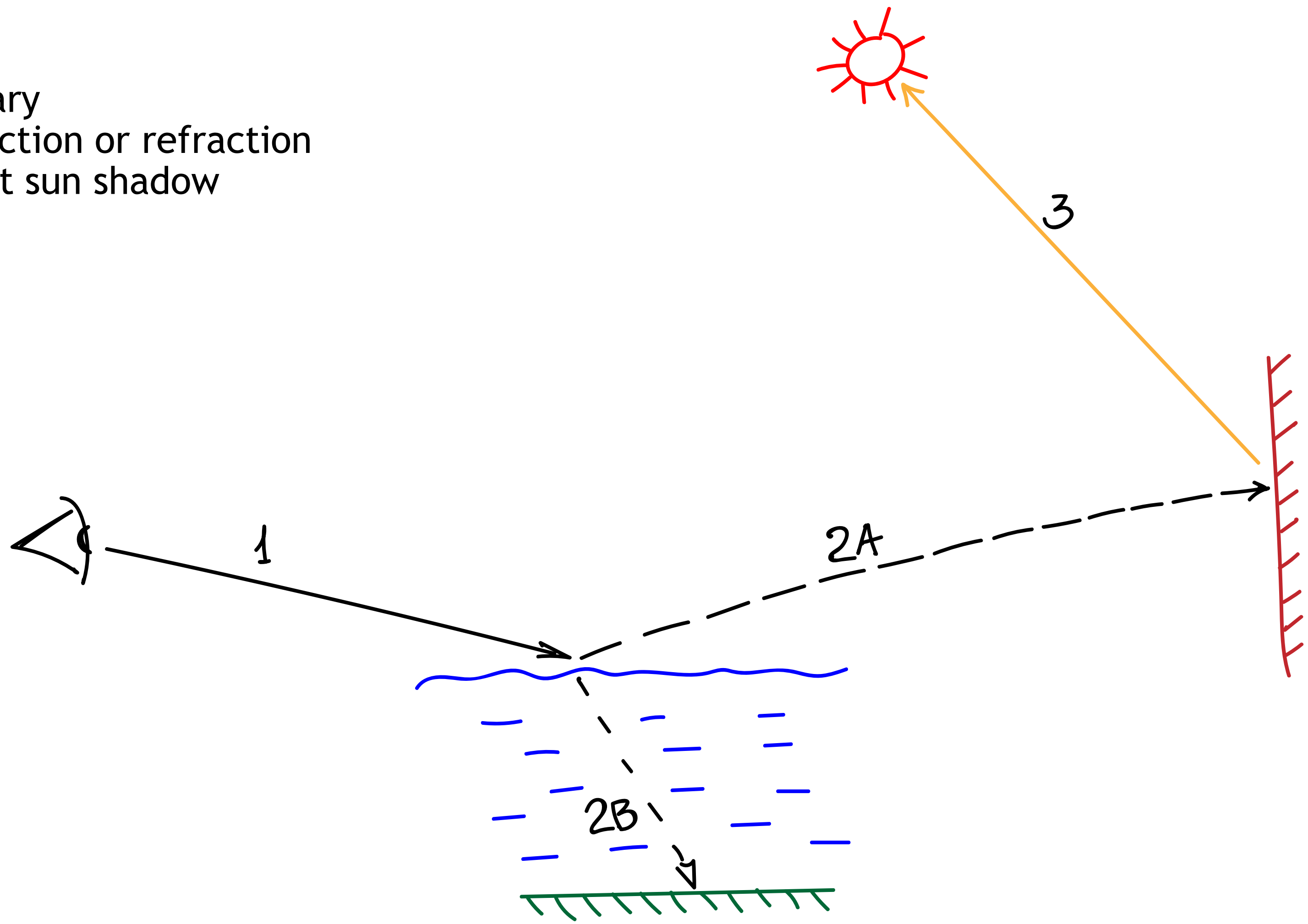
Path Tracer Overview

1. Primary
2. Reflection or refraction



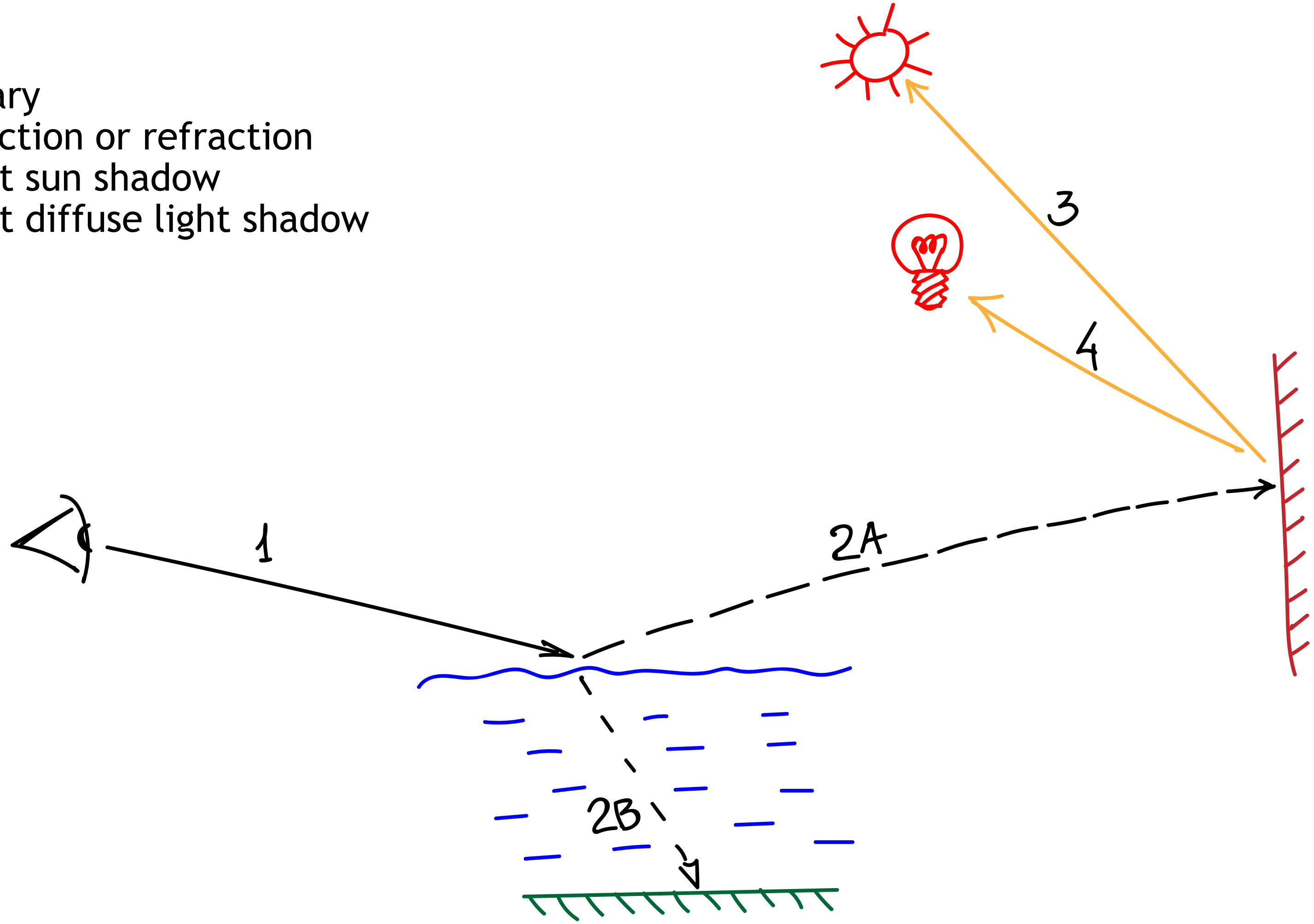
Path Tracer Overview

1. Primary
2. Reflection or refraction
3. Direct sun shadow



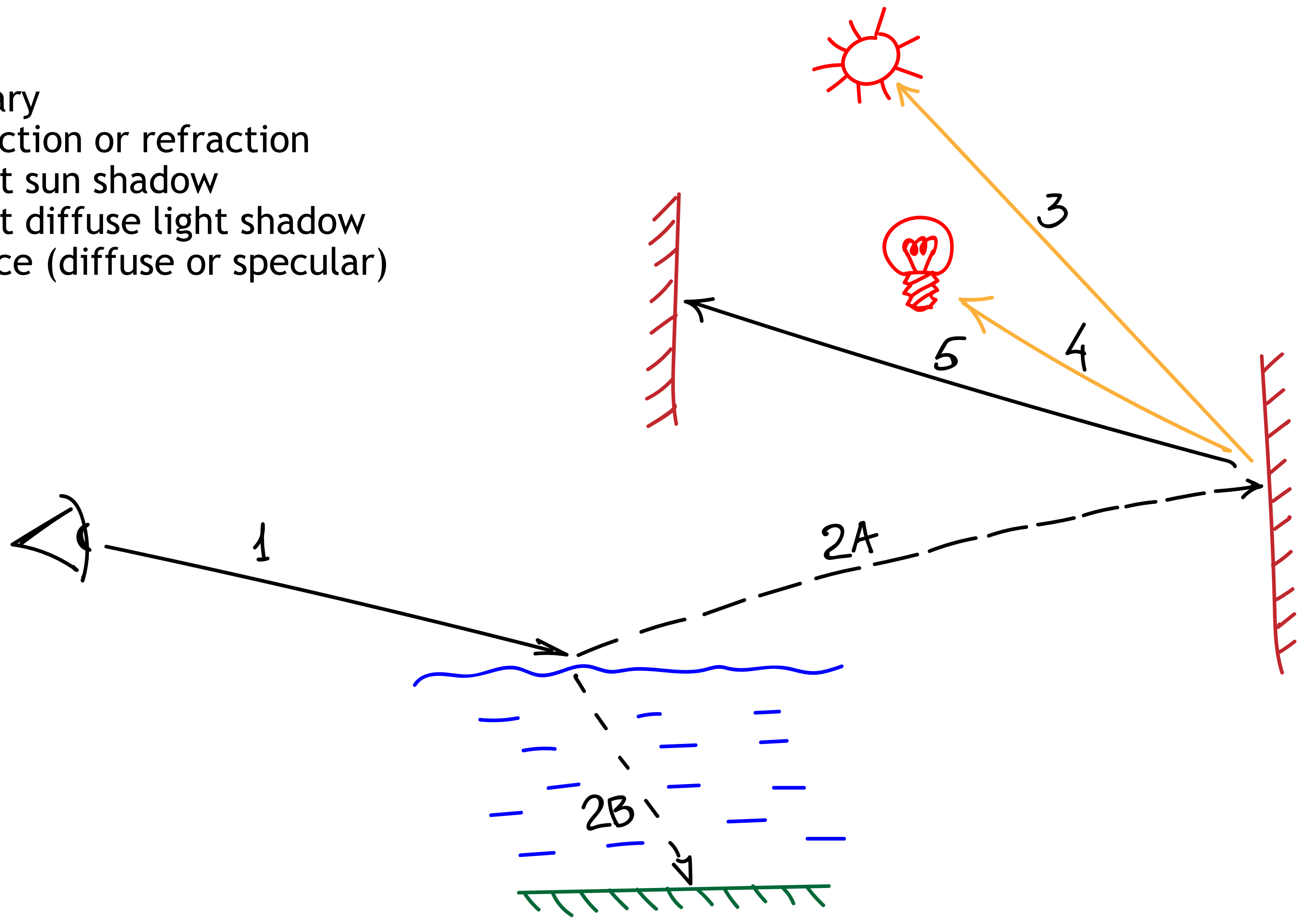
Path Tracer Overview

1. Primary
2. Reflection or refraction
3. Direct sun shadow
4. Direct diffuse light shadow



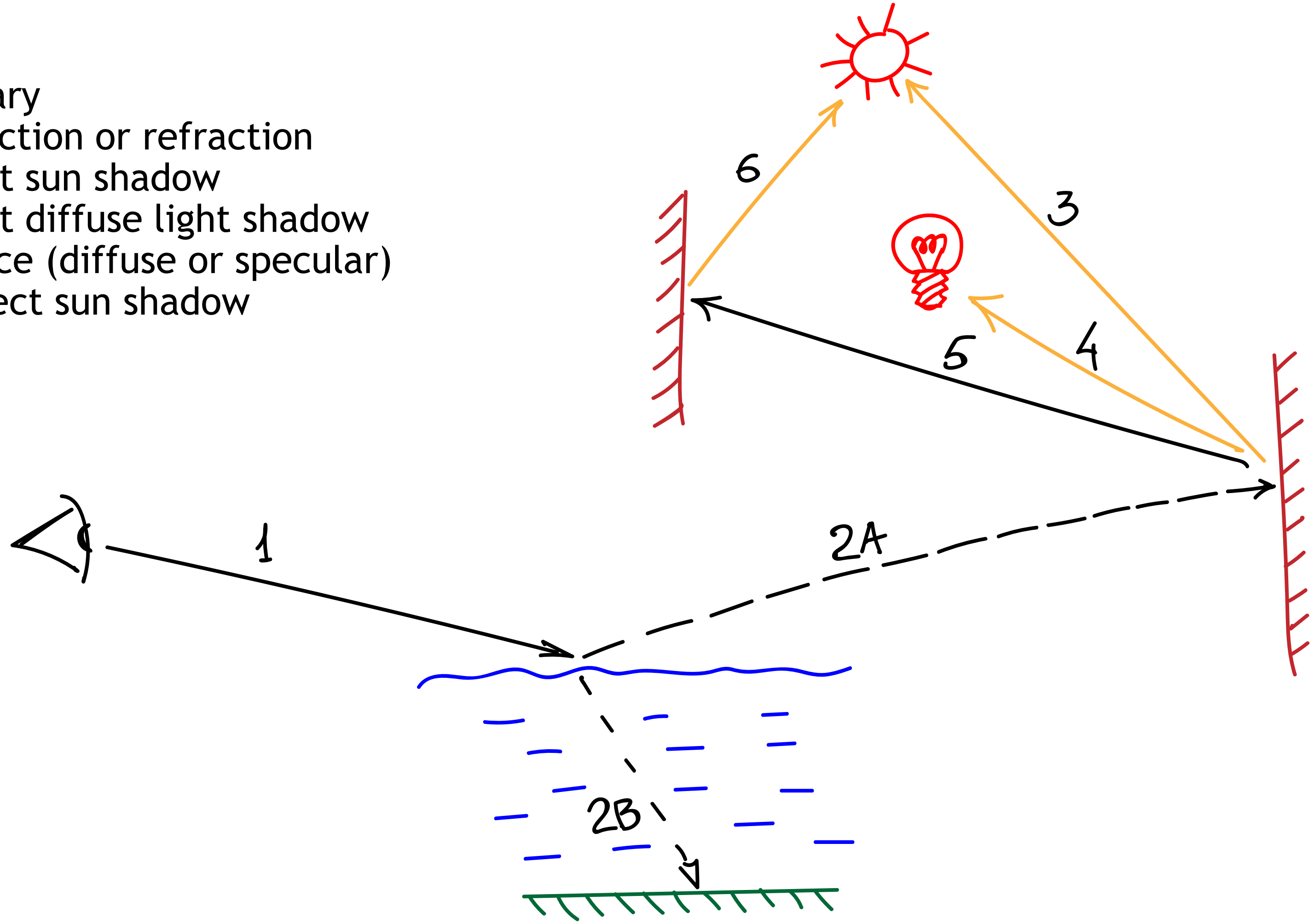
Path Tracer Overview

1. Primary
2. Reflection or refraction
3. Direct sun shadow
4. Direct diffuse light shadow
5. Bounce (diffuse or specular)



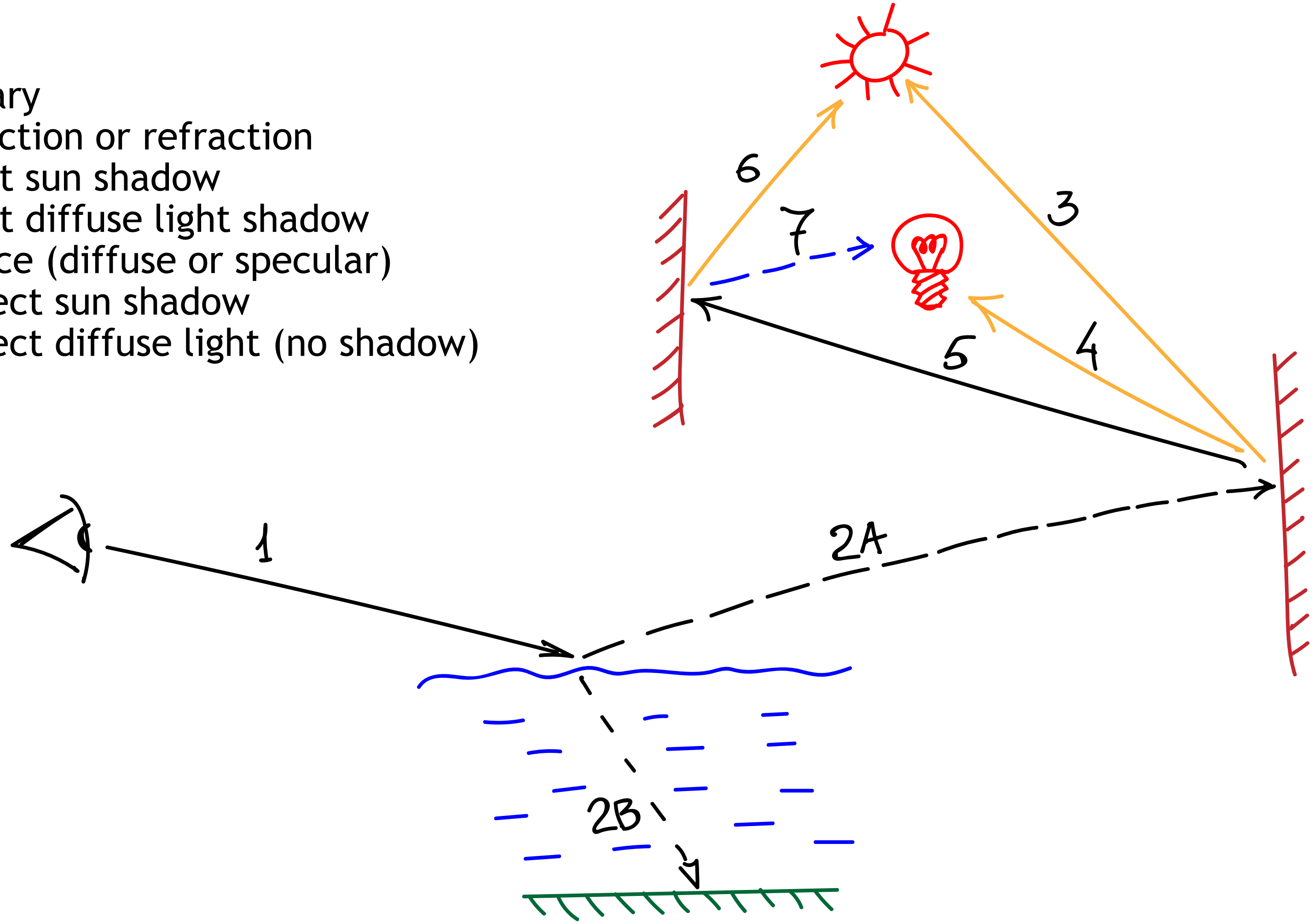
Path Tracer Overview

1. Primary
2. Reflection or refraction
3. Direct sun shadow
4. Direct diffuse light shadow
5. Bounce (diffuse or specular)
6. Indirect sun shadow



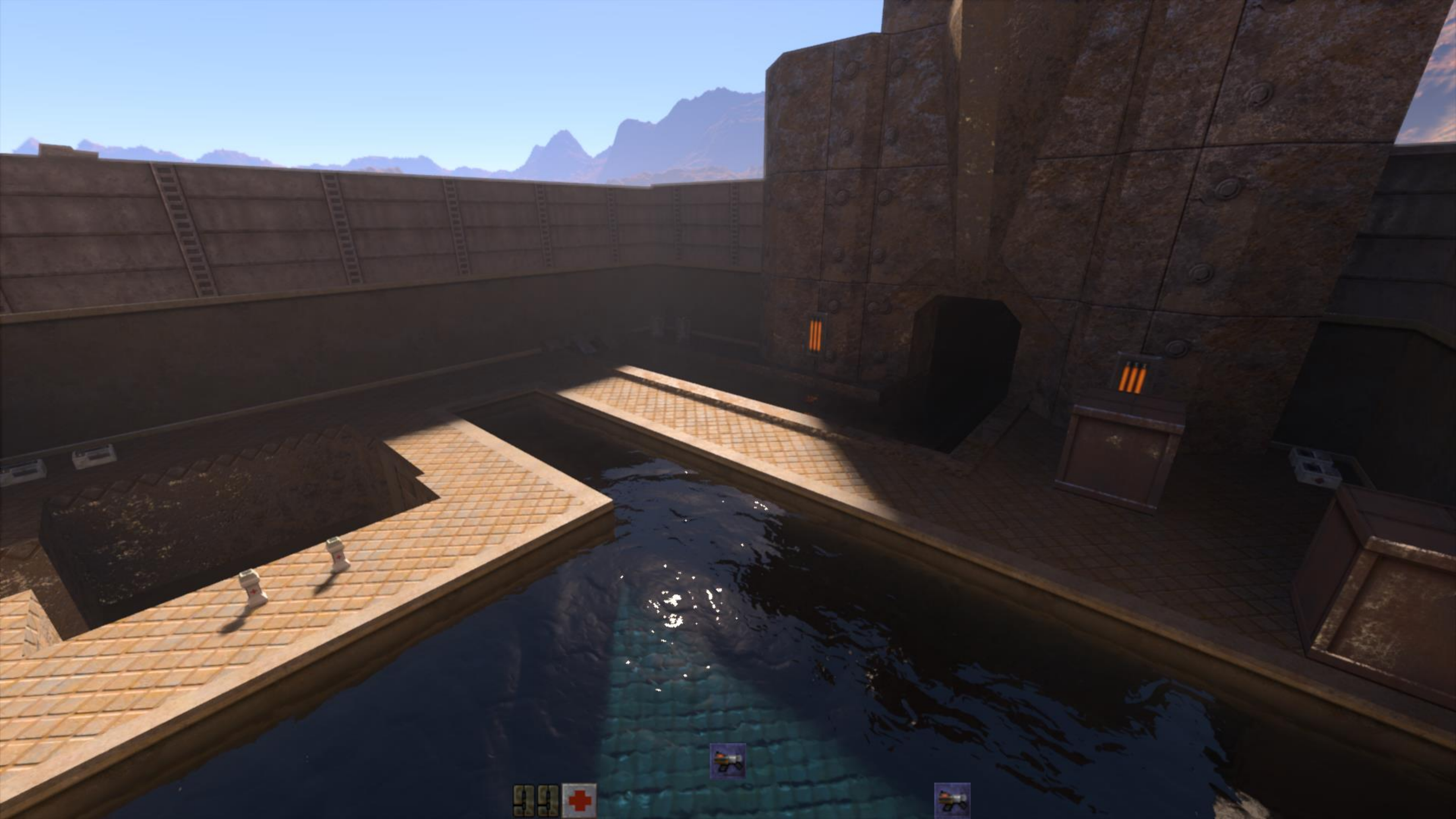
Path Tracer Overview

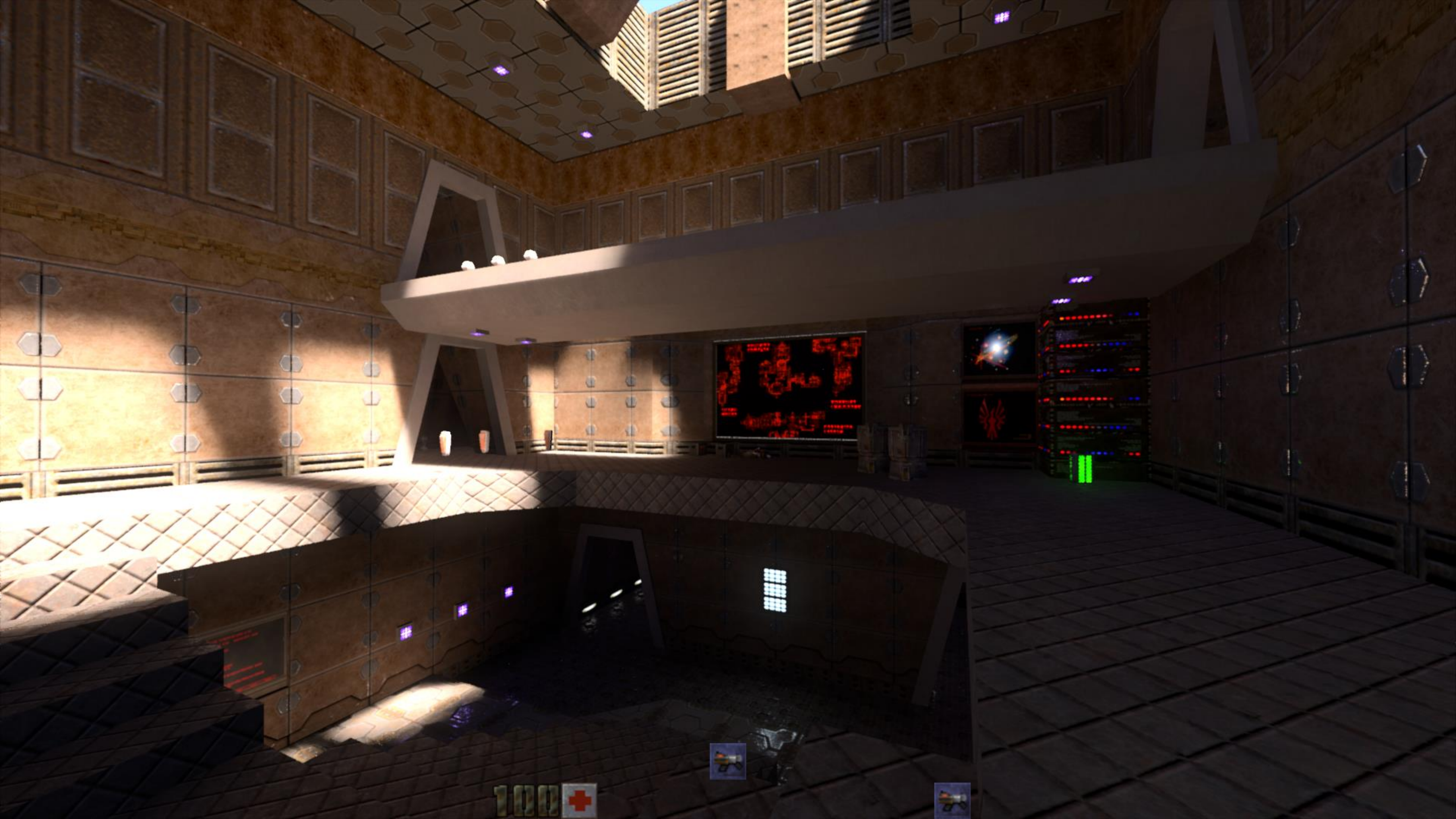
1. Primary
2. Reflection or refraction
3. Direct sun shadow
4. Direct diffuse light shadow
5. Bounce (diffuse or specular)
6. Indirect sun shadow
7. Indirect diffuse light (no shadow)




100

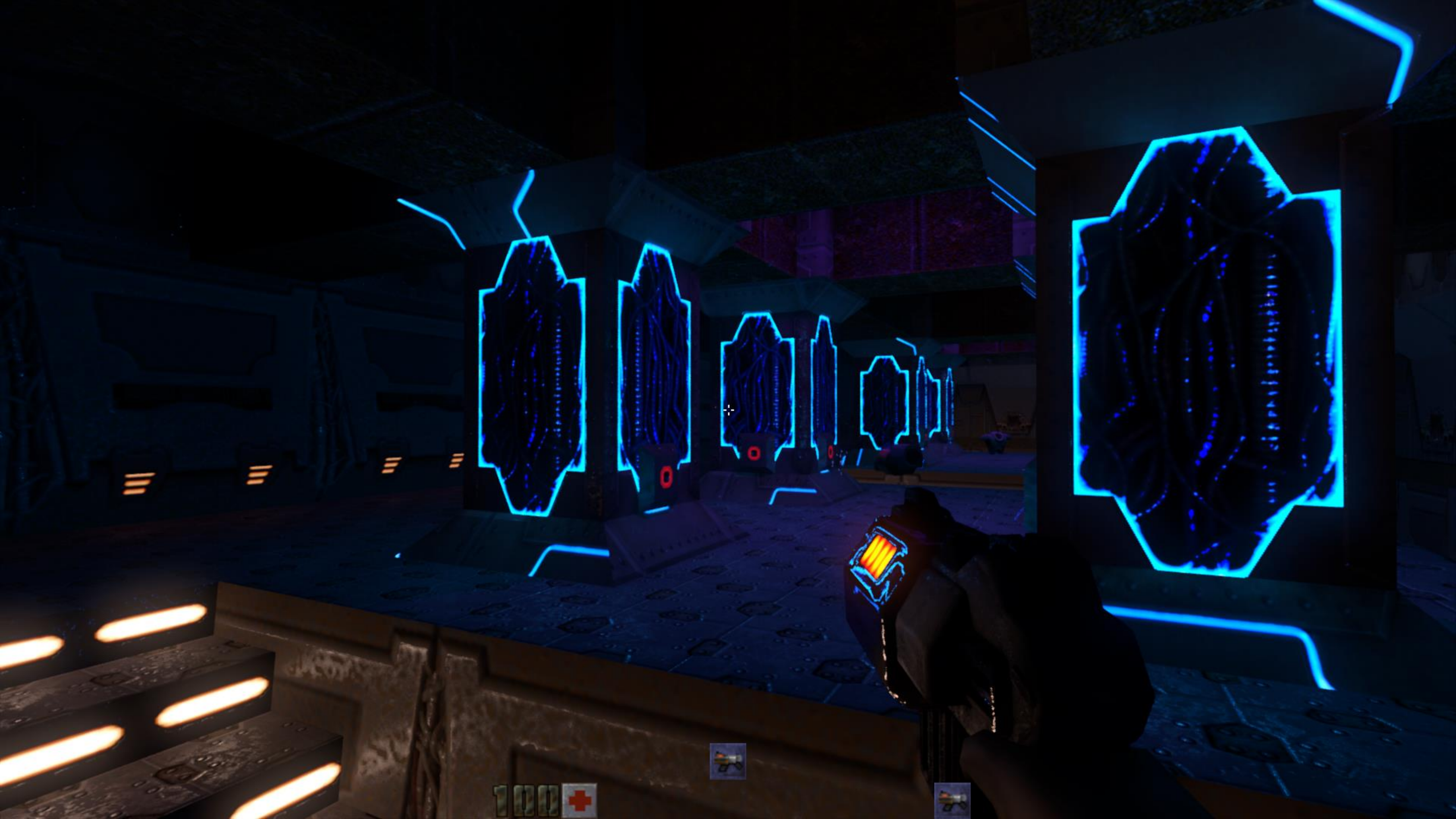

100


99


100


100


100


100


COMM
INSTALLATION
100


99


100

Development Team

- Q2VKPT to keynote version: 6 weeks (Feb-Mar 2019)
- A group of passionate engineers and artists:

Adam Moss	Gavriil Klimov	Liam Middlebrook
Alex Dunn	Gregor Kopka	Manuel Kraemer
Alexey Panteleev	Grigoriy Odegov	Matthew Rusch
Andrew Reidmeyer	Ivan Fedorov	Matthijs De Smedt
Dane Johnston	James Jones	Nuno Subtil
Eric Werness	Johnny Costello	Oleg Arutyunyan

Acknowledgements

- We’d like to thank the modding community:
  - “Kirk Barnes” for Quake II XP
  - Arthur “Turic” Galactionov for the weapon models
  - D Scott Boyce for some of the textures
  - Potentially others whose assets found their way into Quake II RTX through a chain of mods
- We’d also like to thank id Software for the original Quake II

QUESTIONS

Christoph Schied | schied@brechpunkt.de @c_schied

Alexey Panteleev | alpanteleev@nvidia.com