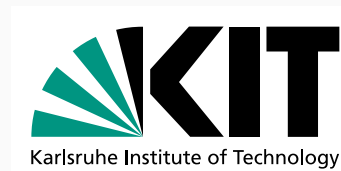


# ReSTIR Subsurface Scattering for Real-Time Path Tracing

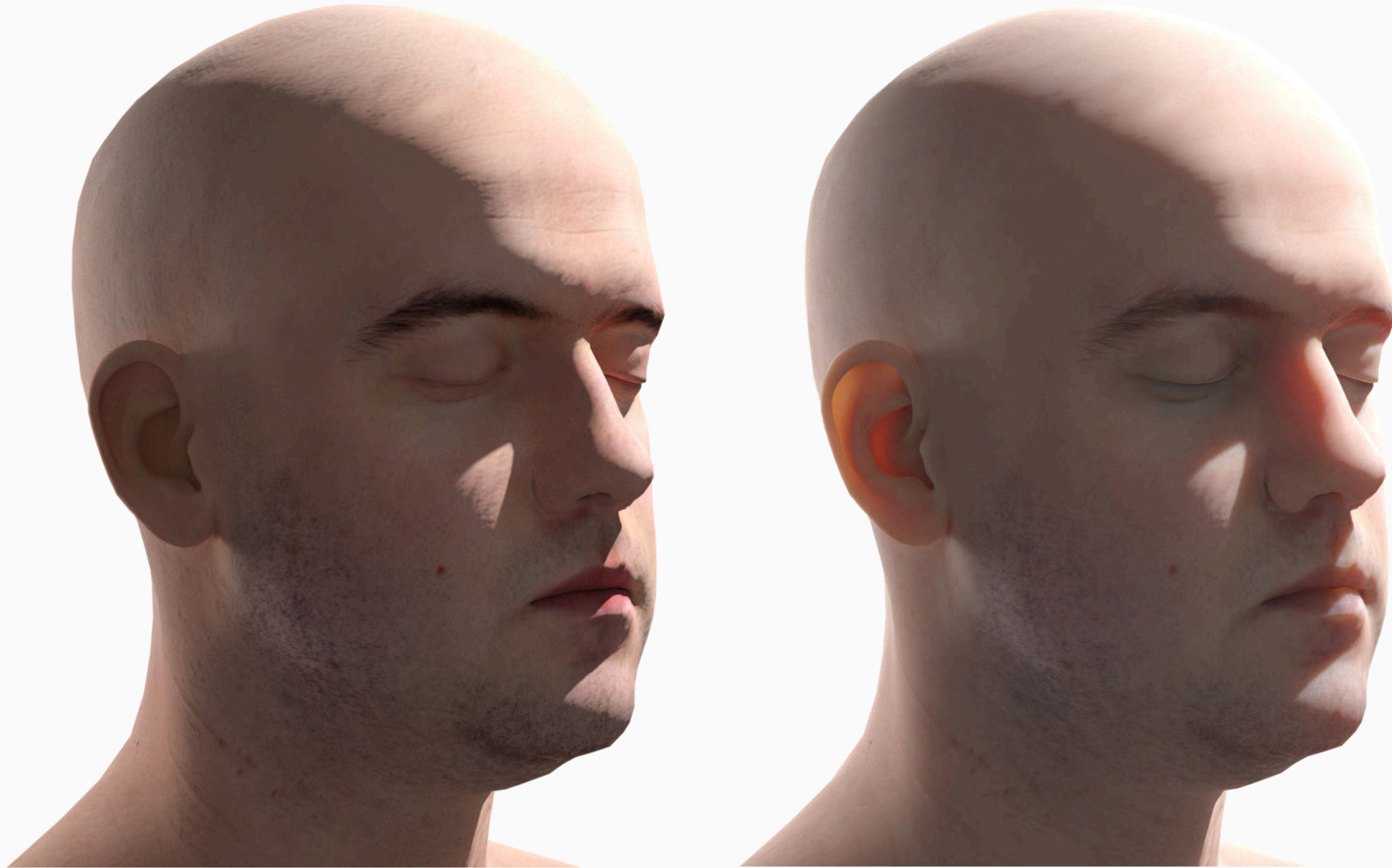
Mirco Werner, Vincent Schüßler, and Carsten Dachsbacher

Karlsruhe Institute of Technology



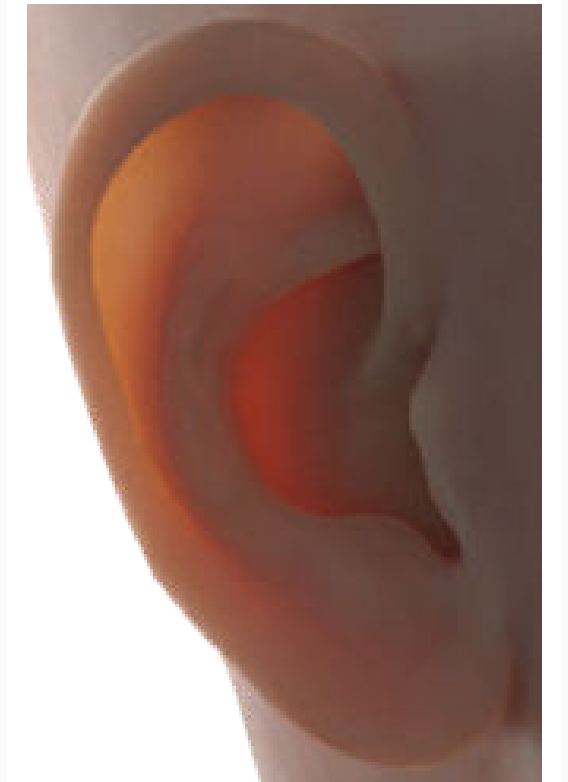
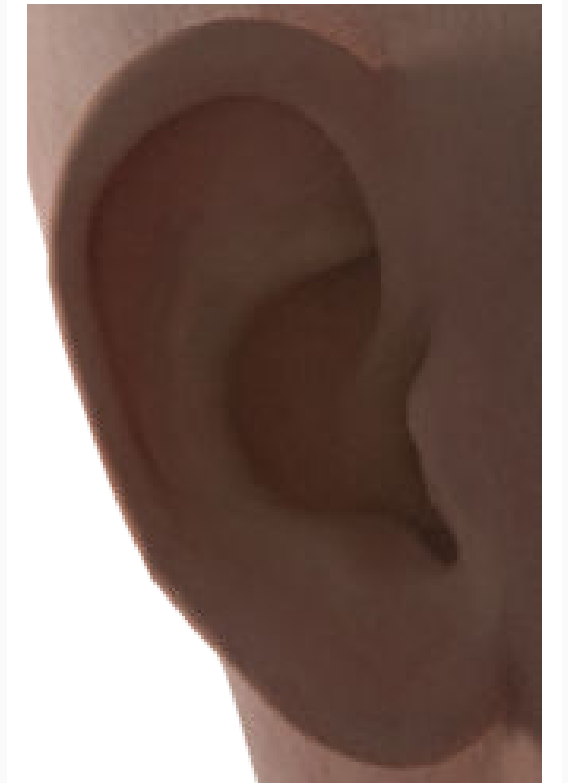
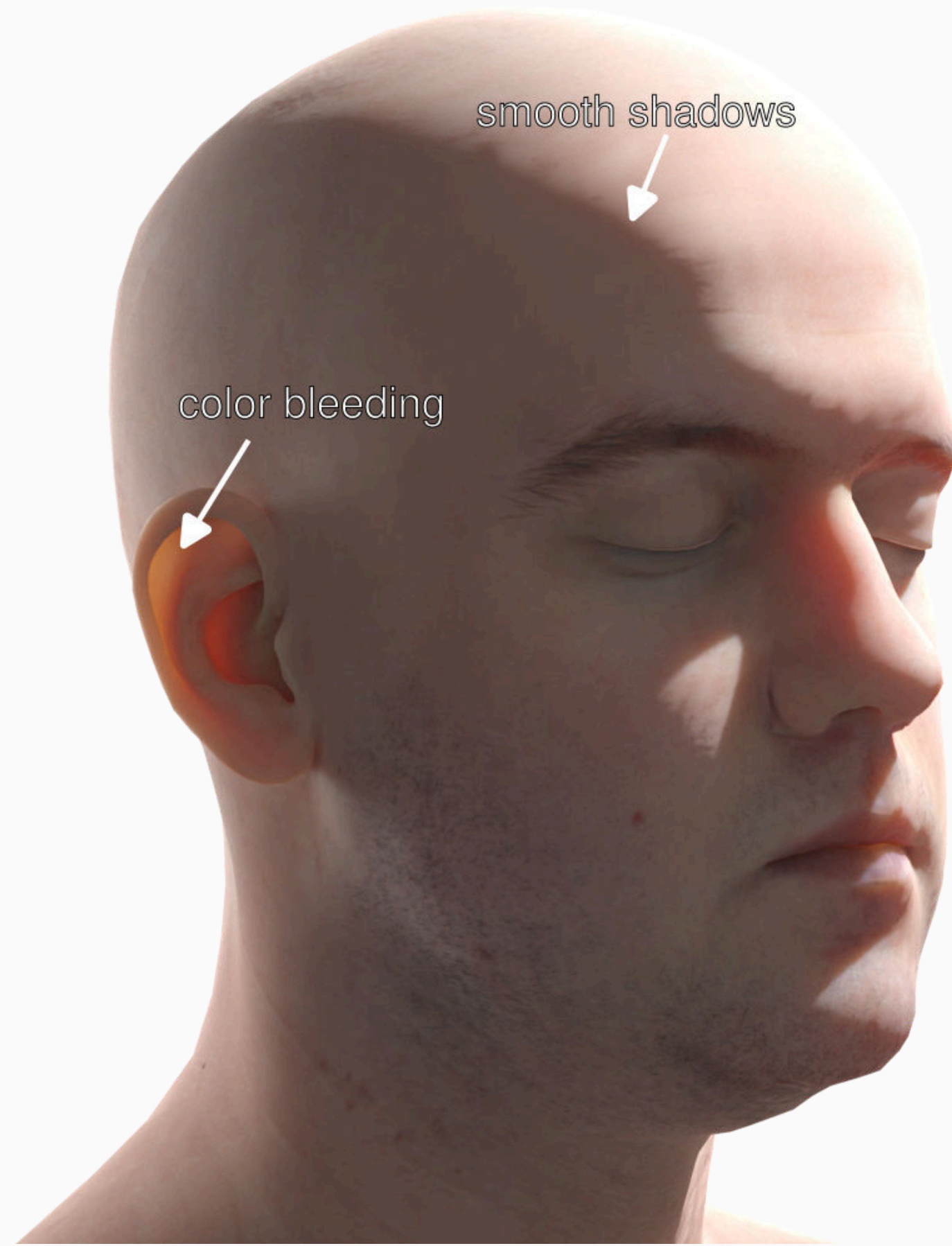


# Surface Light Transport vs. Subsurface Scattering (SSS)





# Surface Light Transport vs. Subsurface Scattering (SSS)

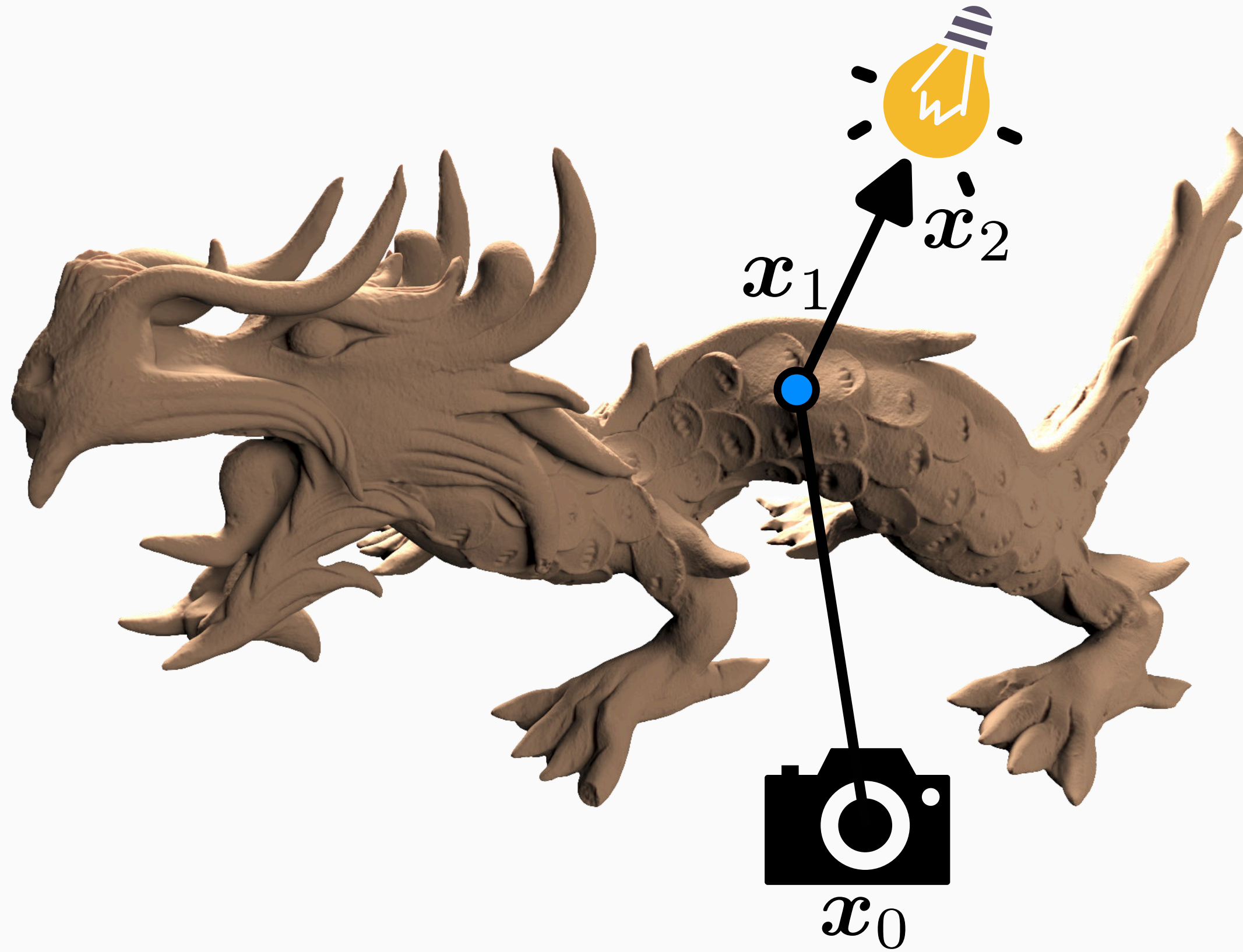


# Surface Light Transport vs. Subsurface Scattering (SSS)



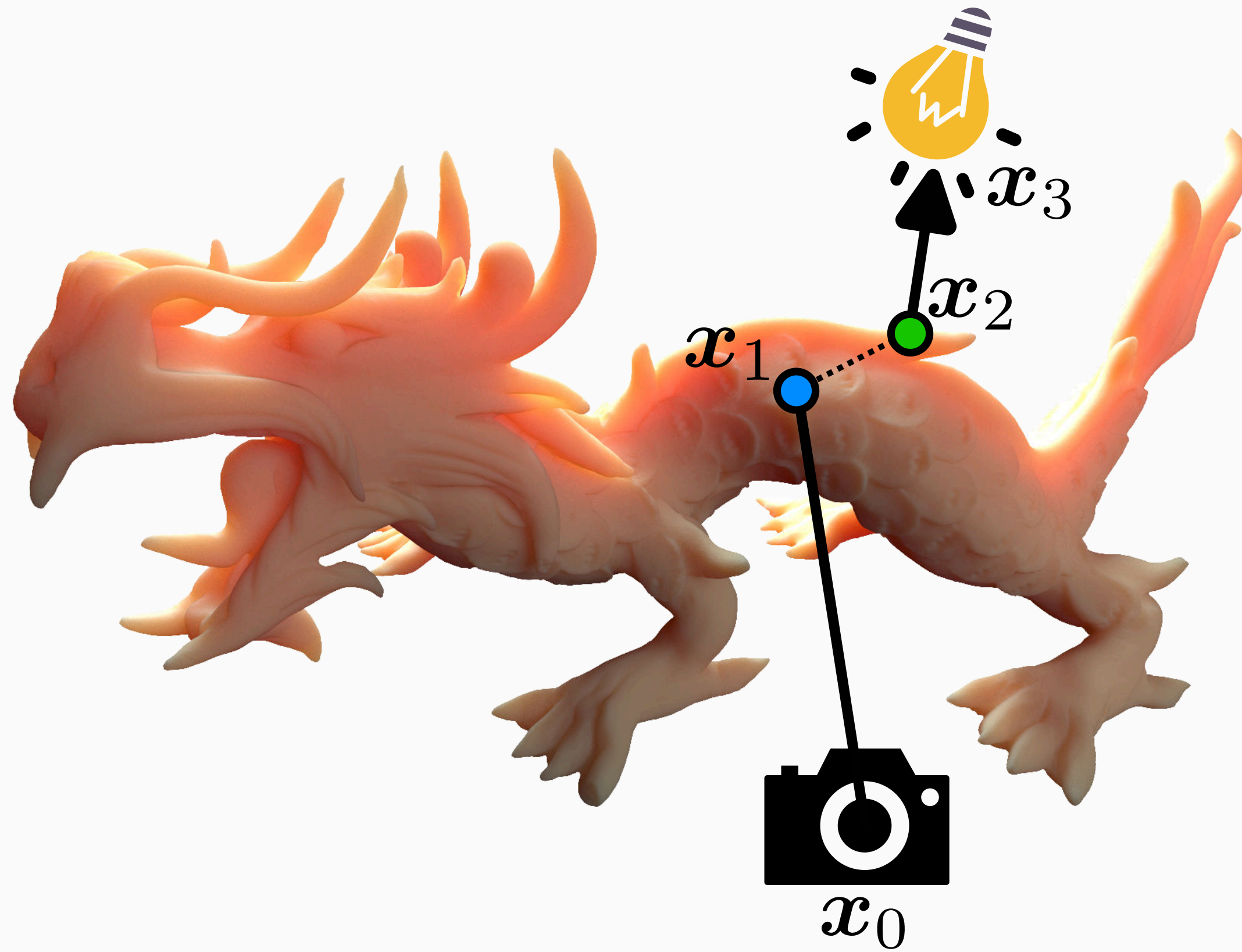


# Diffusion-based SSS for Path Tracing





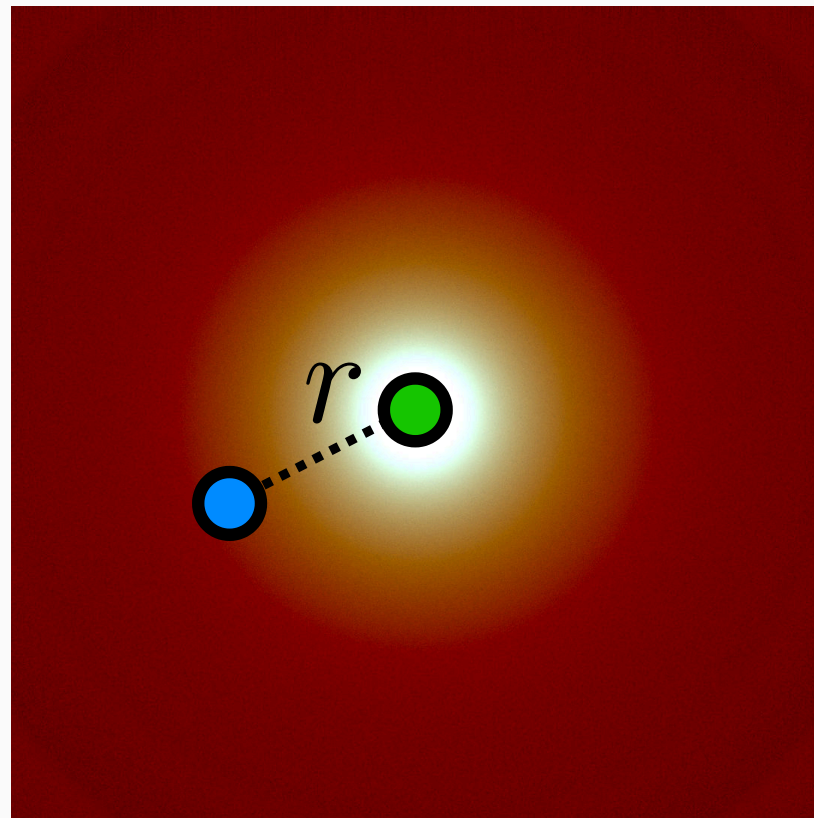
# Diffusion-based SSS for Path Tracing



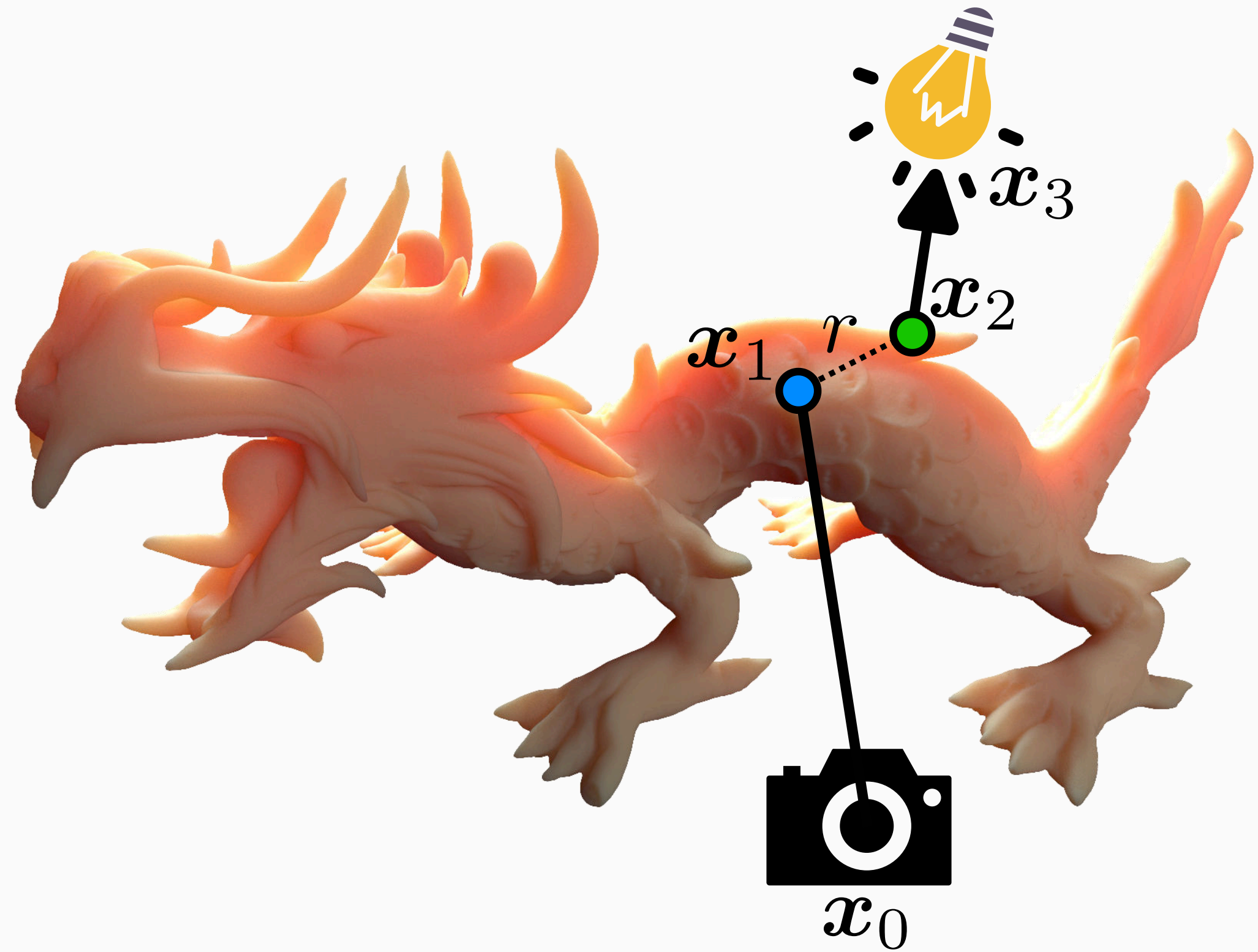


# Diffusion-based SSS for Path Tracing

- $f(\mathbf{x}) \propto R_d(r) \cdot L(\mathbf{x}_3)$



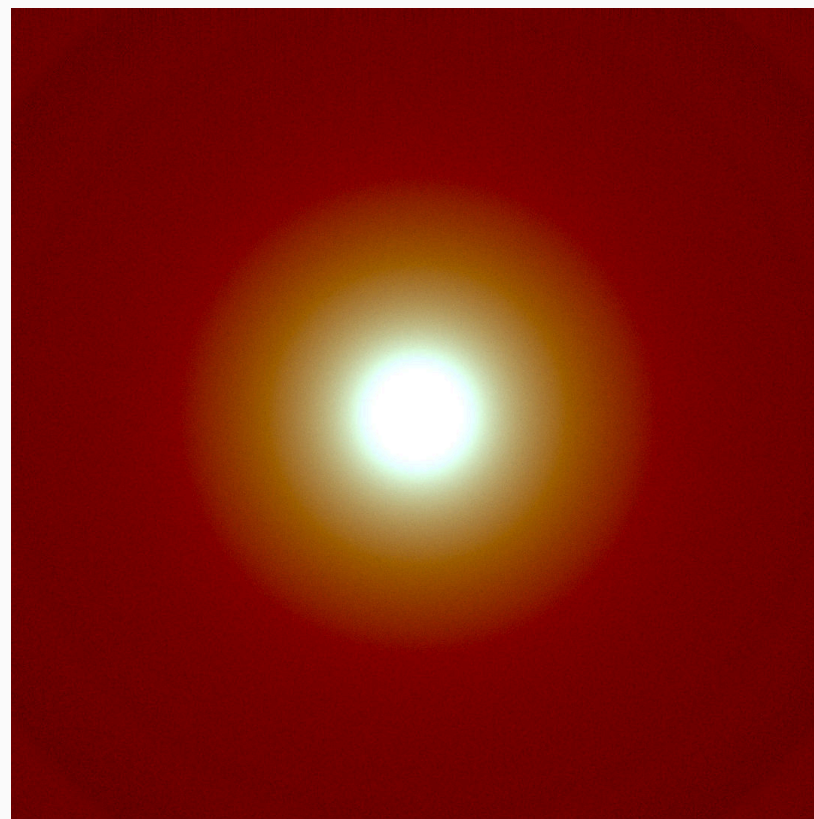
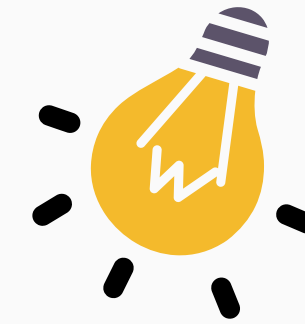
diffusion profile  $R_d(r)$



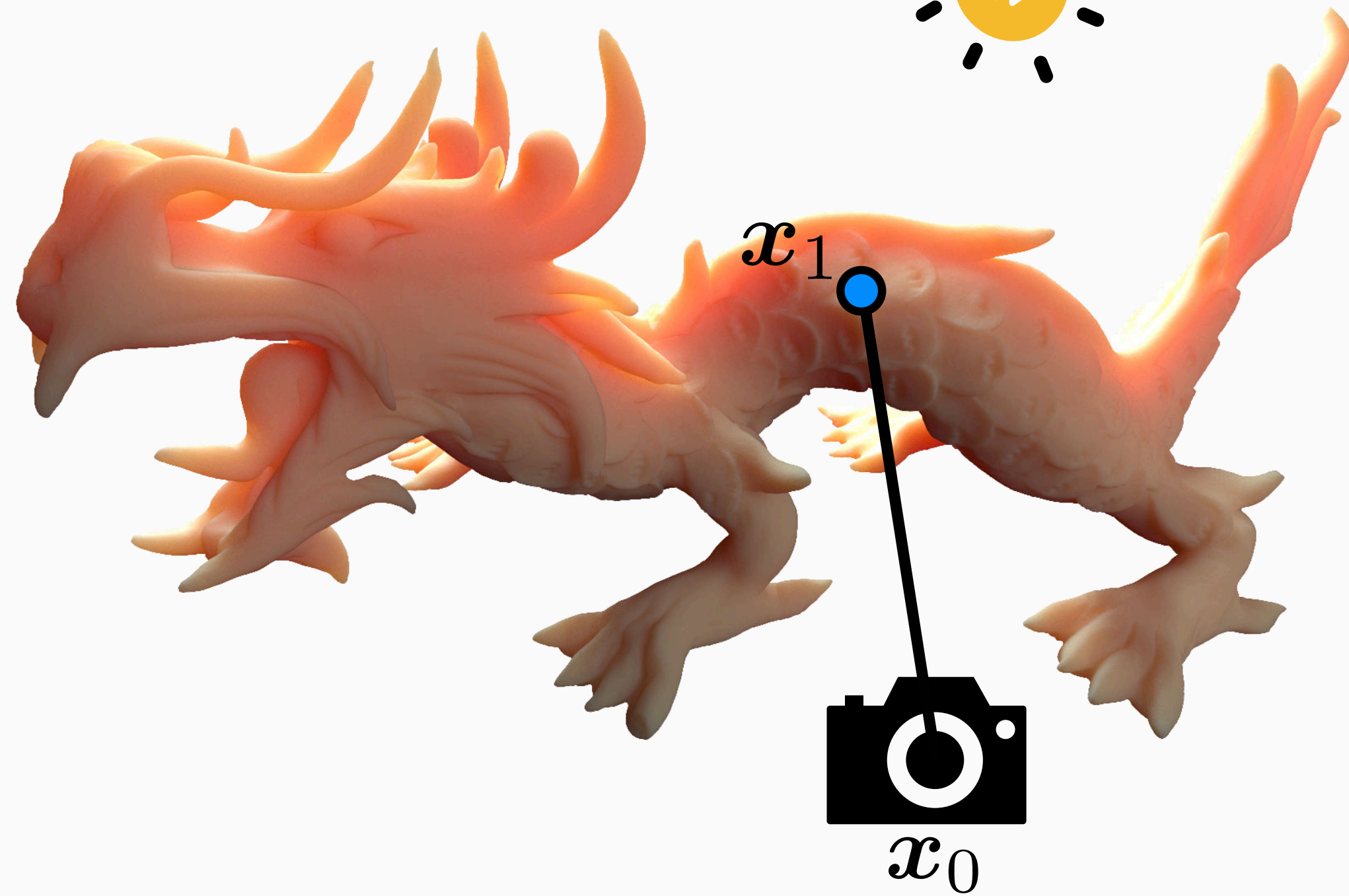


# Diffusion-based SSS for Path Tracing

- BSSRDF importance sampling [King et al. 2013]



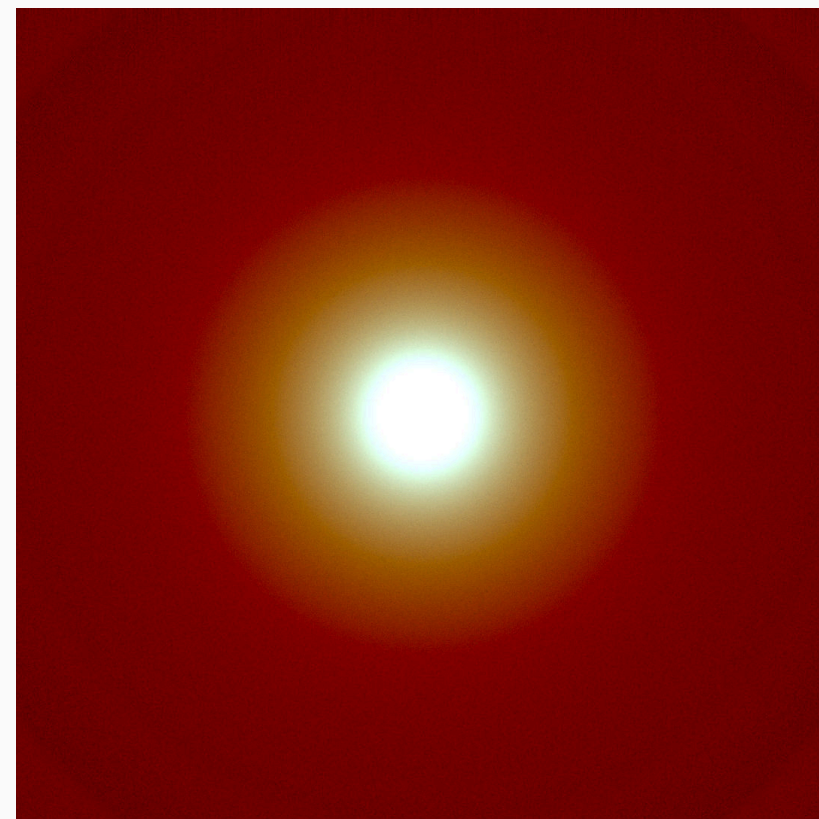
diffusion profile  $R_d(r)$



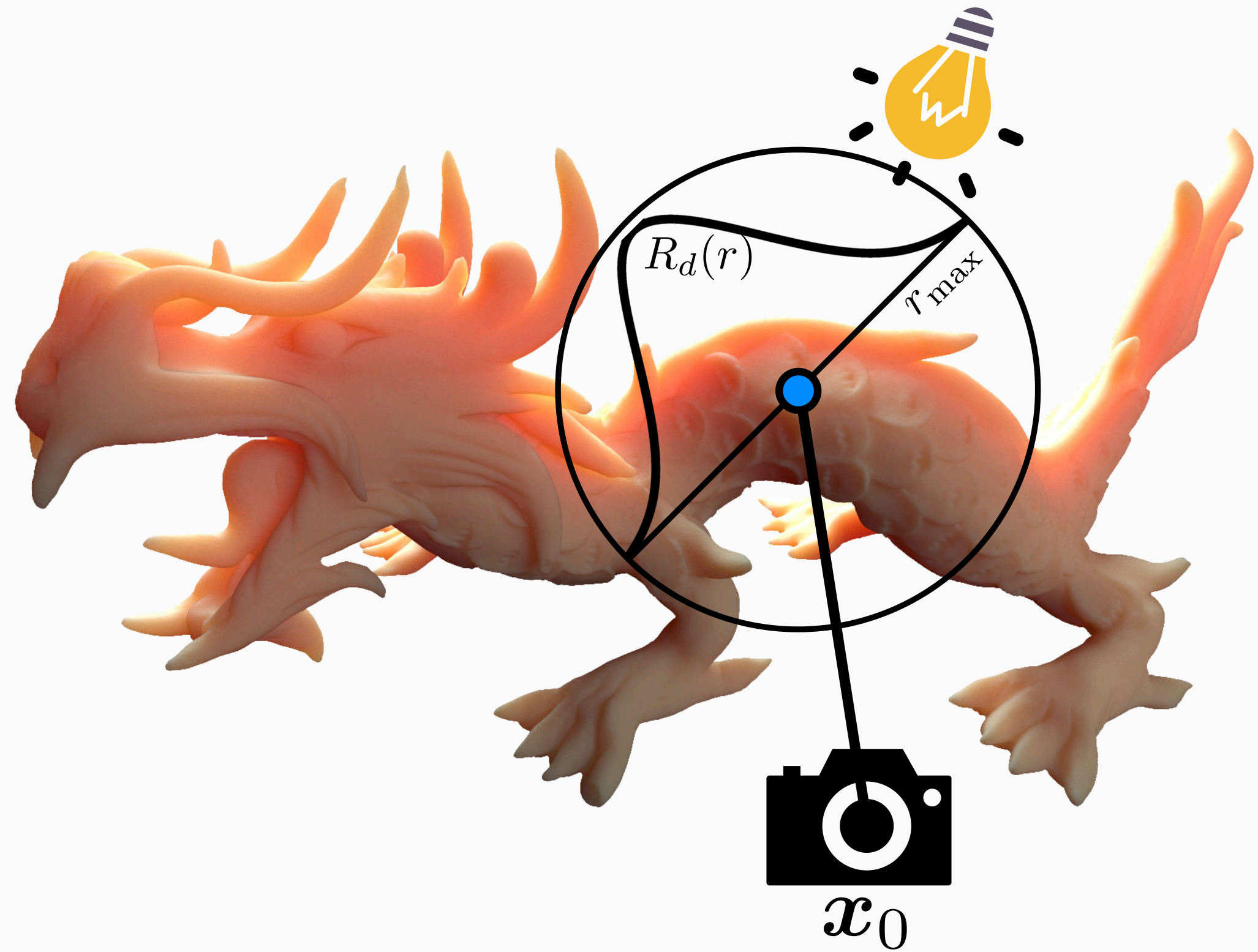


# Diffusion-based SSS for Path Tracing

- BSSRDF importance sampling [King et al. 2013]



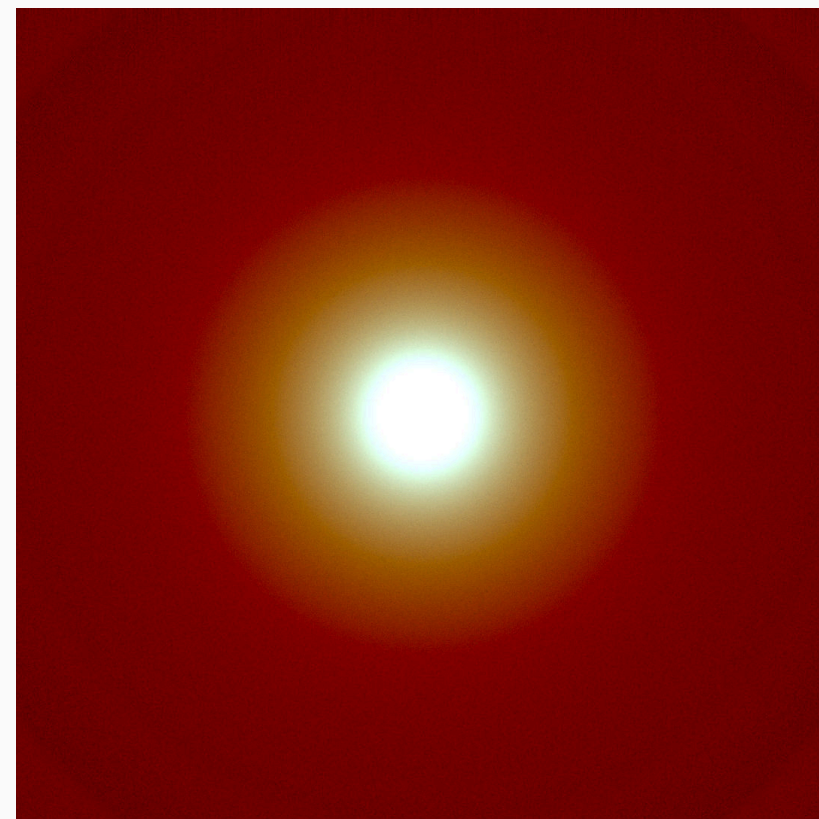
diffusion profile  $R_d(r)$



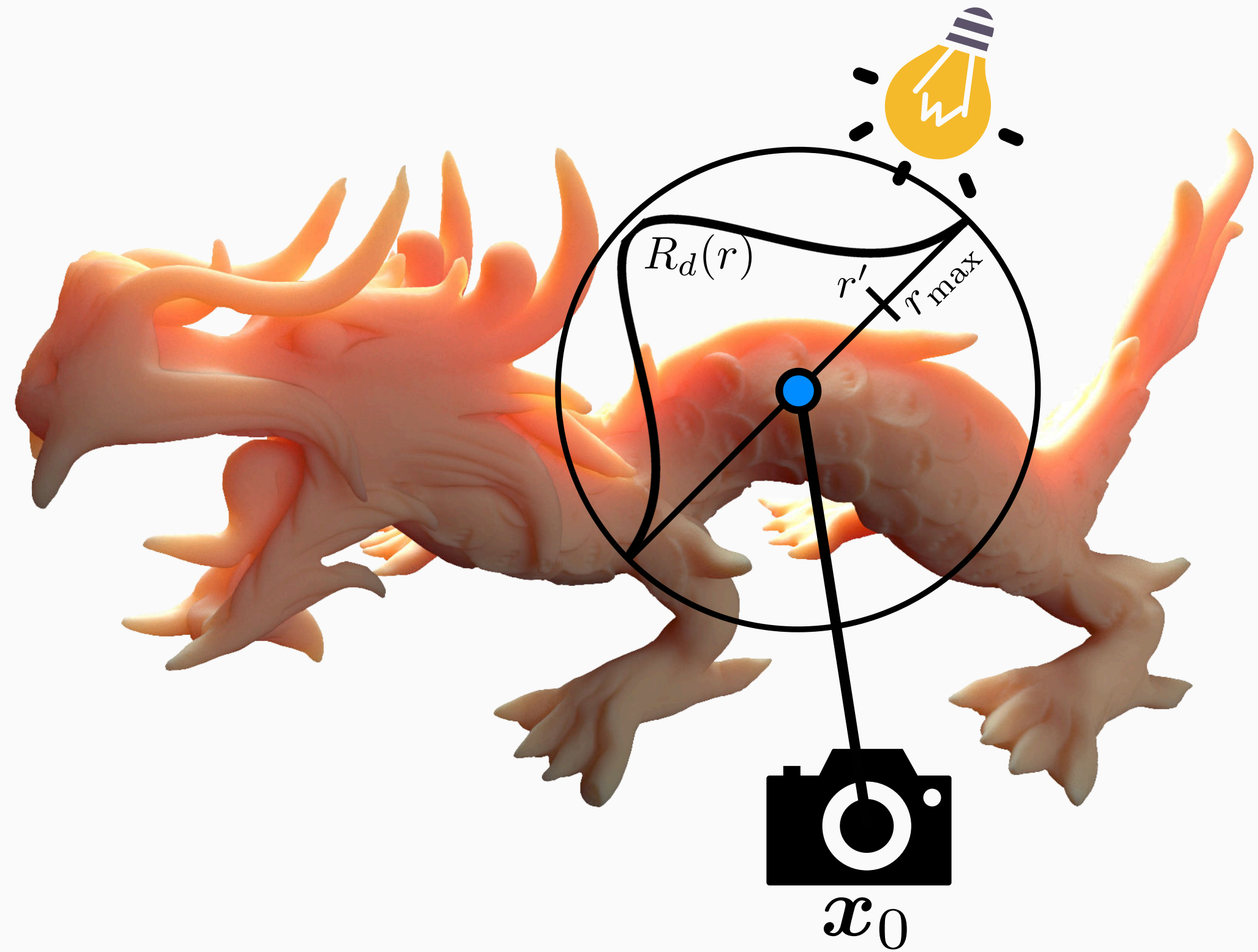


# Diffusion-based SSS for Path Tracing

- BSSRDF importance sampling [King et al. 2013]



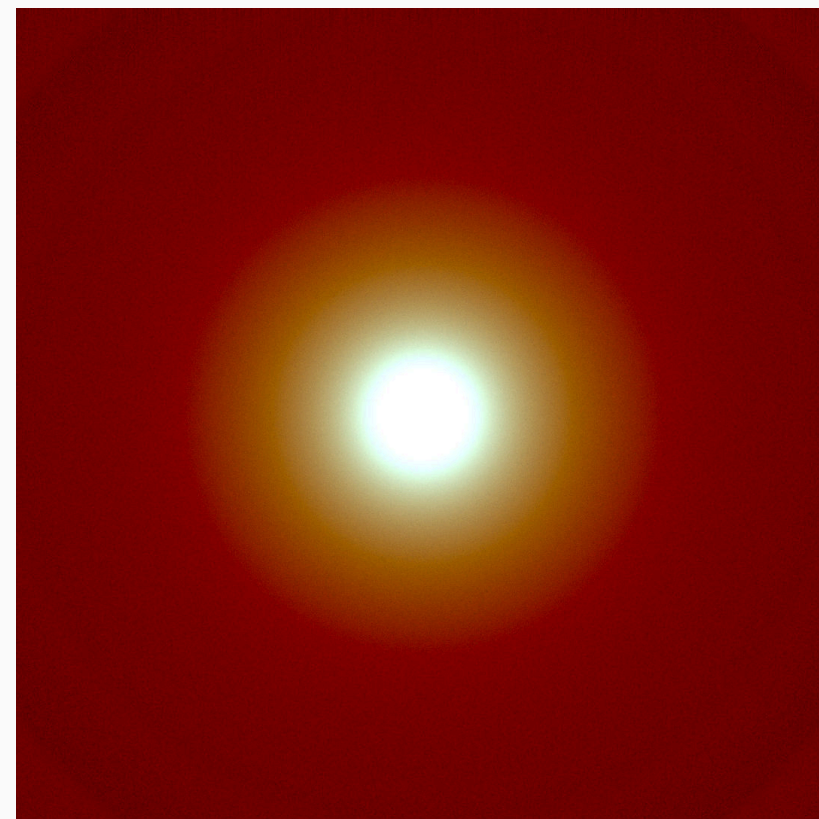
diffusion profile  $R_d(r)$



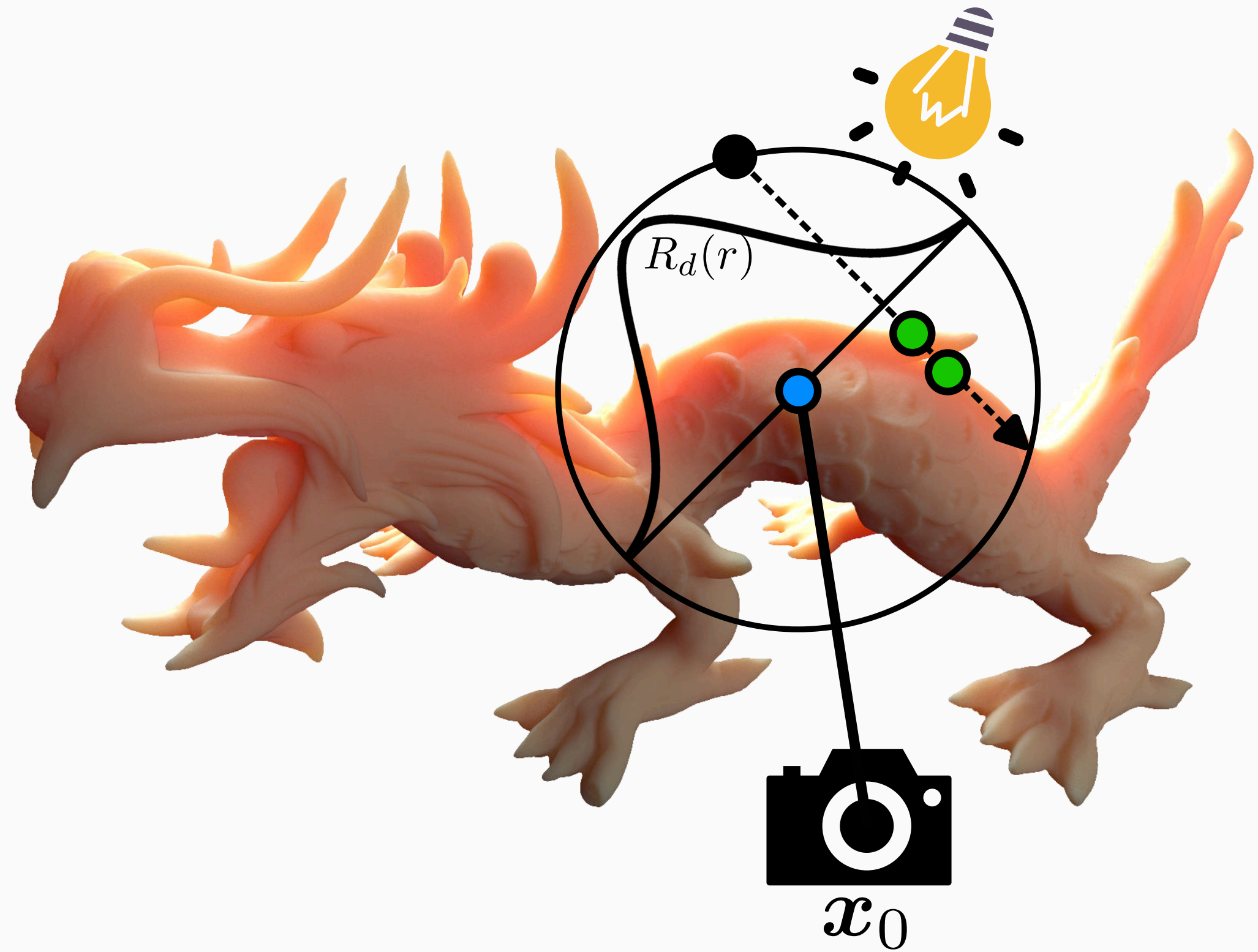


# Diffusion-based SSS for Path Tracing

- BSSRDF importance sampling [King et al. 2013]



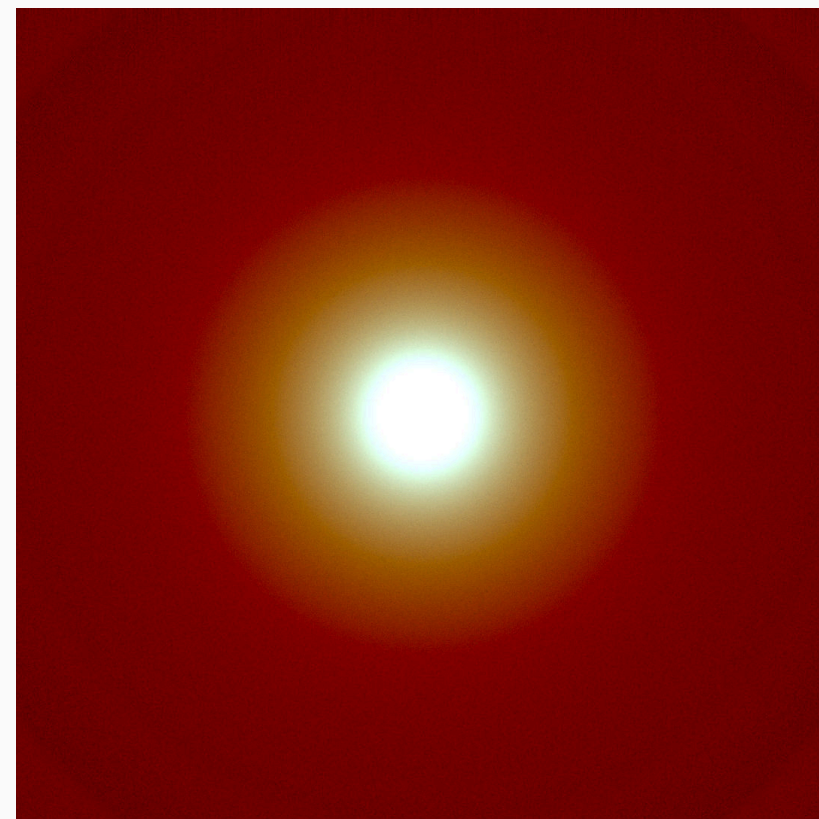
diffusion profile  $R_d(r)$



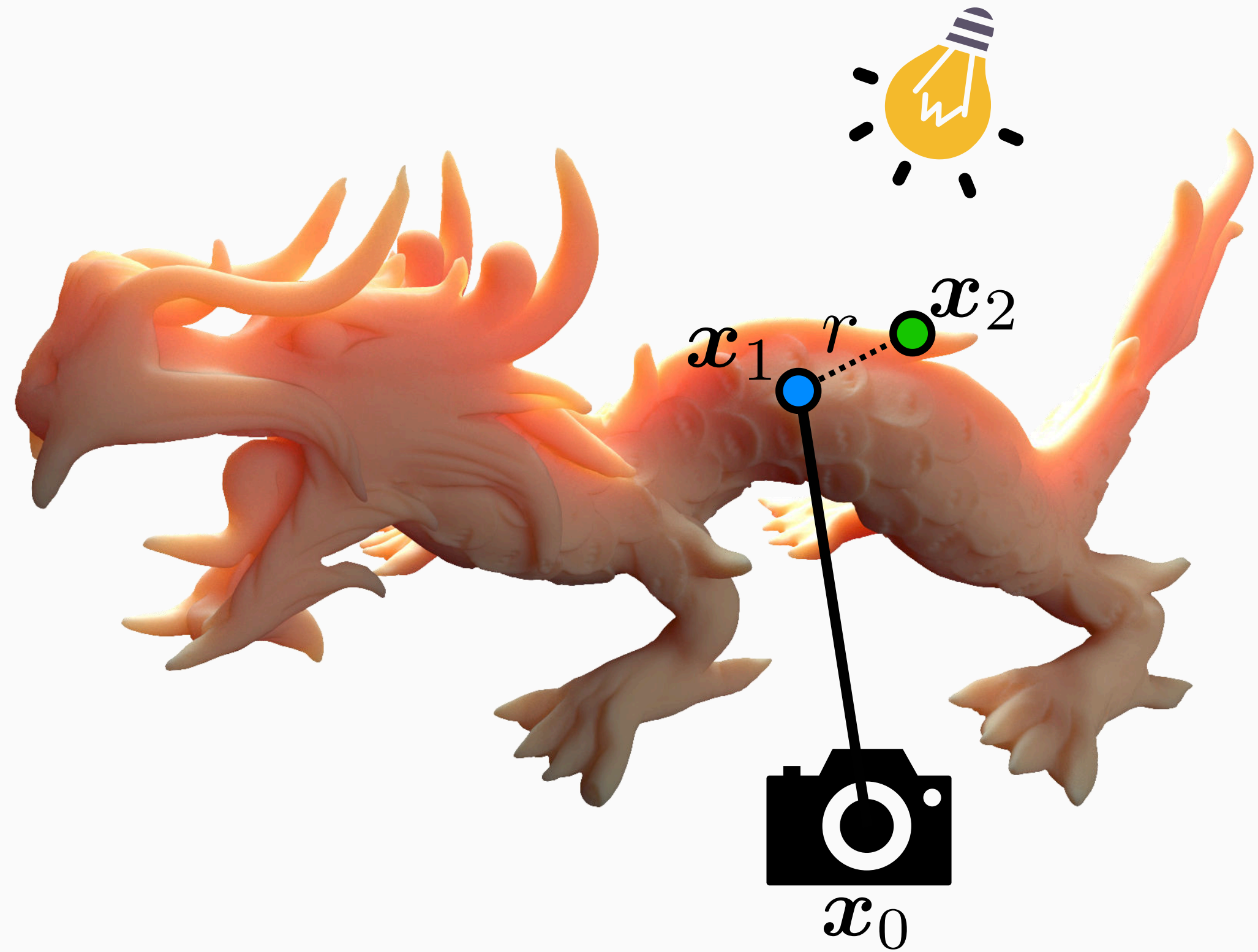


# Diffusion-based SSS for Path Tracing

- BSSRDF importance sampling [King et al. 2013]



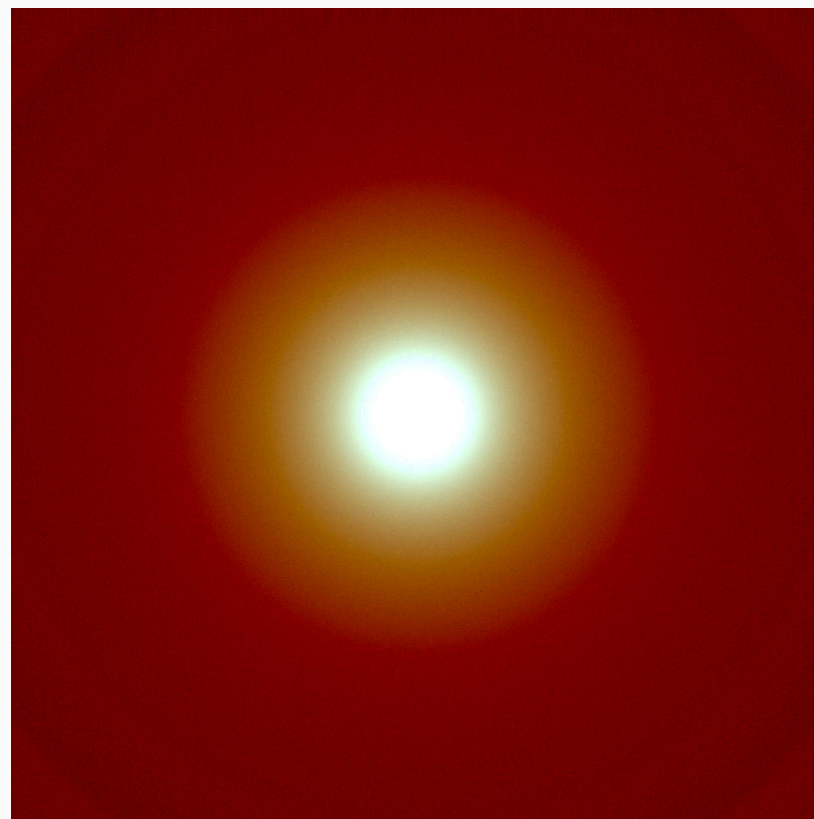
diffusion profile  $R_d(r)$



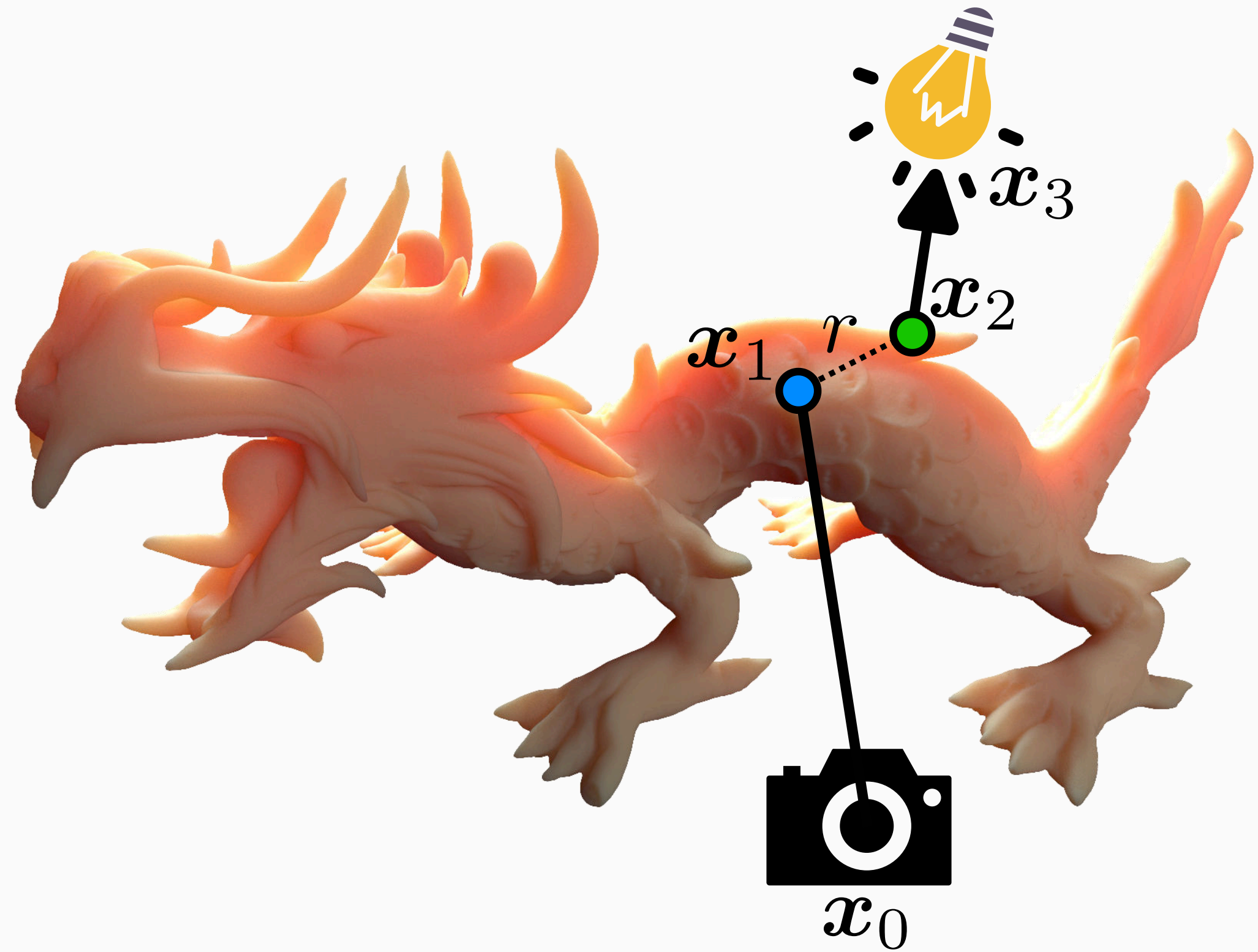


# Diffusion-based SSS for Path Tracing

- BSSRDF importance sampling [King et al. 2013]



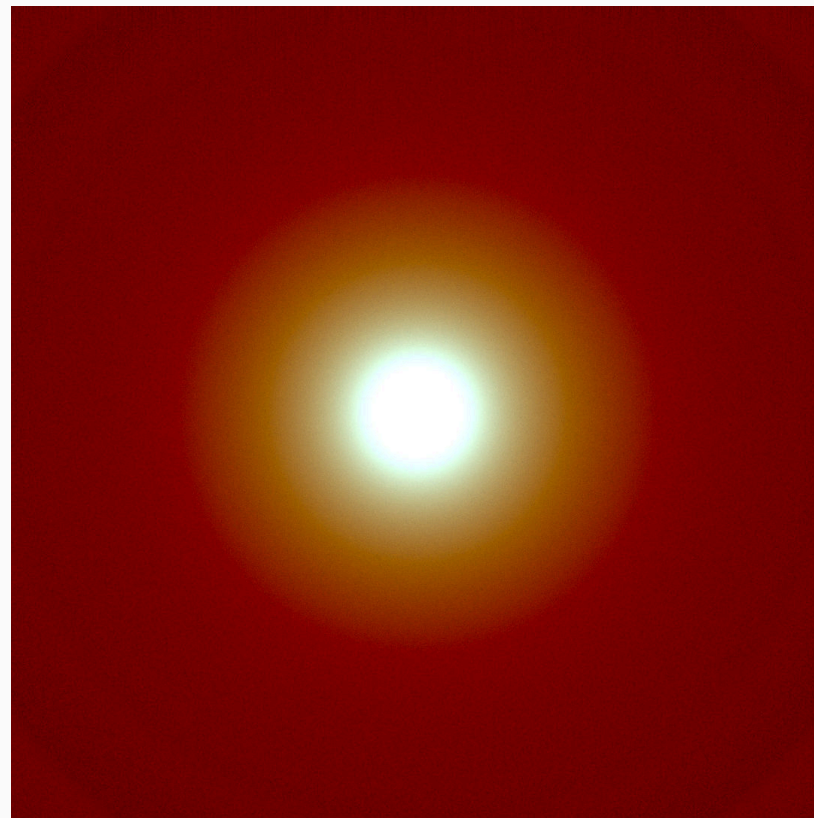
diffusion profile  $R_d(r)$



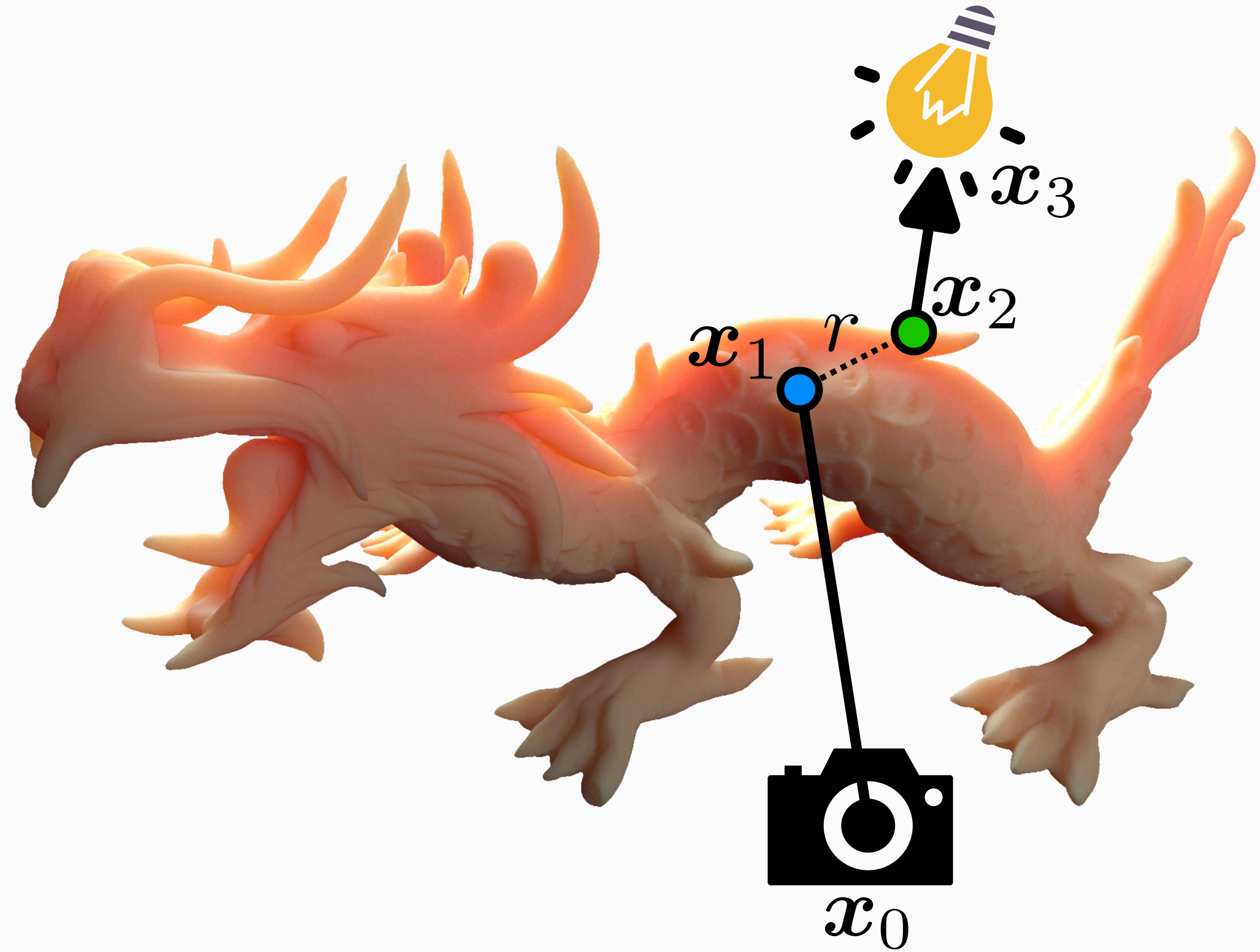


# Diffusion-based SSS for Path Tracing

- additional source of variance



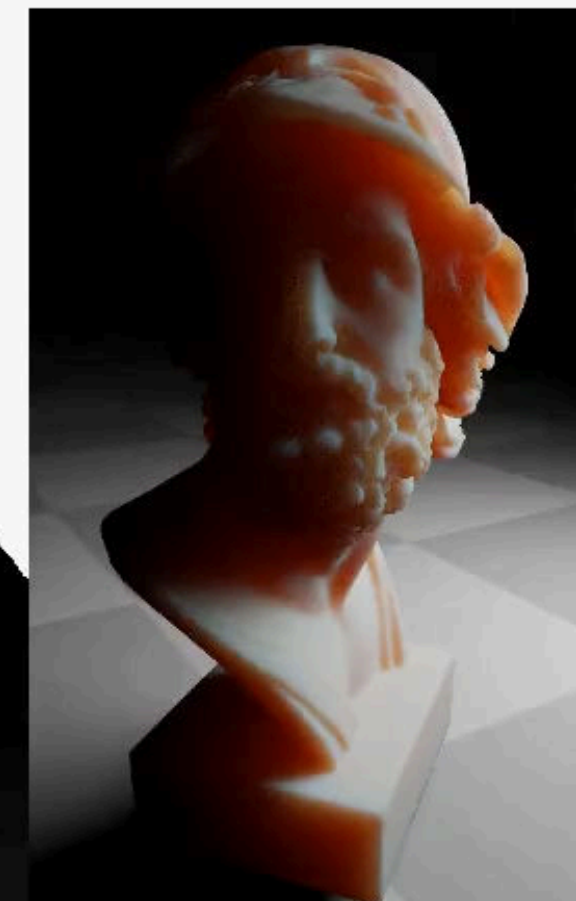
diffusion profile  $R_d(r)$







path tracing (1 spp)



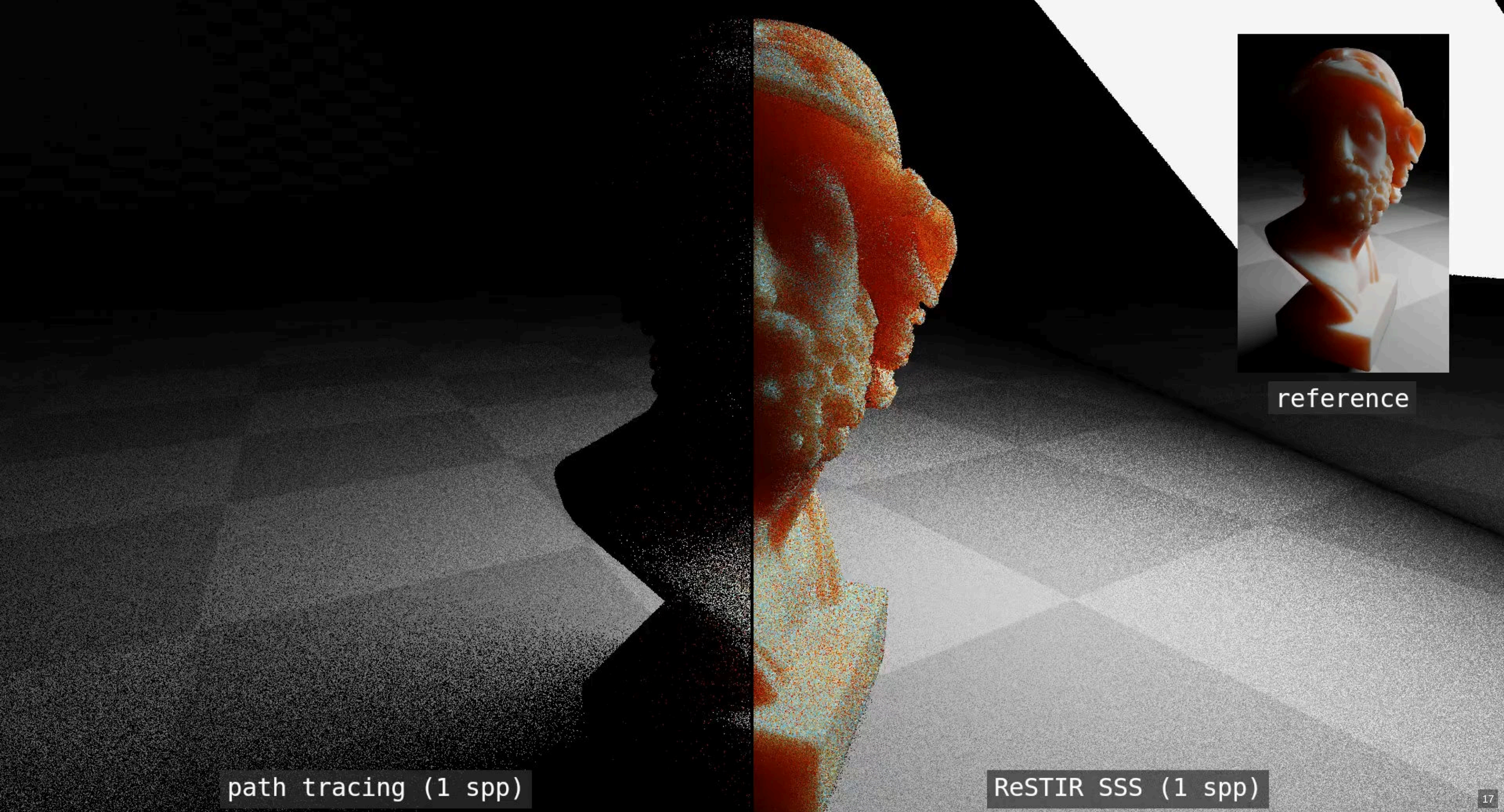
reference



# ReSTIR Subsurface Scattering

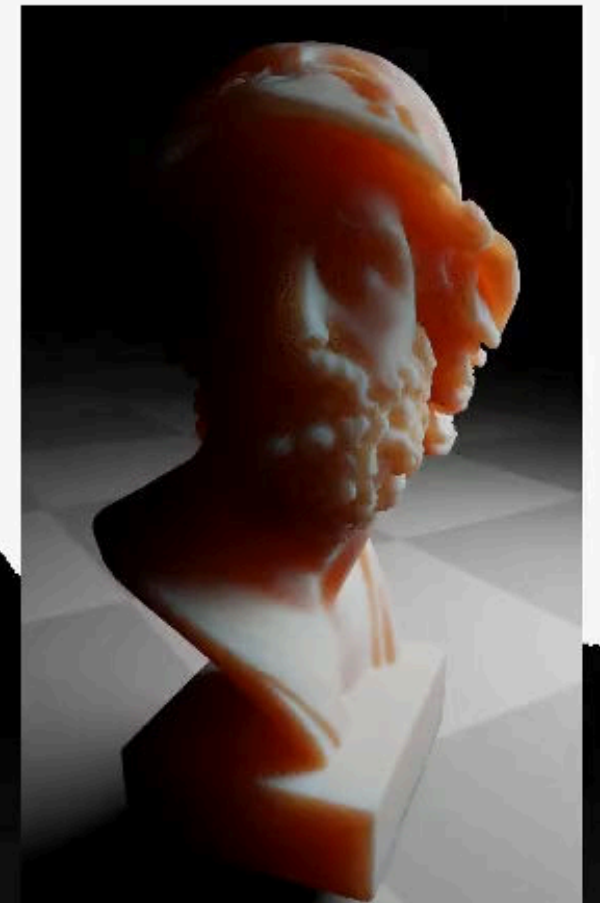
(ReSTIR SSS)





path tracing (1 spp)

ReSTIR SSS (1 spp)



reference



# ReSTIR SSS

- ReSTIR [Bitterli et al. 2020]
  - reuse samples by sharing across pixels and frames

## Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting

BENEDIKT BITTERLI, Dartmouth College  
CHRIS WYMAN, NVIDIA  
MATT PHARR, NVIDIA  
PETER SHIRLEY, NVIDIA  
AARON LEFOHN, NVIDIA  
WOJCIECH JAROSZ, Dartmouth College

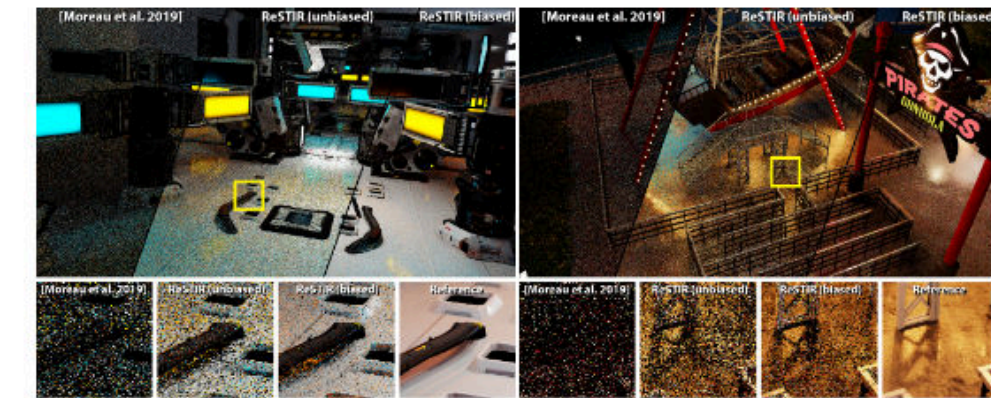


Fig. 1. Two complex scenes ray traced with direct lighting from many dynamic lights. (Left) A still from the Zeno Day video [Winkelmann 2015] with 11,000 dynamic emissive triangles. (Right) A view of one ride in an AMUSEMENT PARK scene containing 3.4 million dynamic emissive triangles. Both images show three methods running in equal time on a modern GPU, from left to right: Moreau et al. [2019]’s efficient light-sampling BVH, our new unbiased estimator, and our new biased estimator. The Zeno Day image is rendered in 15 ms and AMUSEMENT PARK in 50 ms, both at  $1920 \times 1080$  resolution. Zeno Day ©beepie, Pirate Ship ©sema edis

Efficiently rendering direct lighting from millions of dynamic light sources using Monte Carlo integration remains a challenging problem, even for off-line rendering systems. We introduce a new algorithm—ReSTIR—that renders such lighting interactively, at high quality, and without needing to maintain complex data structures. We repeatedly resample a set of candidate

Authors’ addresses: Benedikt Bitterli, Dartmouth College, [benedikt.bitterli@dartmouth.edu](mailto:benedikt.bitterli@dartmouth.edu); Chris Wyman, NVIDIA, [chris.wyman@nvidia.com](mailto:chris.wyman@nvidia.com); Matt Pharr, NVIDIA, [matt.pharr@gmail.com](mailto:matt.pharr@gmail.com); Peter Shirley, NVIDIA, [pshirley@gmail.com](mailto:pshirley@gmail.com); Aaron Lefohn, NVIDIA, 2780 San Tomas Expressway, Santa Clara, CA, 95051, [alefohn@nvidia.com](mailto:alefohn@nvidia.com); Wojciech Jarosz, Dartmouth College, Department of Computer Science, 9 Maynard St. Hanover, NH, 03755, [wojciech.k.jarosz@dartmouth.edu](mailto:wojciech.k.jarosz@dartmouth.edu).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).  
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.  
0730-0301/2020/7-ART148 \$15.00  
<https://doi.org/10.1145/3386569.3392481>

light samples and apply further spatial and temporal resampling to leverage information from relevant nearby samples. We derive an unbiased Monte Carlo estimator for this approach, and show that it achieves equal-error  $6\times-60\times$  faster than state-of-the-art methods. A biased estimator reduces noise further and is  $35\times-65\times$  faster, at the cost of some energy loss. We implemented our approach on the GPU, rendering complex scenes containing up to 3.4 million dynamic, emissive triangles in under 50 ms per frame while tracing at most 8 rays per pixel.

CCS Concepts: • Computing methodologies → Ray tracing.

Additional Key Words and Phrases: Photorealistic rendering, resampled importance sampling, real-time rendering, reservoir sampling

### ACM Reference Format:

Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. *ACM Trans. Graph.* 39, 4, Article 148 (July 2020), 17 pages. <https://doi.org/10.1145/3386569.3392481>

ACM Trans. Graph., Vol. 39, No. 4, Article 148. Publication date: July 2020.



# ReSTIR SSS

- ReSTIR [Bitterli et al. 2020]
  - reuse samples by sharing across pixels and frames
- why ReSTIR SSS? why not use...
  - ReSTIR GI [Ouyang et al. 2021] or
  - ReSTIR PT [Lin et al. 2022]?

High-Performance Graphics 2021  
N. Binder and T. Ritschel  
(Guest Editors)

Volume 40 (2021), Number 8

## ReSTIR GI: Path Resampling for Real-Time Path Tracing

Y. Ouyang<sup>1</sup>, S. Liu<sup>1</sup>, M. Kettunen<sup>1</sup>, M. Pharr<sup>1</sup>, J. Pantaleoni<sup>1</sup>

<sup>1</sup>NVIDIA Corporation, Santa Clara, CA, USA

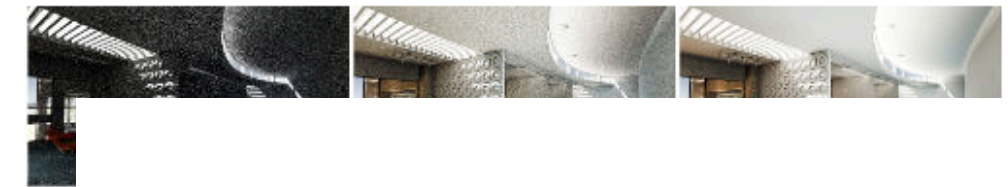


Figure 3090 R  
resamp  
This is  
paths. i

## Generalized Resampled Importance Sampling: Foundations of ReSTIR

DAQI LIN<sup>\*</sup>, University of Utah, USA  
MARKUS KETTUNEN<sup>\*</sup>, NVIDIA, Finland  
BENEDIKT BITTERLI, NVIDIA, USA  
JACOPO PANTALEONI, NVIDIA, Germany  
CEM YUKSEL, University of Utah, USA  
CHRIS WYMAN, NVIDIA, USA

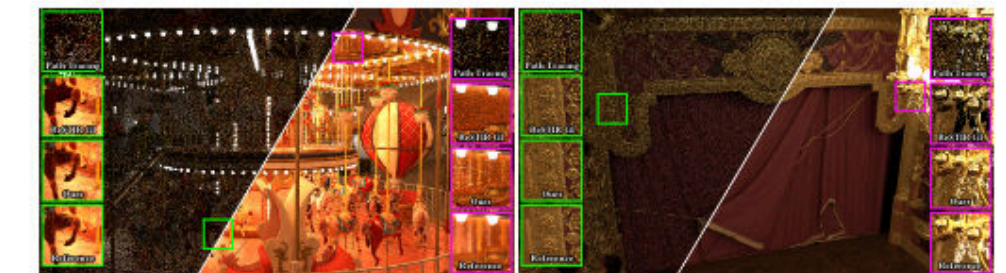


Fig. 1. Our new generalized resampled importance sampling (GRIS) theory extends resampled importance sampling [Talbot 2005] to guarantee convergence even when applied to correlated samples arising from spatiotemporal reuse (i.e., Bitterli et al. [2020]). GRIS allows applying ReSTIR to reuse arbitrary paths, shown with paths of length 10 in the CAROUSEL and PARIS OPERA HOUSE. Main images compare naive path tracing and our new ReSTIR PT in equal time (80 ms at 1920 × 1080). Insets show equal-time path tracing, ReSTIR GI [Ouyang et al. 2021], our ReSTIR PT, plus a converged reference. We significantly improve quality for glossy interreflection, reflections, and other high-frequency lighting. For CAROUSEL, MAPE errors: path tracing (1.63), ReSTIR GI (0.45), and ReSTIR PT (0.39). Corresponding errors in OPERA HOUSE: 1.28, 0.39, and 0.33. (CAROUSEL: © carousel\_world; PARIS OPERA HOUSE: courtesy © GoldSmooth from TurboSquid.)

### 1. Intro

The file  
highly  
single  
scenes  
path tr  
due to  
vent of  
a few t

© 2021 TI  
Computer  
Wiley & S

As scenes become ever more complex and real-time applications embrace ray tracing, path sampling algorithms that maximize quality at low sample counts become vital. Recent resampling algorithms building on Talbot et al.'s [2005] resampled importance sampling (RIS) reuse paths spatiotemporally to render surprisingly complex light transport with a few samples per pixel. These reservoir-based spatiotemporal importance resamplers (ReSTIR) and their underlying RIS theory make various assumptions, including sample independence. But sample reuse introduces correlation, so ReSTIR-style iterative reuse loses most convergence guarantees that RIS theoretically provides.

We introduce generalized resampled importance sampling (GRIS) to extend the theory, allowing RIS on correlated samples, with unknown PDFs.

\*Joint first authors; equal contribution.  
Authors' addresses: Daqi Lin, University of Utah, USA, daqi@cs.utah.edu; Markus Kettunen, NVIDIA, Finland, m.kettunen@nvidia.com; Benedikt Bitterli, NVIDIA, USA, bbitterli@nvidia.com; Jacopo Pantaleoni, NVIDIA, Germany, pantaleoni@nvidia.com; Cem Yuksel, University of Utah, USA, cem@cs.utah.edu; Chris Wyman, NVIDIA, USA, chris.wyman@nvidia.com.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.  
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM. 0730-0301/2022/7-ART75 \$15.00

and taken from varied domains. This solidifies the theoretical foundation, allowing us to derive variance bounds and convergence conditions in ReSTIR-based samplers. It also guides practical algorithm design and enables advanced path reuse between pixels via complex shift mappings.

We show a path-traced resampler (ReSTIR PT) running interactively on complex scenes, capturing many-bounce diffuse and specular lighting while shading just one path per pixel. With our new theoretical foundation, we can also modify the algorithm to guarantee convergence for offline renderers.

CCS Concepts: • Computing methodologies → Rendering.

### ACM Reference Format:

Daqi Lin, Markus Kettunen, Benedikt Bitterli, Jacopo Pantaleoni, Cem Yuksel, and Chris Wyman. 2022. Generalized Resampled Importance Sampling: Foundations of ReSTIR. *ACM Trans. Graph.* 41, 4, Article 75 (July 2022), 23 pages. <https://doi.org/10.1145/3528223.3530158>

### 1 INTRODUCTION

Monte Carlo algorithms form the core of modern rendering. While originally only feasible in offline renderers, ray-tracing hardware [Kilgariff et al. 2018] has made such algorithms practical in real-time systems as well. However, strict real-time constraints in games limit feasible per-frame ray counts [Halen et al. 2021], giving many modern real-time path tracers budgets of at most one path per pixel. Importance sampling reduces variance at low sample counts by



# ReSTIR SSS

- ReSTIR [Bitterli et al. 2020]
  - reuse samples by sharing across pixels and frames
- why ReSTIR SSS? why not use...
  - ReSTIR GI [Ouyang et al. 2021] or
  - ReSTIR PT [Lin et al. 2022]?
- sampling of surface vs. subsurface light transport paths
- specialized techniques for reusing paths

## ReSTIR Subsurface Scattering for Real-Time Path Tracing

MIRCO WERNER, Karlsruhe Institute of Technology, Germany  
VINCENT SCHÜSSLER, Karlsruhe Institute of Technology, Germany  
CARSTEN DACHSBACHER, Karlsruhe Institute of Technology, Germany



Fig. 1. A dragon with a material that exhibits noticeable subsurface scattering. We apply ReSTIR for subsurface scattering using our hybrid and sequential shift in real-time path tracing to significantly reduce noise and denoising artifacts in regions with visible scattered light.

Subsurface scattering is an important visual cue and in real-time rendering it is often approximated using screen-space algorithms. Path tracing with the diffusion approximation can easily overcome the limitations of these algorithms, but increases image noise. We improve its efficiency by applying reservoir-based spatiotemporal importance resampling (ReSTIR) to subsurface light transport paths. For this, we adopt BSSRDF importance sampling for generating candidates. Further, spatiotemporal reuse requires shifting paths between domains. We observe that different image regions benefit most from either reconnecting through the translucent object (*reconnection shift*), or one vertex later (*delayed reconnection shift*). We first introduce a local subsurface scattering specific criterion for a *hybrid* shift that deterministically selects one of the two shifts for a path. Due to the locality, it cannot always choose the most efficient shift, e.g. near shadow boundaries. Therefore, we additionally propose a novel *sequential* shift to combine multiple shift mappings: We execute subsequent resampling passes, each one using a different shift, which does not require to deterministically choose a shift for a path. Instead, resampling can pick the most successful shift implicitly. Our method achieves real-time performance and significantly reduces noise and denoising artifacts in regions with visible subsurface scattering compared to standard path tracing with equal render time.

CCS Concepts: • Computing methodologies → Ray tracing.

Authors' addresses: Mirco Werner, mirco.werner@student.kit.edu, Karlsruhe Institute of Technology, Karlsruhe, Germany; Vincent Schüssler, vincent.schuessler@kit.edu, Karlsruhe Institute of Technology, Karlsruhe, Germany; Carsten Dachsbacher, dachsbacher@kit.edu, Karlsruhe Institute of Technology, Karlsruhe, Germany.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

© 2024 Copyright held by the owner/author(s).

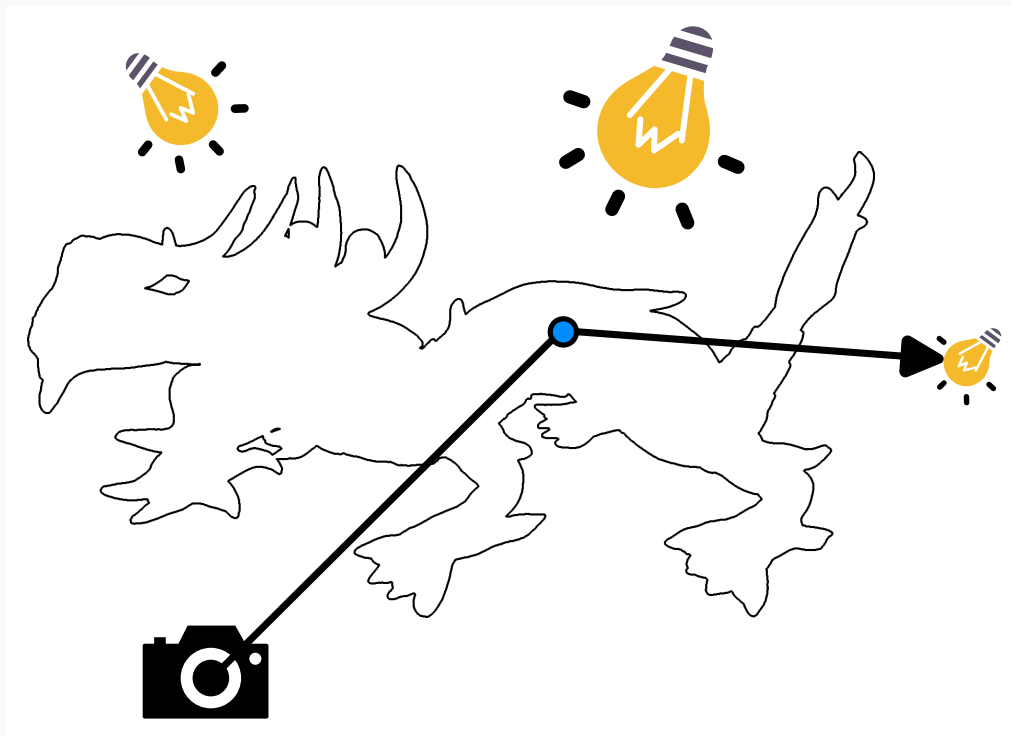
ACM 2577-6193/2024/7-ART1

<https://doi.org/10.1145/3675372>

Proc. ACM Comput. Graph. Interact. Tech., Vol. 7, No. 3, Article 1. Publication date: July 2024.

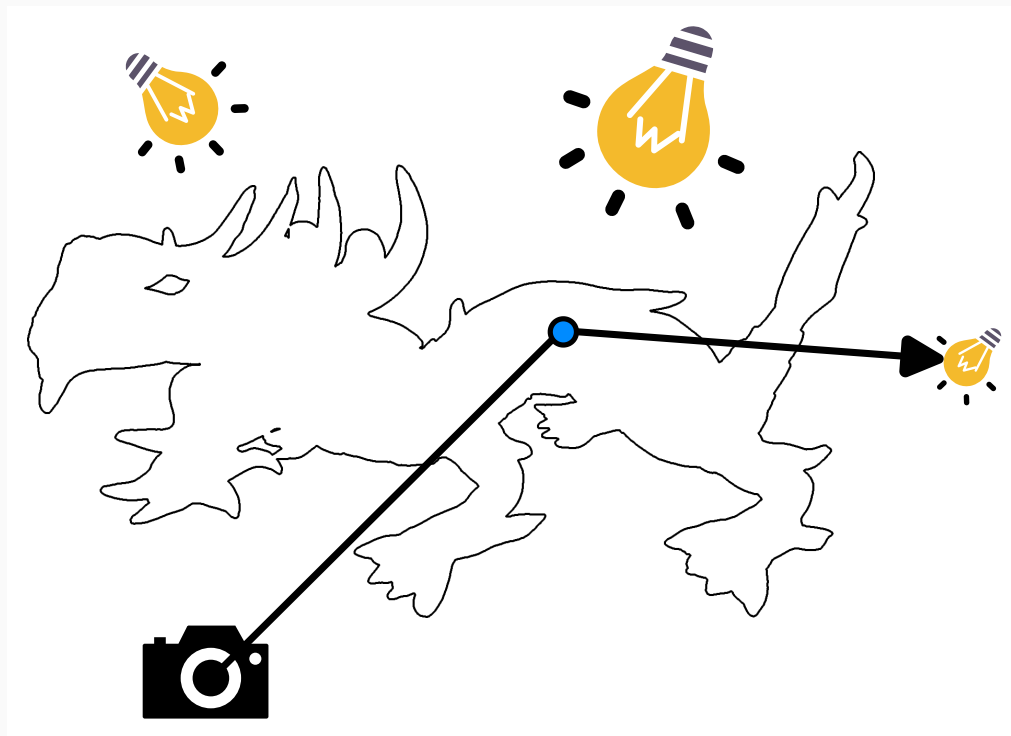


# Recap: ReSTIR



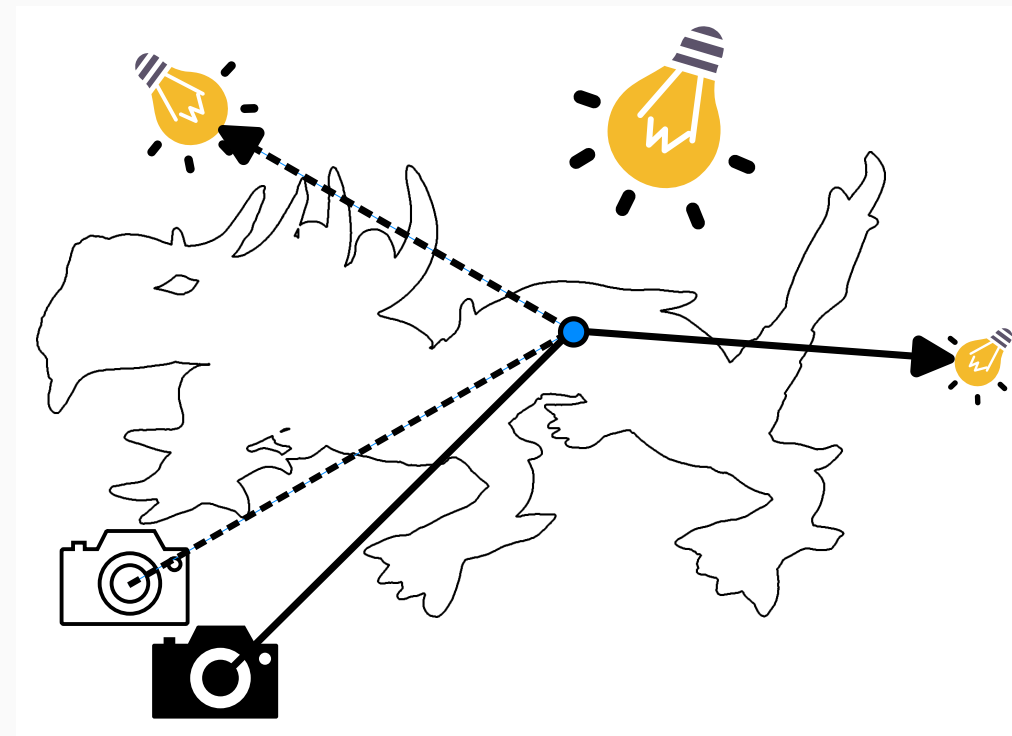
candidate generation

# Recap: ReSTIR



candidate generation

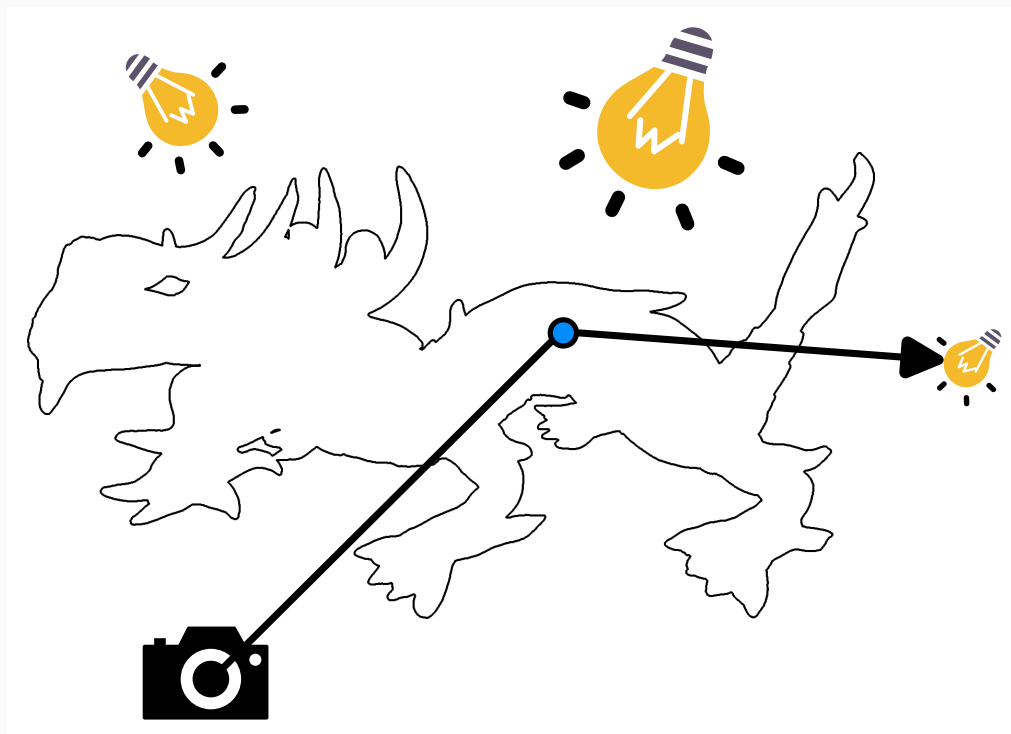
↑  
last frame  
↑



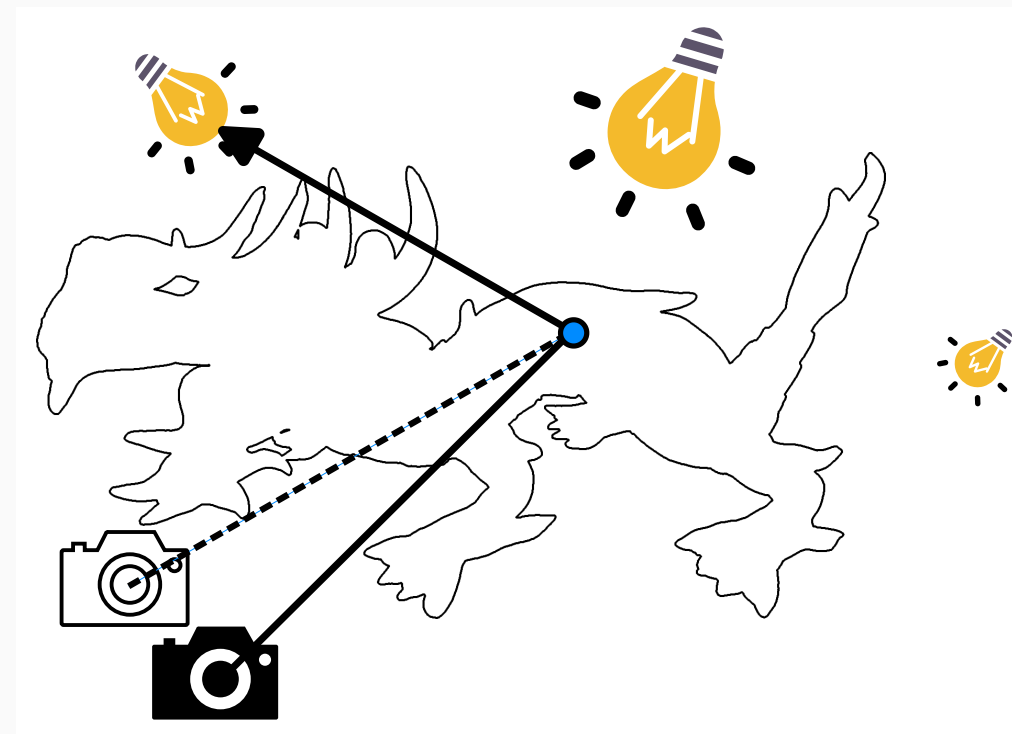
temporal reuse



# Recap: ReSTIR



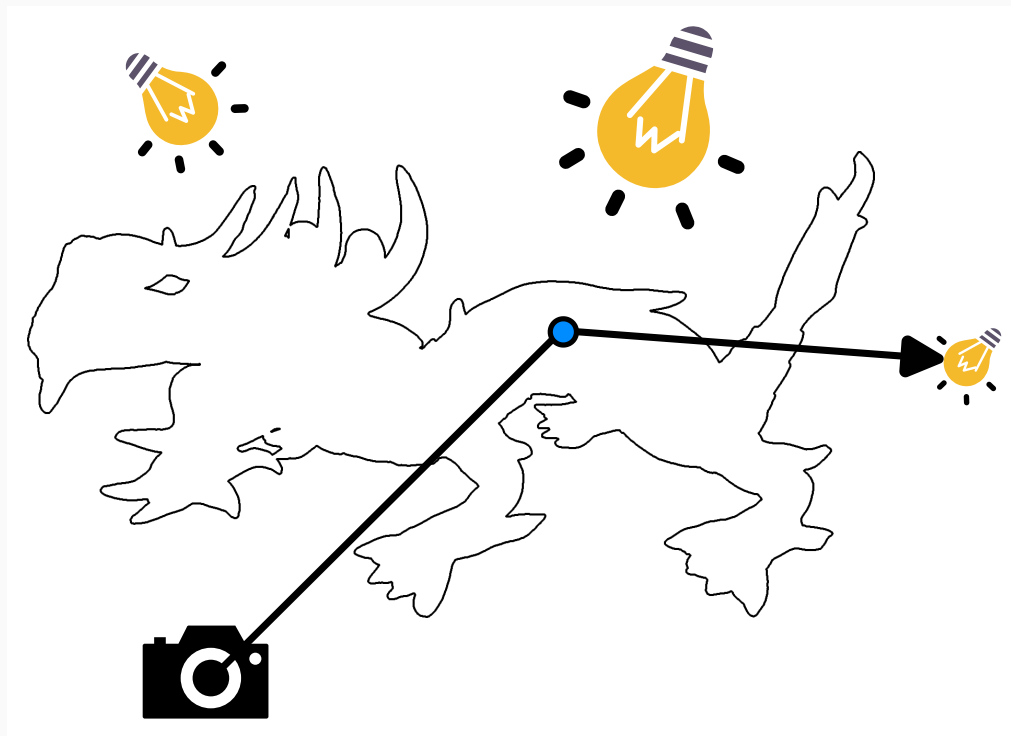
candidate generation



temporal reuse

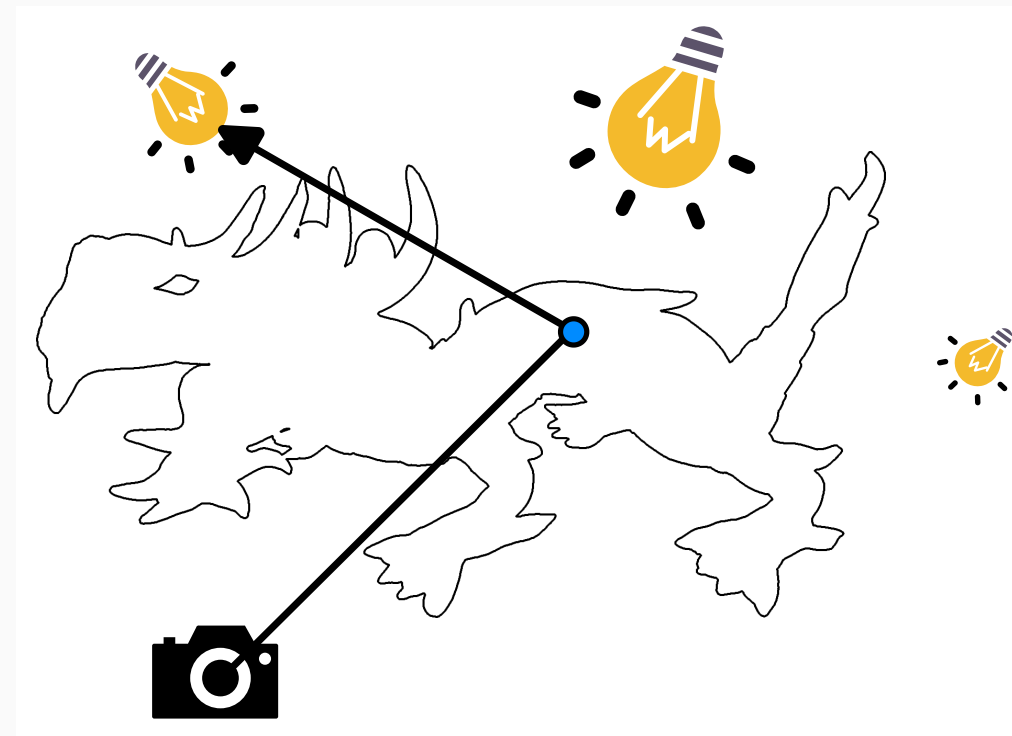


# Recap: ReSTIR



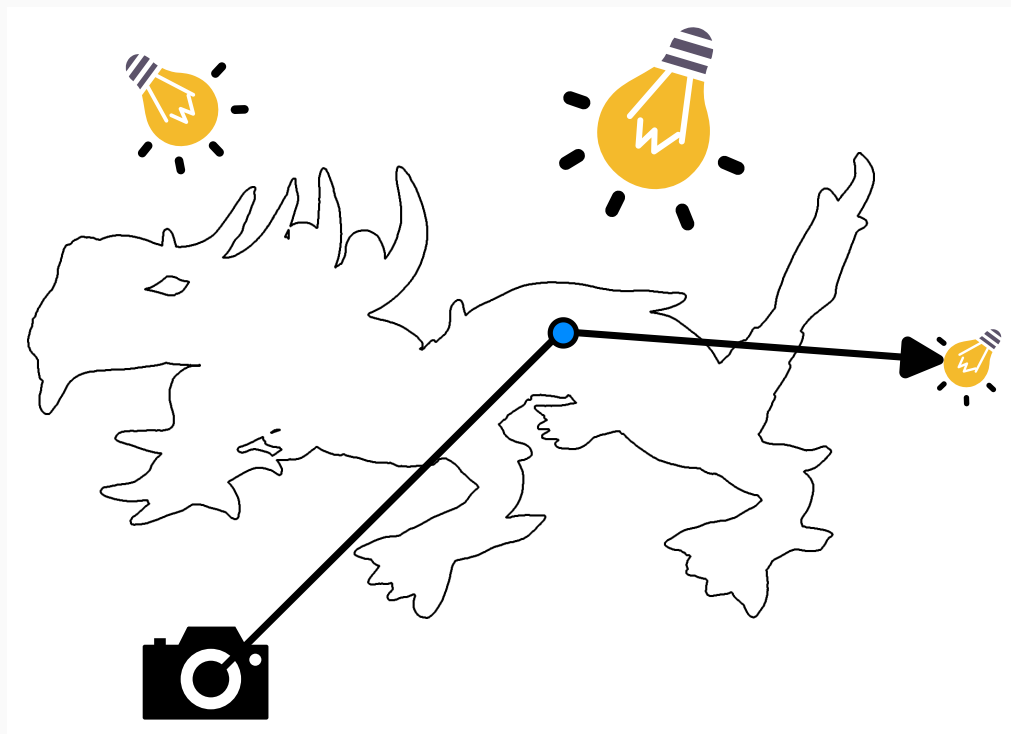
candidate generation

↑  
last frame

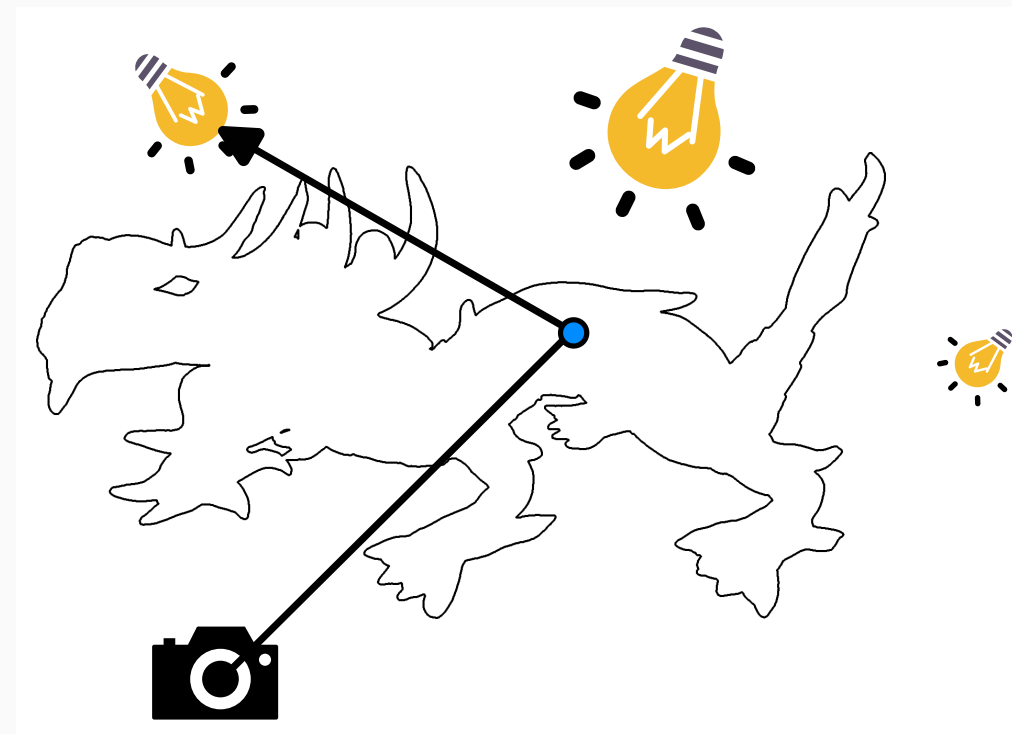
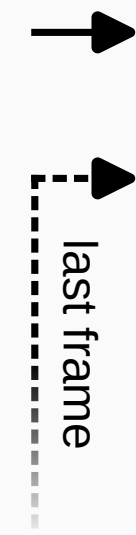


temporal reuse

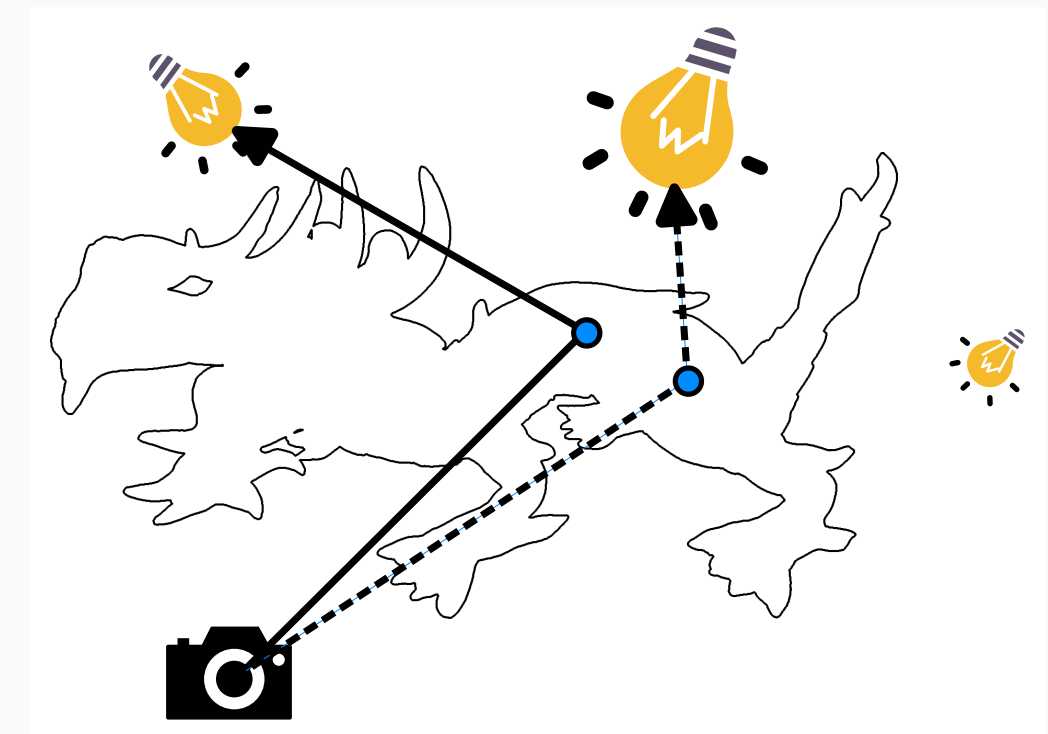
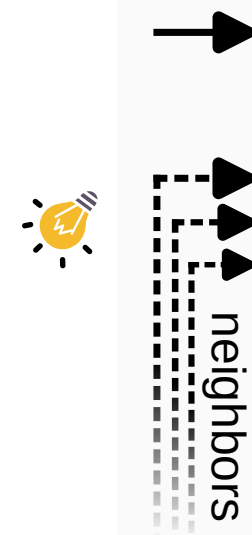
# Recap: ReSTIR



candidate generation



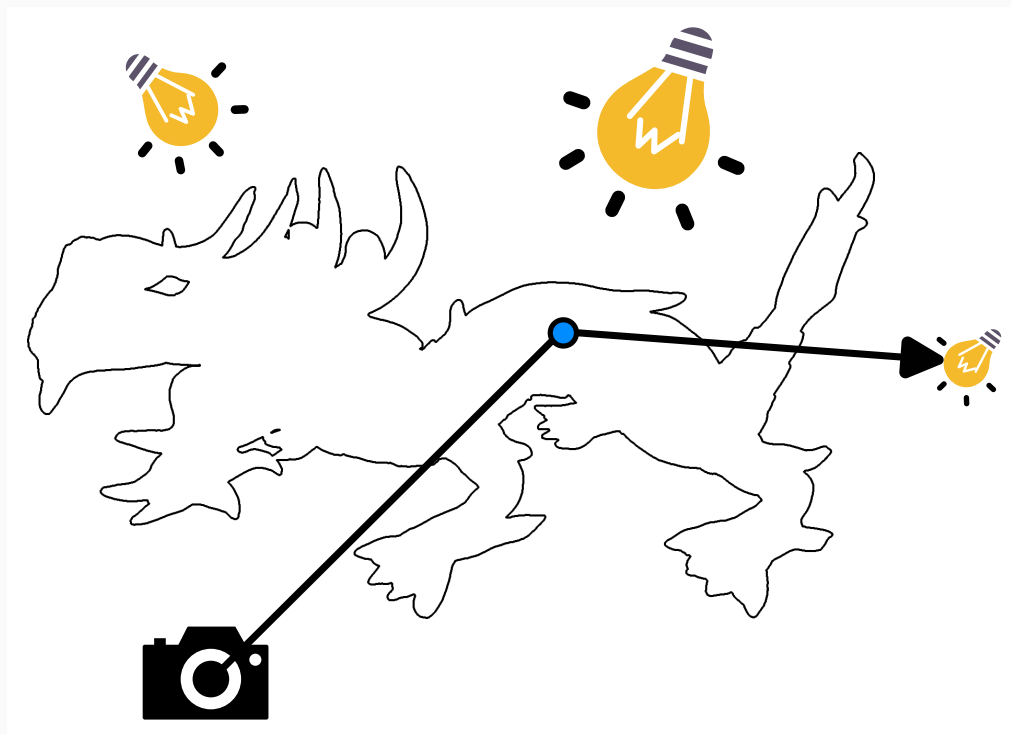
temporal reuse



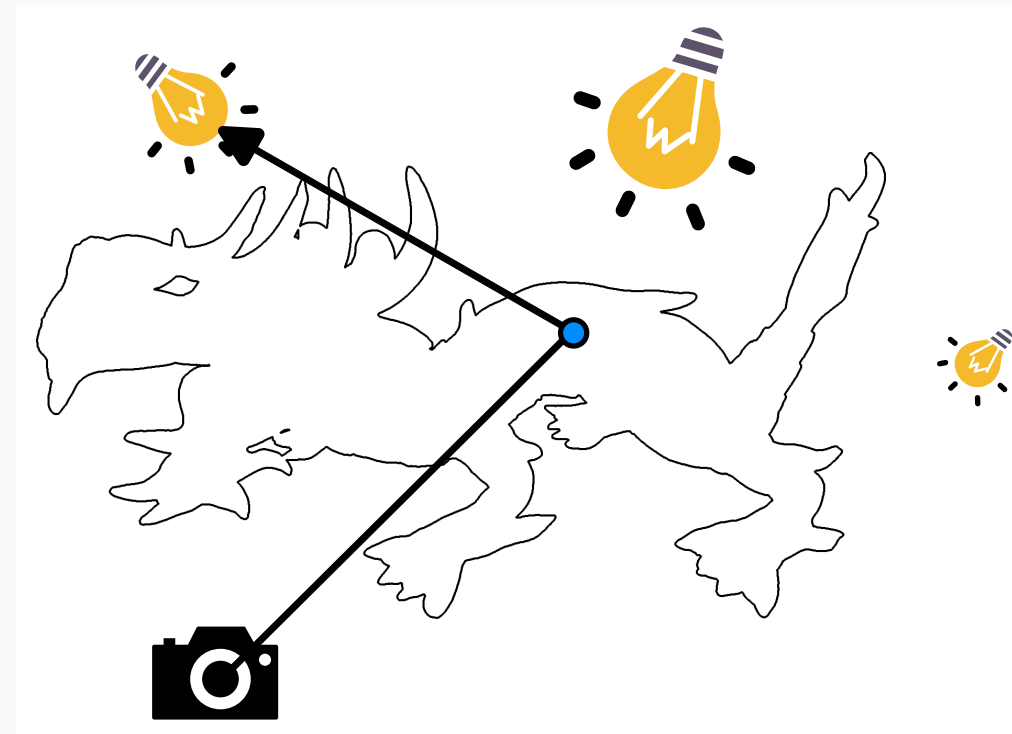
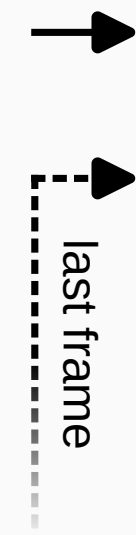
spatial reuse



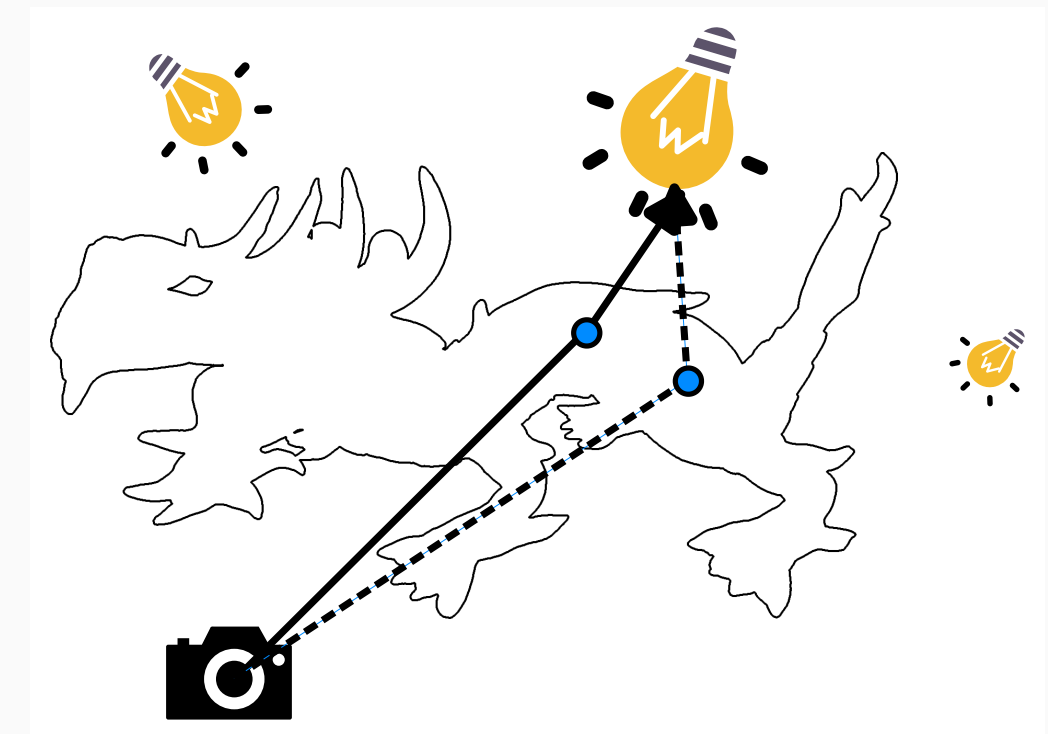
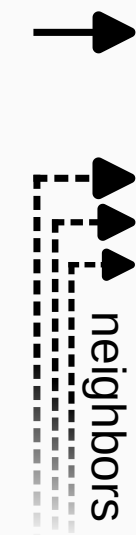
# Recap: ReSTIR



candidate generation

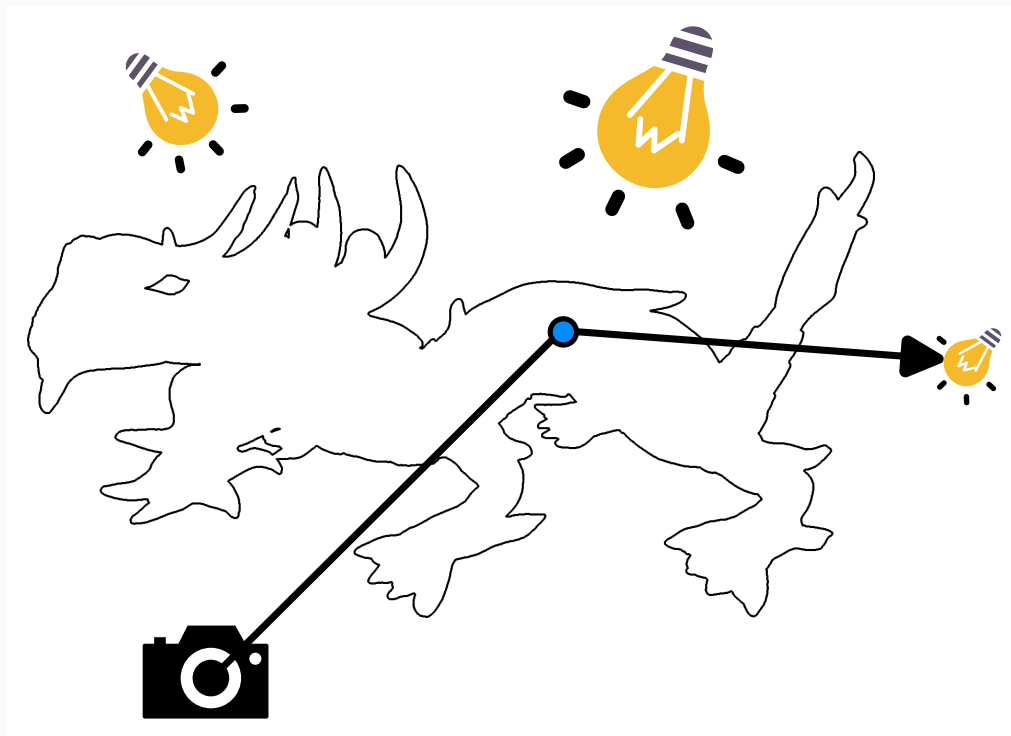


temporal reuse

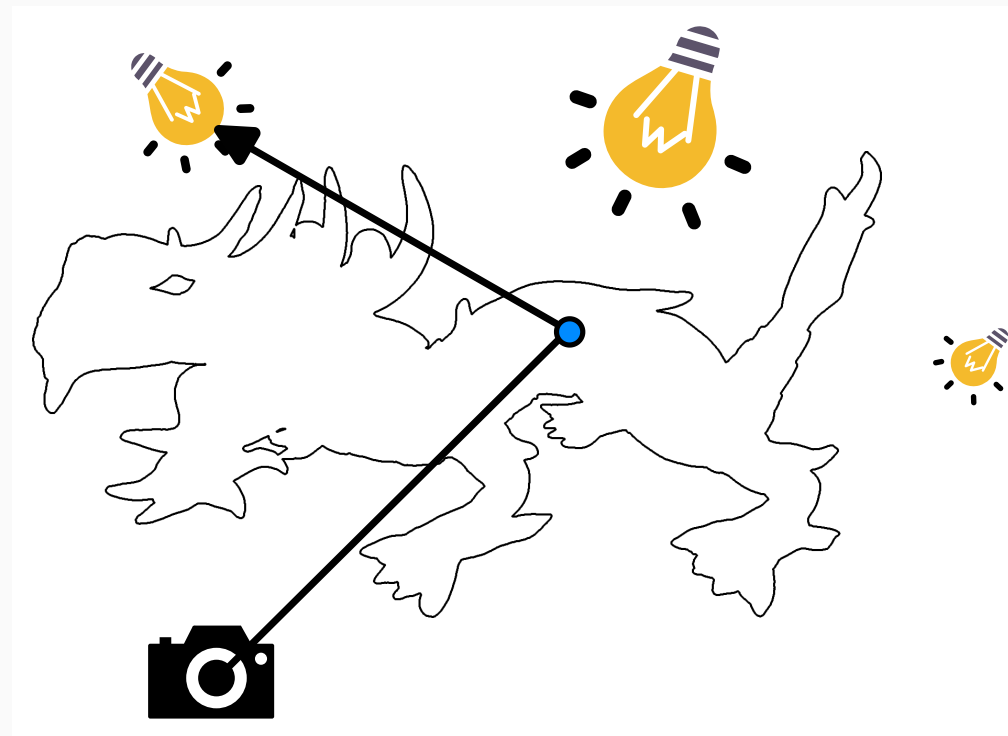
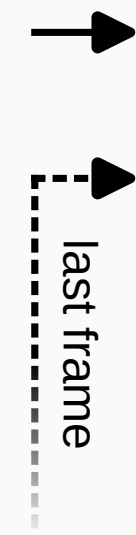


spatial reuse

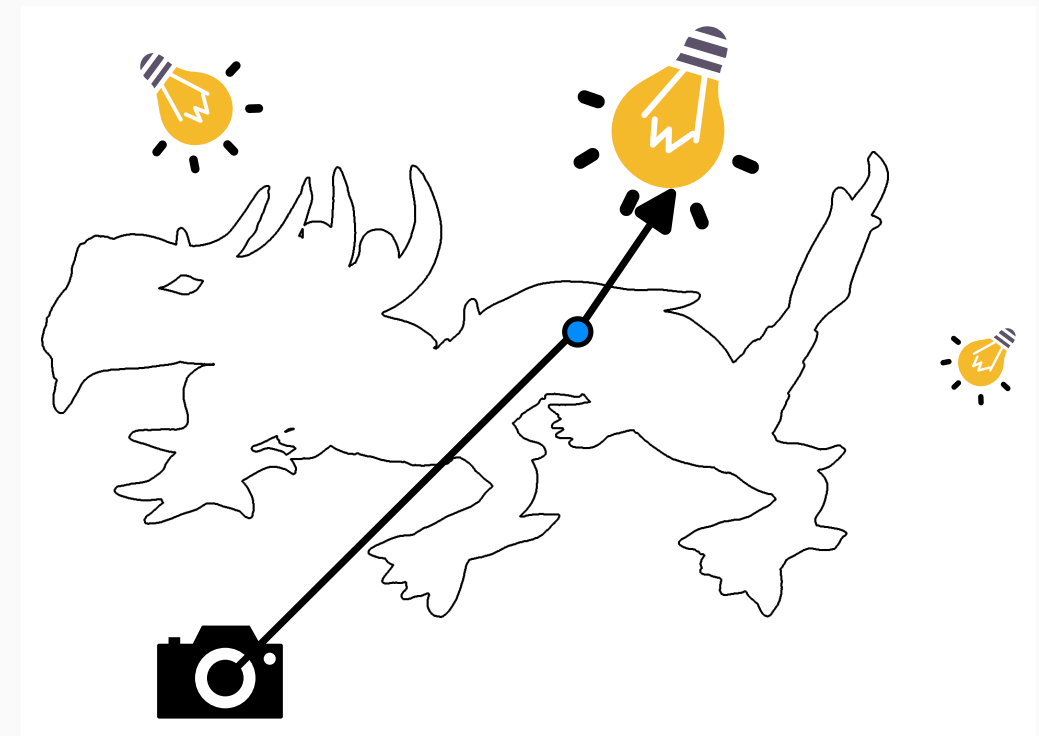
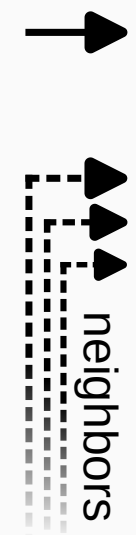
# Recap: ReSTIR



candidate generation



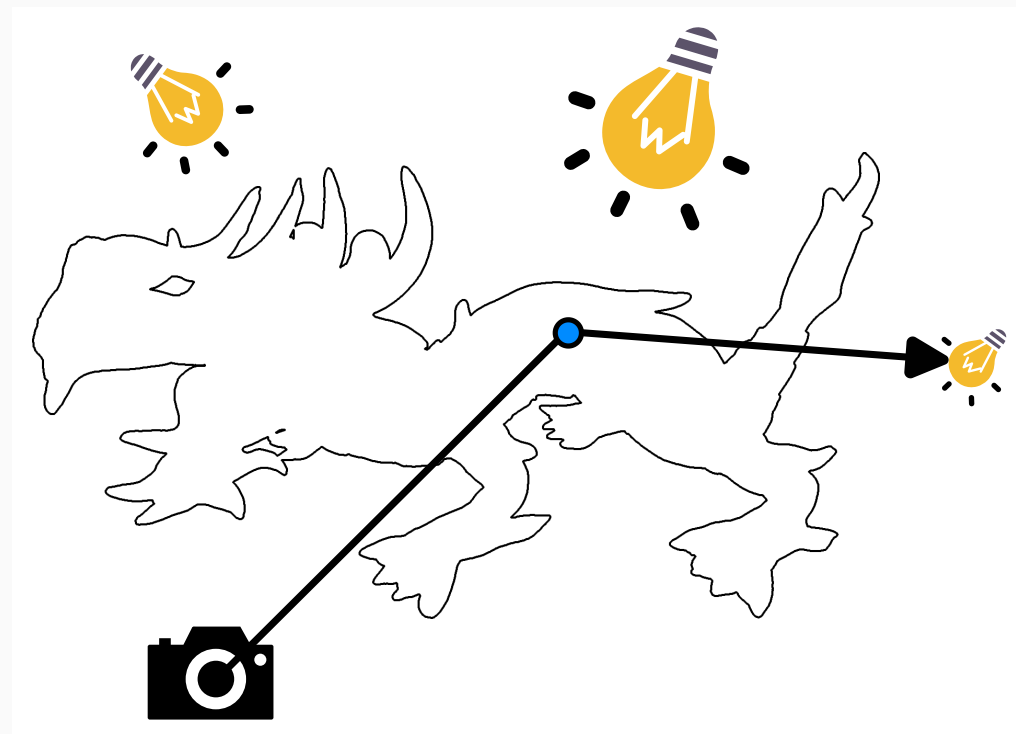
temporal reuse



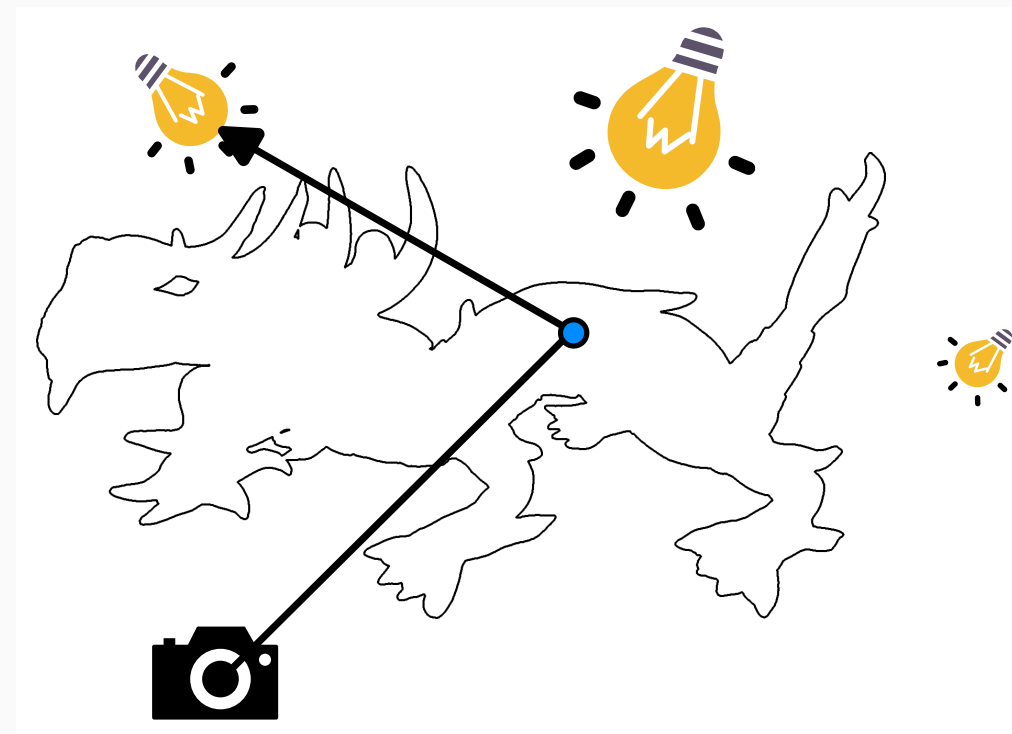
spatial reuse



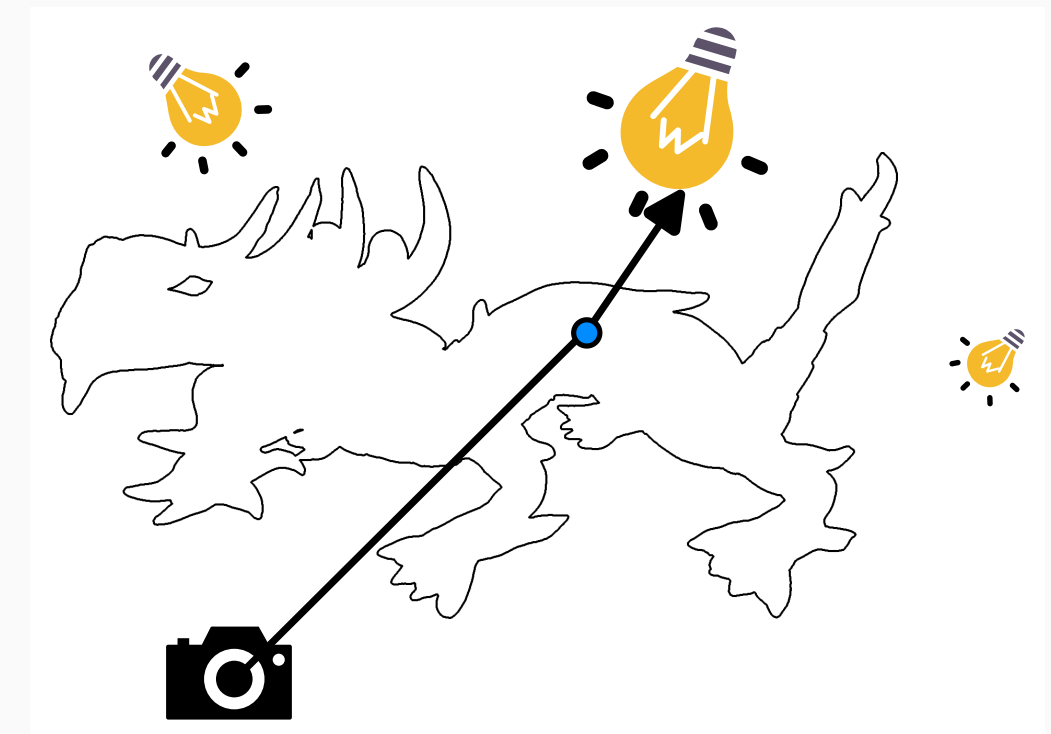
# Recap: ReSTIR



candidate generation

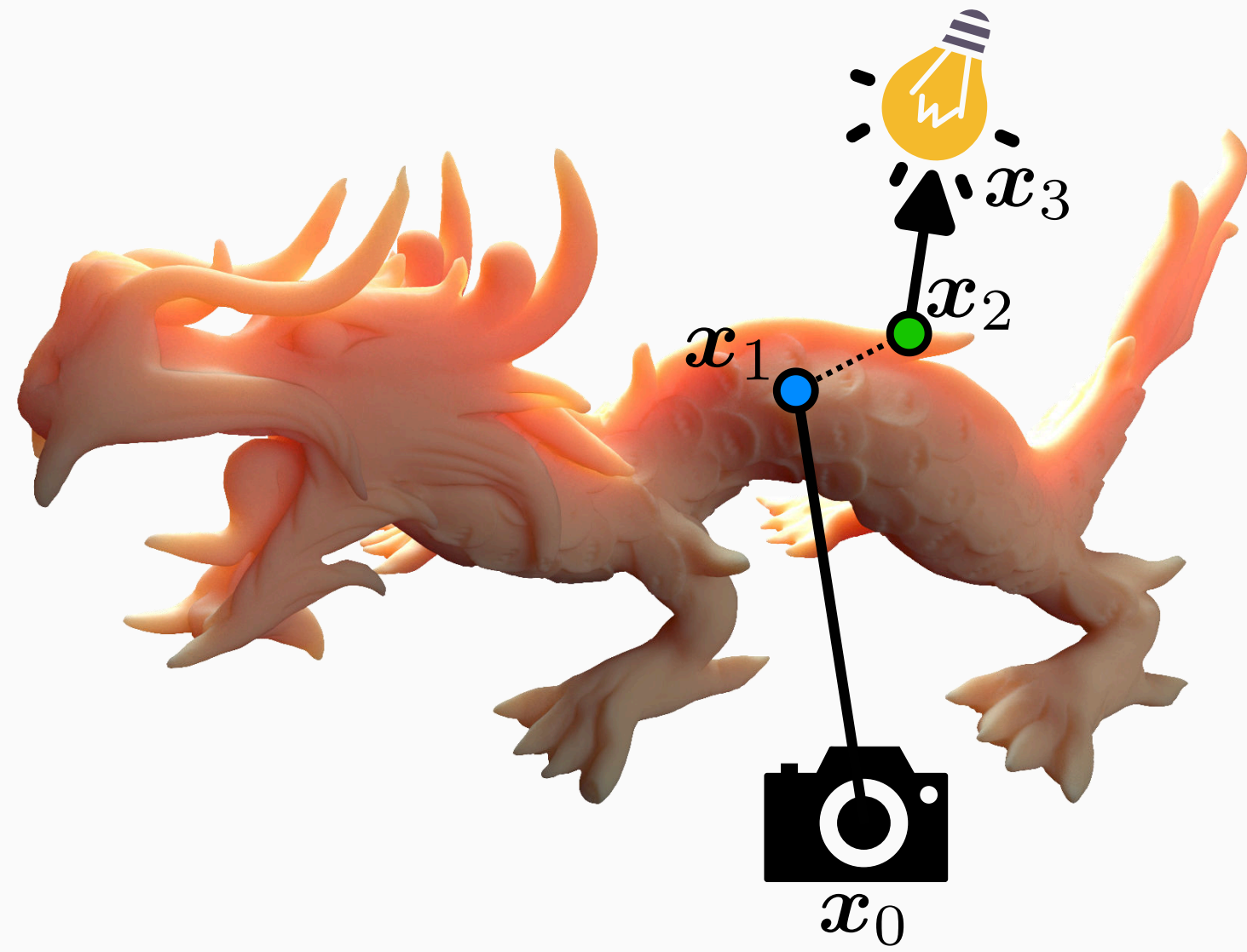


temporal reuse



spatial reuse

# ReSTIR SSS

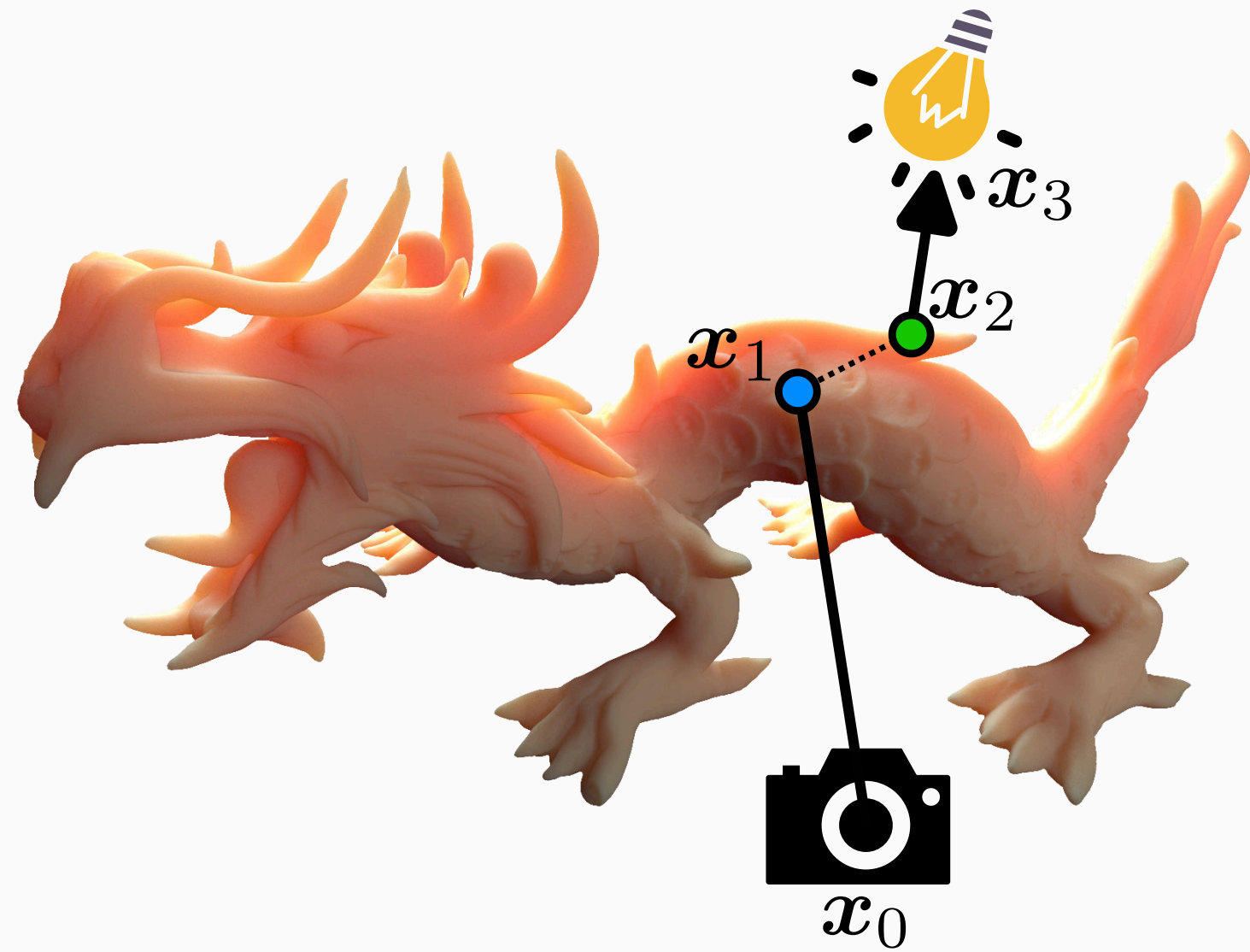


candidate generation\*

\*more details in the paper

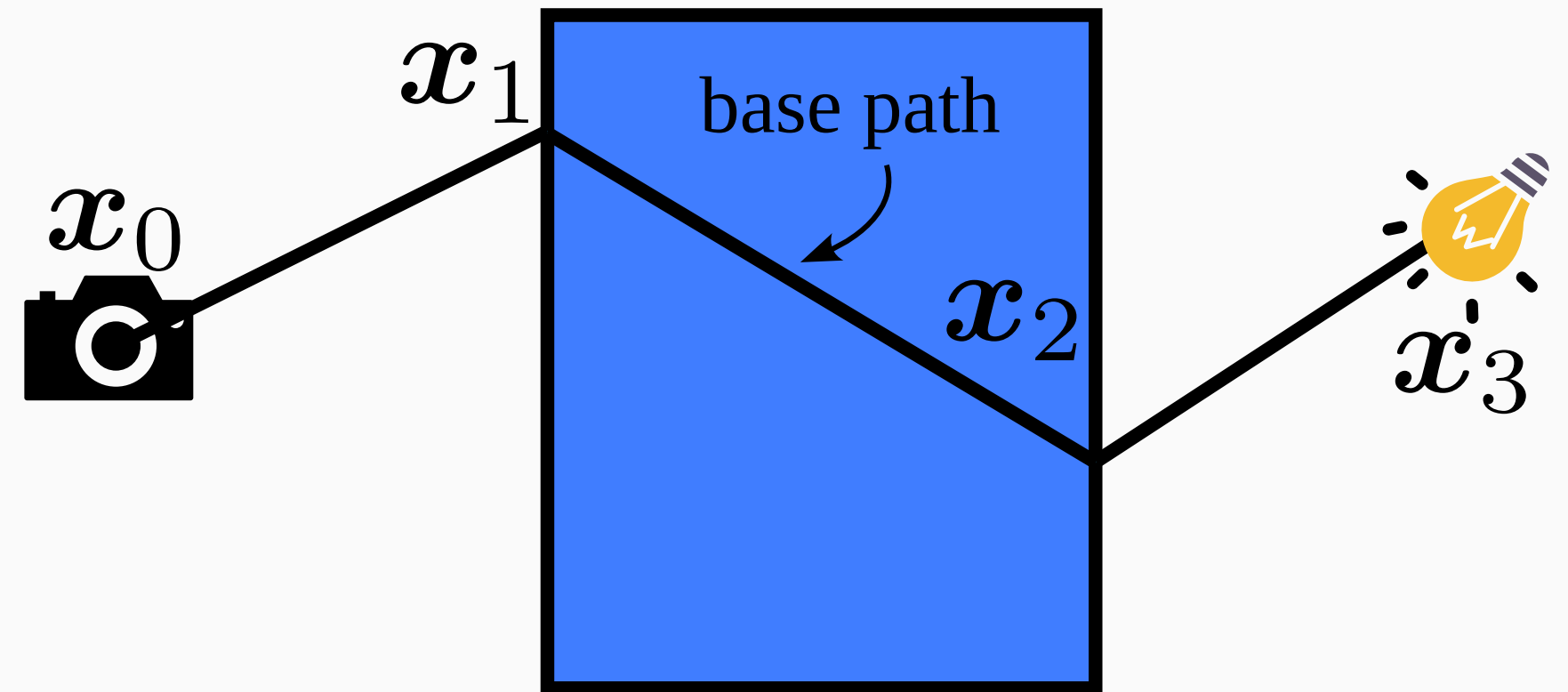


# ReSTIR SSS



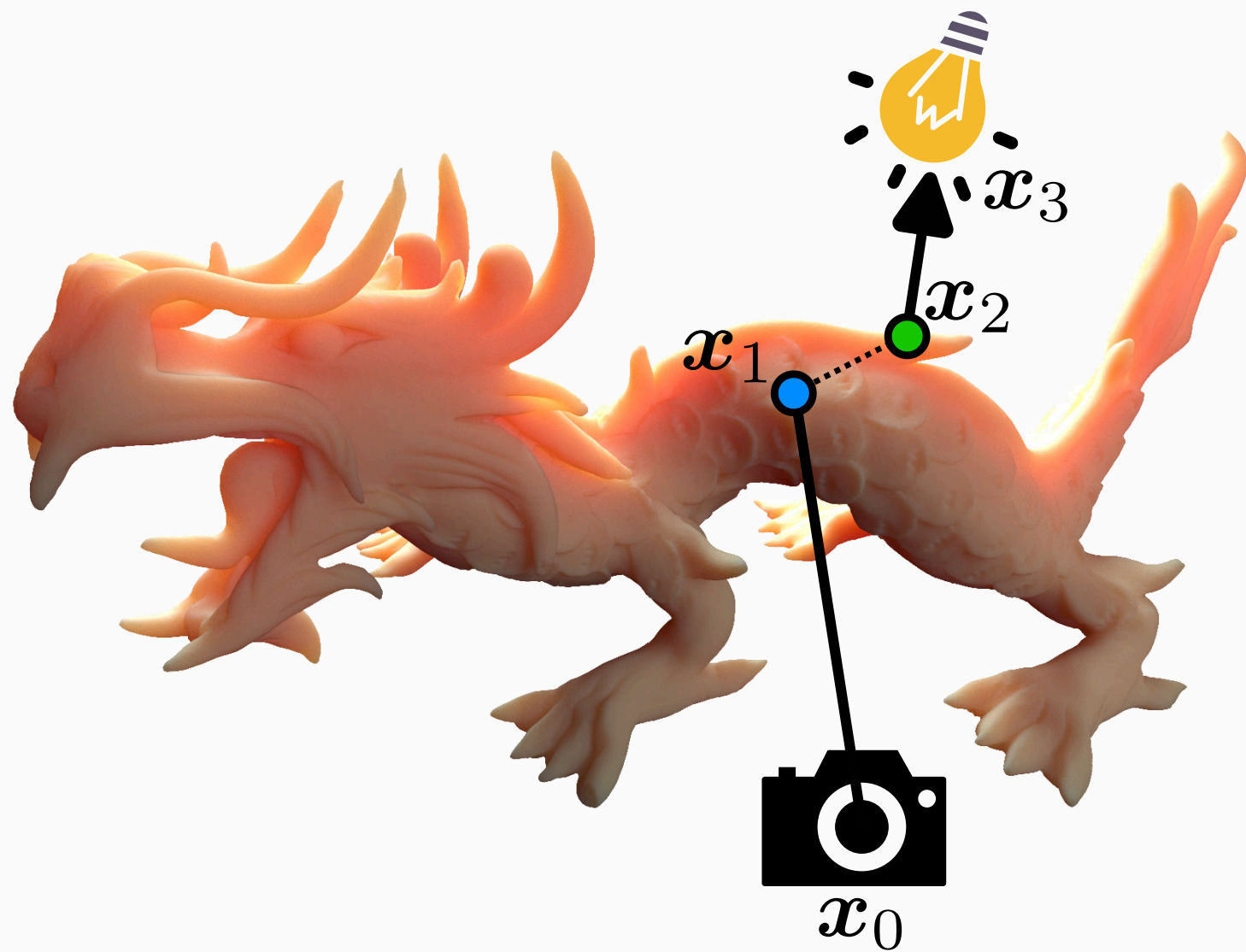
candidate generation\*

\*more details in the paper



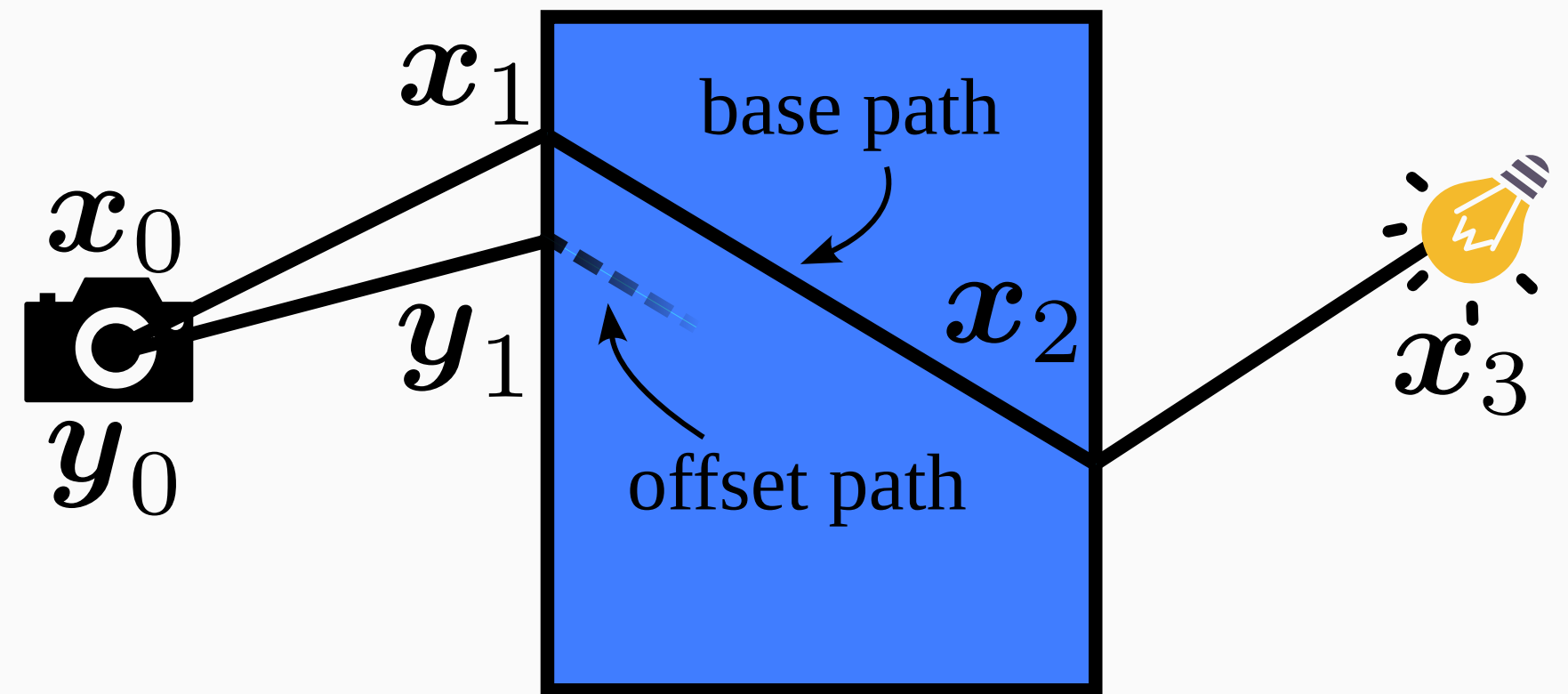
ability to shift paths

# ReSTIR SSS



candidate generation\*

\*more details in the paper

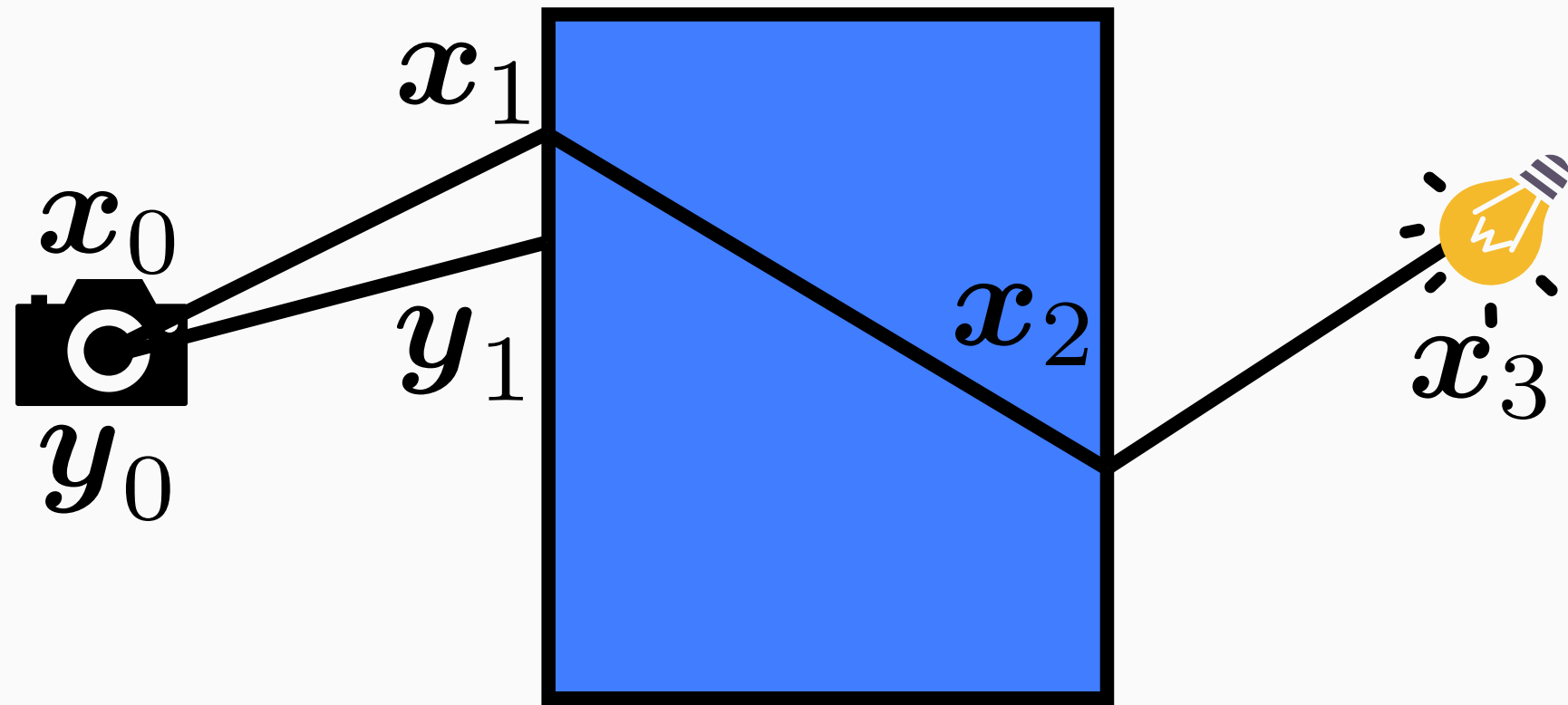


$$T([x_0, x_1, x_2, x_3]) = [y_0, y_1, y_2, y_3]$$

ability to shift paths



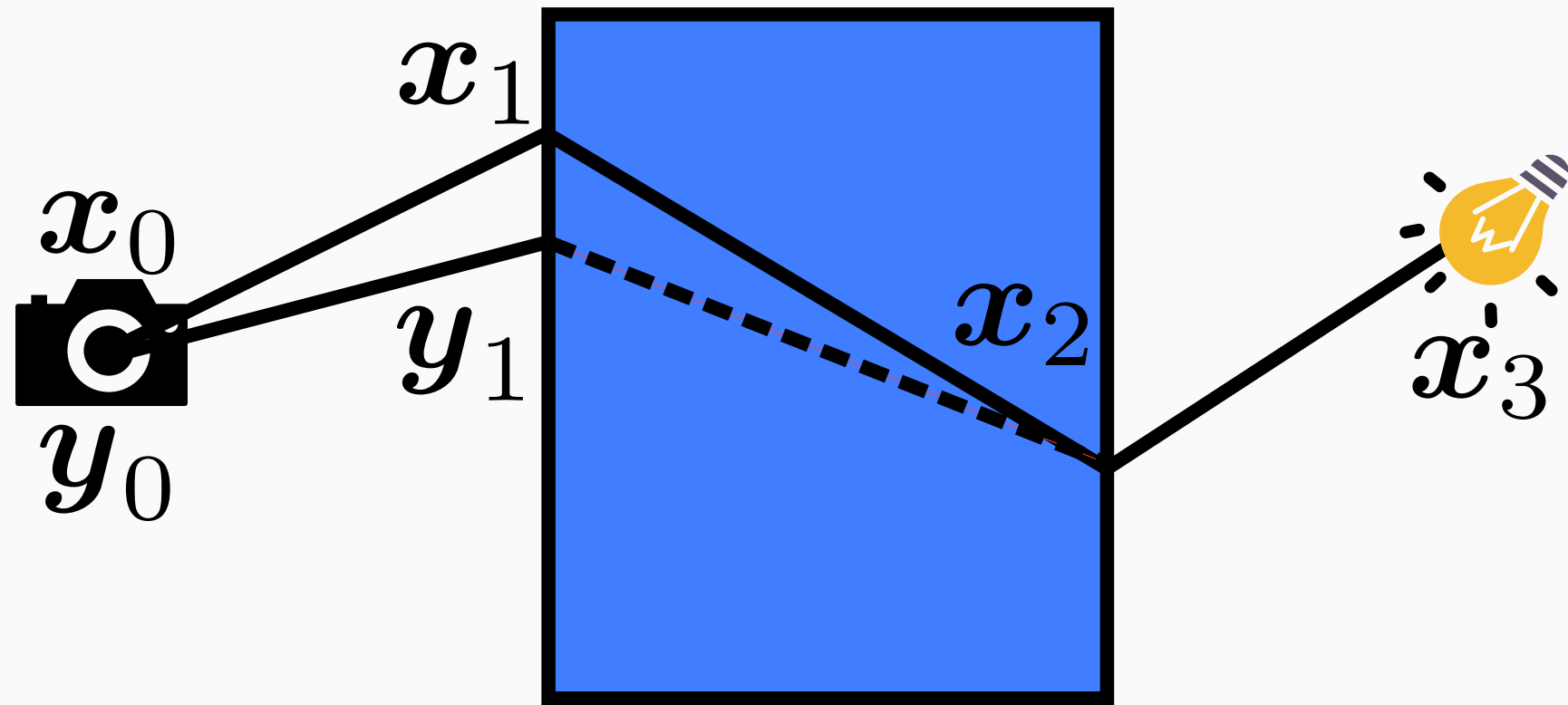
# ReSTIR SSS: Shifting Paths (naive)



$$T([x_0, x_1, x_2, x_3]) = [y_0, y_1, \quad , \quad ]$$

reconnection

# ReSTIR SSS: Shifting Paths (naive)

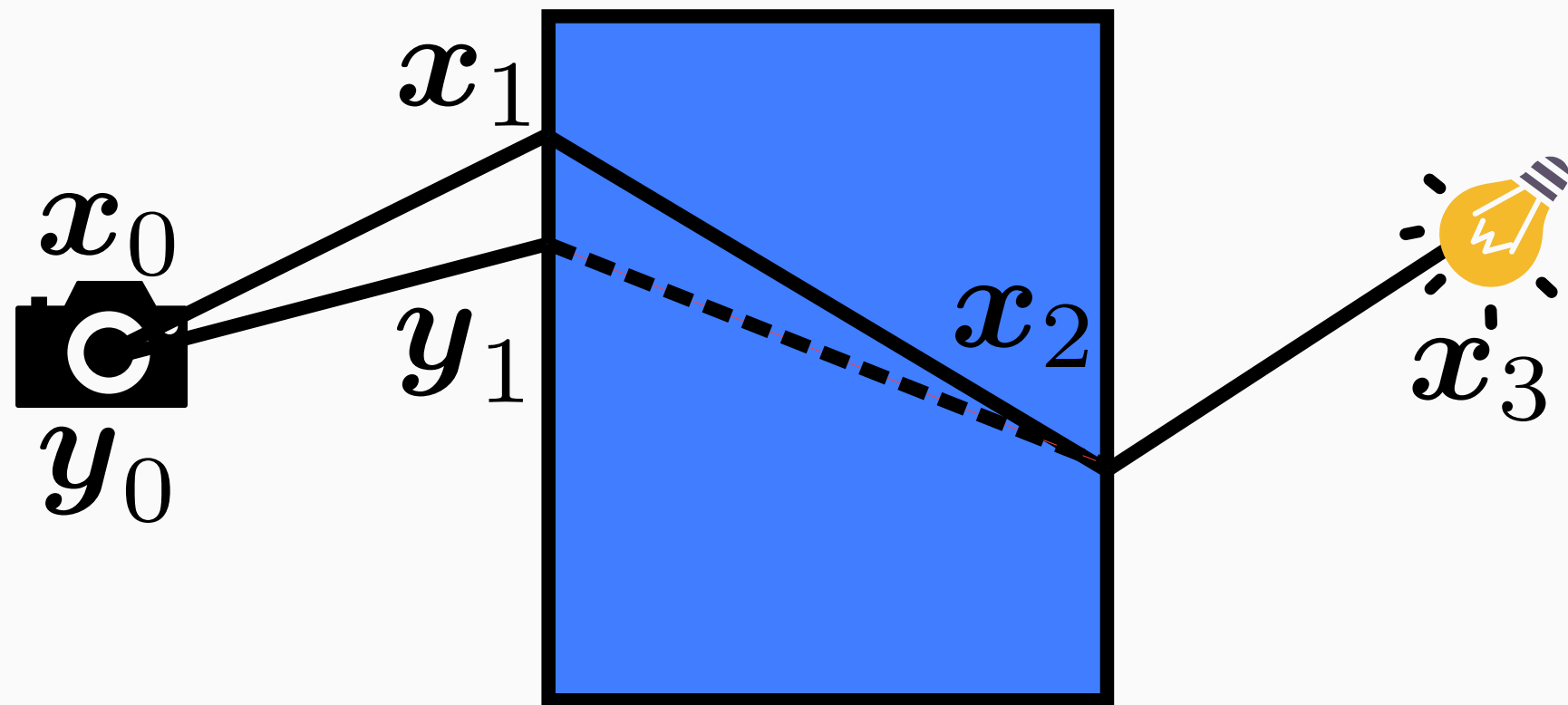


$$T([x_0, x_1, x_2, x_3]) = [y_0, y_1, x_2, x_3]$$

reconnection

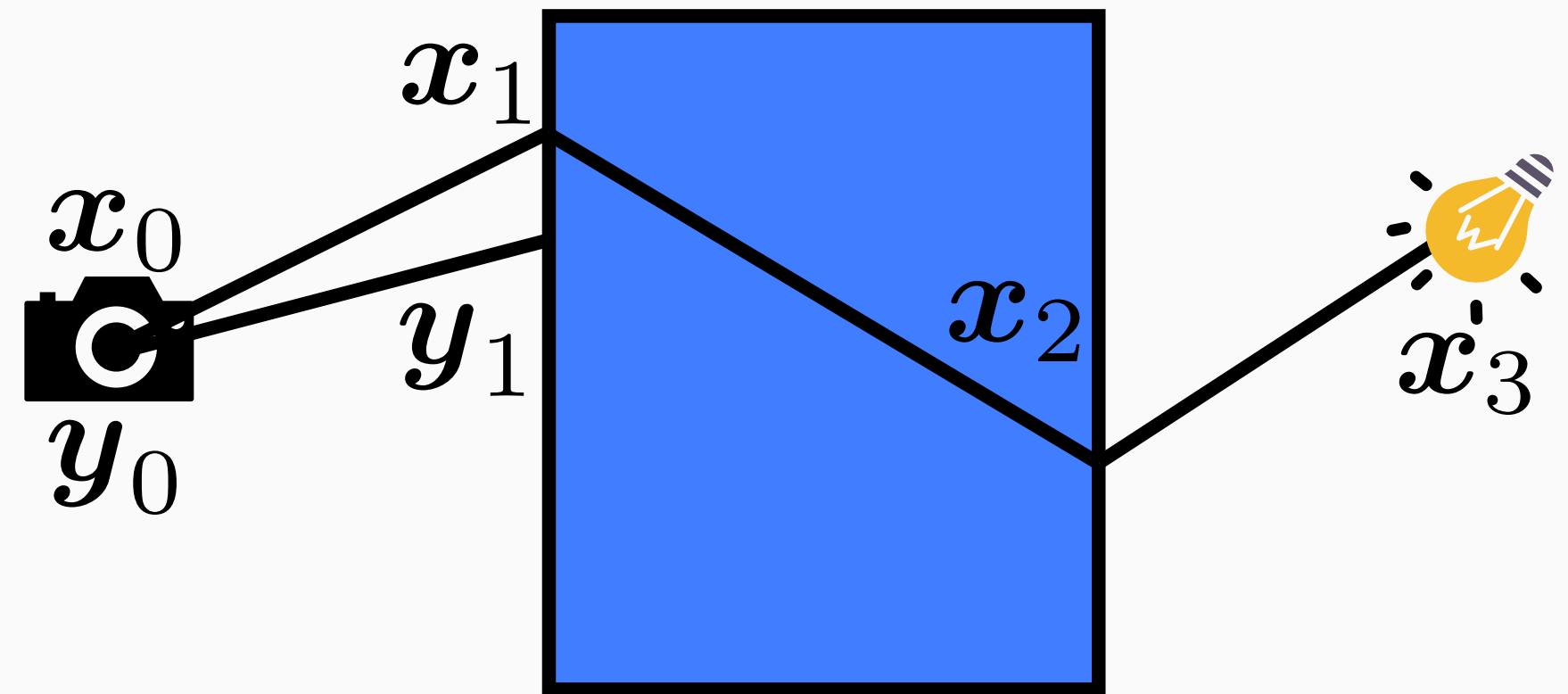


# ReSTIR SSS: Shifting Paths (naive)



$$T([x_0, x_1, x_2, x_3]) = [y_0, y_1, x_2, x_3]$$

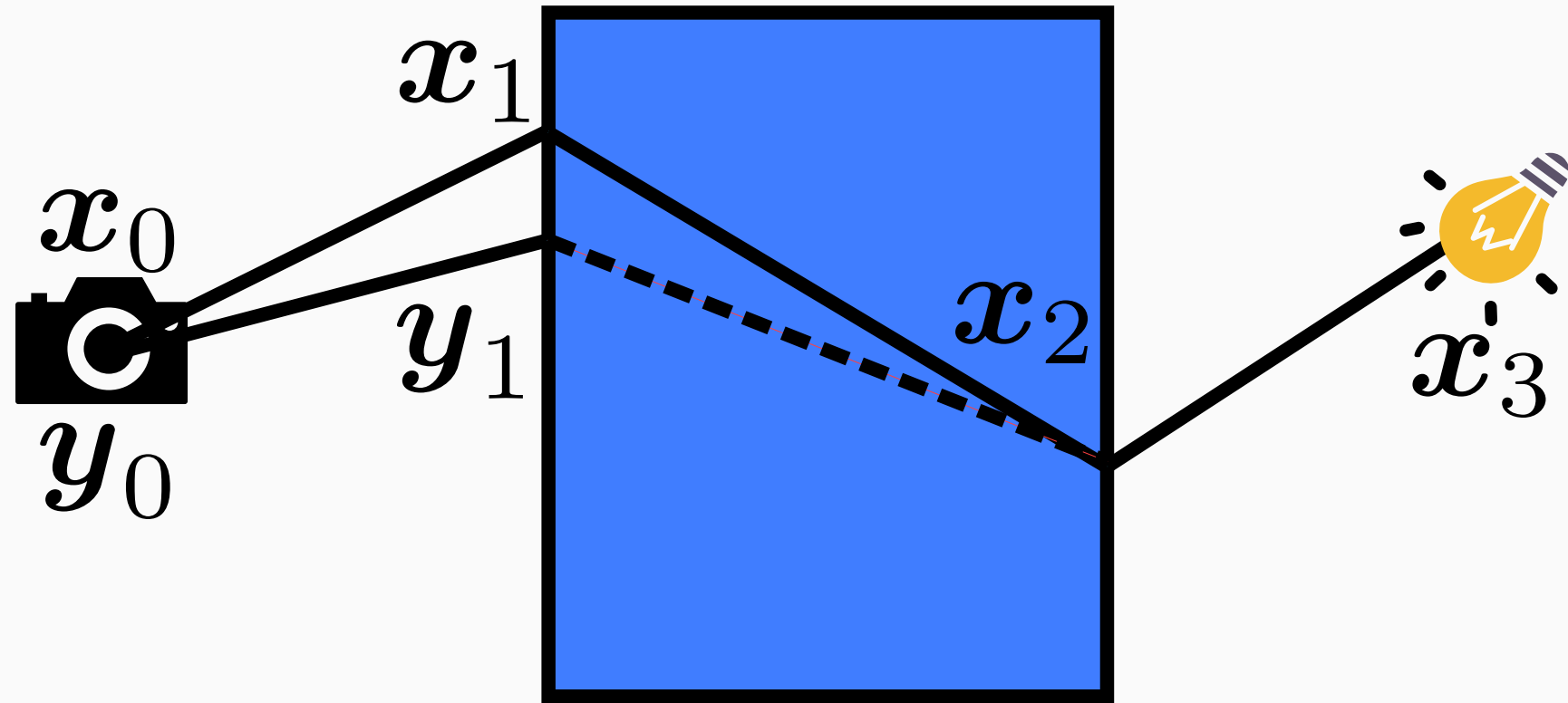
reconnection



$$T([x_0, x_1, x_2, x_3]) = [y_0, y_1, \quad, \quad]$$

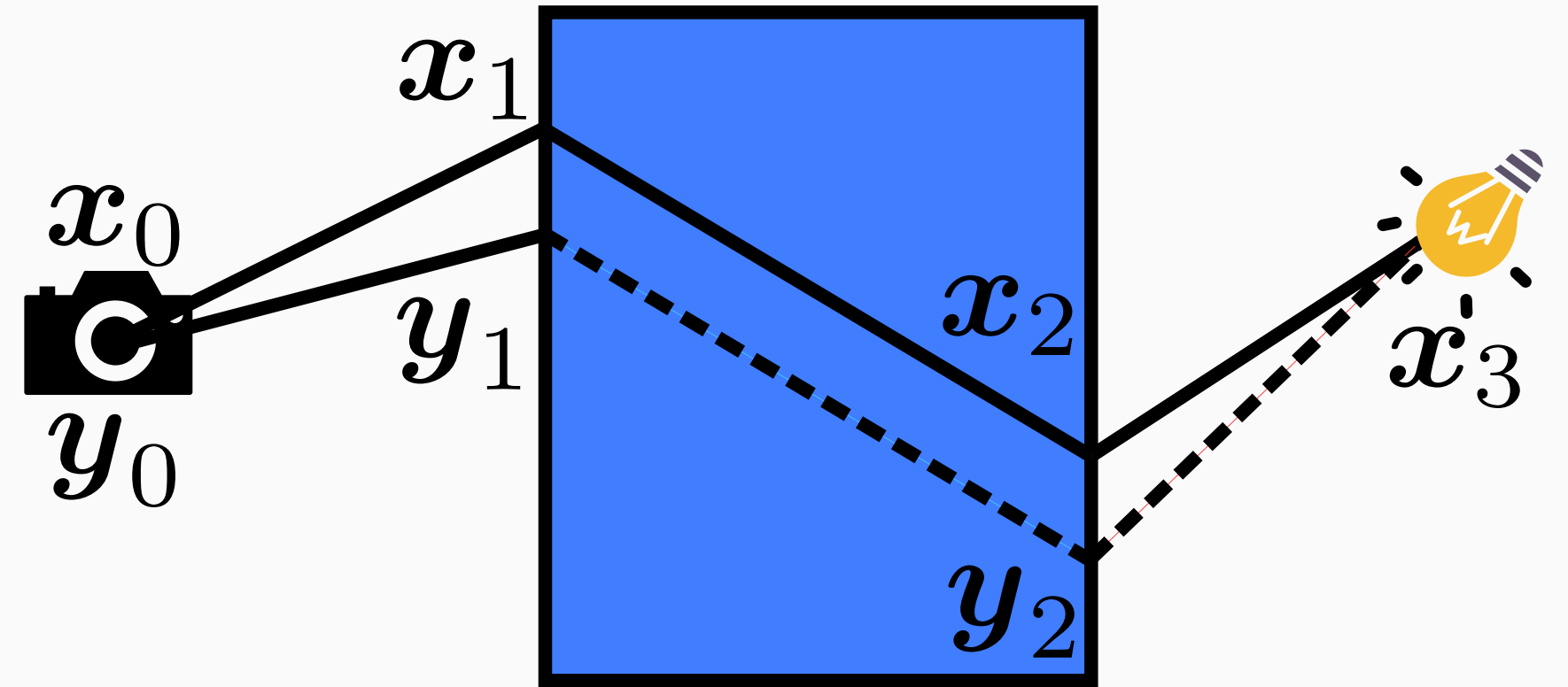
delayed reconnection

# ReSTIR SSS: Shifting Paths (naive)



$$T([x_0, x_1, x_2, x_3]) = [y_0, y_1, x_2, x_3]$$

reconnection

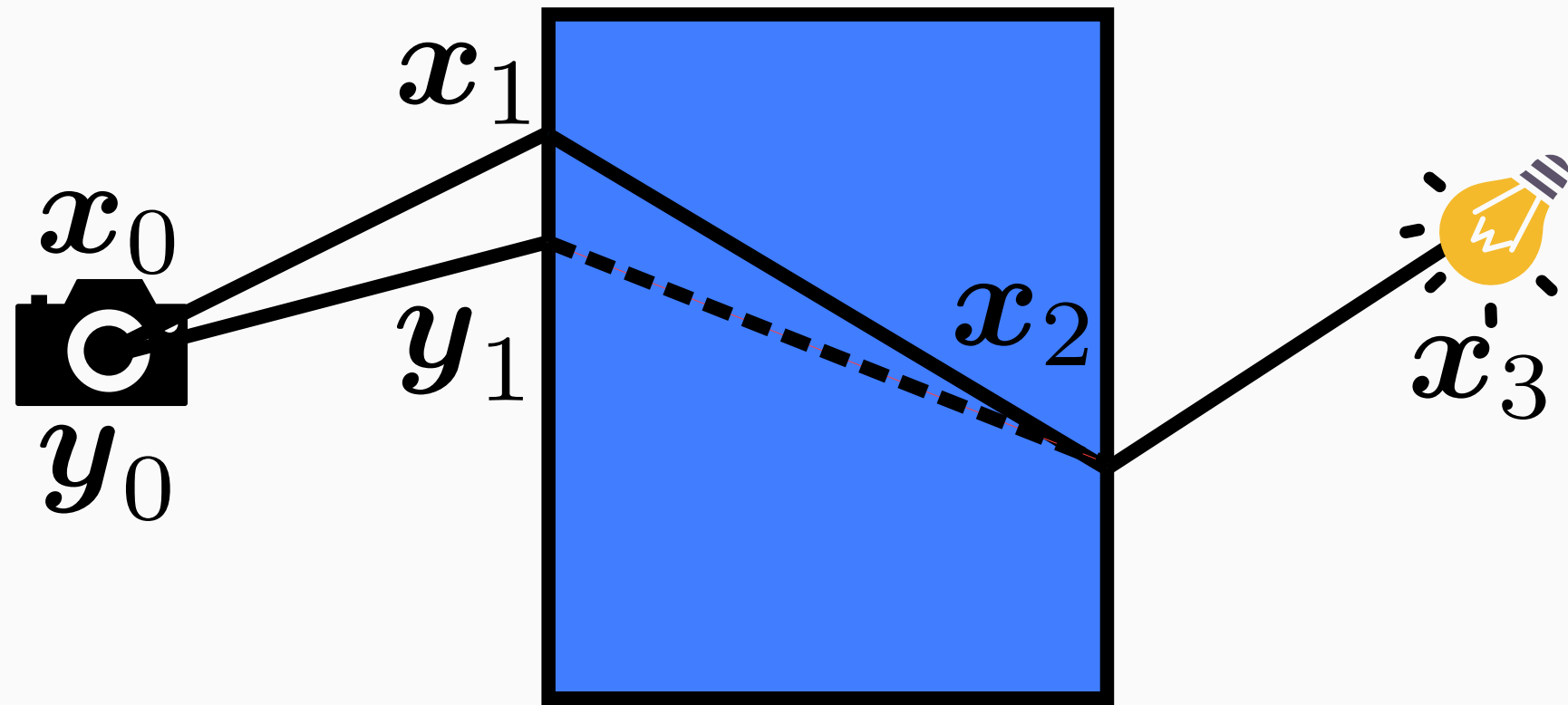


$$T([x_0, x_1, x_2, x_3]) = [y_0, y_1, y_2, \quad ]$$

delayed reconnection  
(random replay)

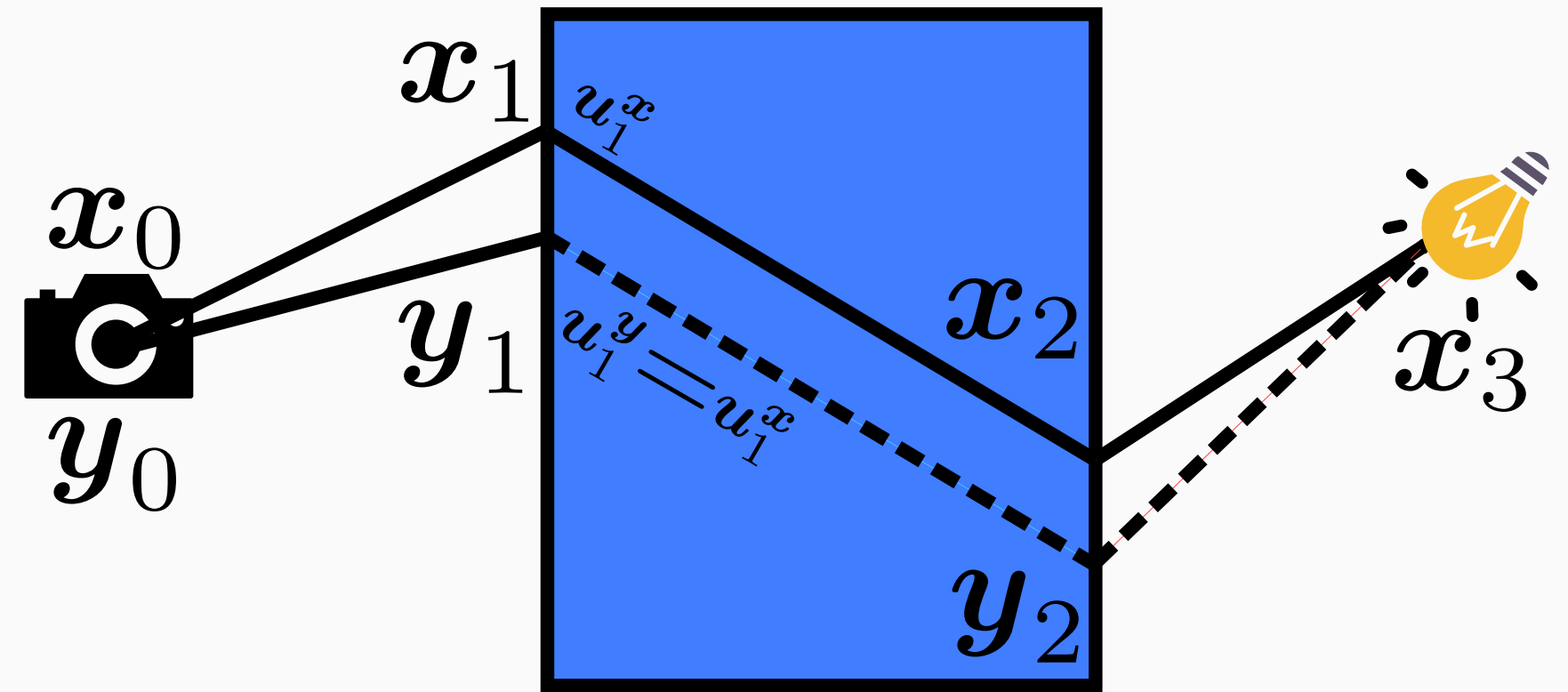


# ReSTIR SSS: Shifting Paths (naive)



$$T([x_0, x_1, x_2, x_3]) = [y_0, y_1, x_2, x_3]$$

reconnection



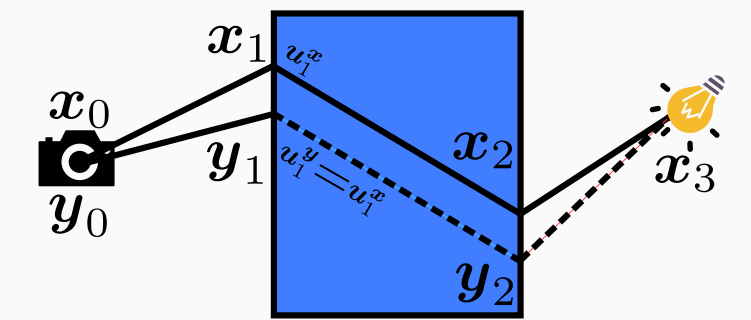
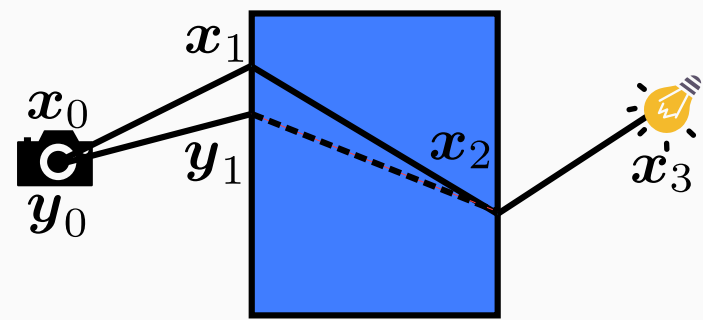
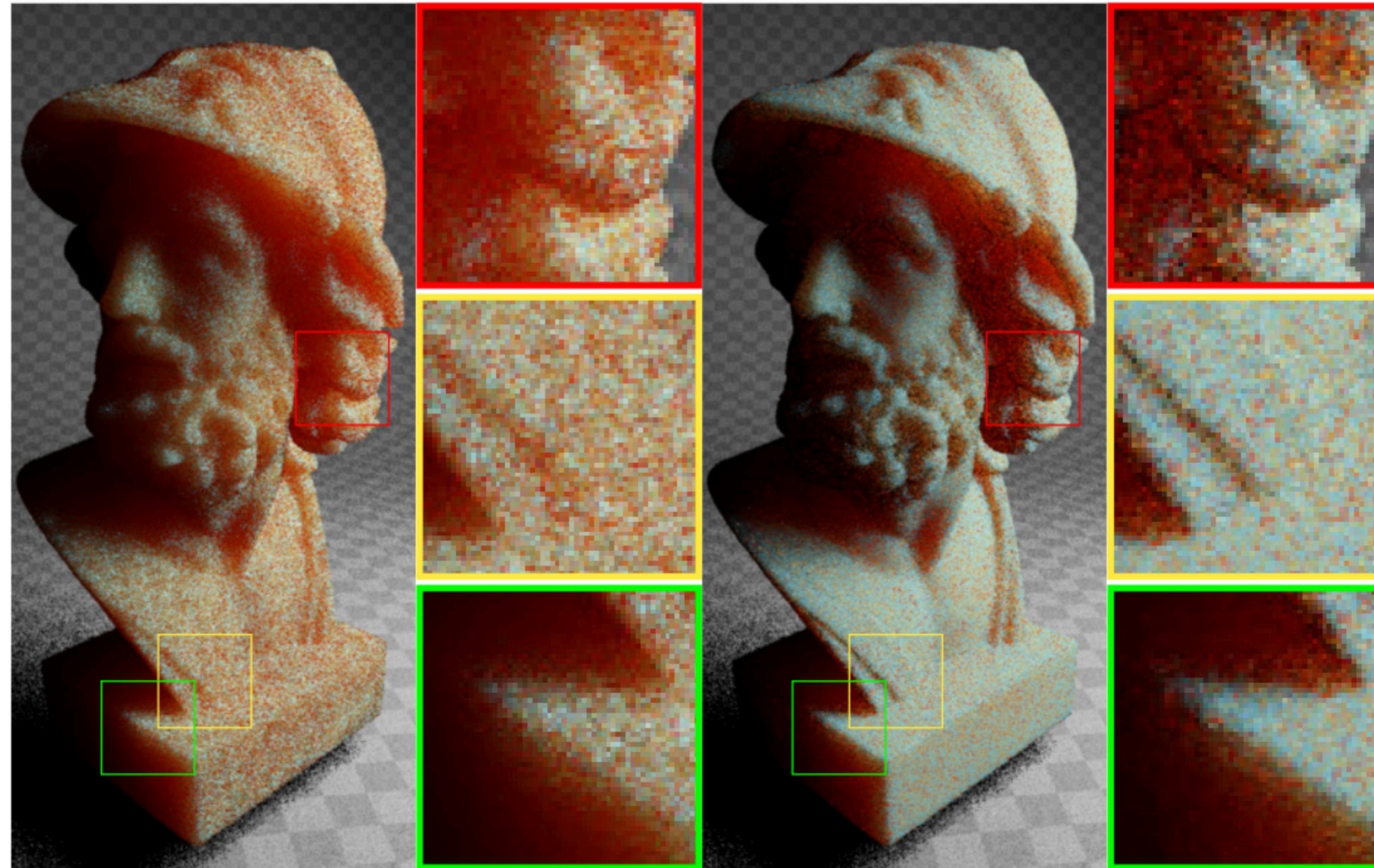
$$T([x_0, x_1, x_2, x_3]) = [y_0, y_1, y_2, x_3]$$

delayed reconnection  
(random replay + reconnection)

# ReSTIR SSS: Shifting Paths (naive)

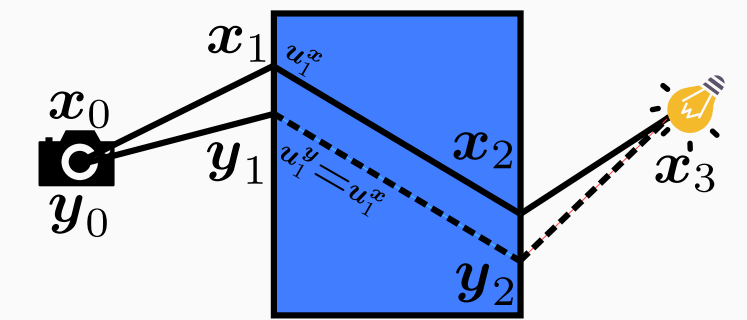
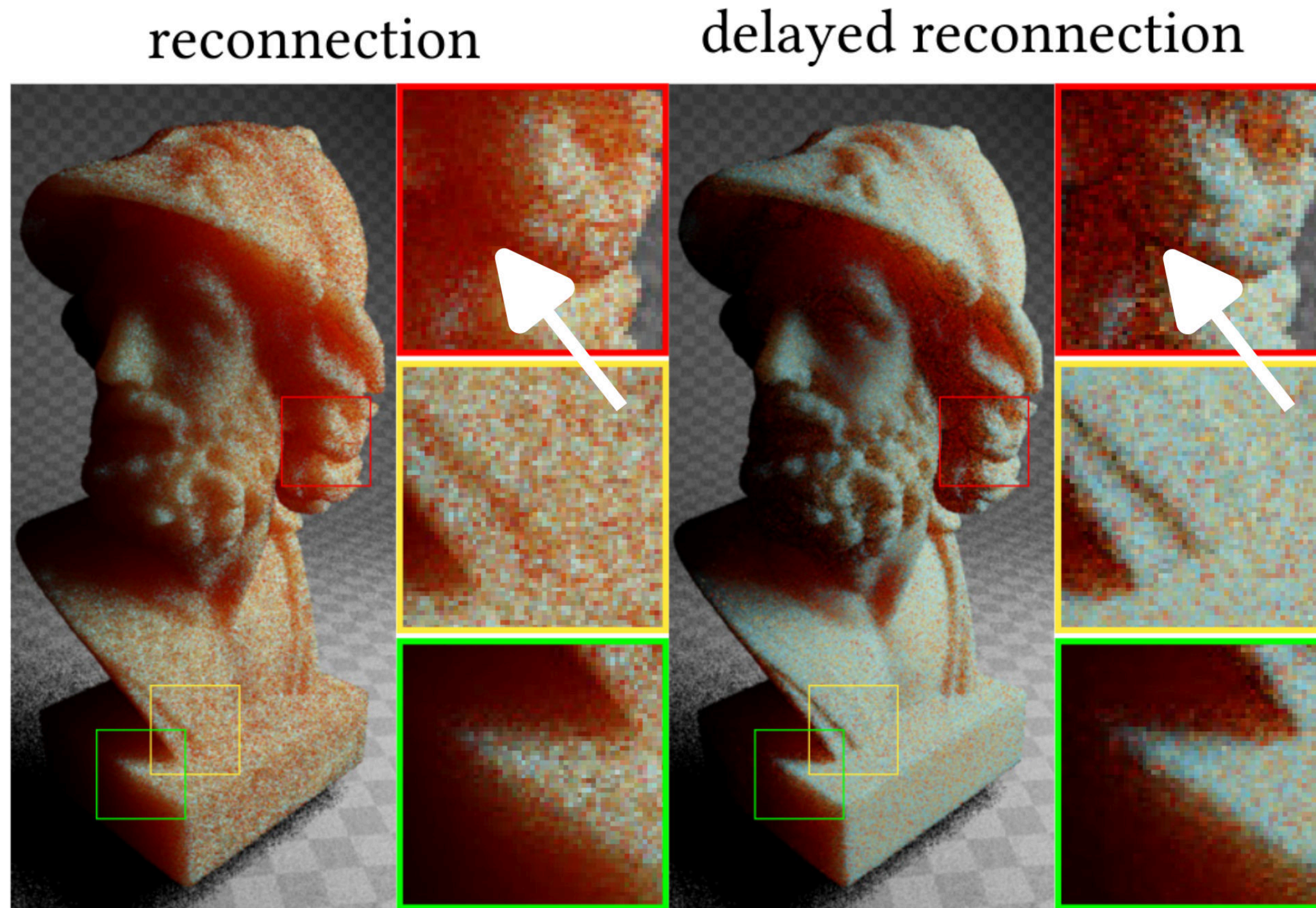
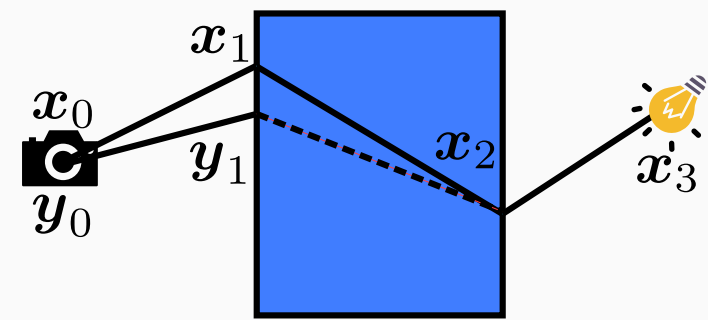
reconnection

delayed reconnection





# ReSTIR SSS: Shifting Paths (naive)

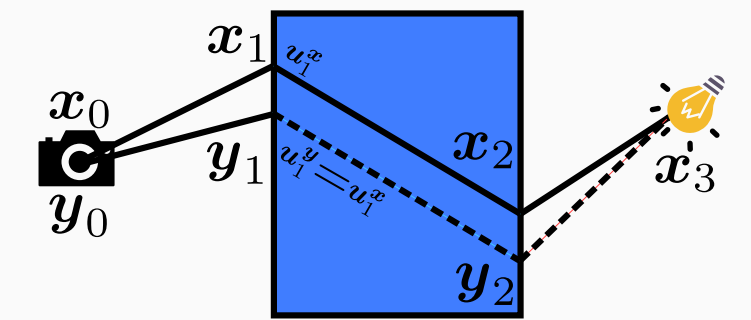
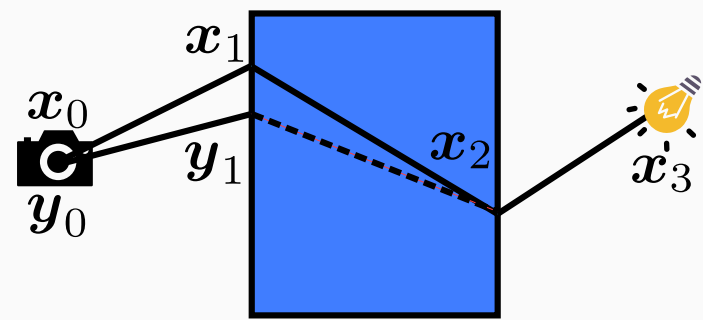
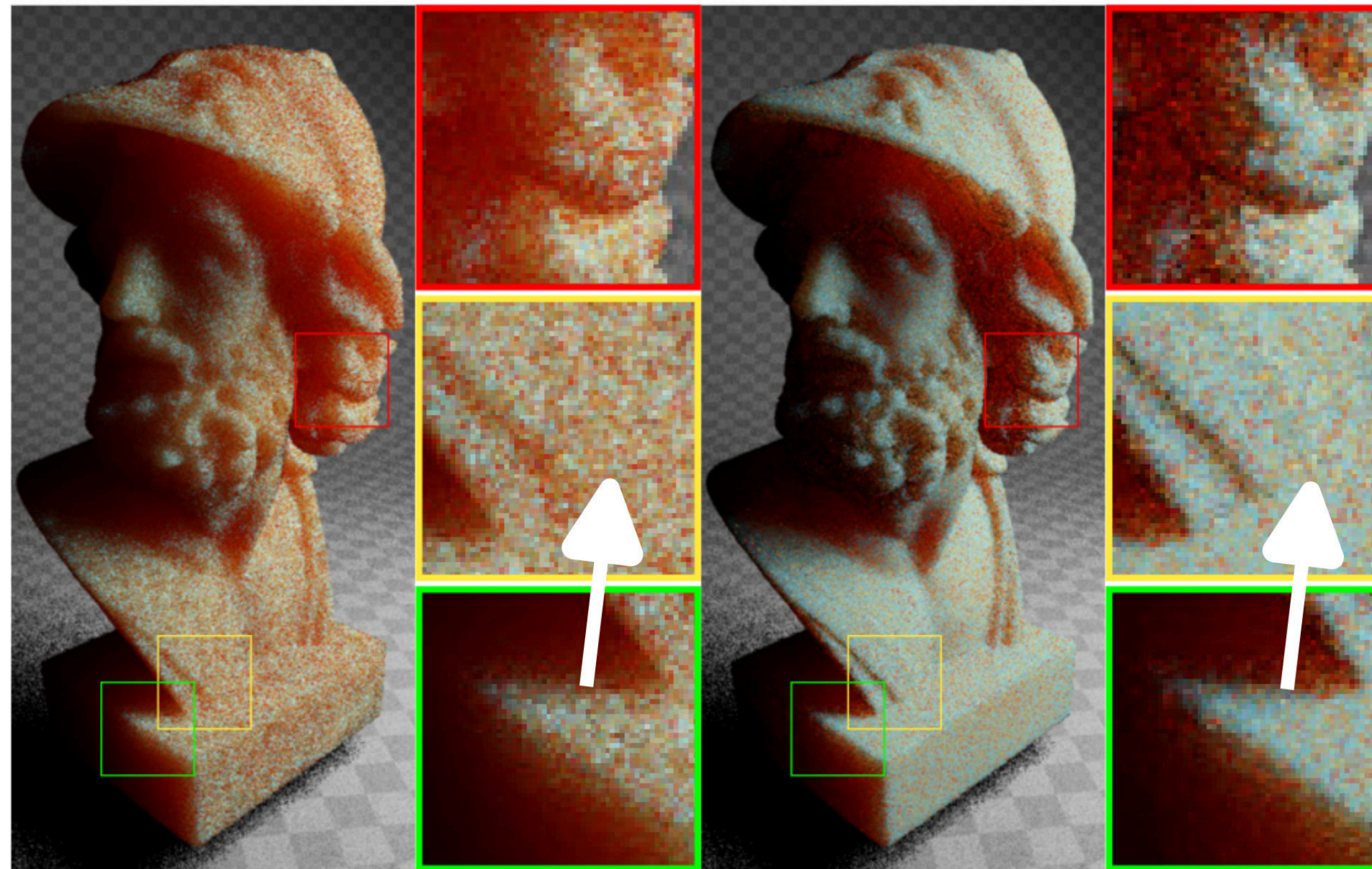




# ReSTIR SSS: Shifting Paths (naive)

reconnection

delayed reconnection

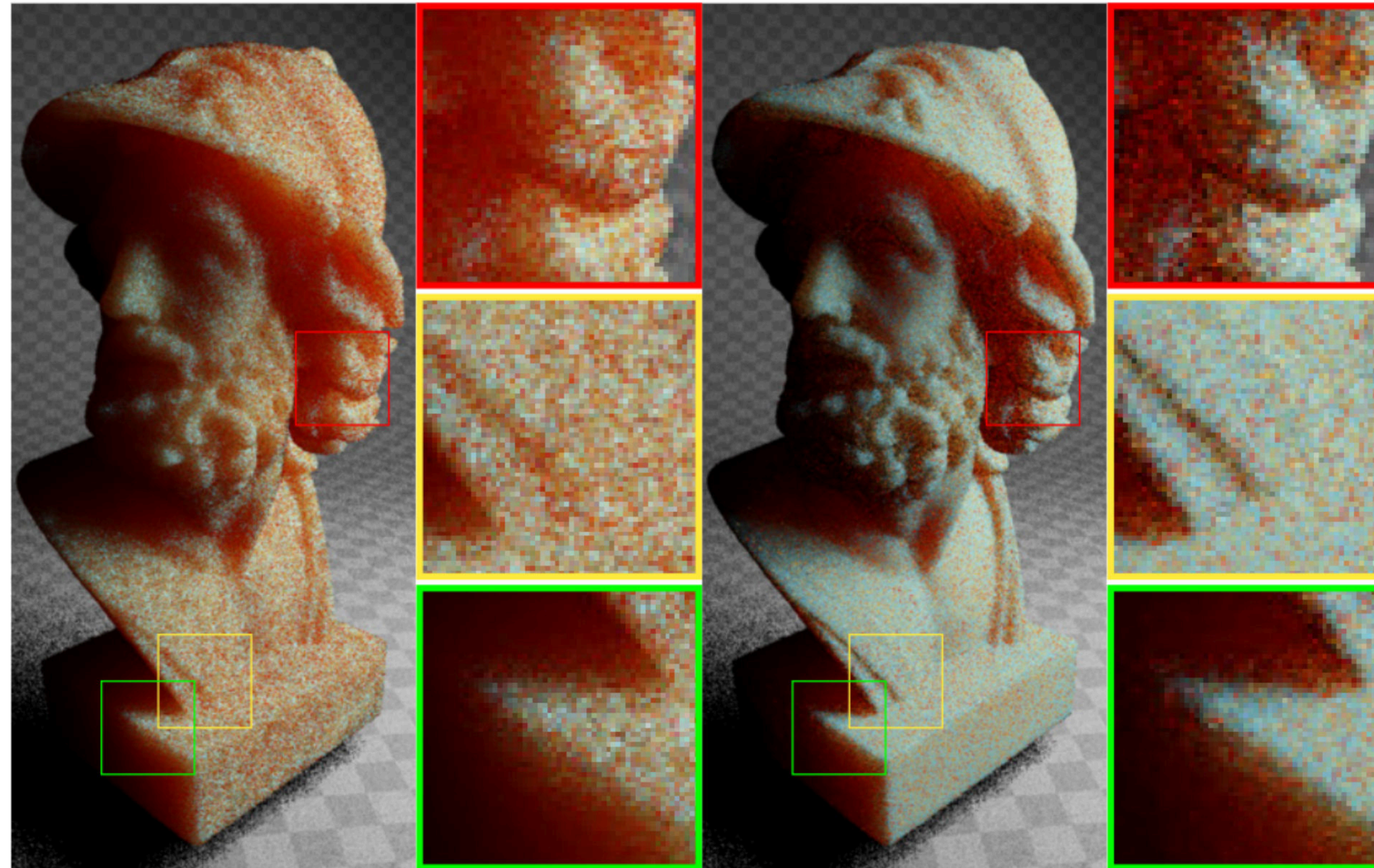




# ReSTIR SSS: Shifting Paths (naive)

reconnection

delayed reconnection



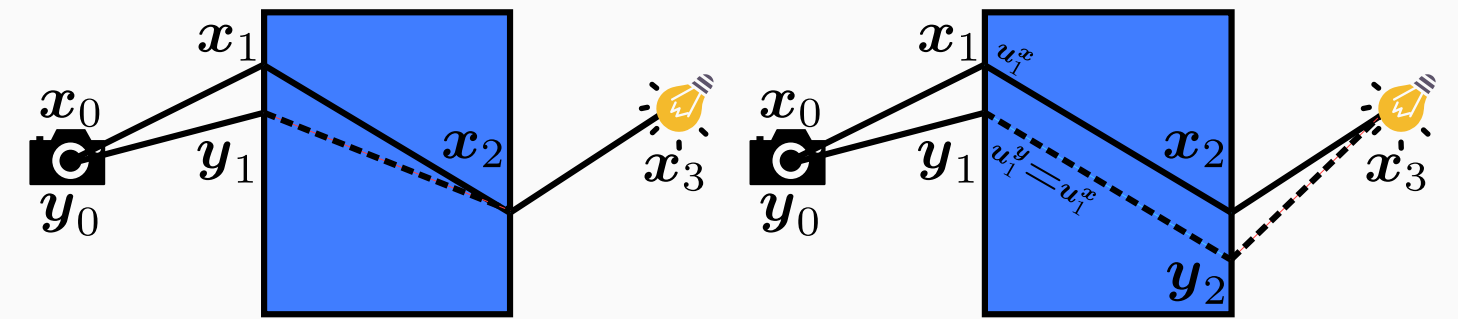
- need to combine both shifts!

# Advanced Shift Strategies (I)

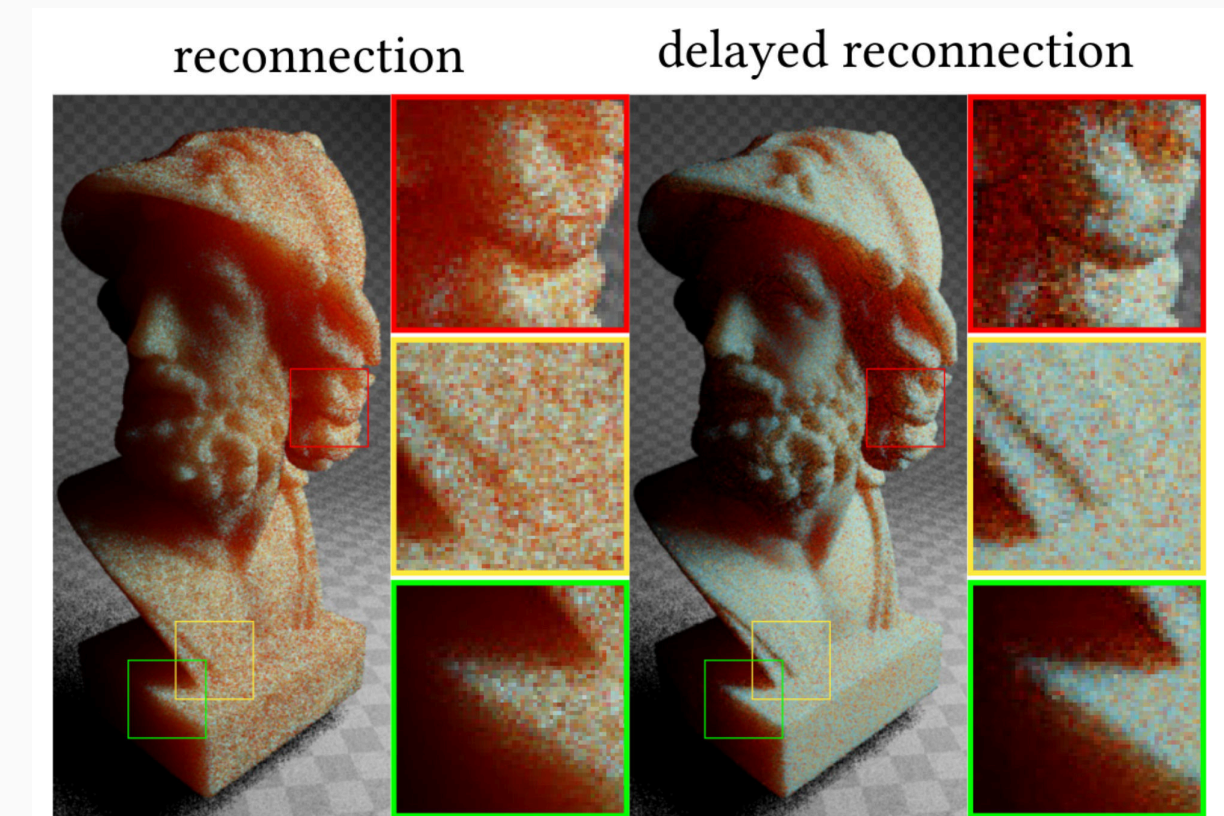
## Hybrid Shift



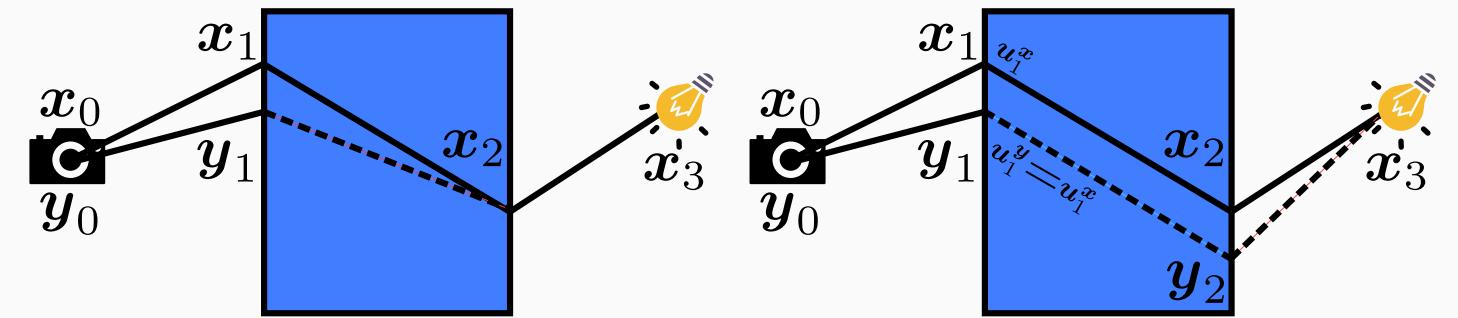
# ReSTIR SSS: Hybrid Shift



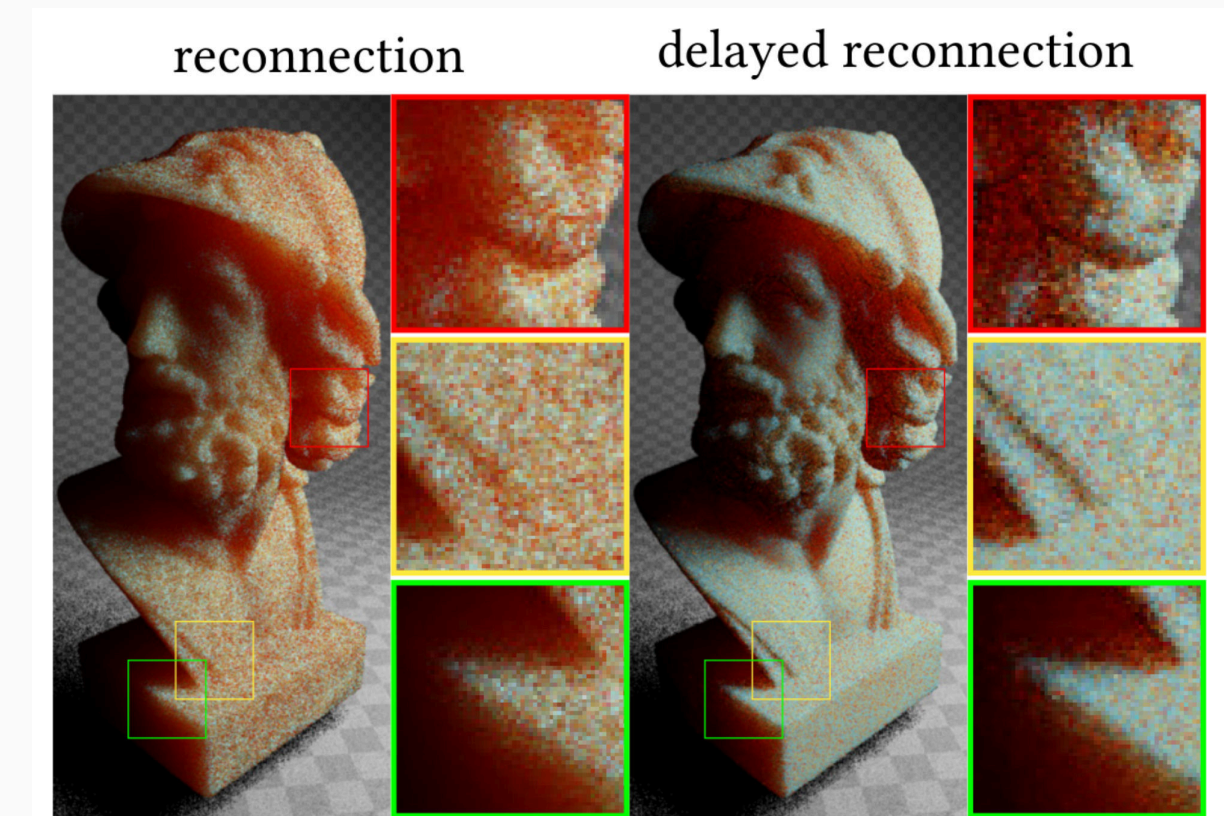
- choose either  $x_2$  or  $x_3$  for reconnection
  - deterministic criterion



# ReSTIR SSS: Hybrid Shift



- choose either  $x_2$  or  $x_3$  for reconnection
  - deterministic criterion
- idea: separate regions
  - shadowed (visible SSS)
  - illuminated

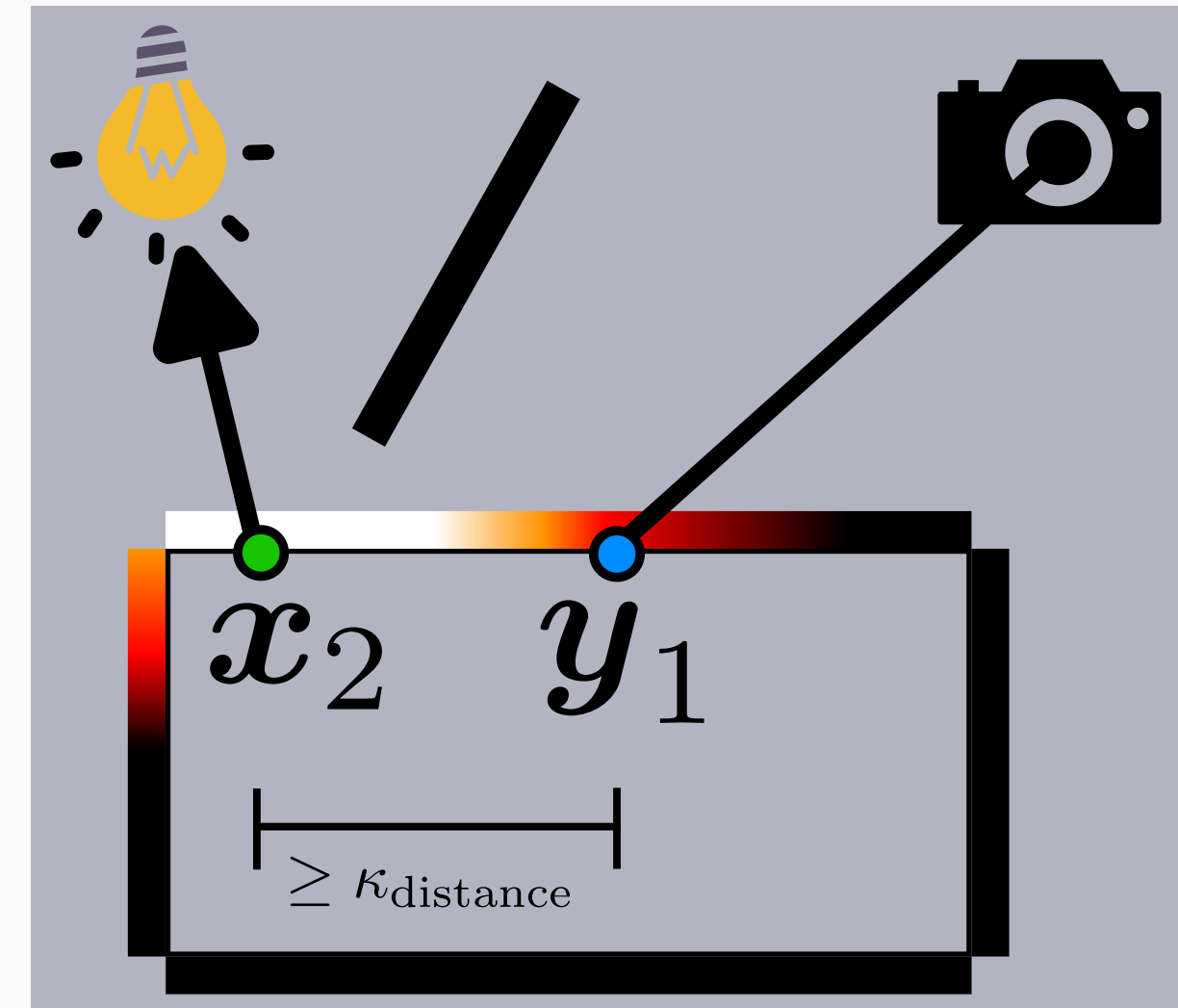




# ReSTIR SSS: Hybrid Shift

- choose either  $x_2$  or  $x_3$  for reconnection
  - deterministic criterion
- idea: separate regions
  - shadowed (visible SSS)
  - illuminated

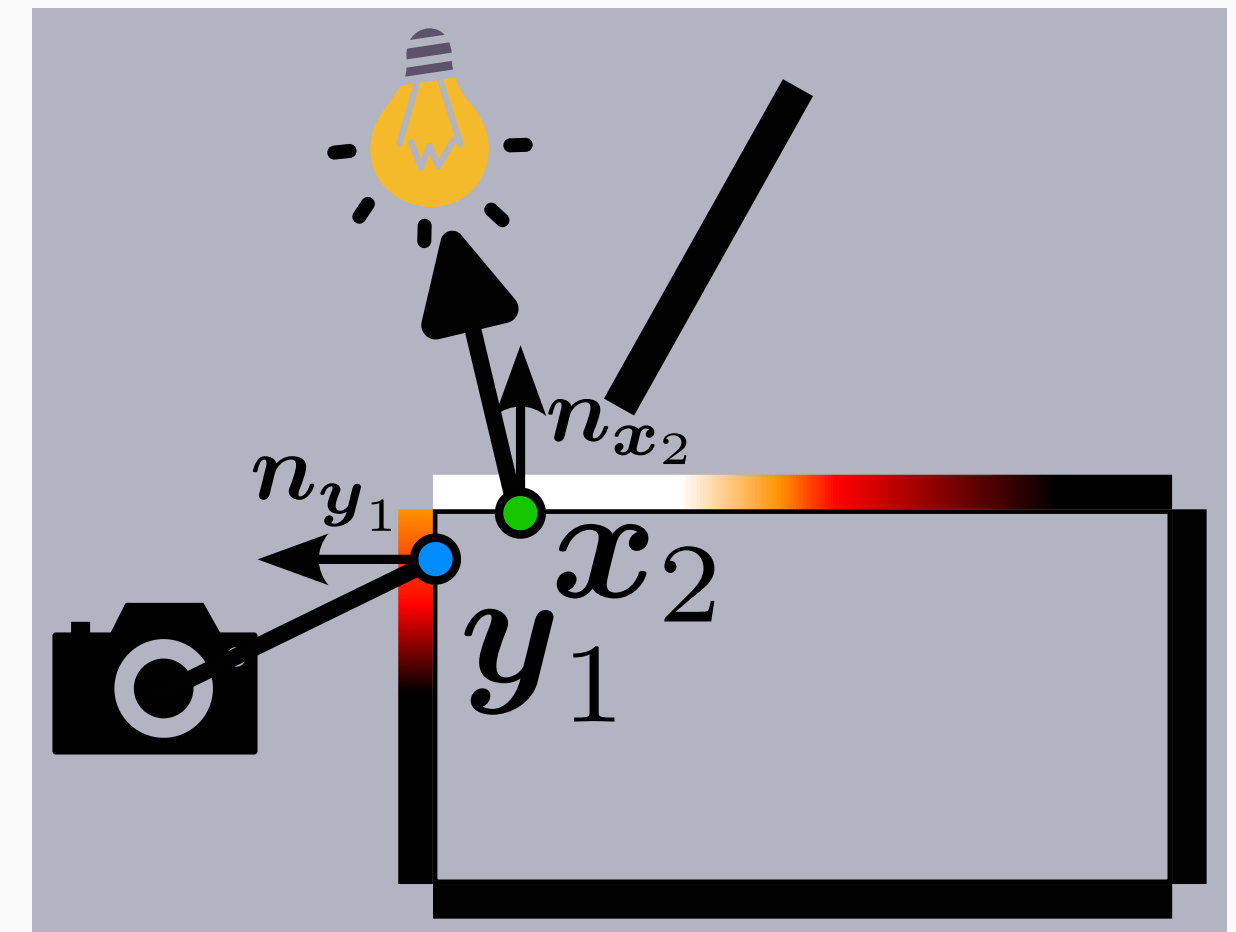
- our criterion: reconnect  $y_1 \rightarrow x_2$  if
  - $\|x_2 - y_1\| \geq \kappa_{\text{distance}}$



# ReSTIR SSS: Hybrid Shift

- choose either  $x_2$  or  $x_3$  for reconnection
  - deterministic criterion
- idea: separate regions
  - shadowed (visible SSS)
  - illuminated

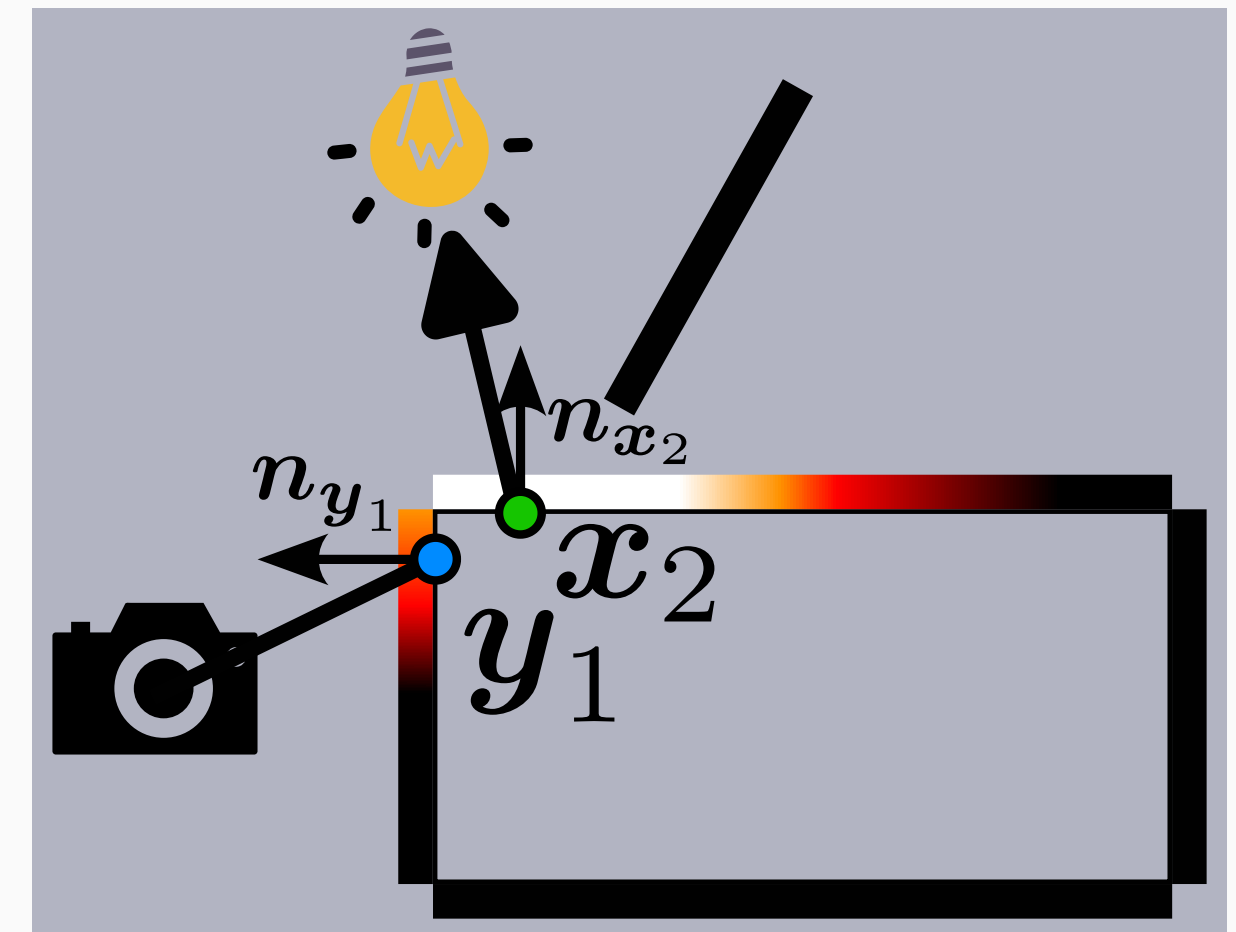
- our criterion: reconnect  $y_1 \rightarrow x_2$  if
  - $\|\mathbf{x}_2 - \mathbf{y}_1\| \geq \kappa_{\text{distance}}$  or
  - $\mathbf{n}_{x_2} \cdot \mathbf{n}_{y_1} \leq \kappa_{\text{orientation}}$



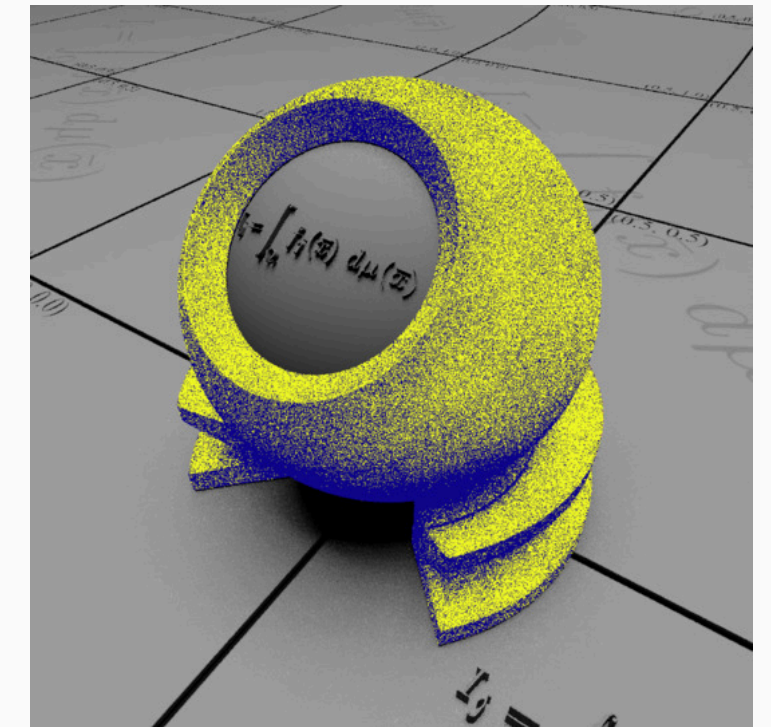
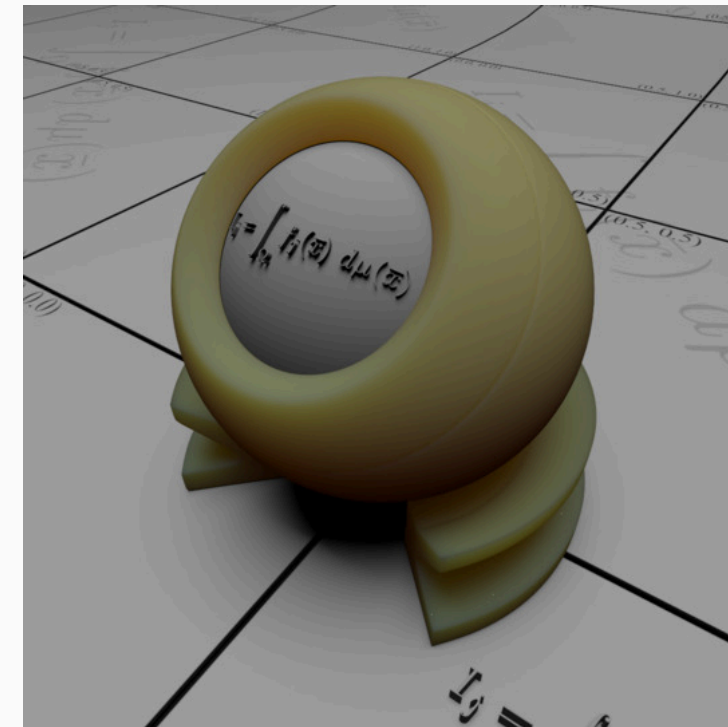
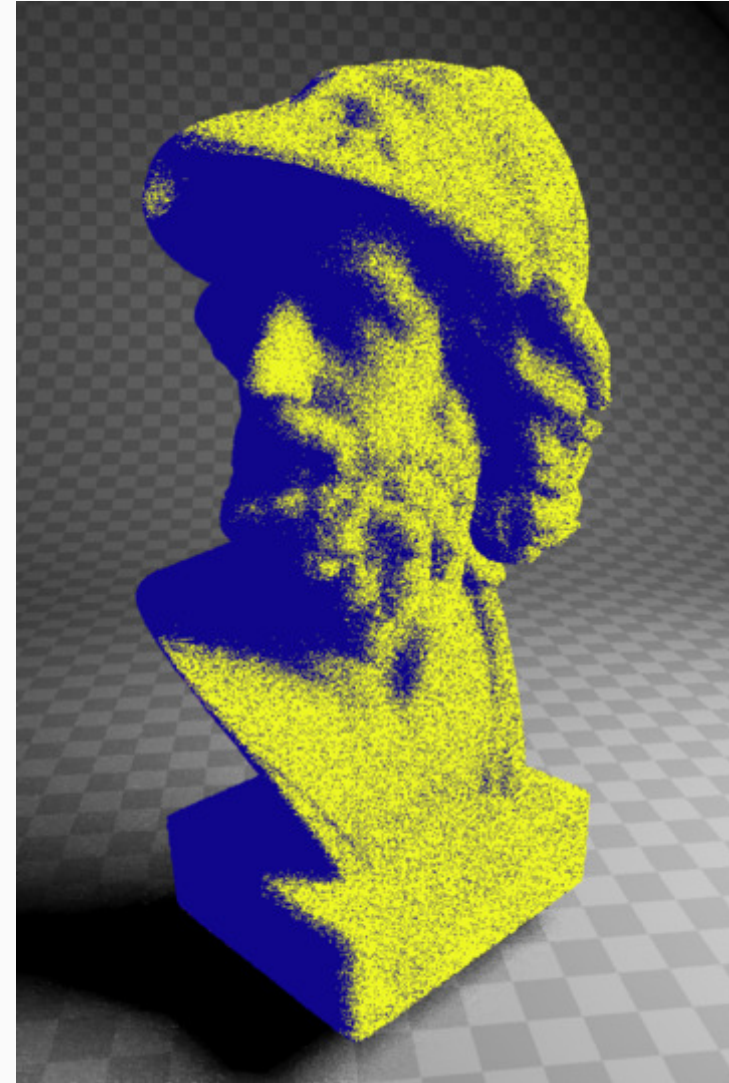
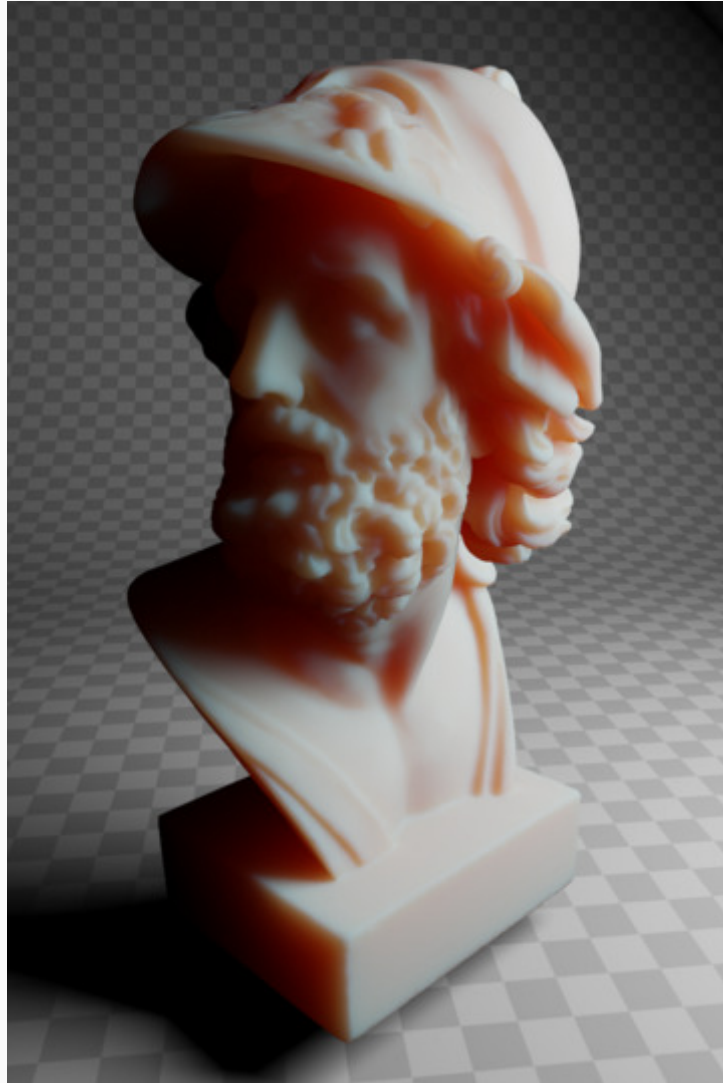


# ReSTIR SSS: Hybrid Shift

- choose either  $x_2$  or  $x_3$  for reconnection
  - deterministic criterion
- idea: separate regions
  - shadowed (visible SSS)
  - illuminated
- our criterion: reconnect  $y_1 \rightarrow x_2$  if
  - $\|\mathbf{x}_2 - \mathbf{y}_1\| \geq \kappa_{\text{distance}}$  or
  - $\mathbf{n}_{x_2} \cdot \mathbf{n}_{y_1} \leq \kappa_{\text{orientation}}$
- otherwise random replay and reconnect  $y_2 \rightarrow x_3$



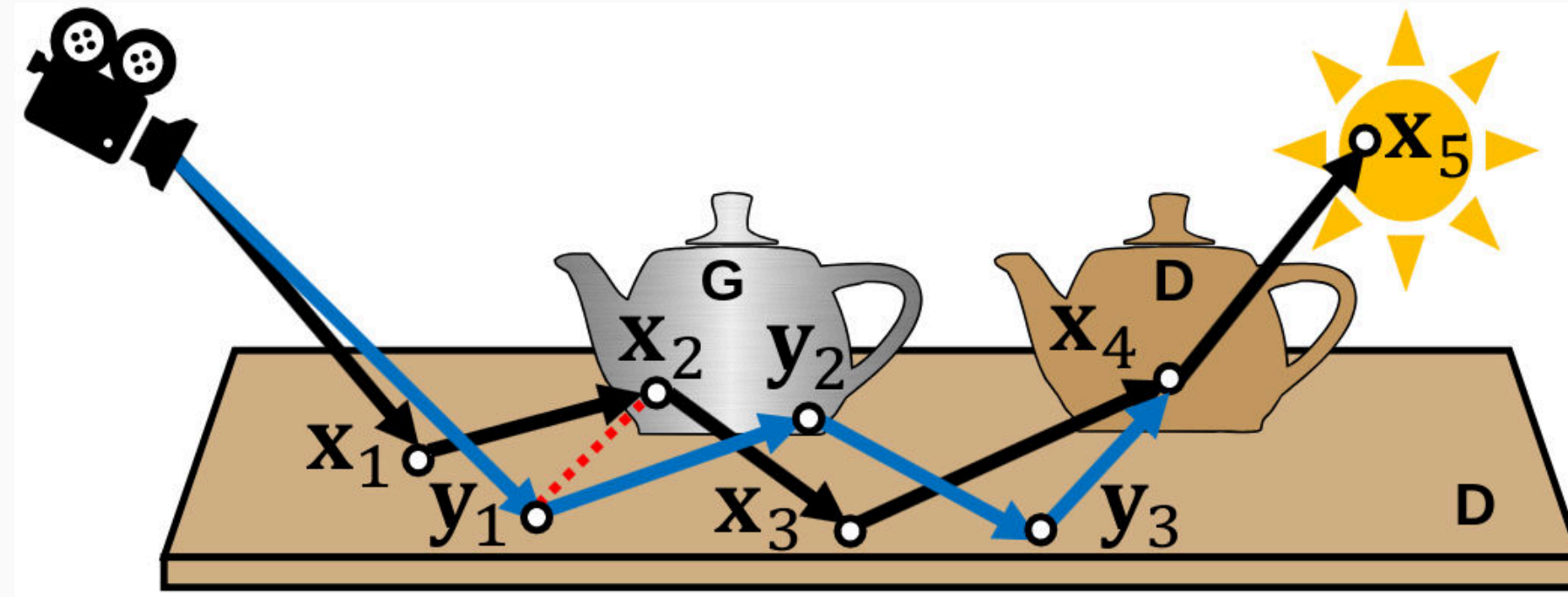
# ReSTIR SSS: Hybrid Shift



—■— reconnection —■— delayed reconnection



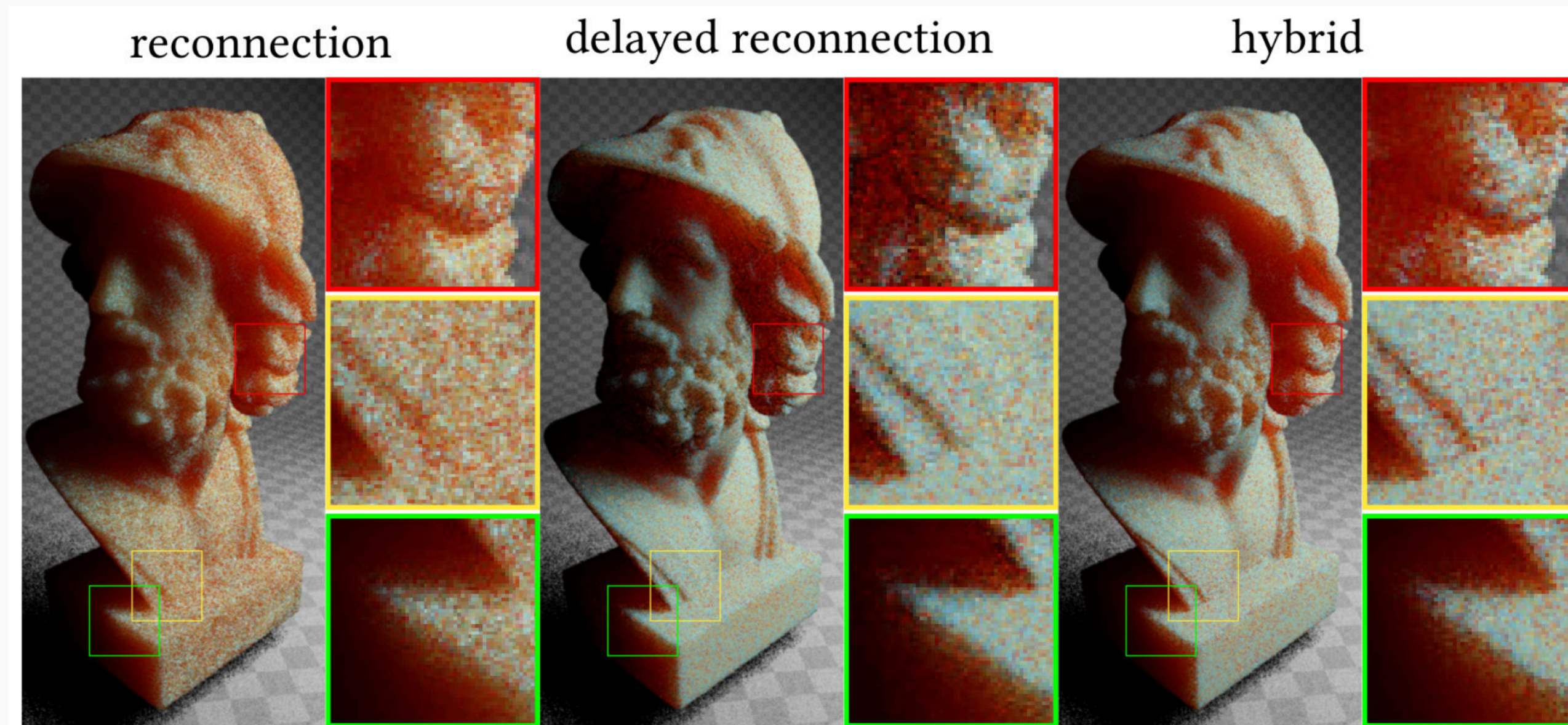
# ReSTIR SSS: Hybrid Shift



(image from [Lin et al. 2022])

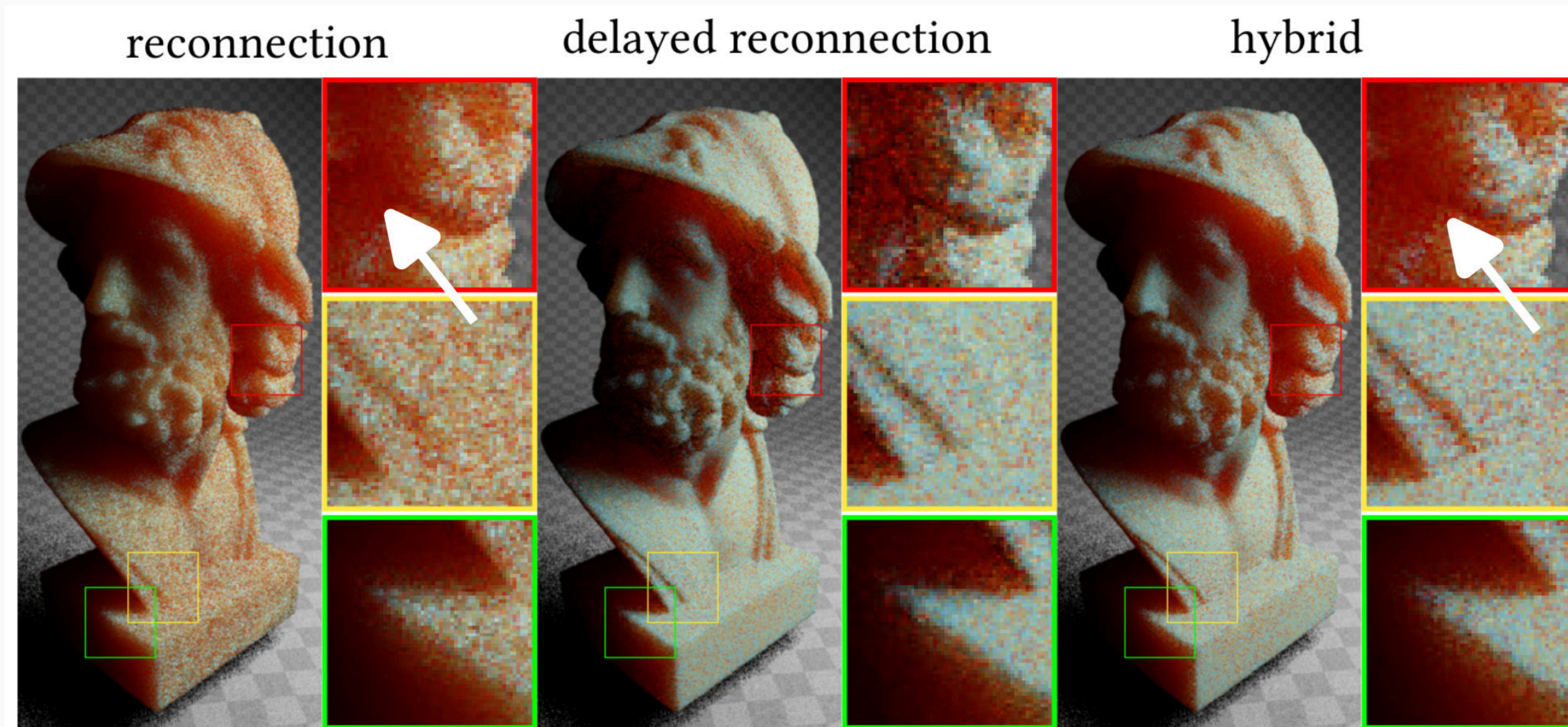
- similar to ReSTIR PT's hybrid shift [Lin et al. 2022]

# ReSTIR SSS: Hybrid Shift



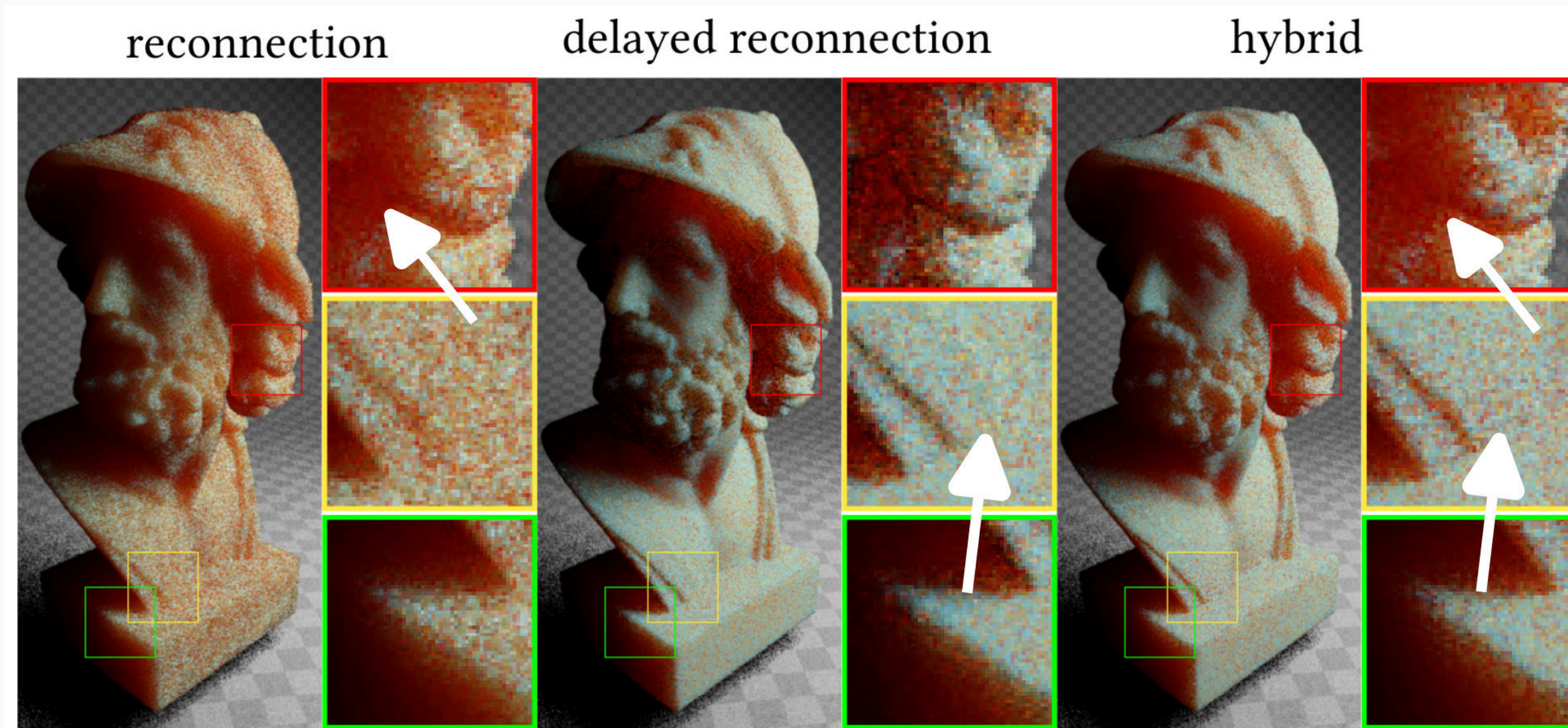


# ReSTIR SSS: Hybrid Shift



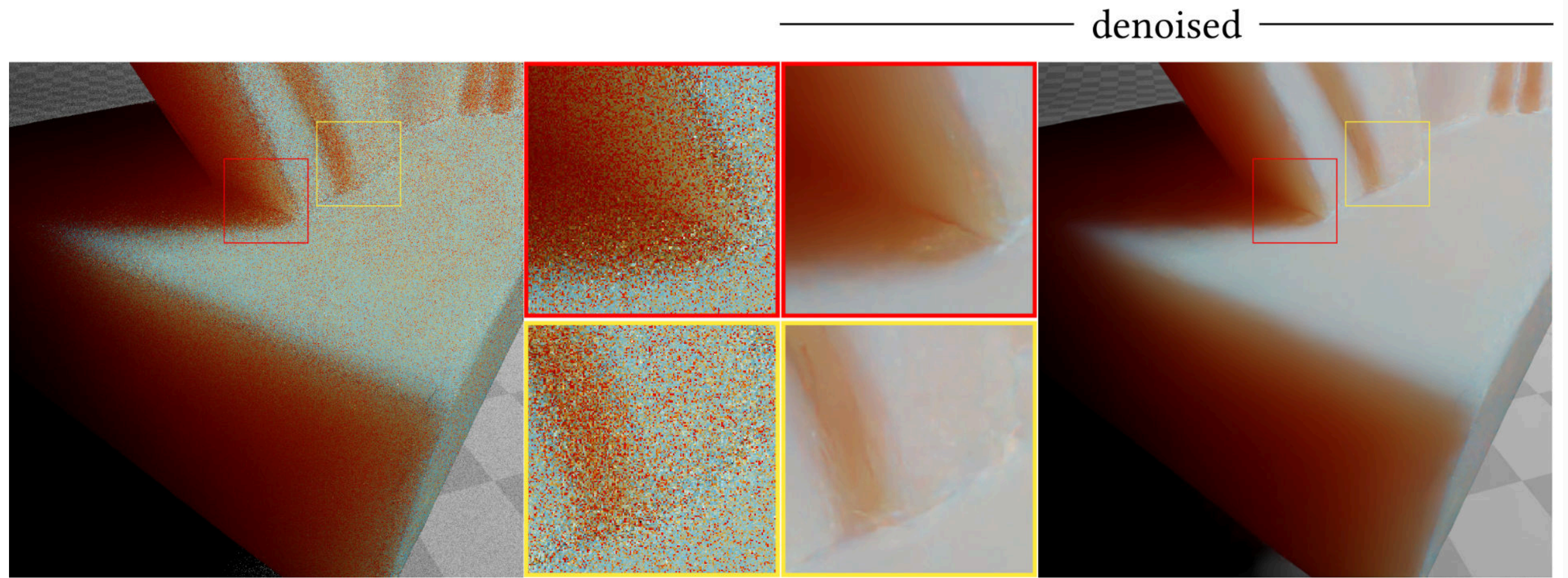


# ReSTIR SSS: Hybrid Shift



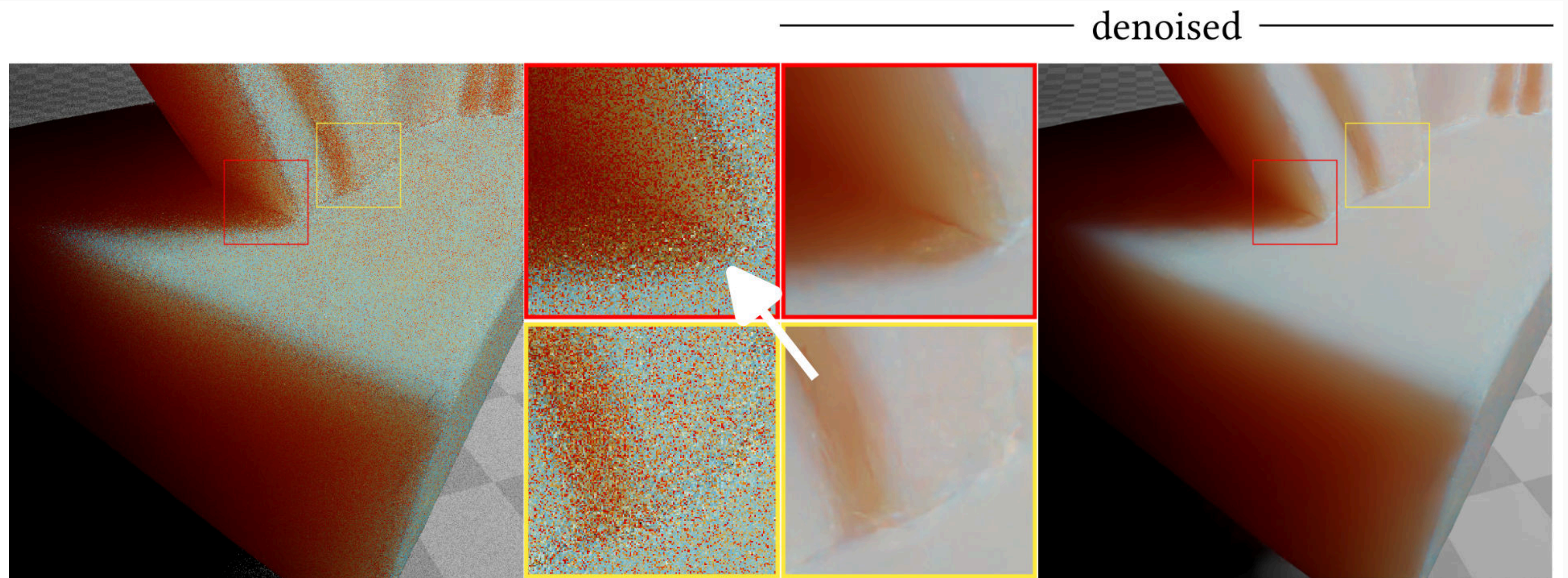


# ReSTIR SSS: Hybrid Shift



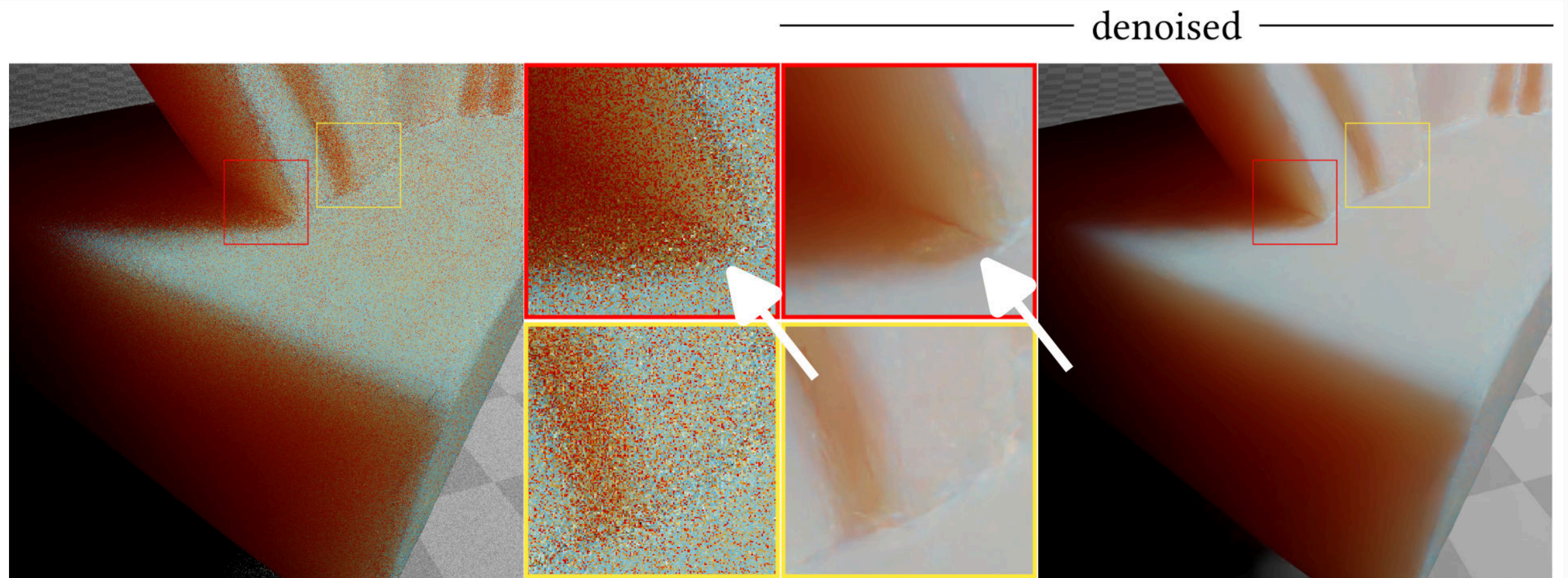


# ReSTIR SSS: Hybrid Shift





# ReSTIR SSS: Hybrid Shift

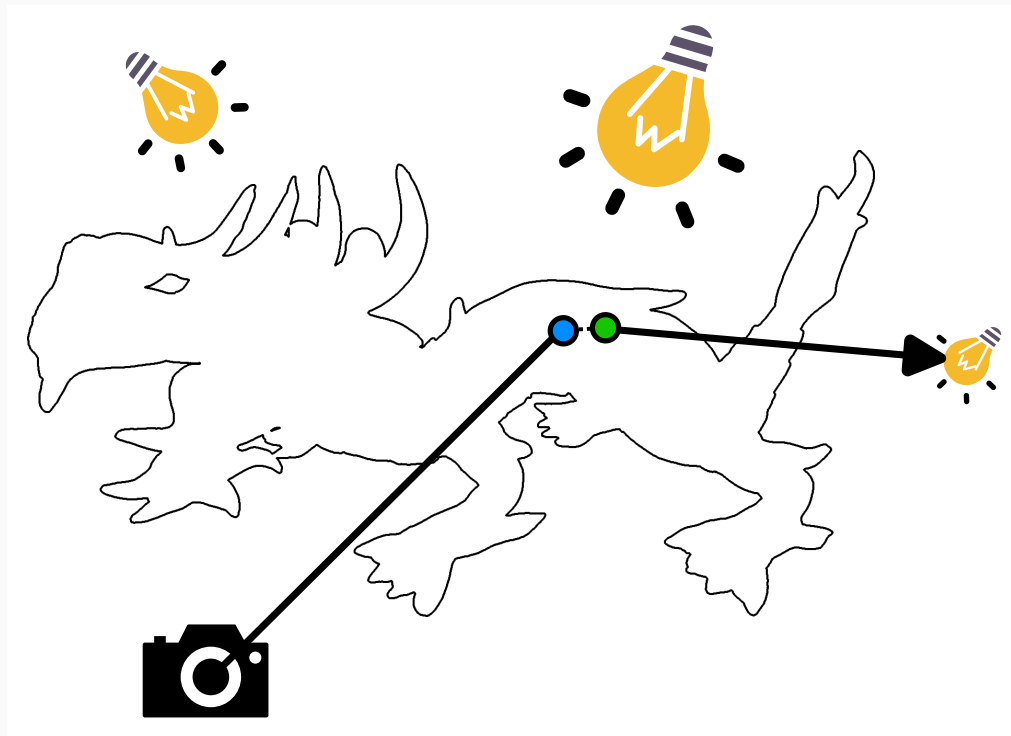


# Advanced Shift Strategies (II)

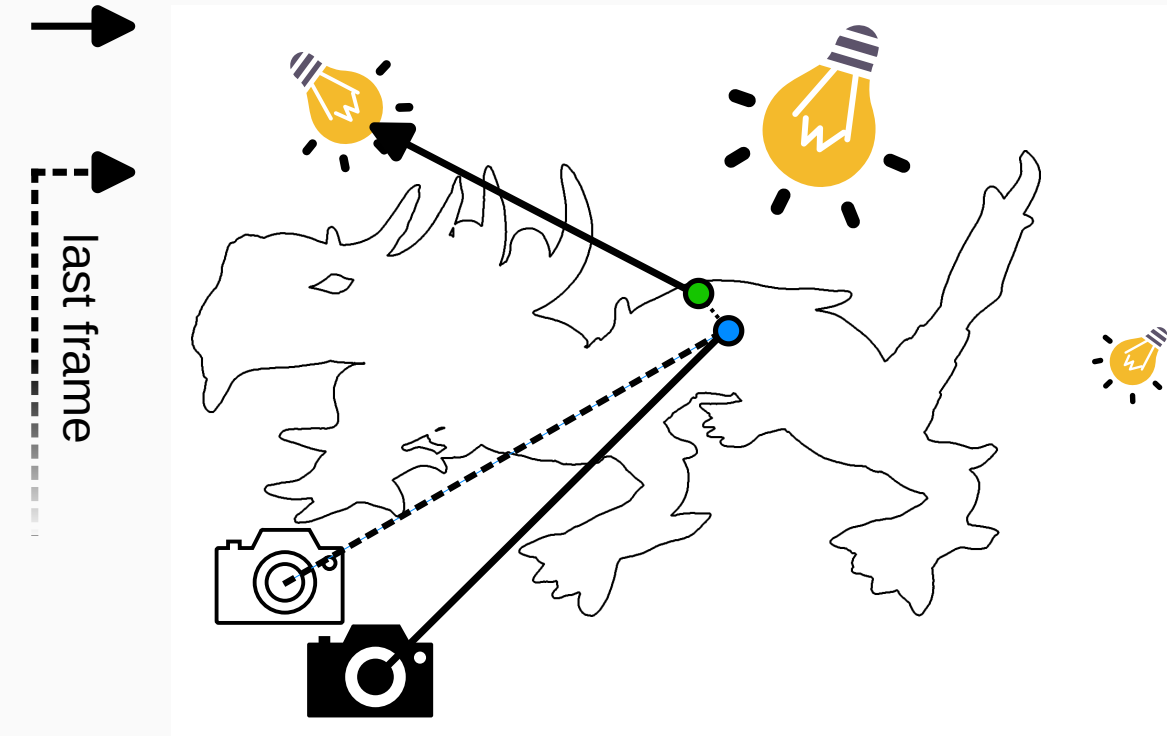
## Sequential Shift



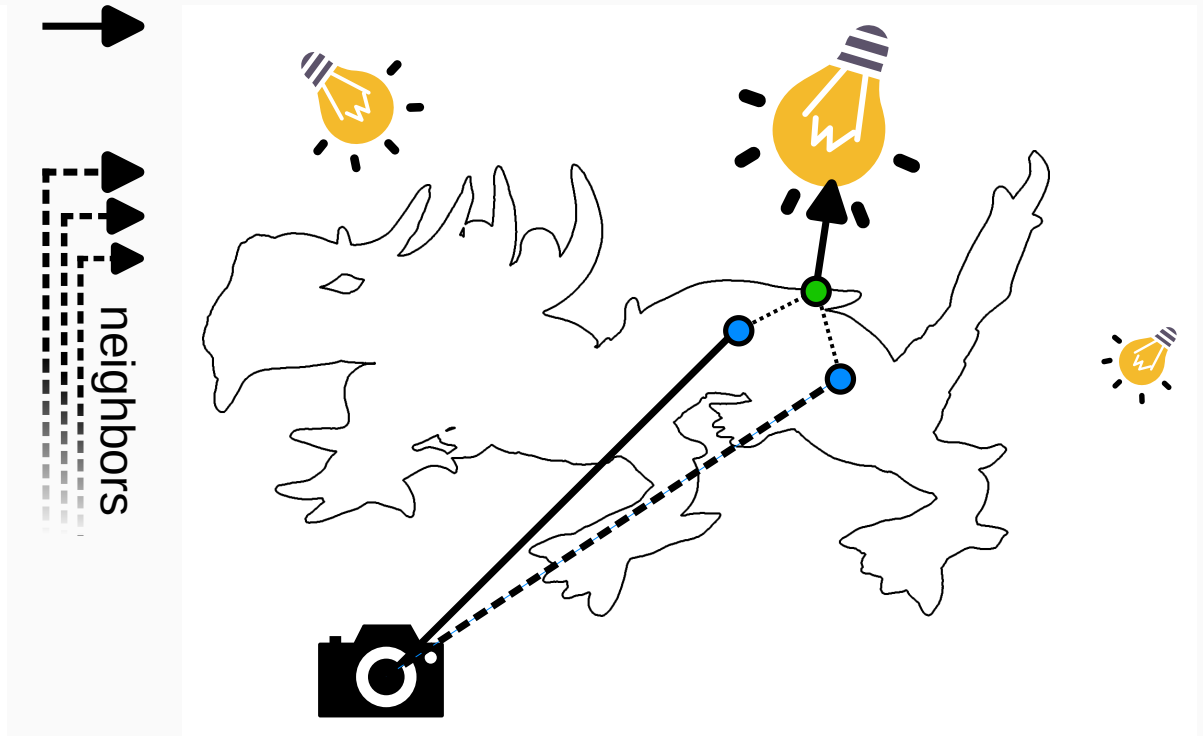
# ReSTIR SSS: Hybrid Shift



candidate generation



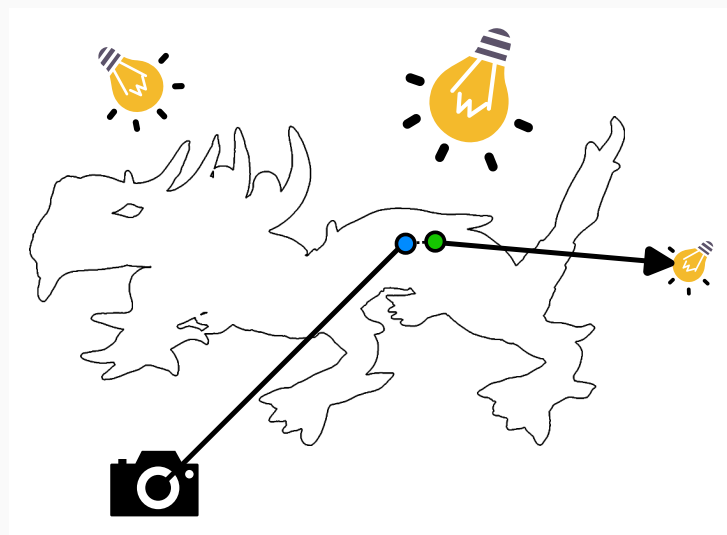
temporal reuse



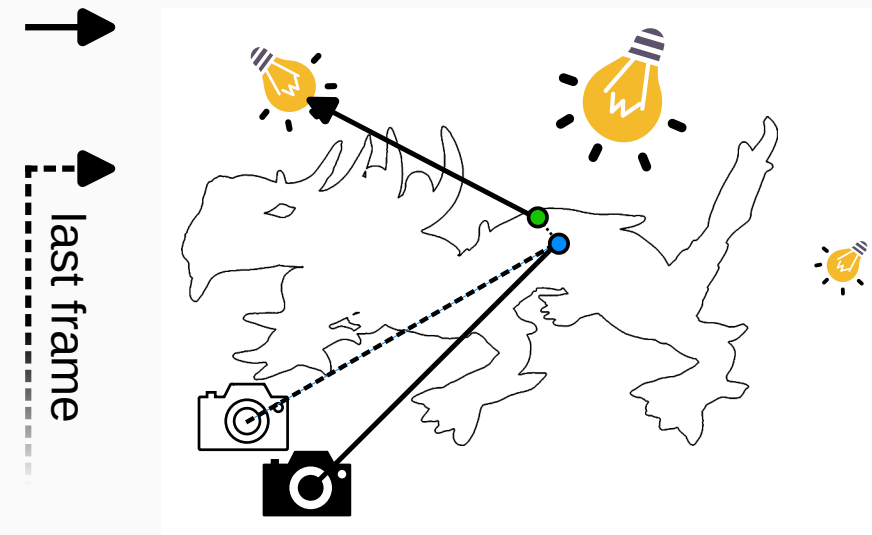
spatial reuse  
(either reconnection or delayed reconnection)

- try to shift a certain sample with only one deterministically selected shift

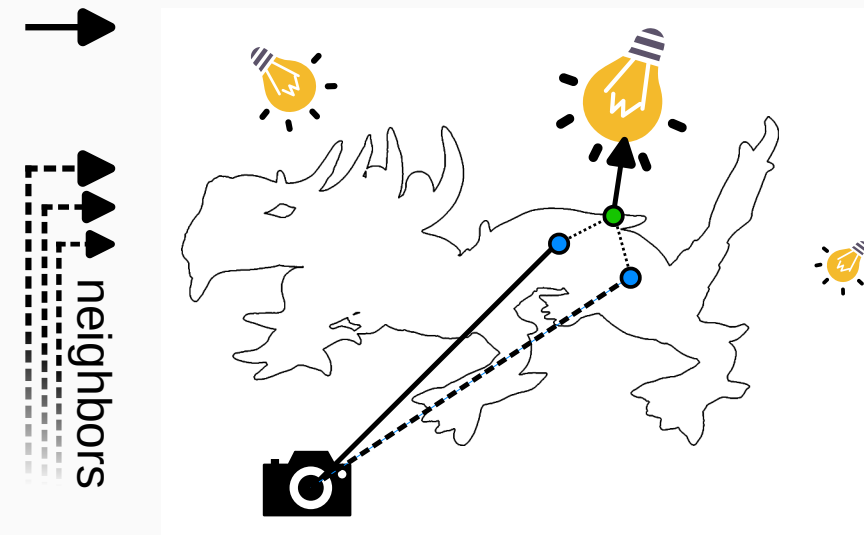
# ReSTIR SSS: Sequential Shift



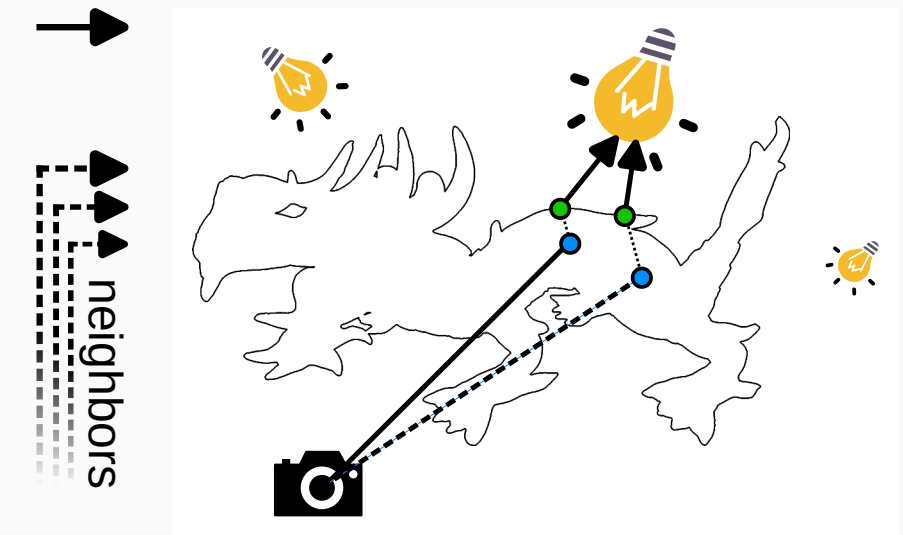
candidate generation



temporal reuse



spatial reuse  
(reconnection)



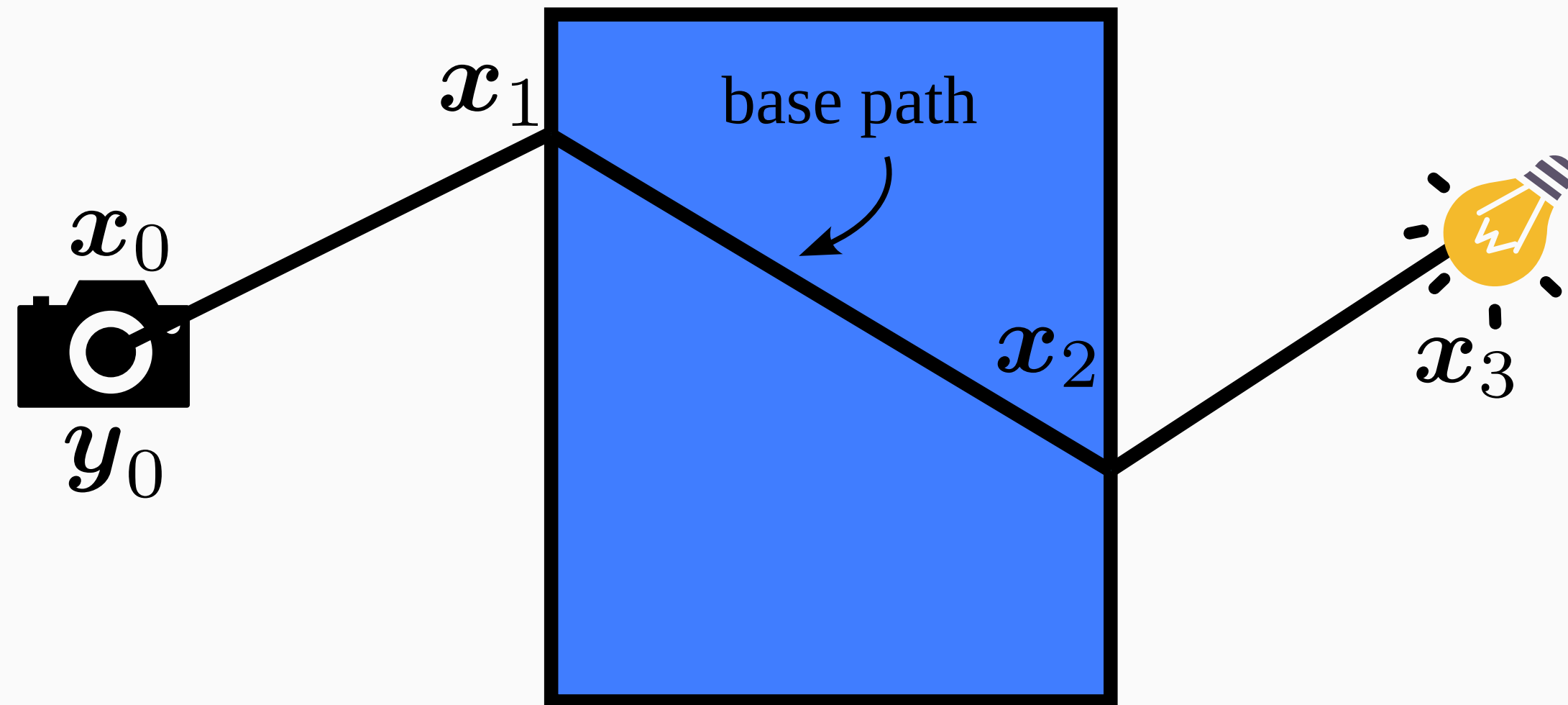
spatial reuse  
(delayed reconnection)

- try to shift a certain sample with both shifts
- best shift is chosen implicitly



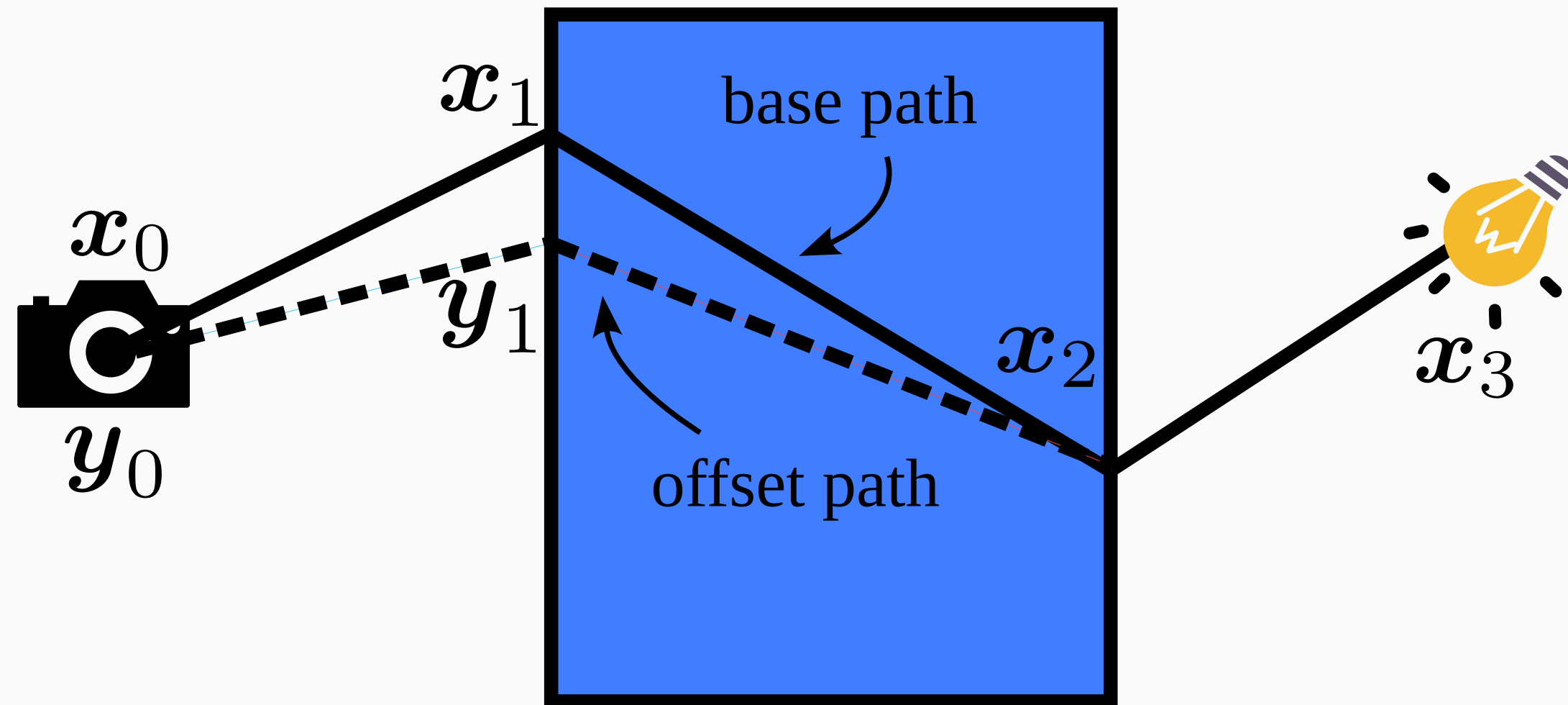
# ReSTIR SSS: Sequential Shift Example

## 1. spatial reuse (reconnection)



# ReSTIR SSS: Sequential Shift Example

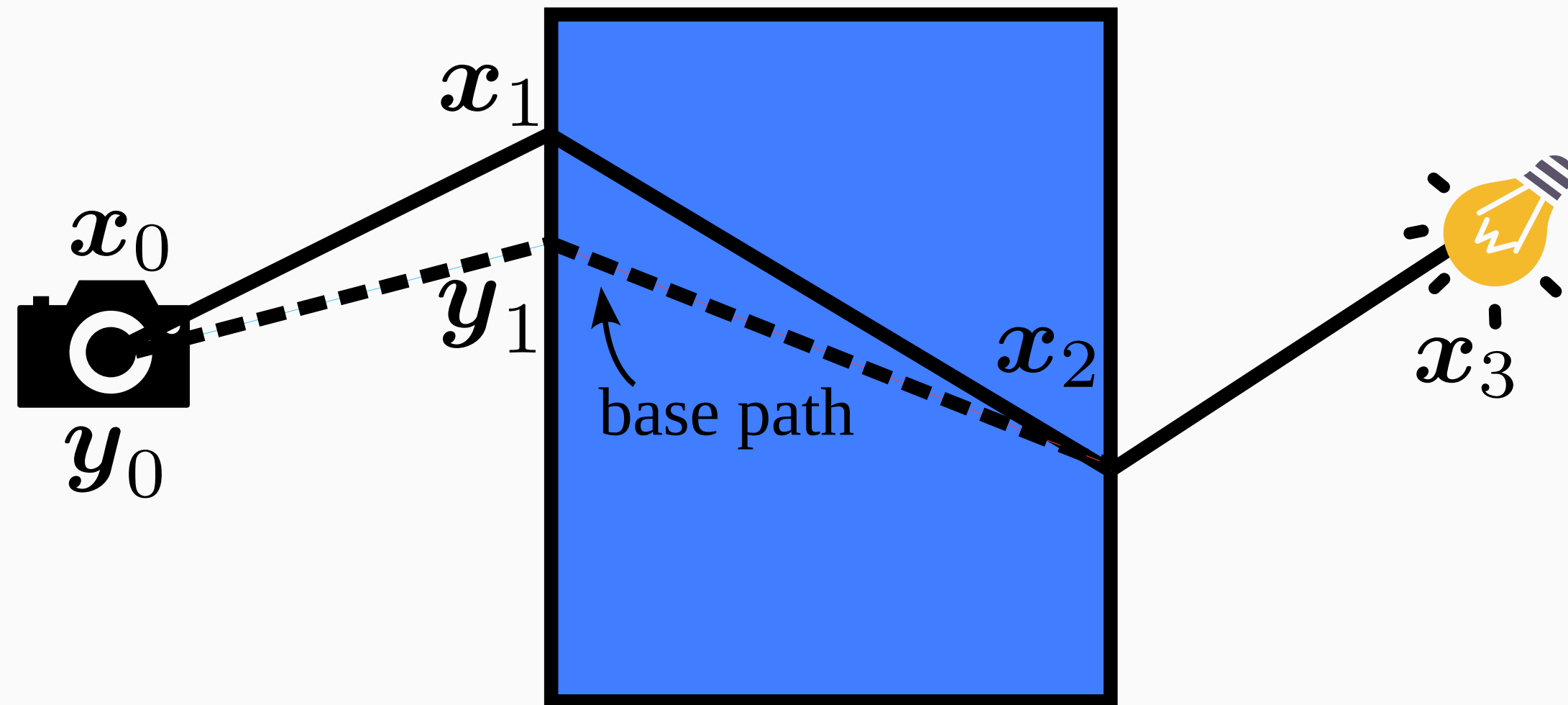
## 1. spatial reuse (reconnection)





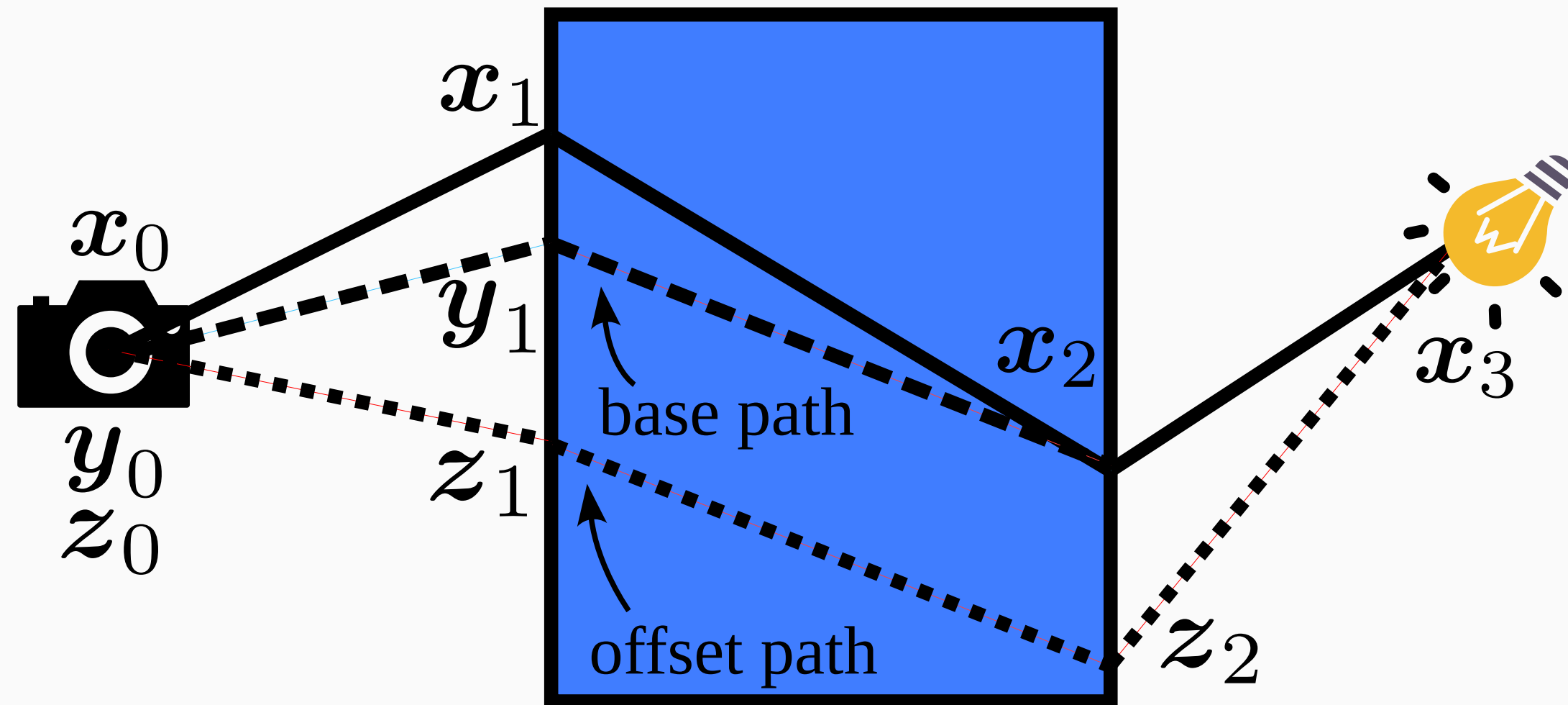
# ReSTIR SSS: Sequential Shift Example

1. spatial reuse (reconnection)
2. spatial reuse (delayed reconnection)



# ReSTIR SSS: Sequential Shift Example

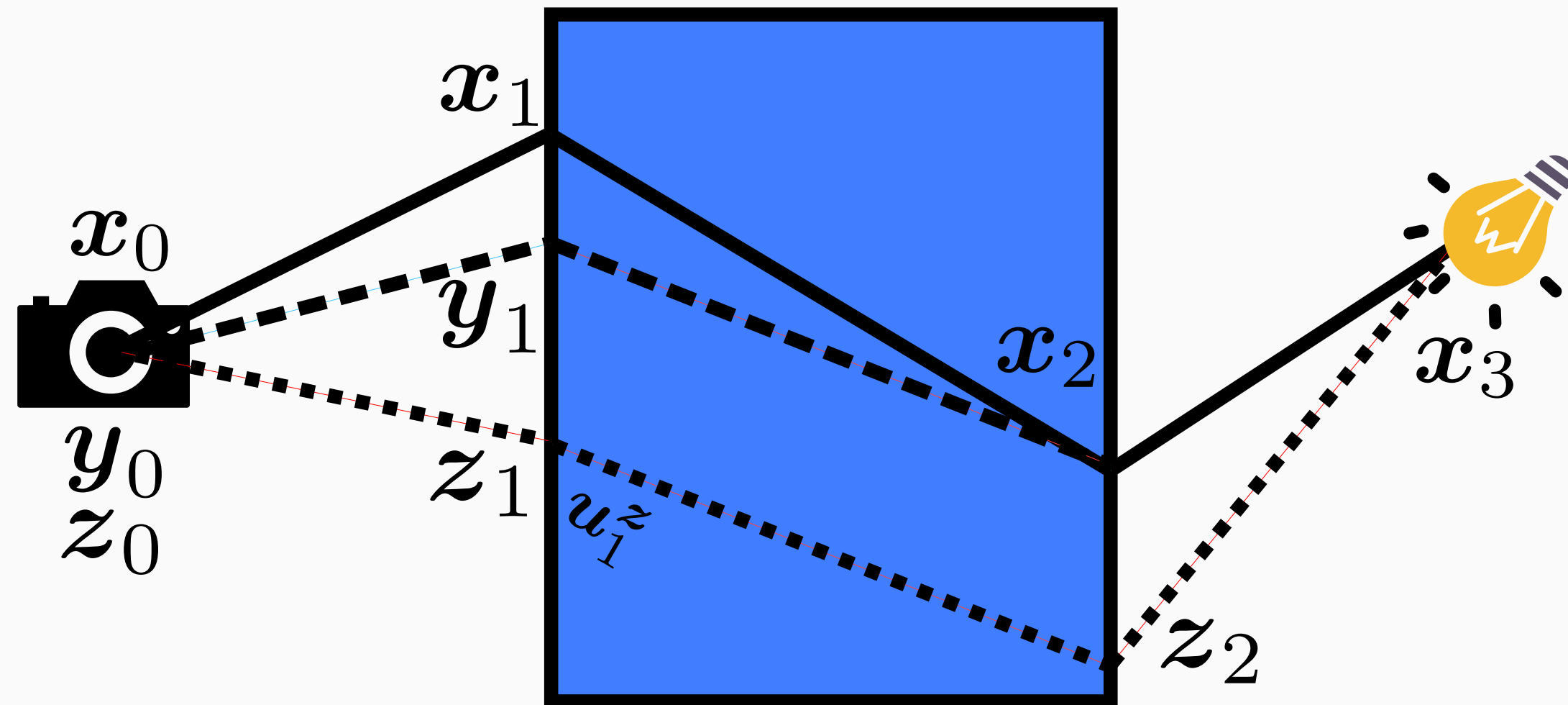
1. spatial reuse (reconnection)
2. spatial reuse (delayed reconnection)





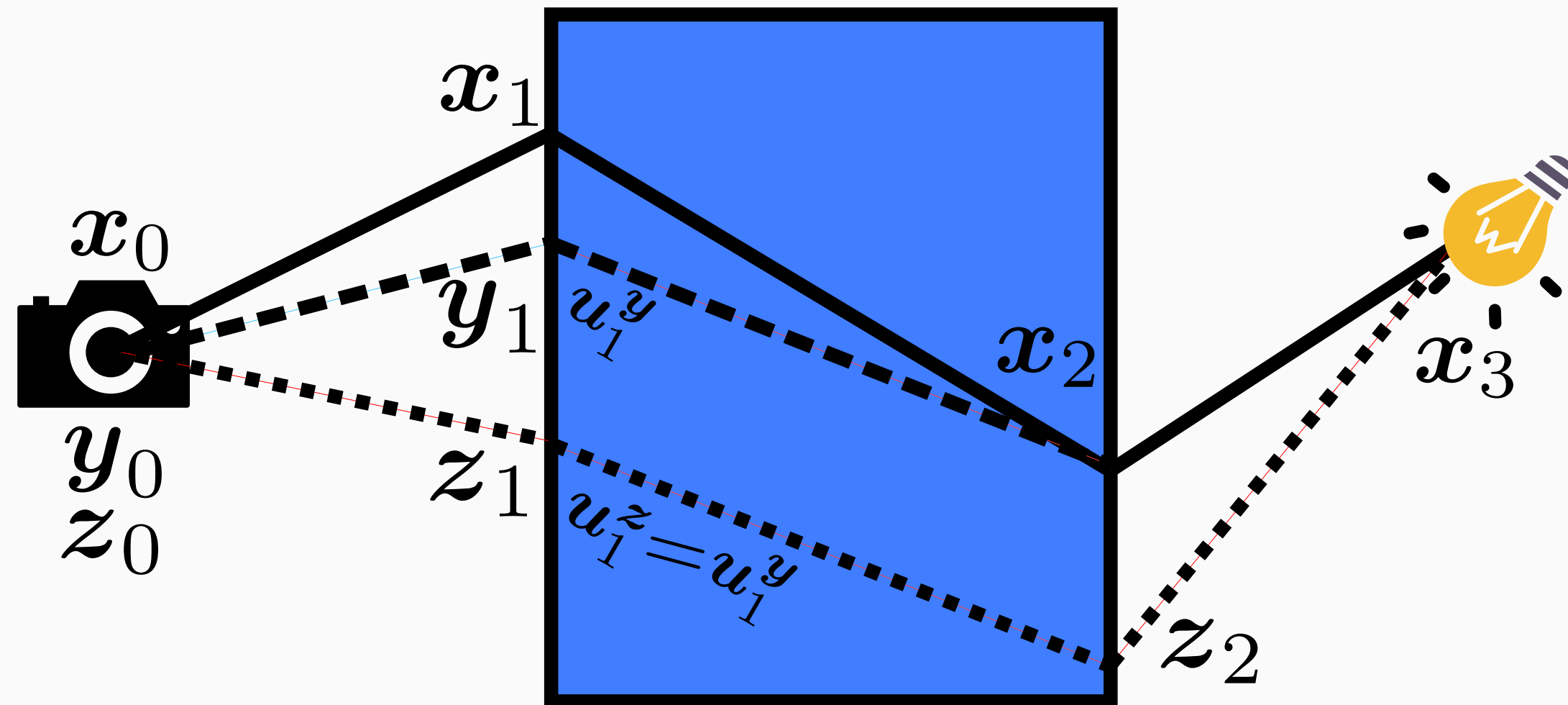
# ReSTIR SSS: Sequential Shift Example

1. spatial reuse (reconnection)
2. spatial reuse (delayed reconnection)



# ReSTIR SSS: Sequential Shift Example

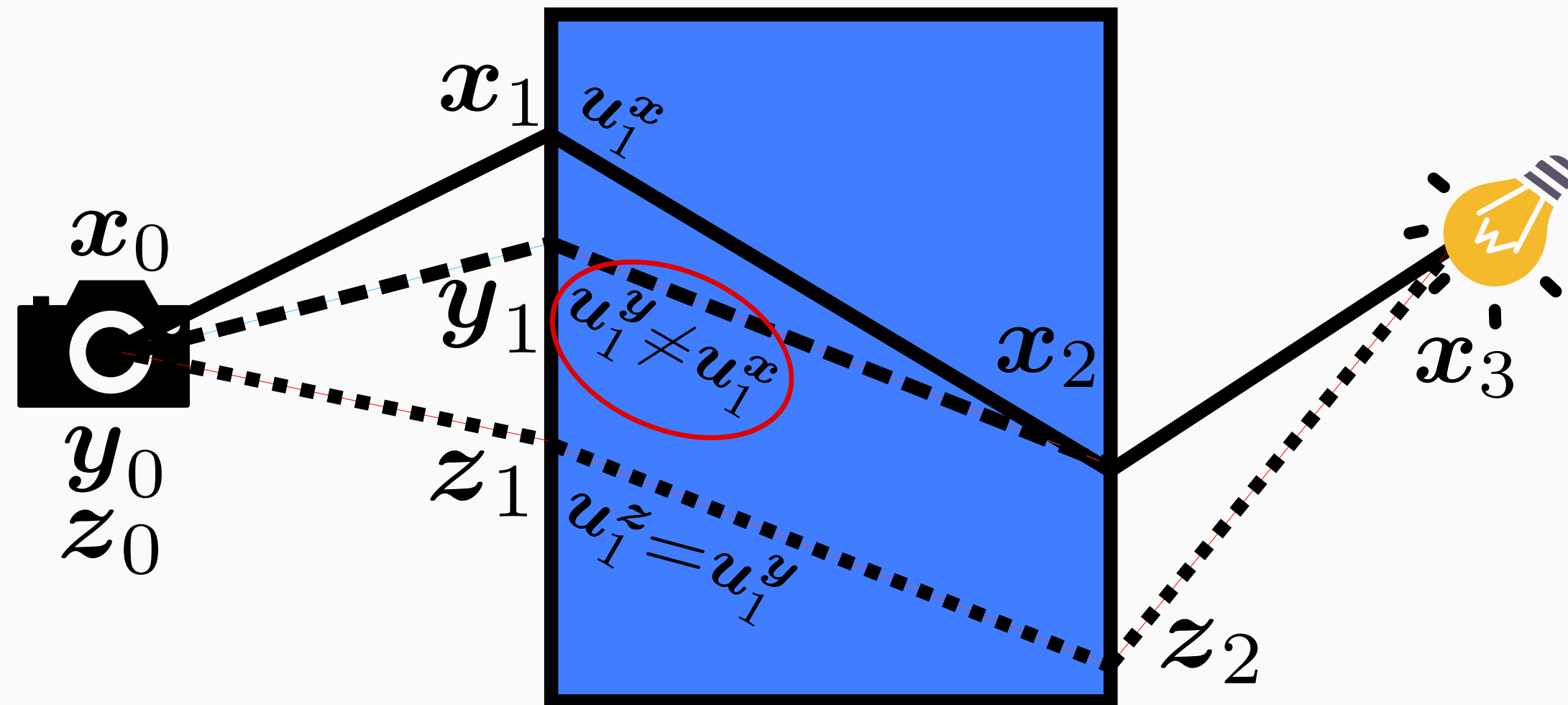
1. spatial reuse (reconnection)
2. spatial reuse (delayed reconnection)





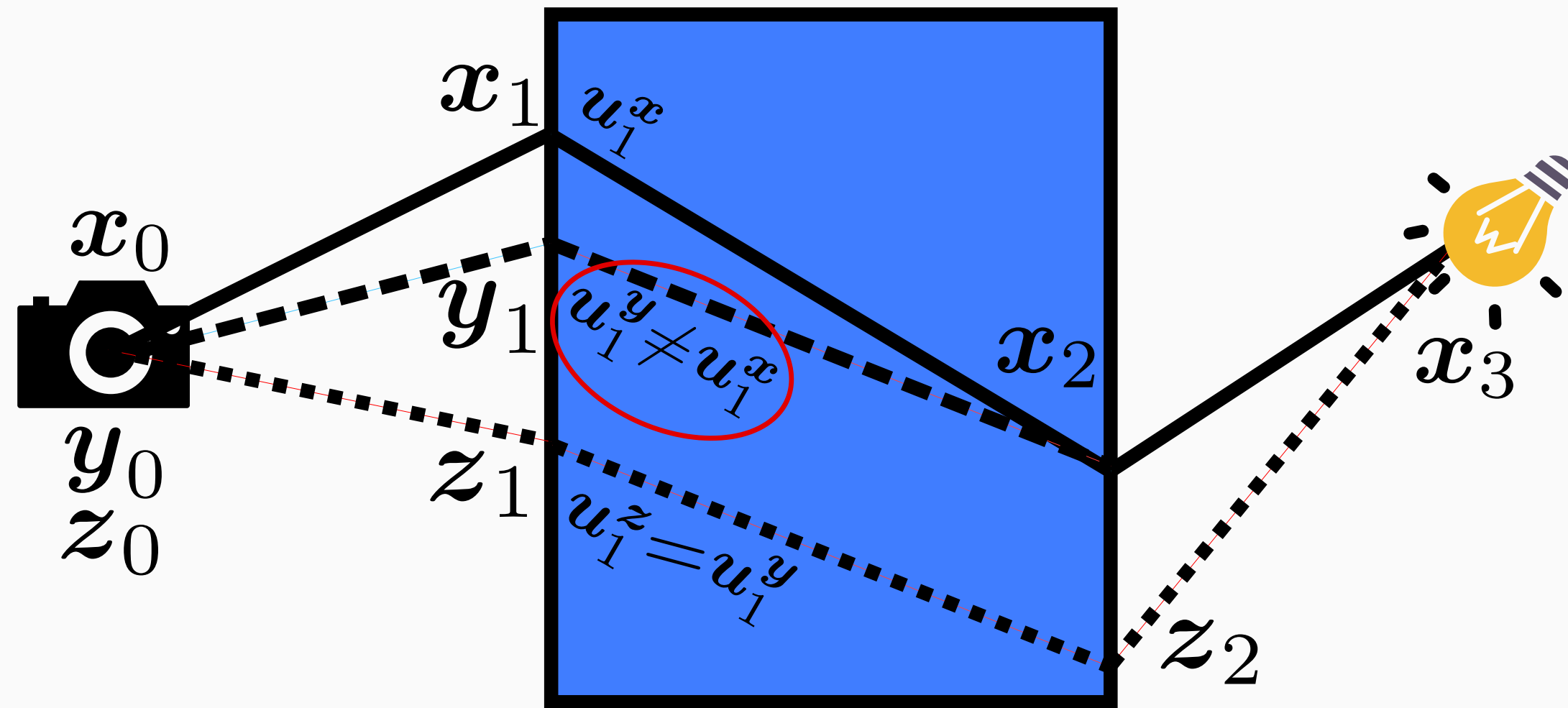
# ReSTIR SSS: Sequential Shift Example

1. spatial reuse (reconnection)
2. spatial reuse (delayed reconnection)



# ReSTIR SSS: Sequential Shift Example

1. spatial reuse (reconnection)
2. spatial reuse (delayed reconnection)

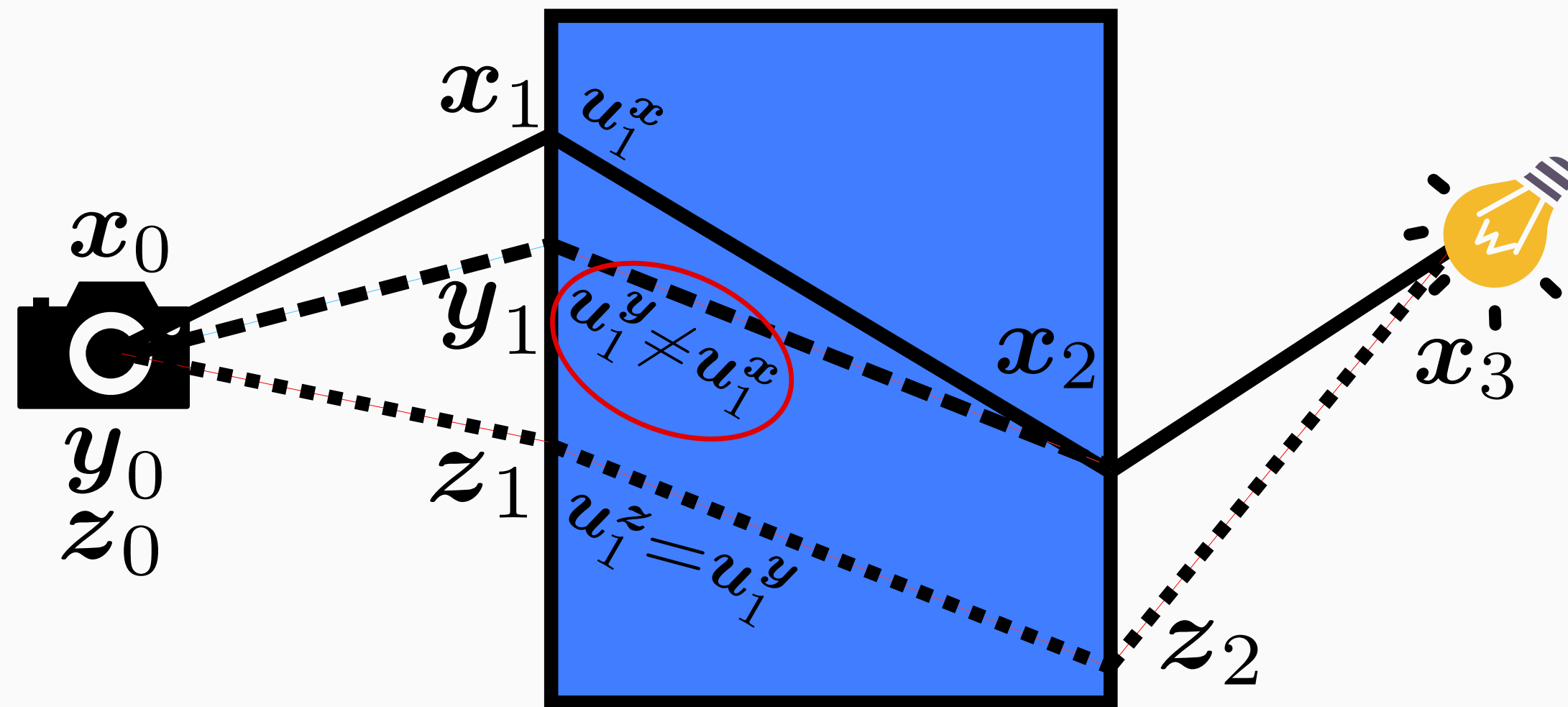


- how do we obtain  $u_1^y$ ?



# ReSTIR SSS: Sequential Shift Example

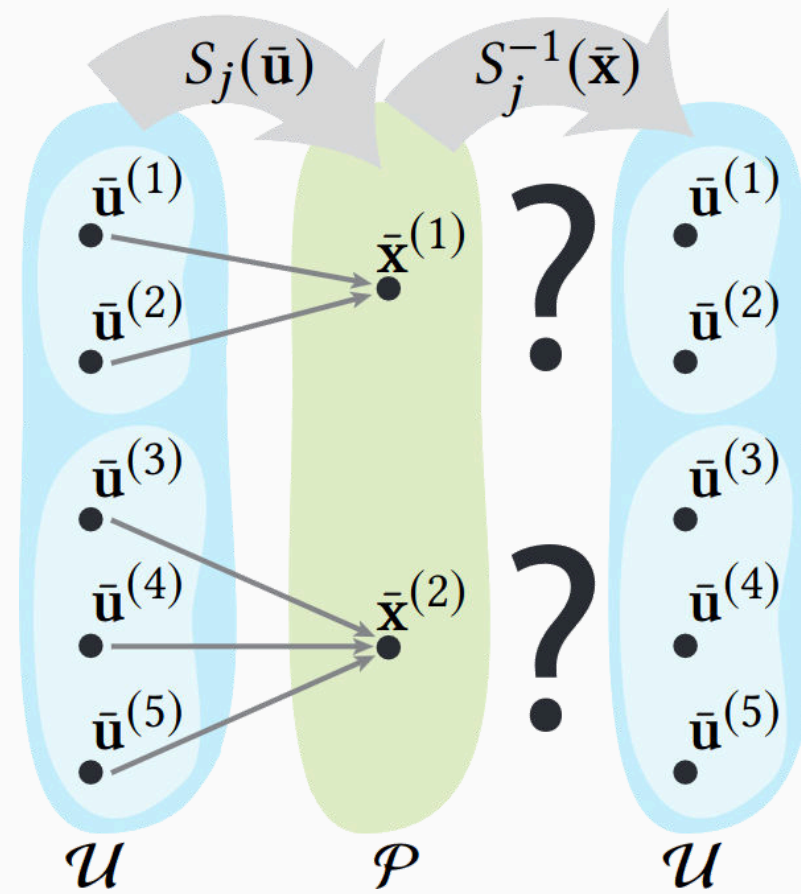
1. spatial reuse (reconnection)
2. spatial reuse (delayed reconnection)



- how do we obtain  $u_1^y$ ?
- can we just invert  $S_{y_1}(u_1^y) = x_2$ , i.e.  $S_{y_1}^{-1}(x_2) = u_1^y$ ?

# ReSTIR SSS: Sequential Shift Invertibility

- usually not invertible

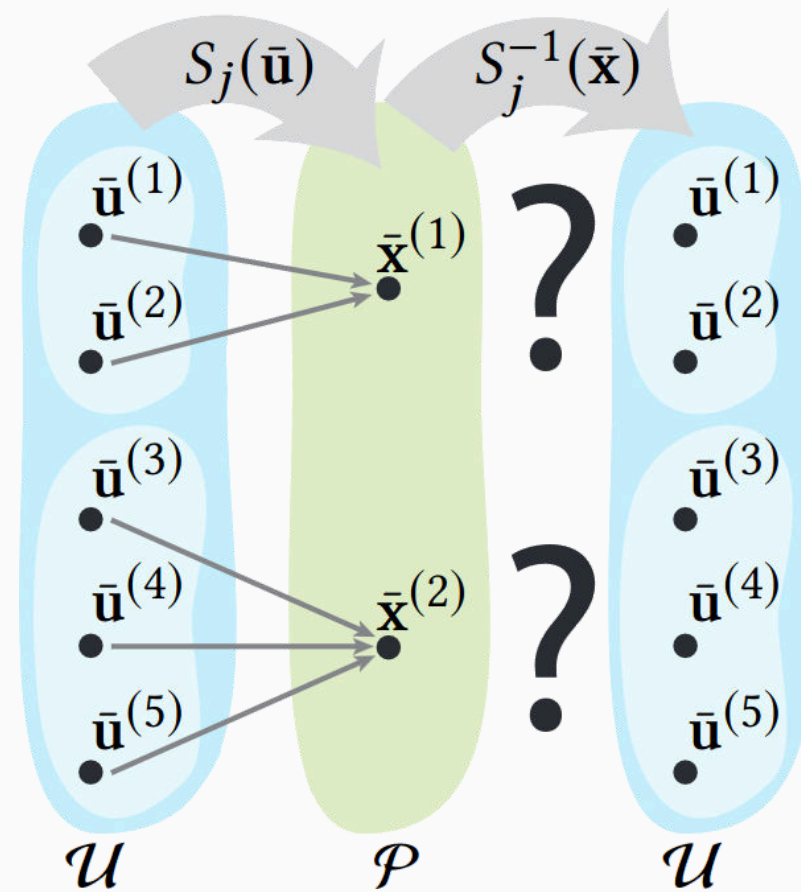


(image from [Bitterli et al. 2017])



# ReSTIR SSS: Sequential Shift Invertibility

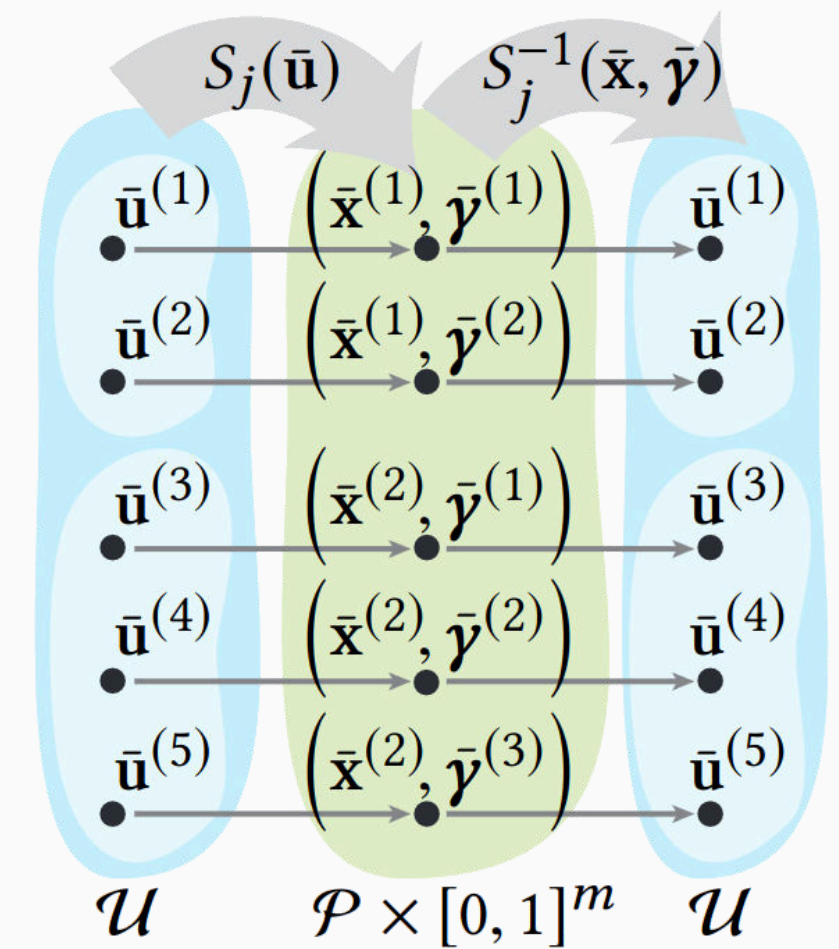
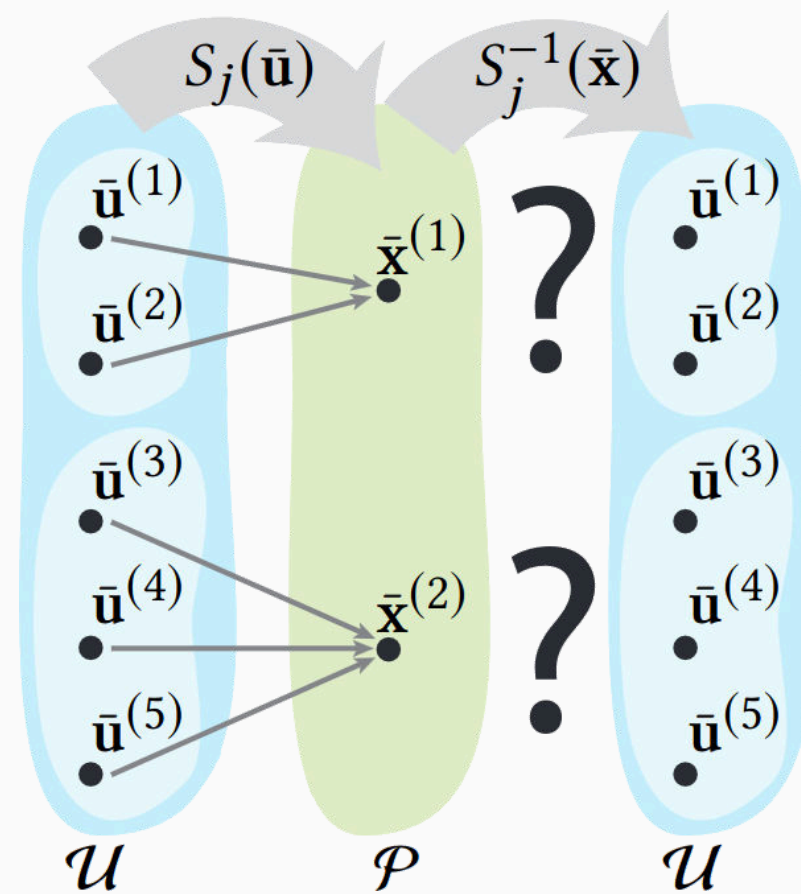
- usually not invertible
  - different "techniques" can generate the same sample
  - techniques: BSDF lobe, projection axis, channel of diffusion profile, etc.



(image from [Bitterli et al. 2017])

# ReSTIR SSS: Sequential Shift Invertibility

- define *extended path space*  $\mathcal{P} \times [0, 1]^m$  [Bitterli et al. 2017]

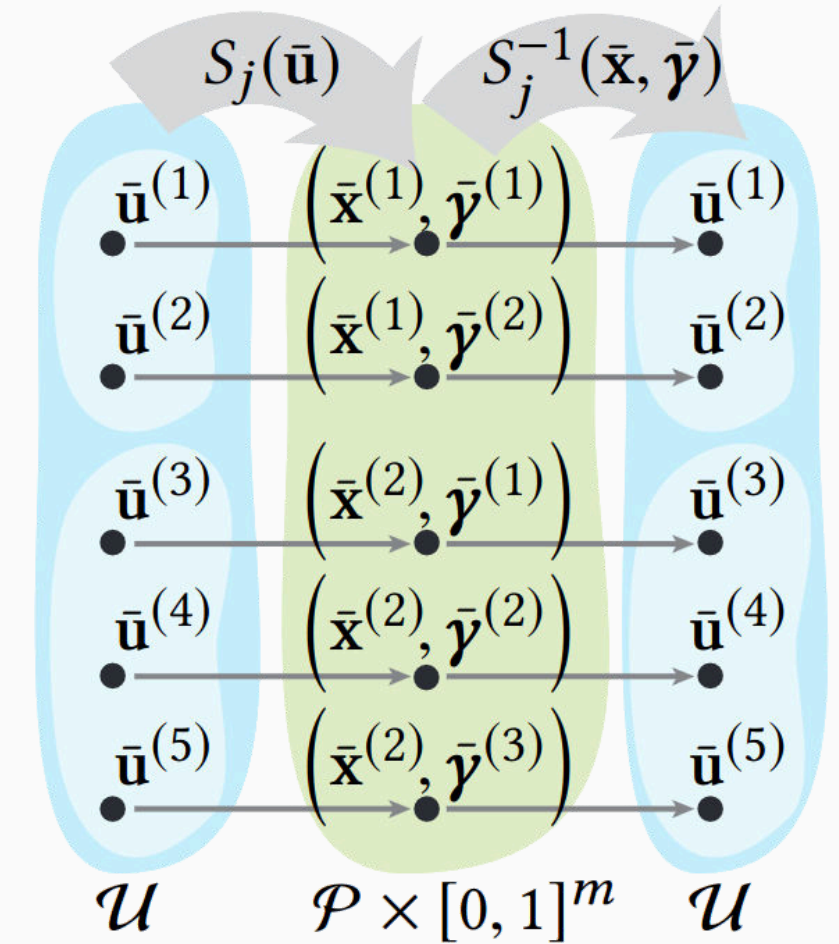
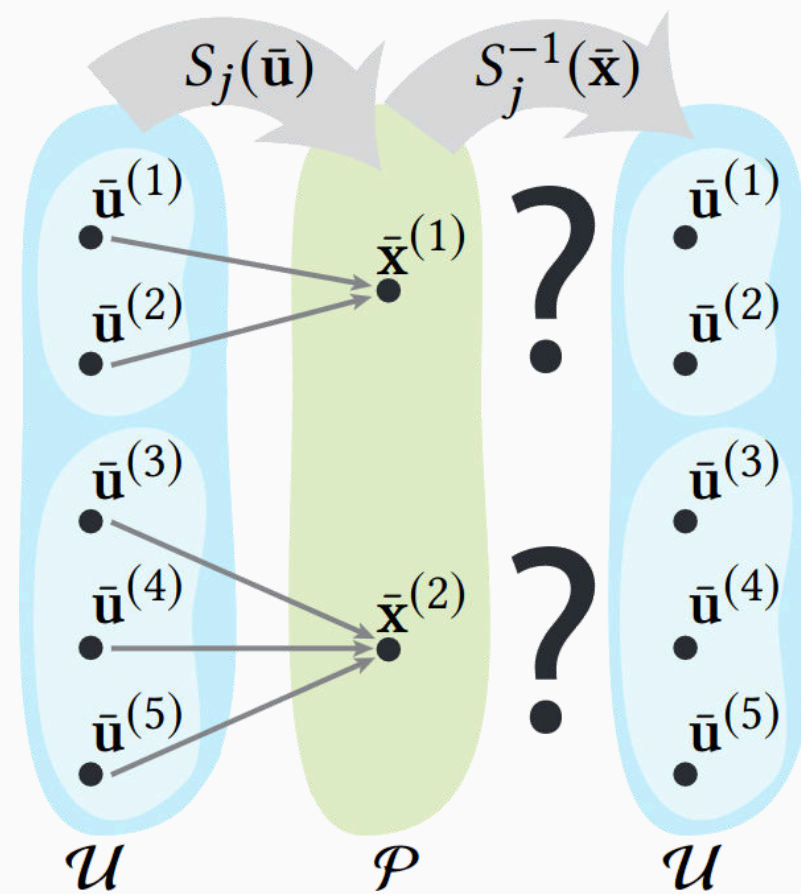


(image from [Bitterli et al. 2017])



# ReSTIR SSS: Sequential Shift Invertibility

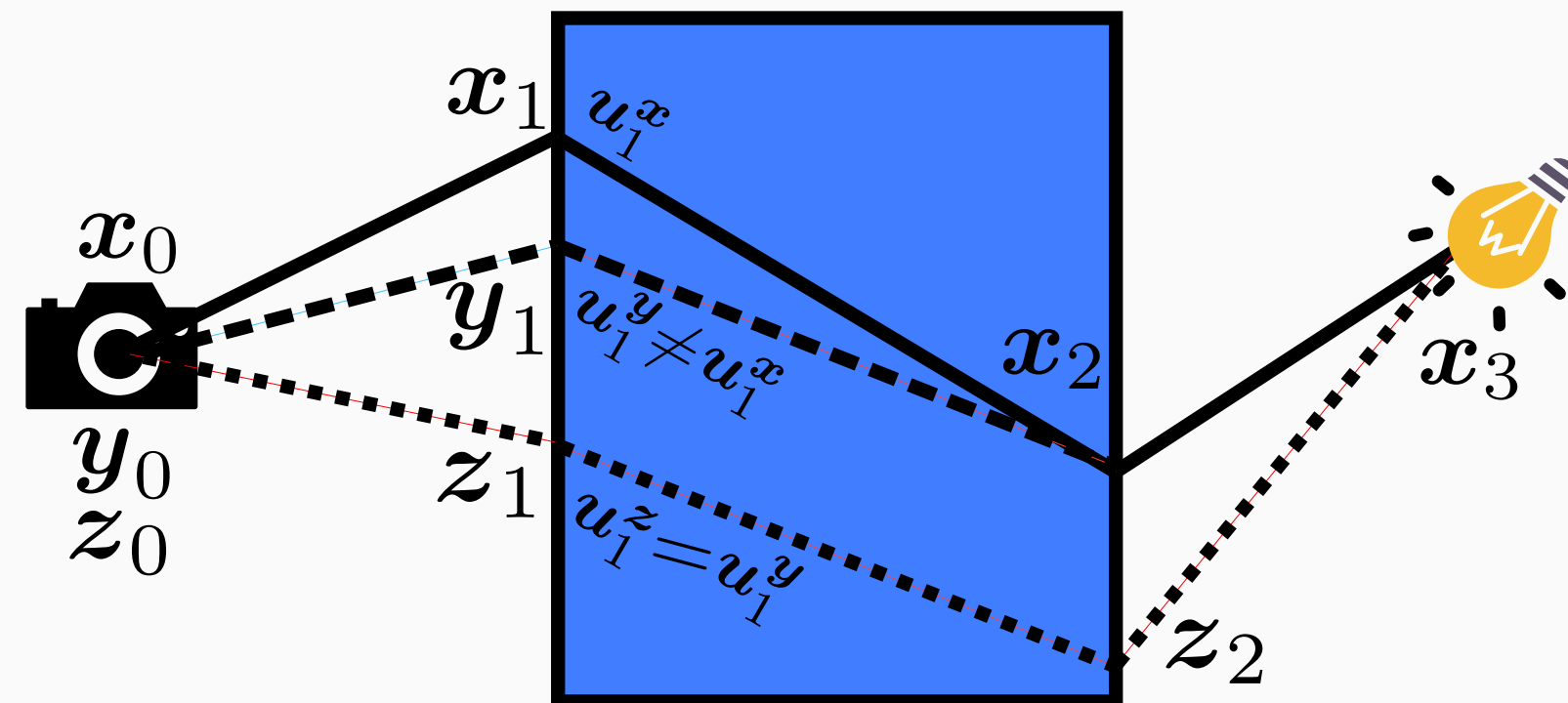
- define *extended path space*  $\mathcal{P} \times [0, 1]^m$  [Bitterli et al. 2017]
- obtain extra dimension (technique) using *probabilistic inversion* [Bitterli et al. 2017]
  - draw independent random numbers
  - sample technique based on the techniques' likelihoods



(image from [Bitterli et al. 2017])

# ReSTIR SSS: Sequential Shift Invertibility

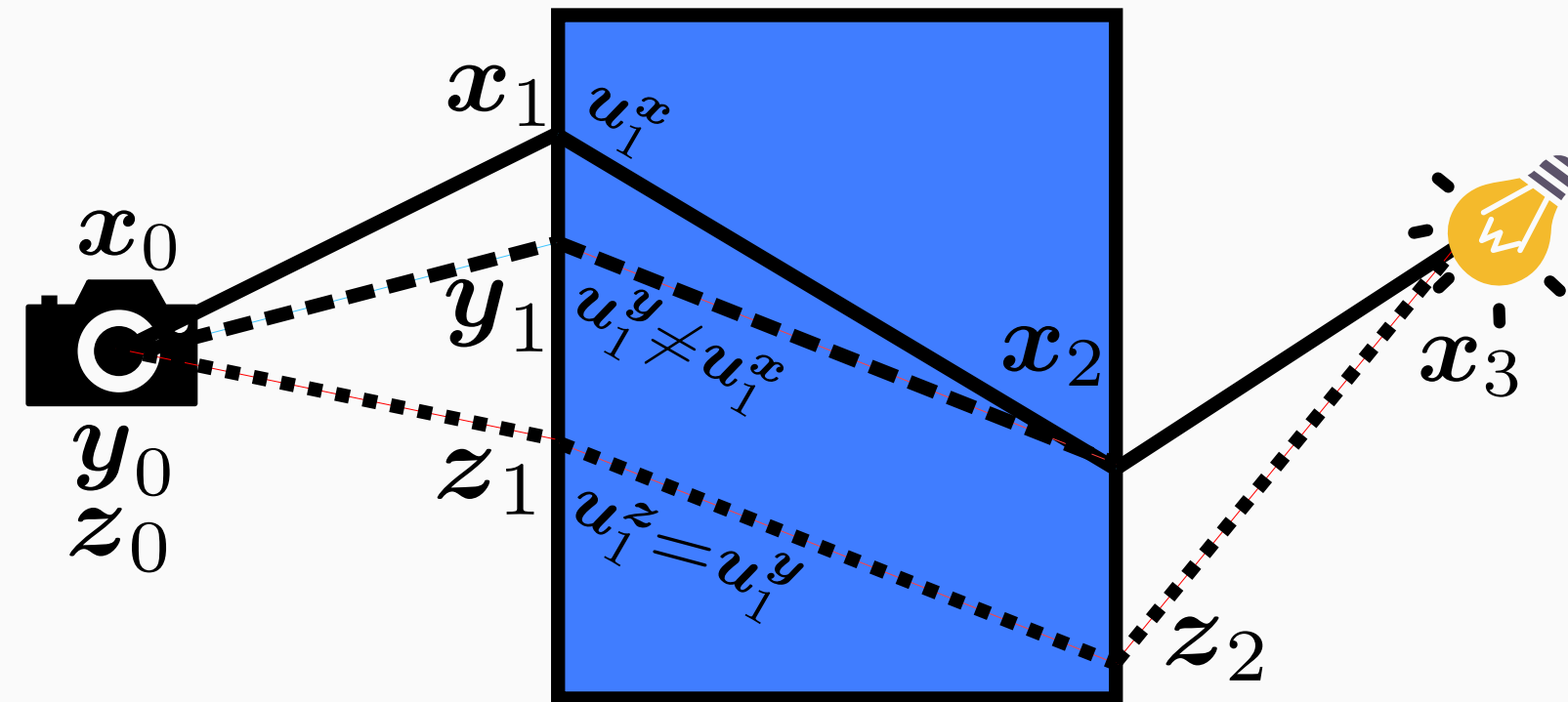
- define *extended path space*  $\mathcal{P} \times [0, 1]^m$  [Bitterli et al. 2017]
- obtain extra dimension (technique) using *probabilistic inversion* [Bitterli et al. 2017]
  - draw independent random numbers
  - sample technique based on the techniques' likelihoods
- applicable to ReSTIR (SSS)





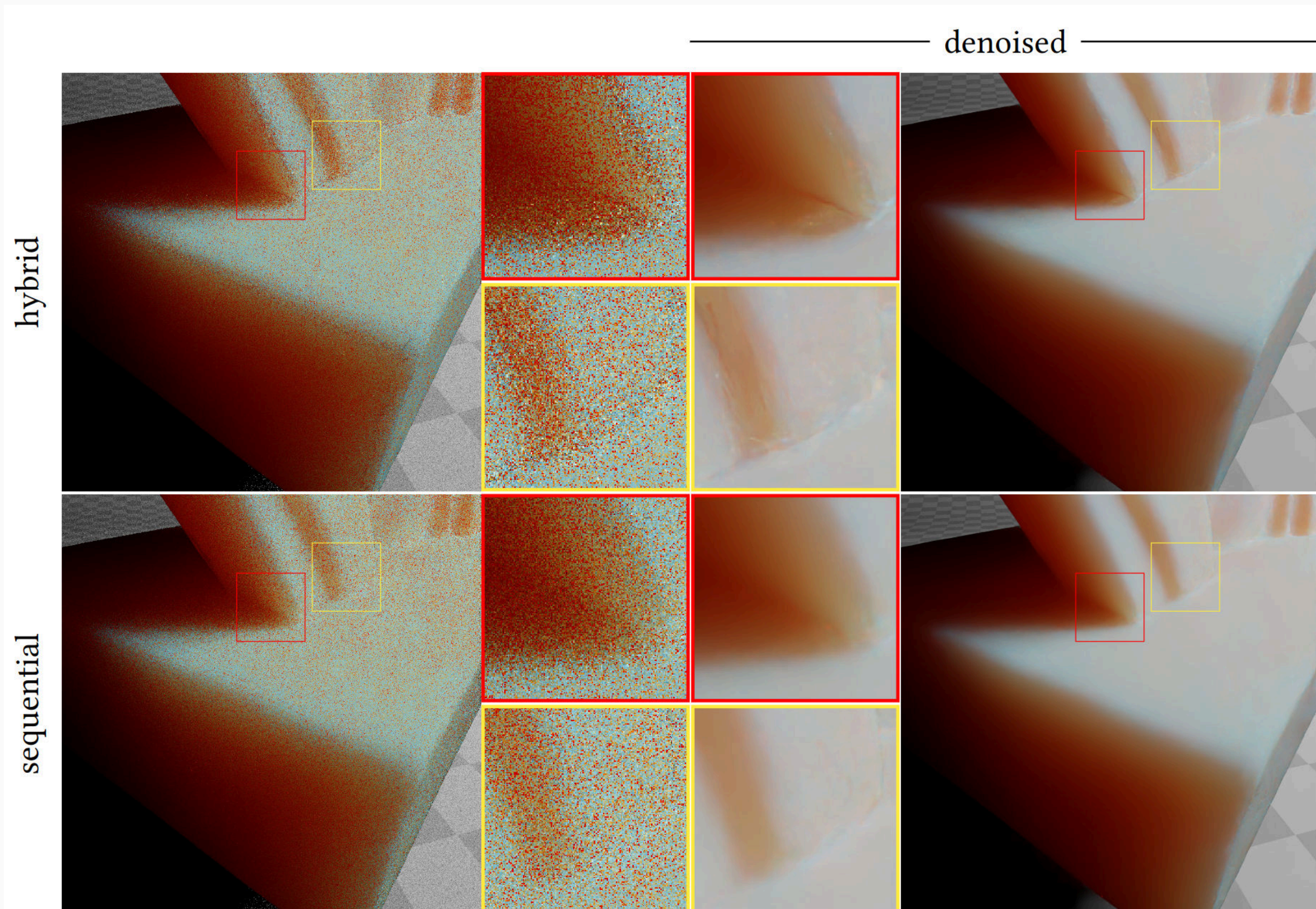
# ReSTIR SSS: Sequential Shift Invertibility

- define *extended path space*  $\mathcal{P} \times [0, 1]^m$  [Bitterli et al. 2017]
- obtain extra dimension (technique) using *probabilistic inversion* [Bitterli et al. 2017]
  - draw independent random numbers
  - sample technique based on the techniques' likelihoods
- applicable to ReSTIR (SSS)
  - solves our problem with the sequential shift (obtaining  $u_1^y$ )



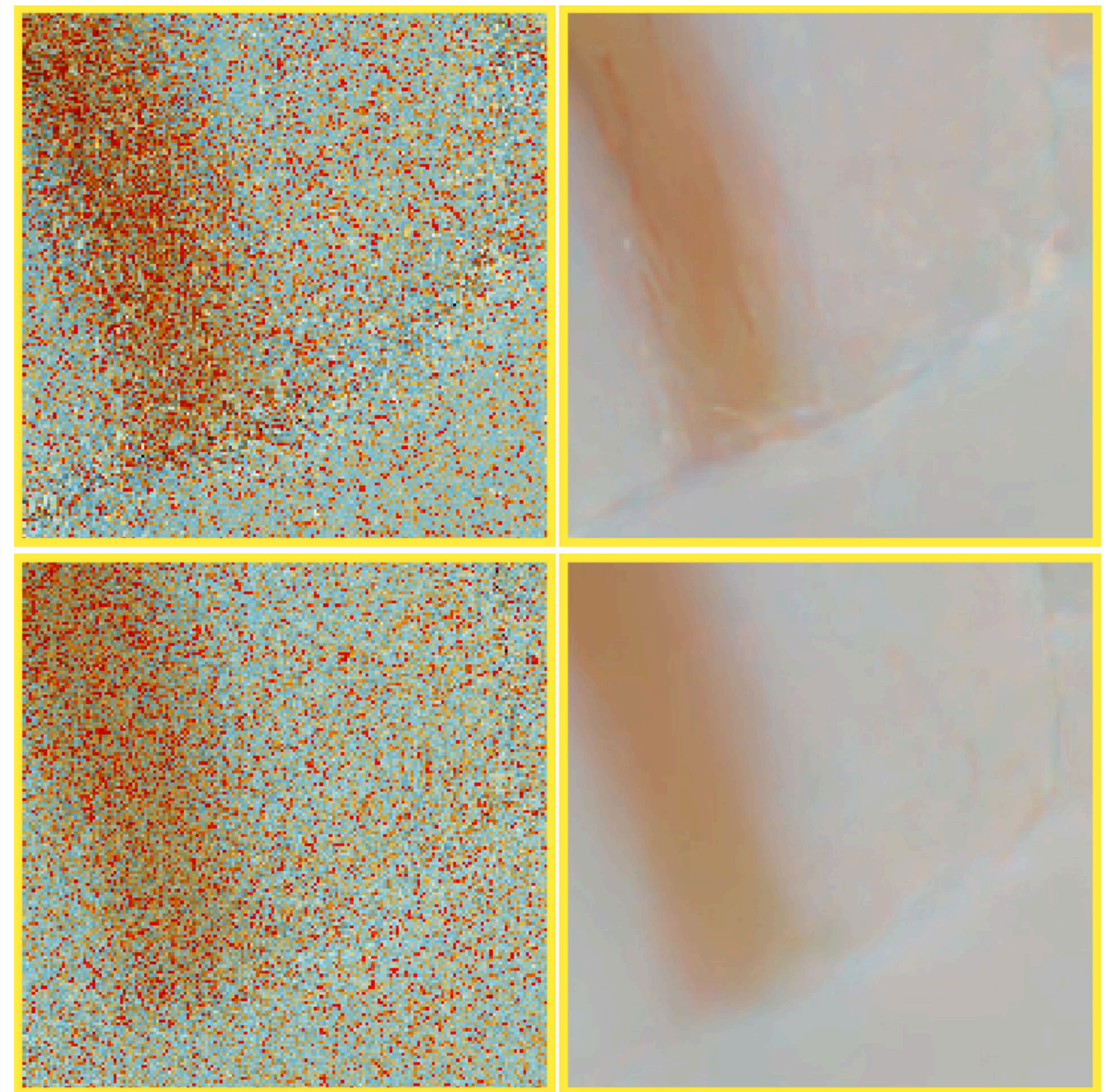
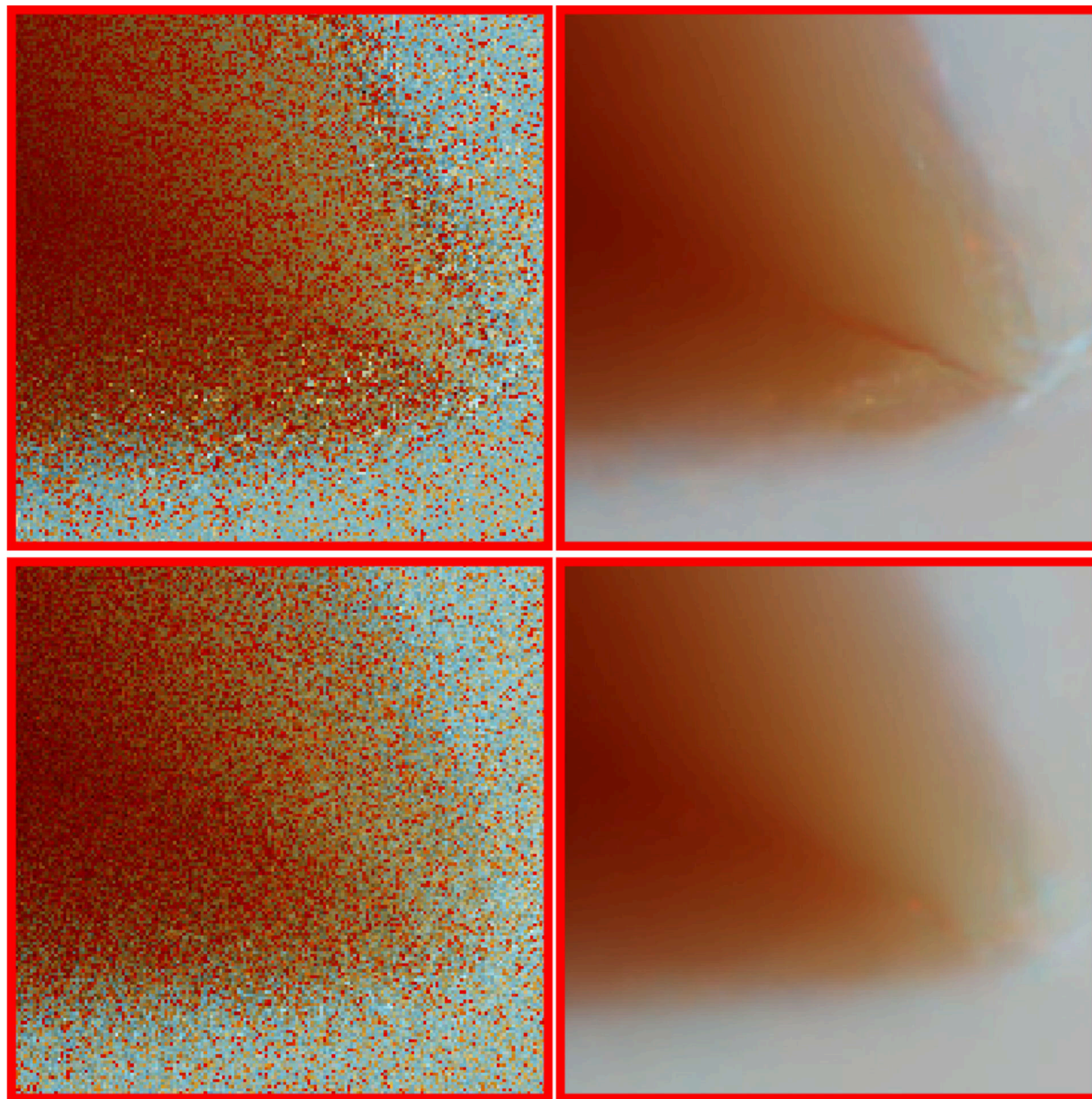


# ReSTIR SSS: Hybrid vs. Sequential Shift



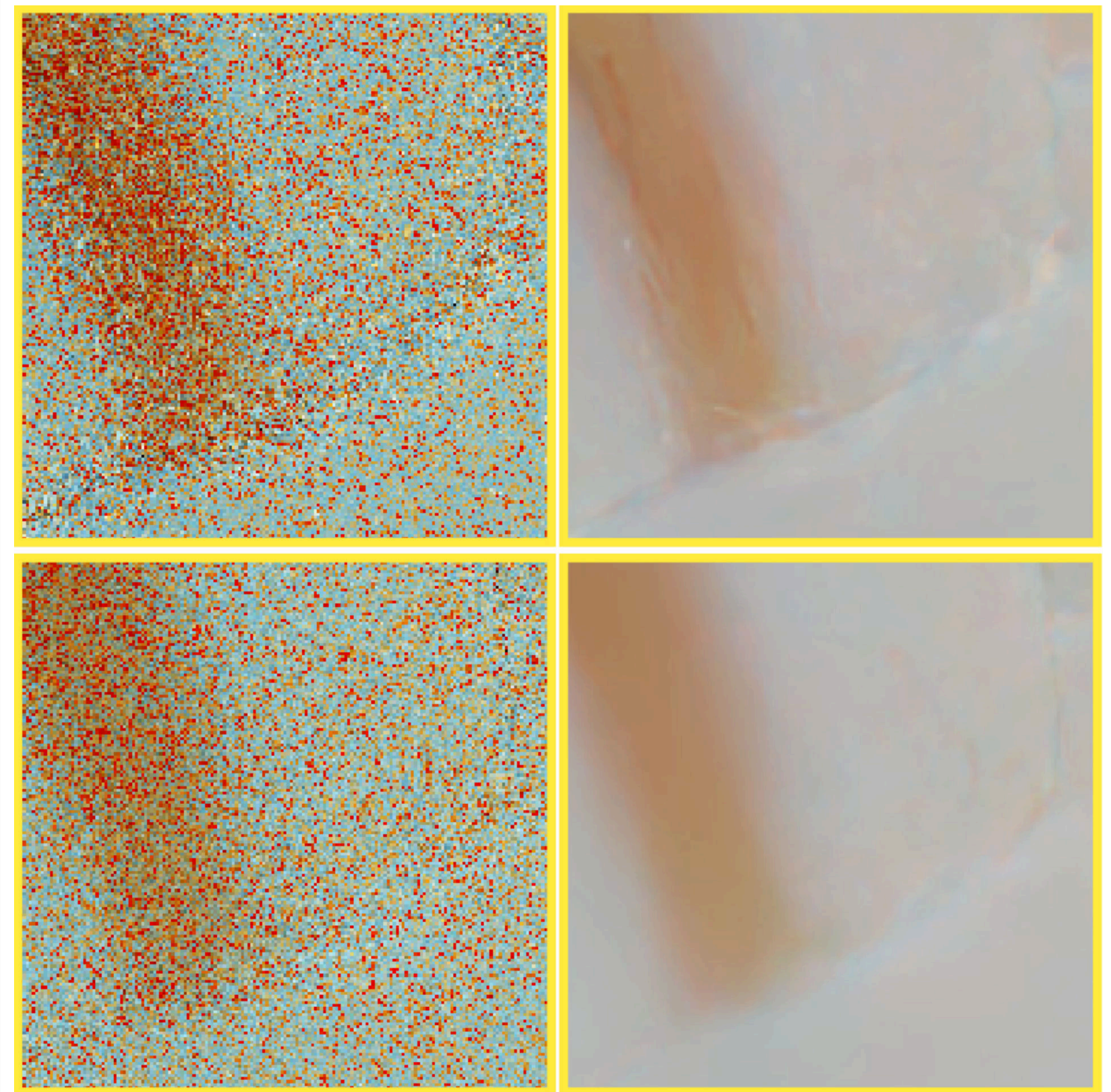
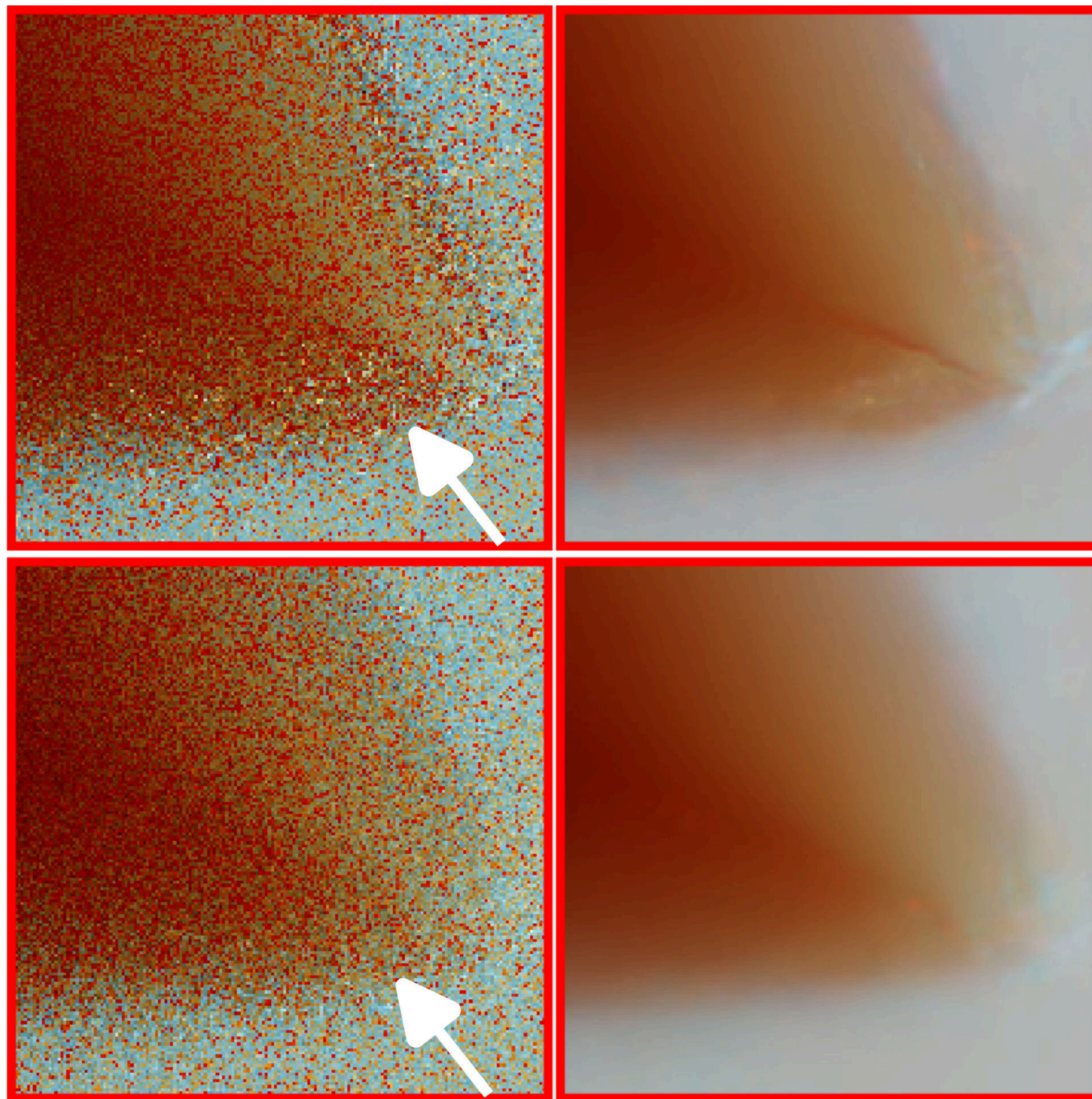


# ReSTIR SSS: Hybrid vs. Sequential Shift



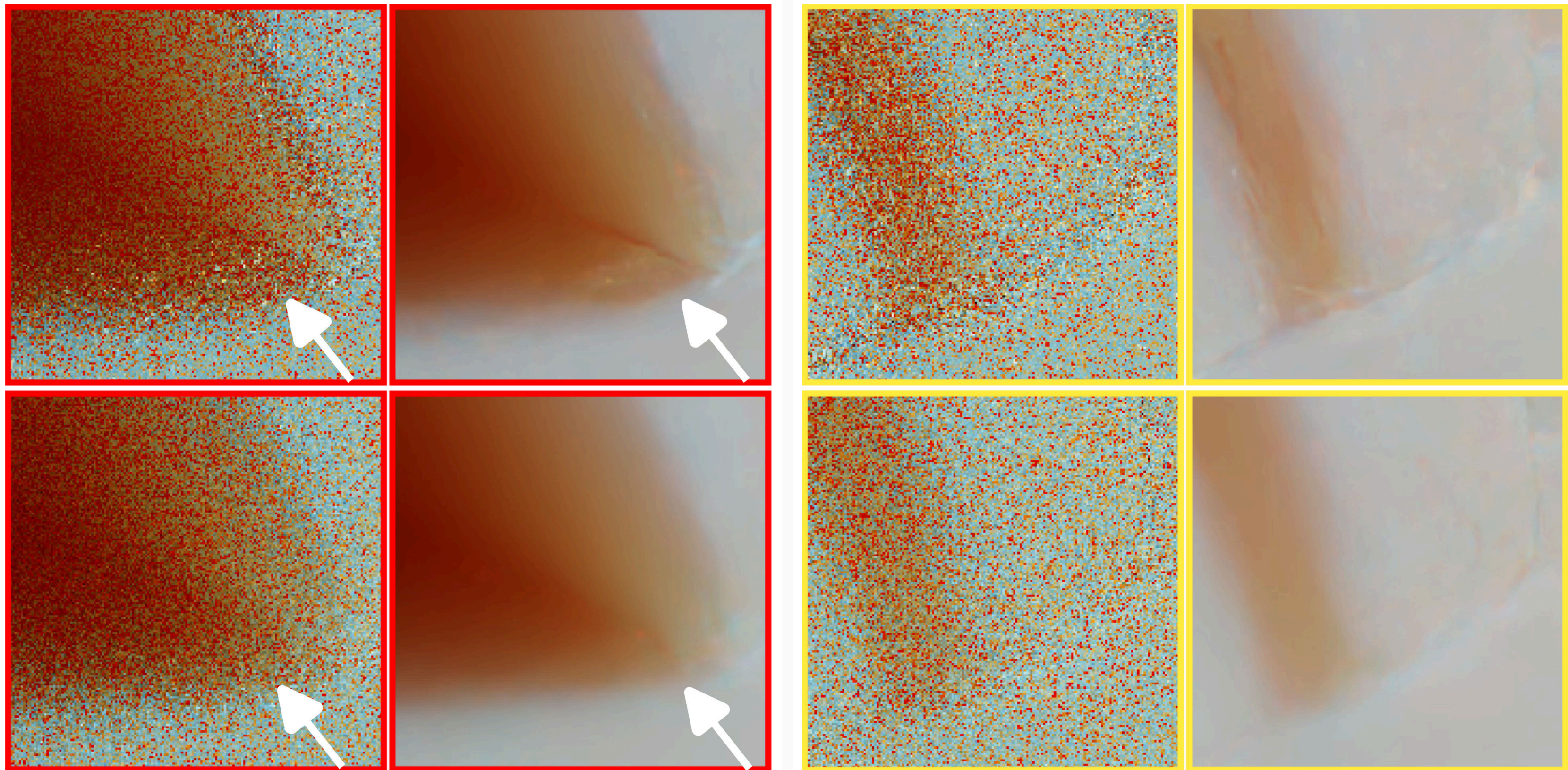


# ReSTIR SSS: Hybrid vs. Sequential Shift





# ReSTIR SSS: Hybrid vs. Sequential Shift





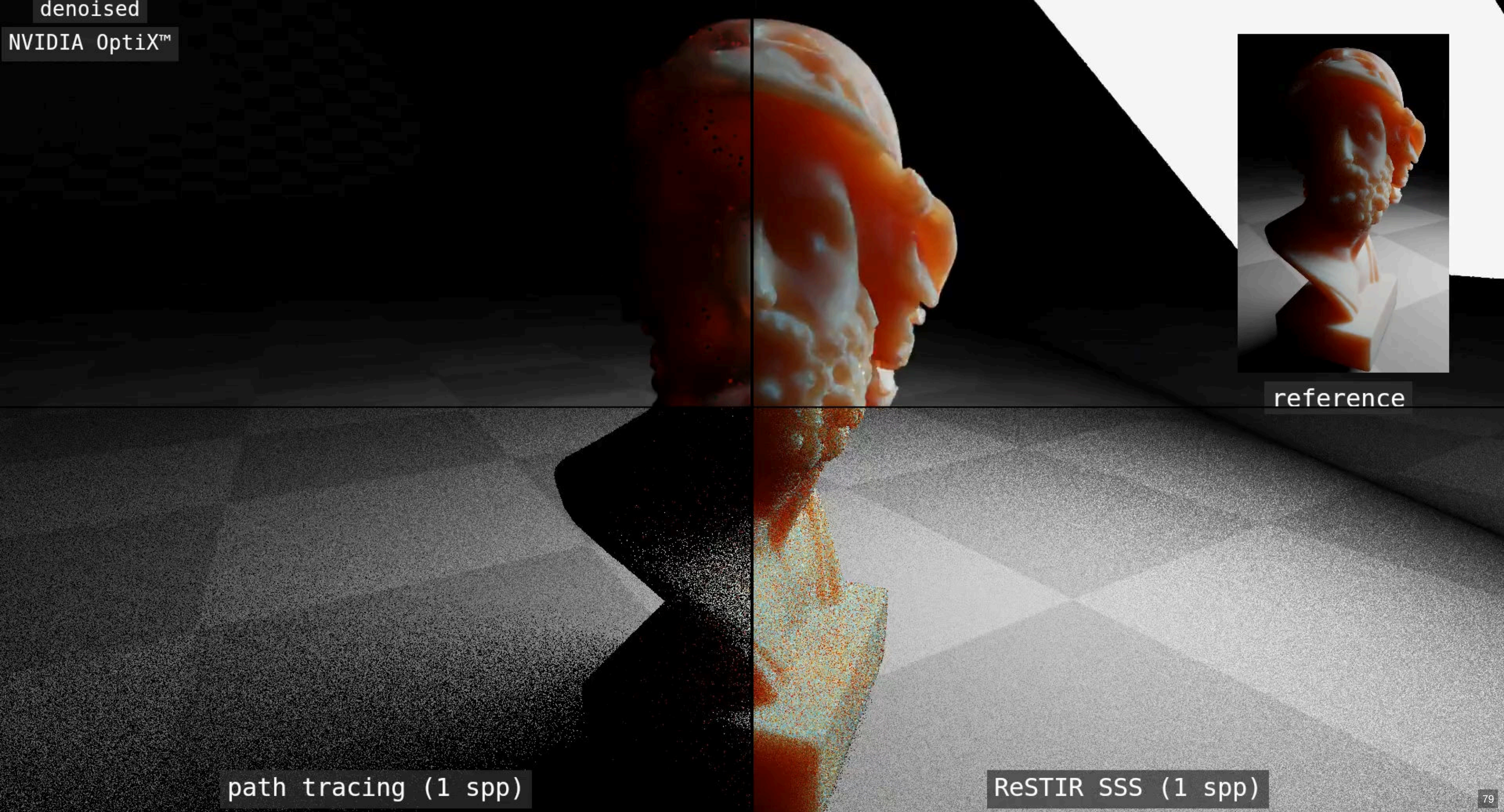
# Results







denoised  
NVIDIA OptiX™



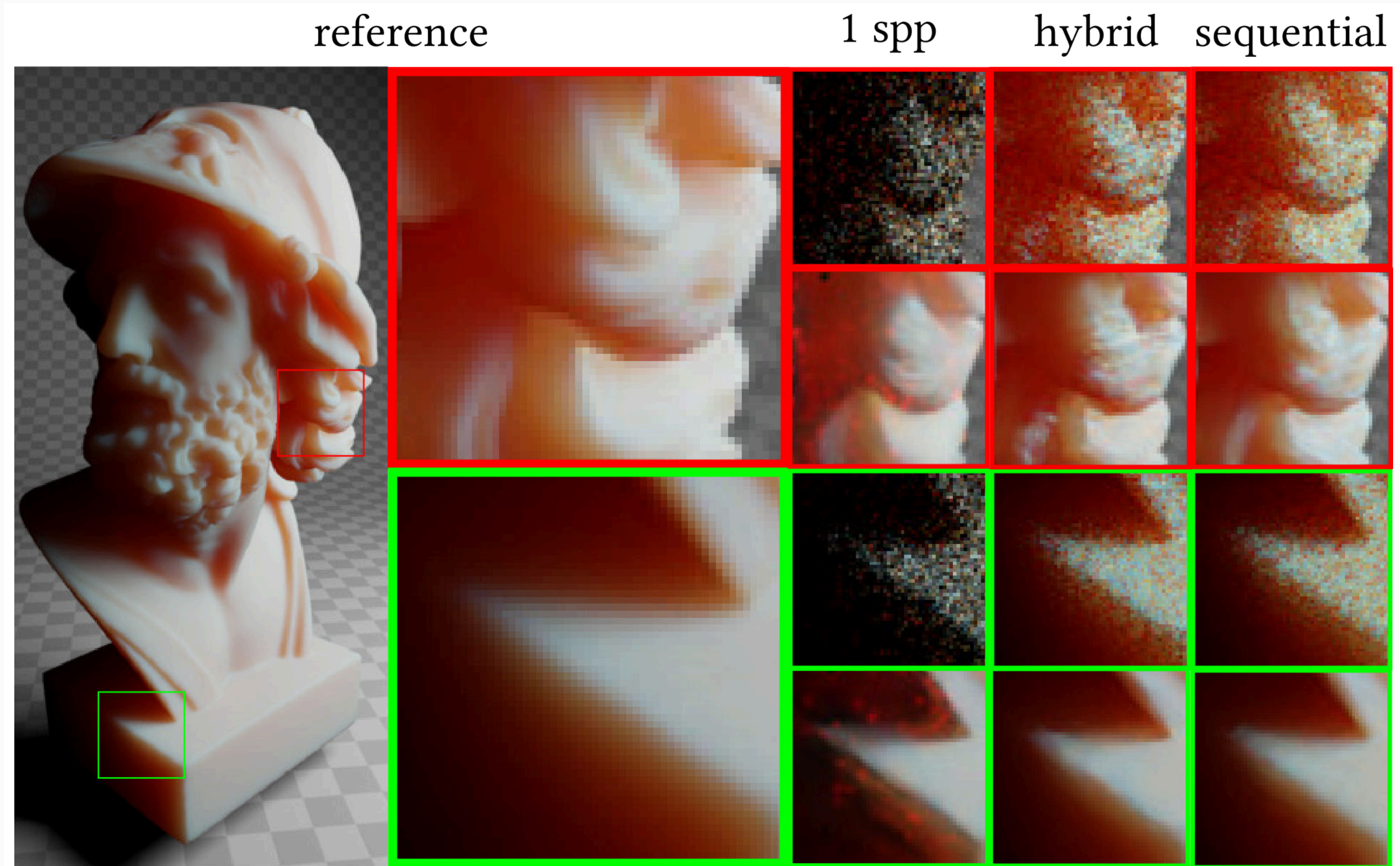
reference

path tracing (1 spp)

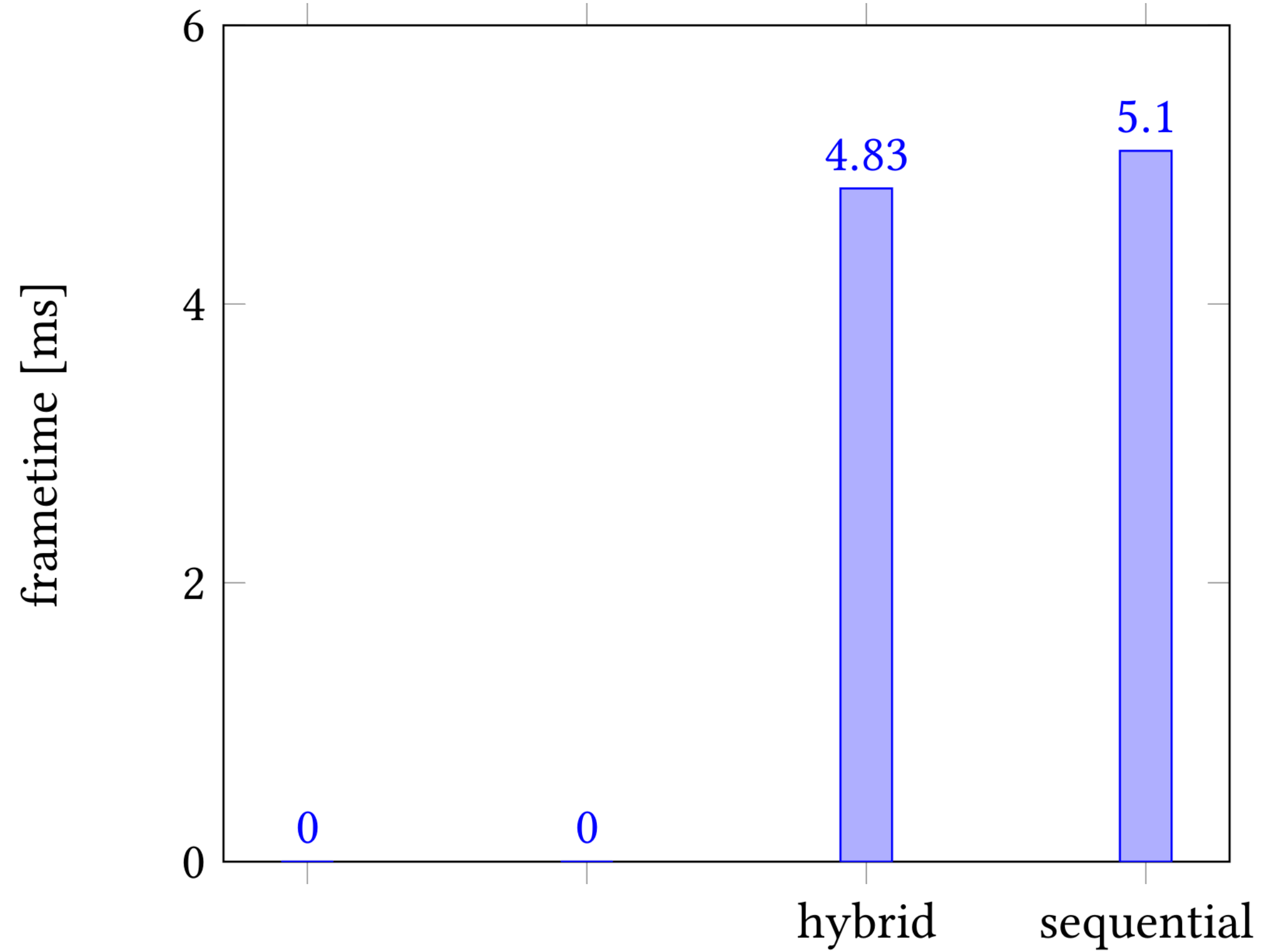
ReSTIR SSS (1 spp)



# ReSTIR SSS: Results (NVIDIA RTX 3070)

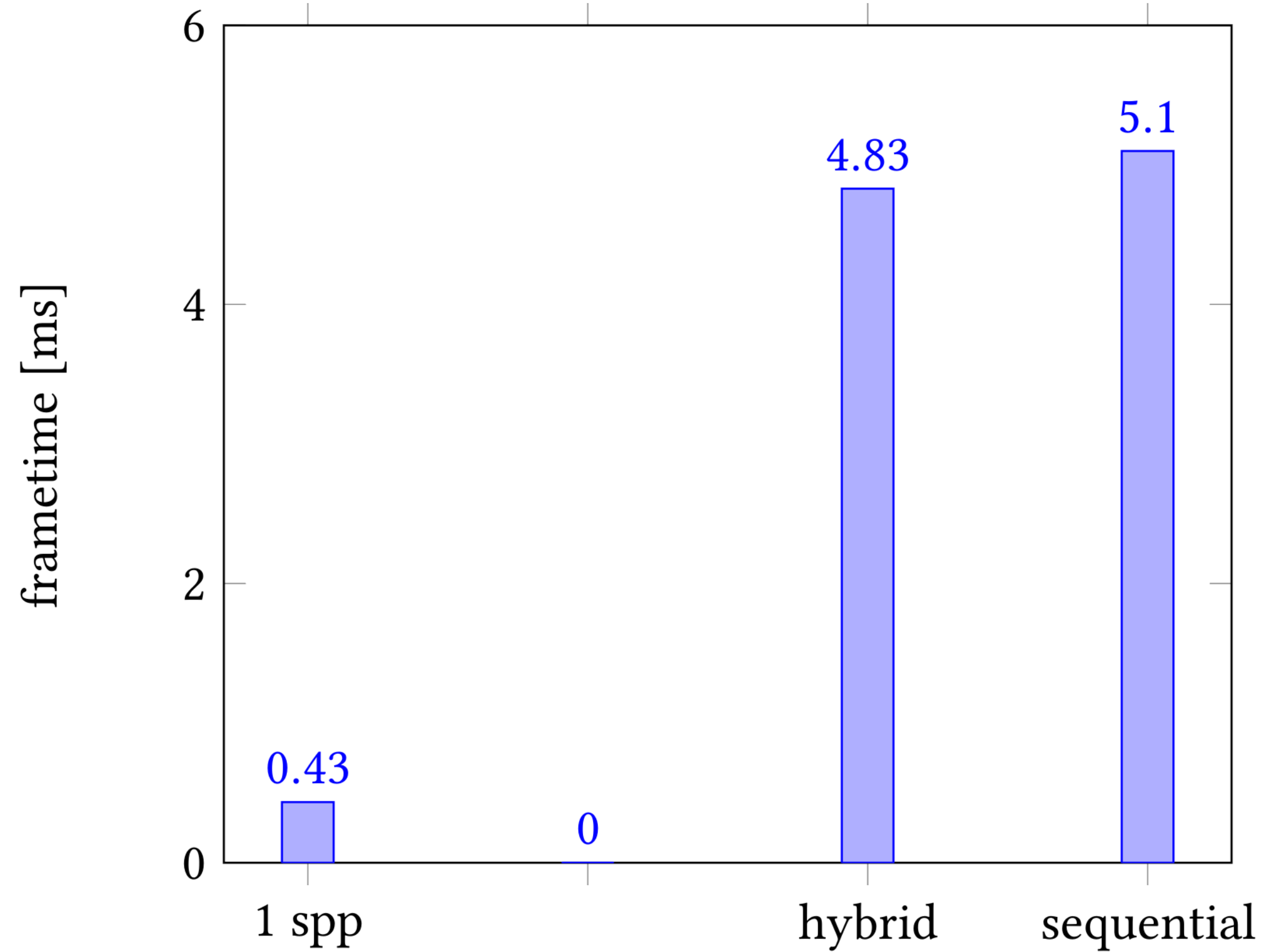


# ReSTIR SSS: Results (NVIDIA RTX 3070)

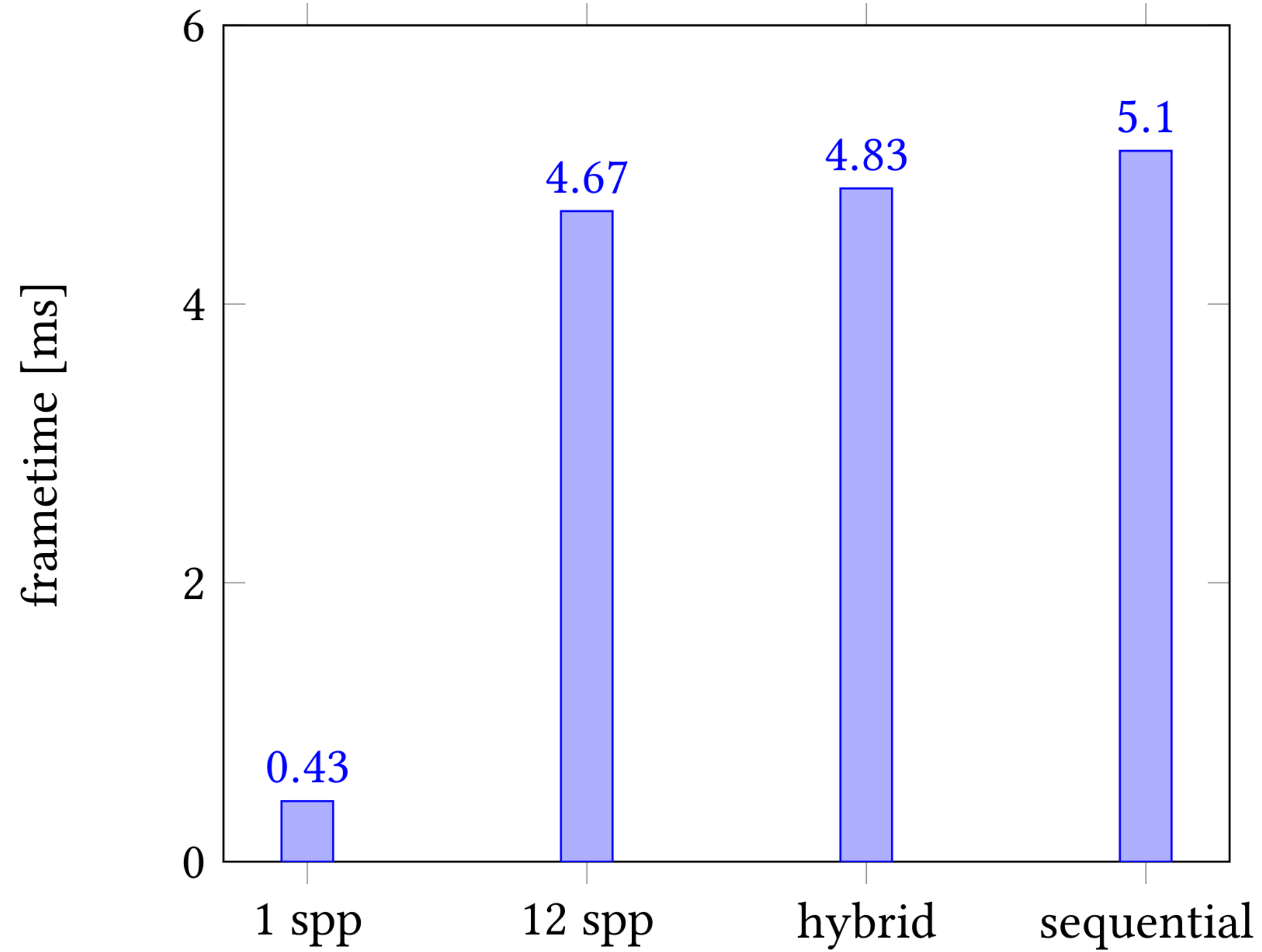




# ReSTIR SSS: Results (NVIDIA RTX 3070)

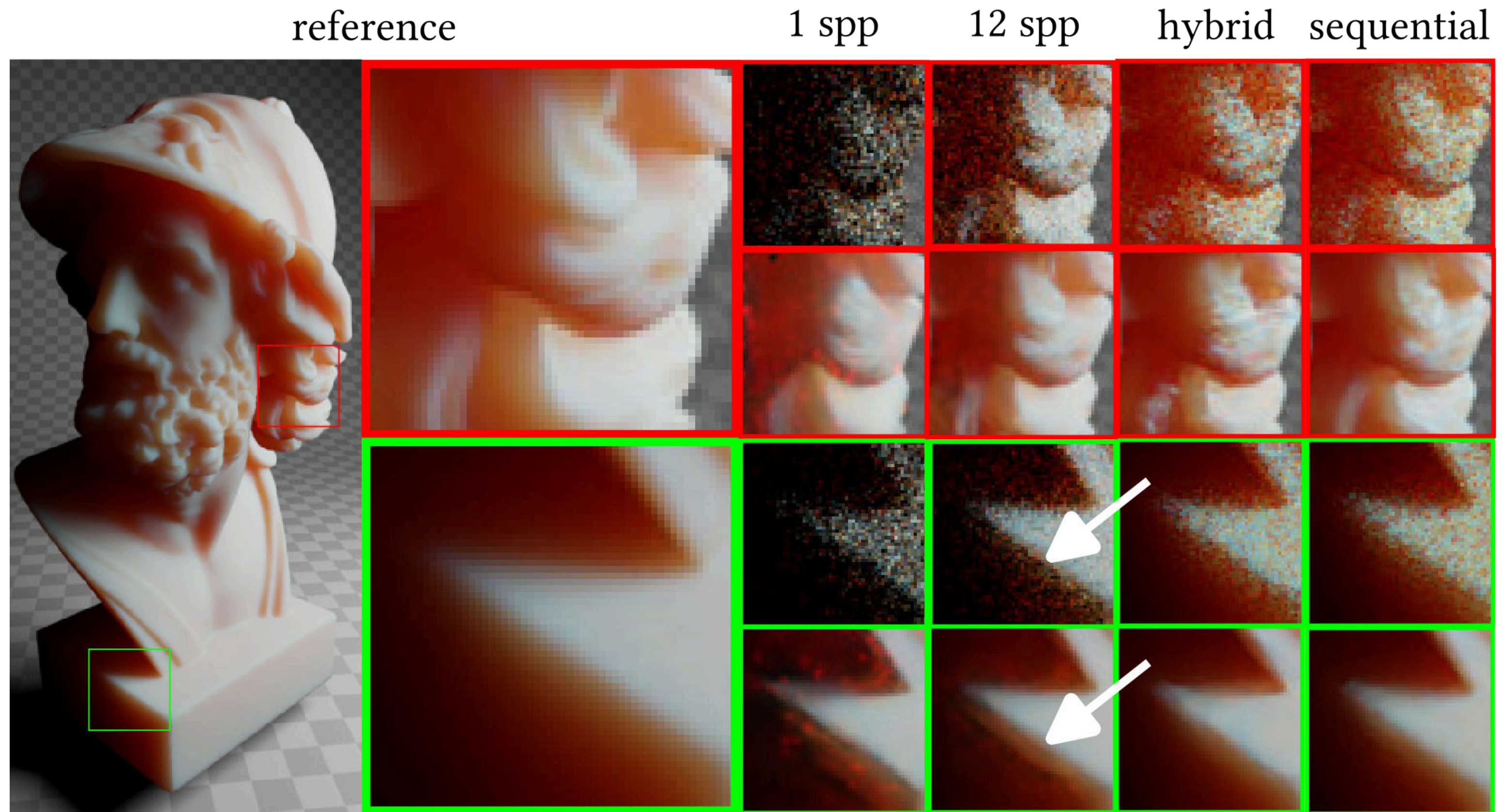


# ReSTIR SSS: Results (NVIDIA RTX 3070)

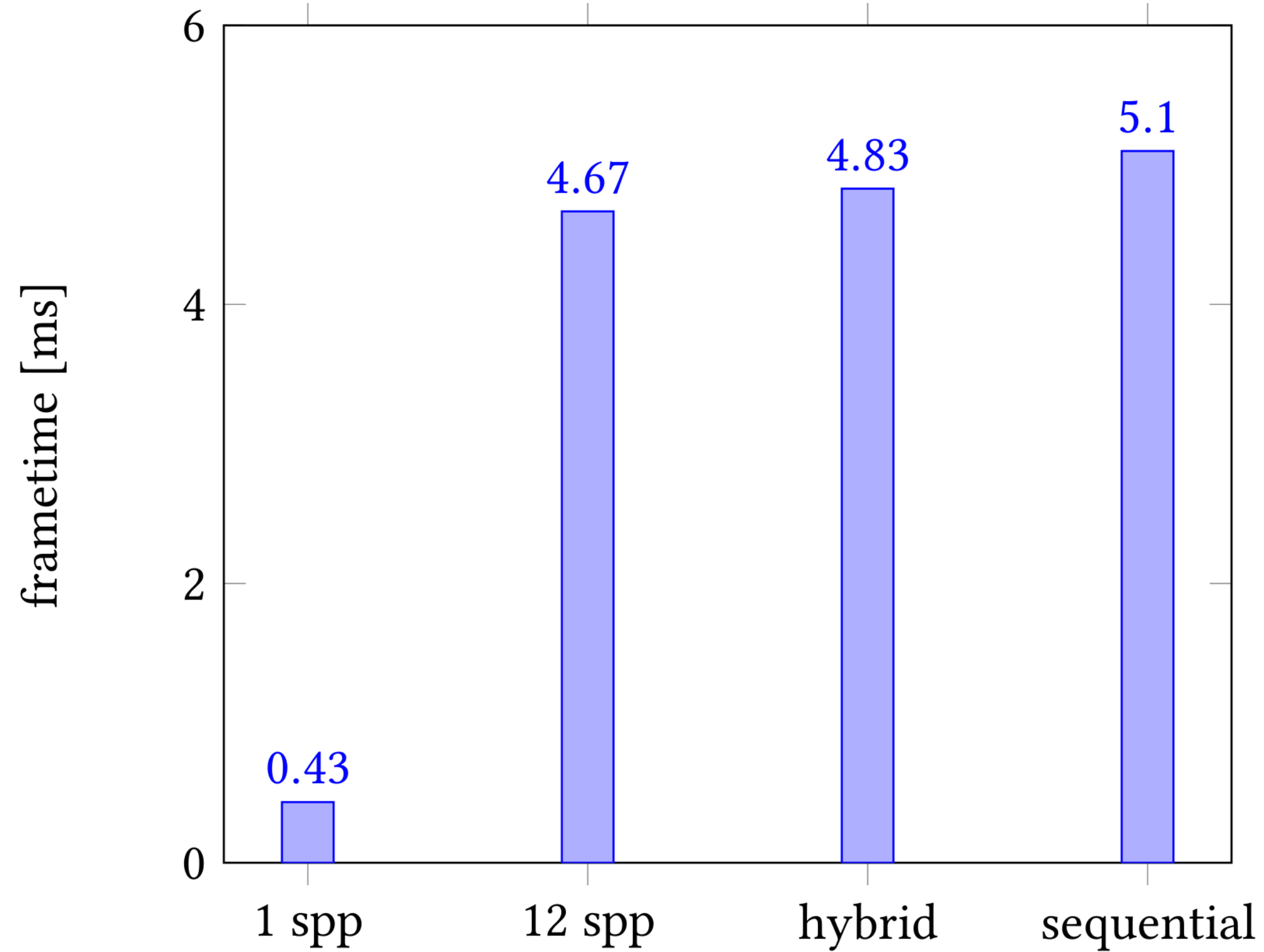




# ReSTIR SSS: Results (NVIDIA RTX 3070)

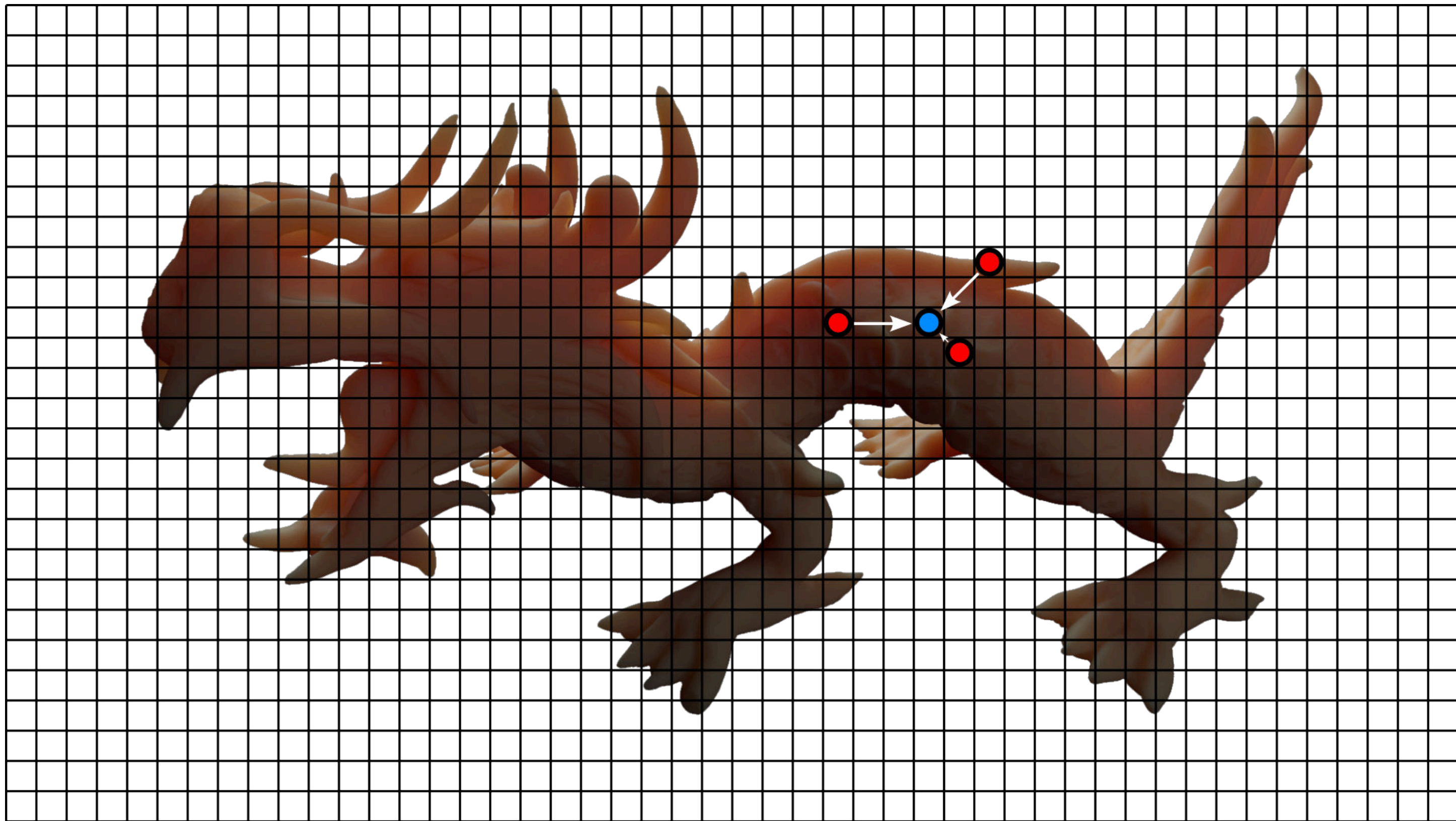


# ReSTIR SSS: Results (NVIDIA RTX 3070)

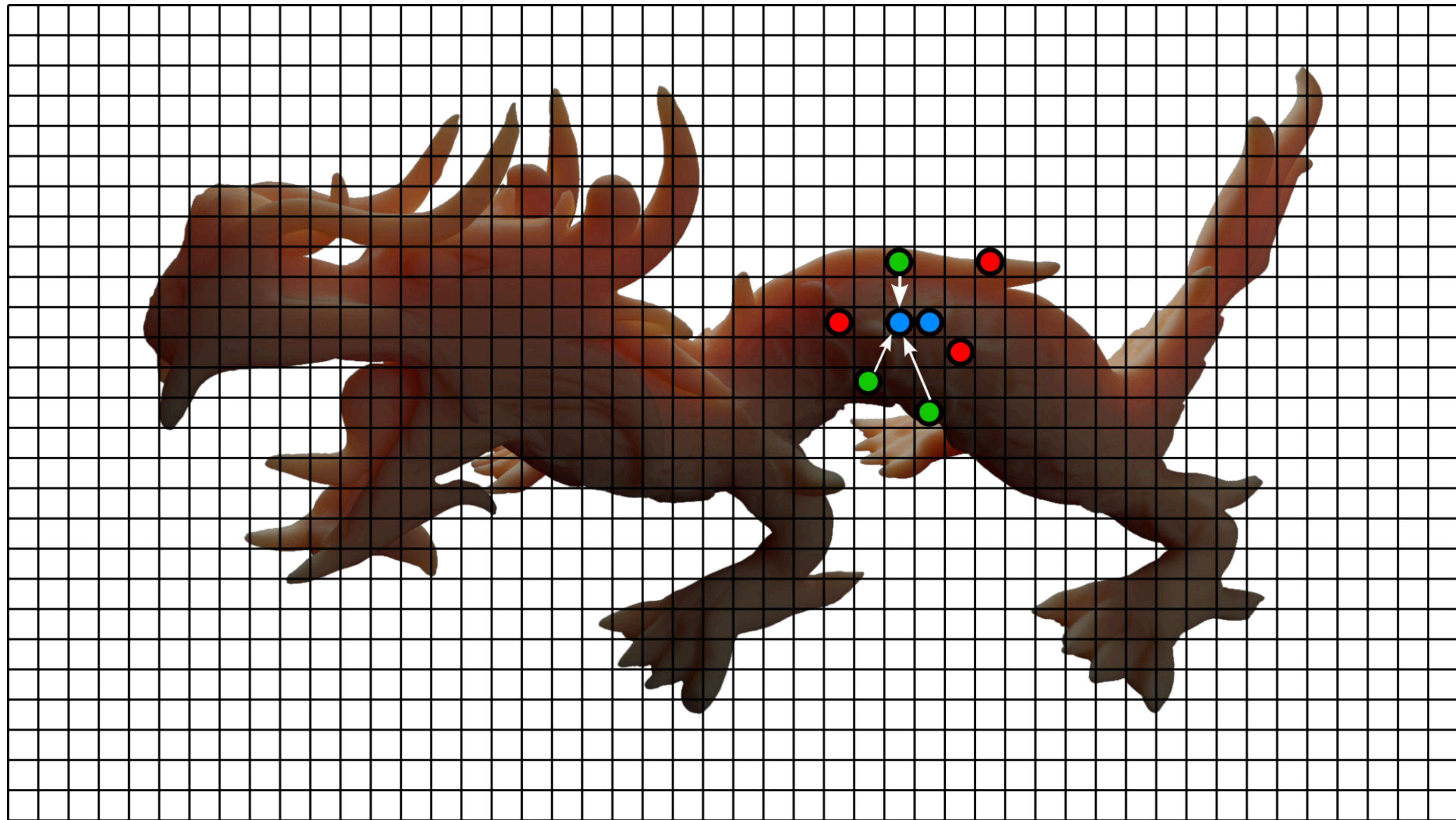




# ReSTIR SSS: Results



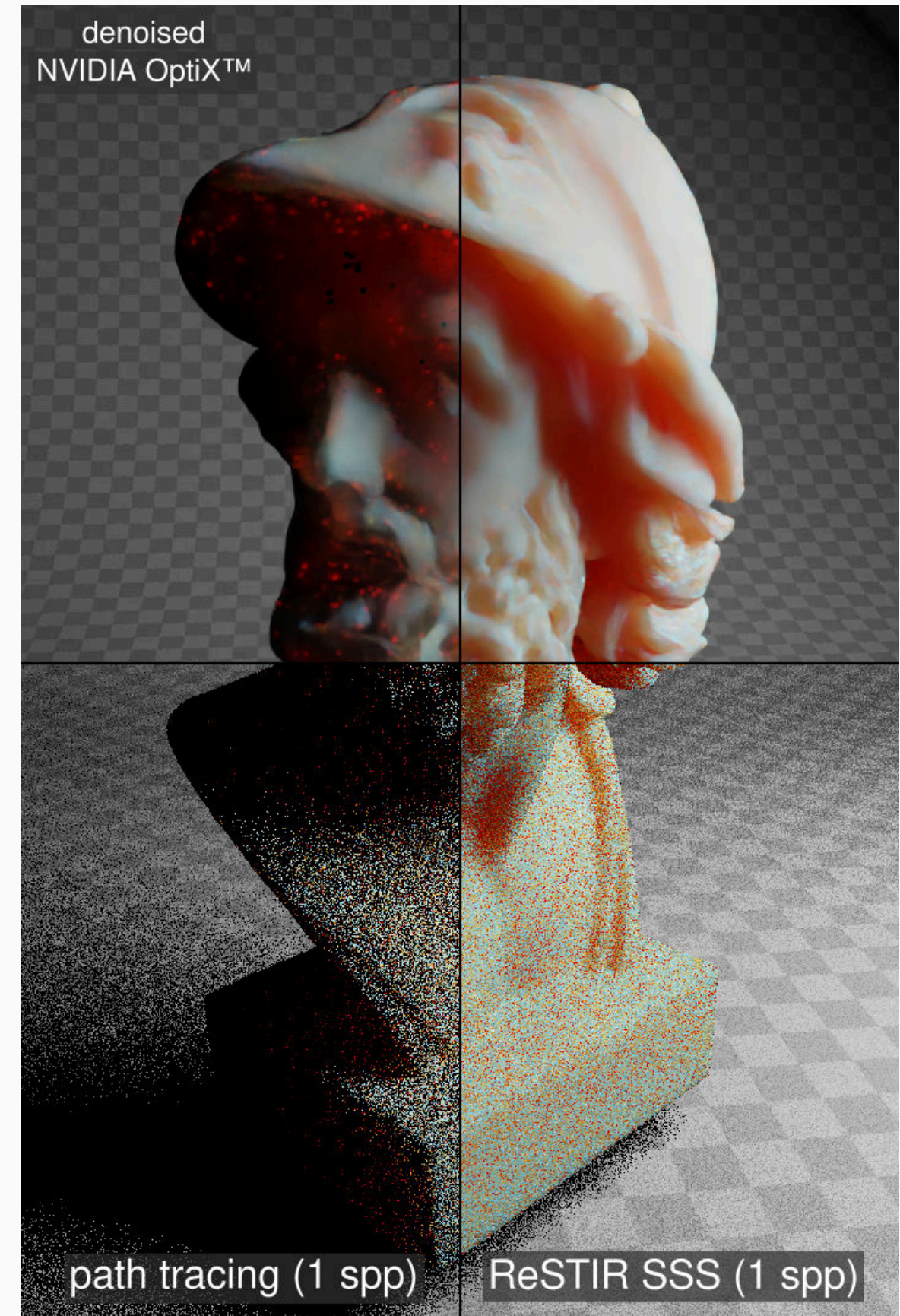
# ReSTIR SSS: Results





# Conclusion

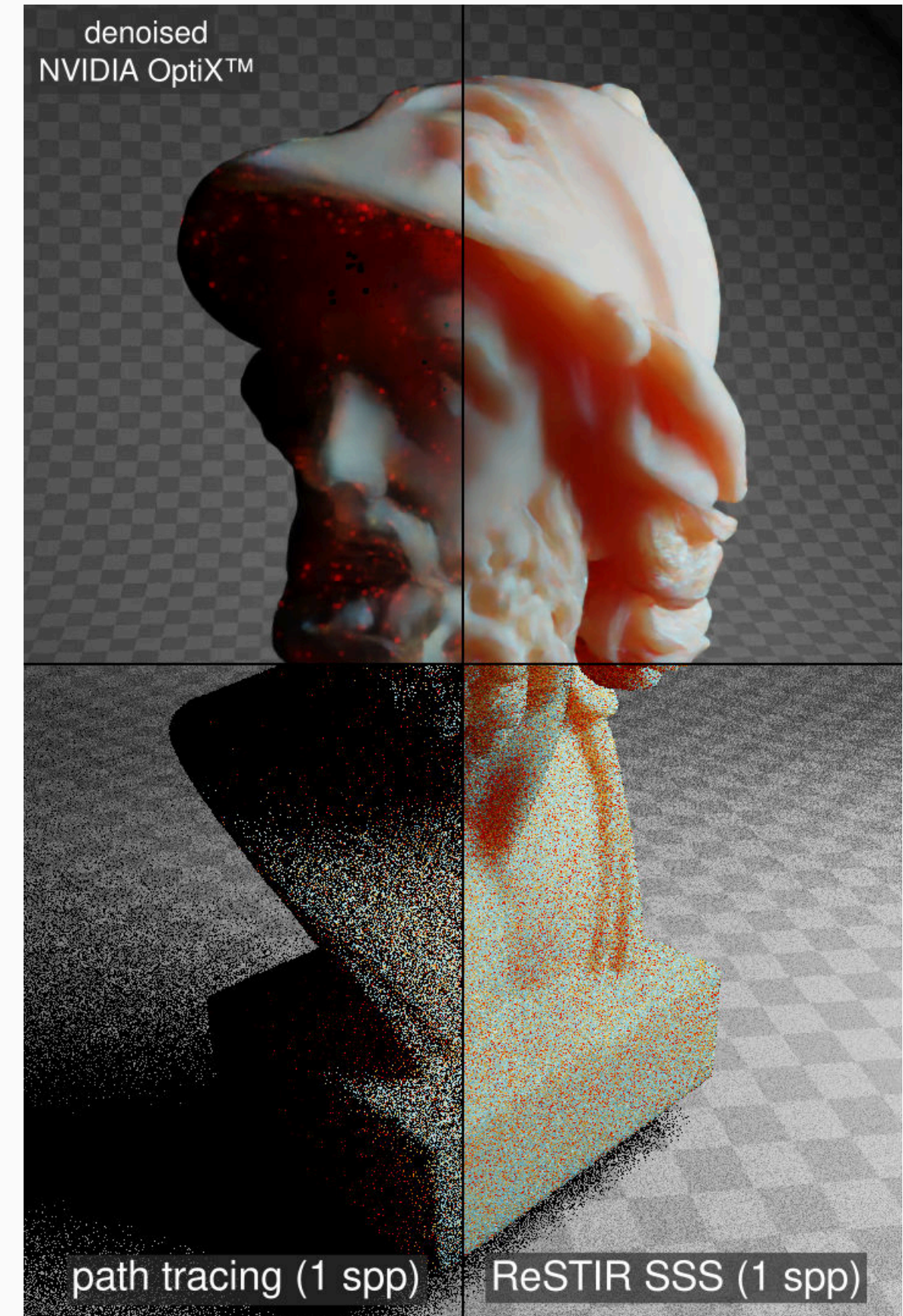
- ReSTIR SSS significantly reduces noise...





# Conclusion

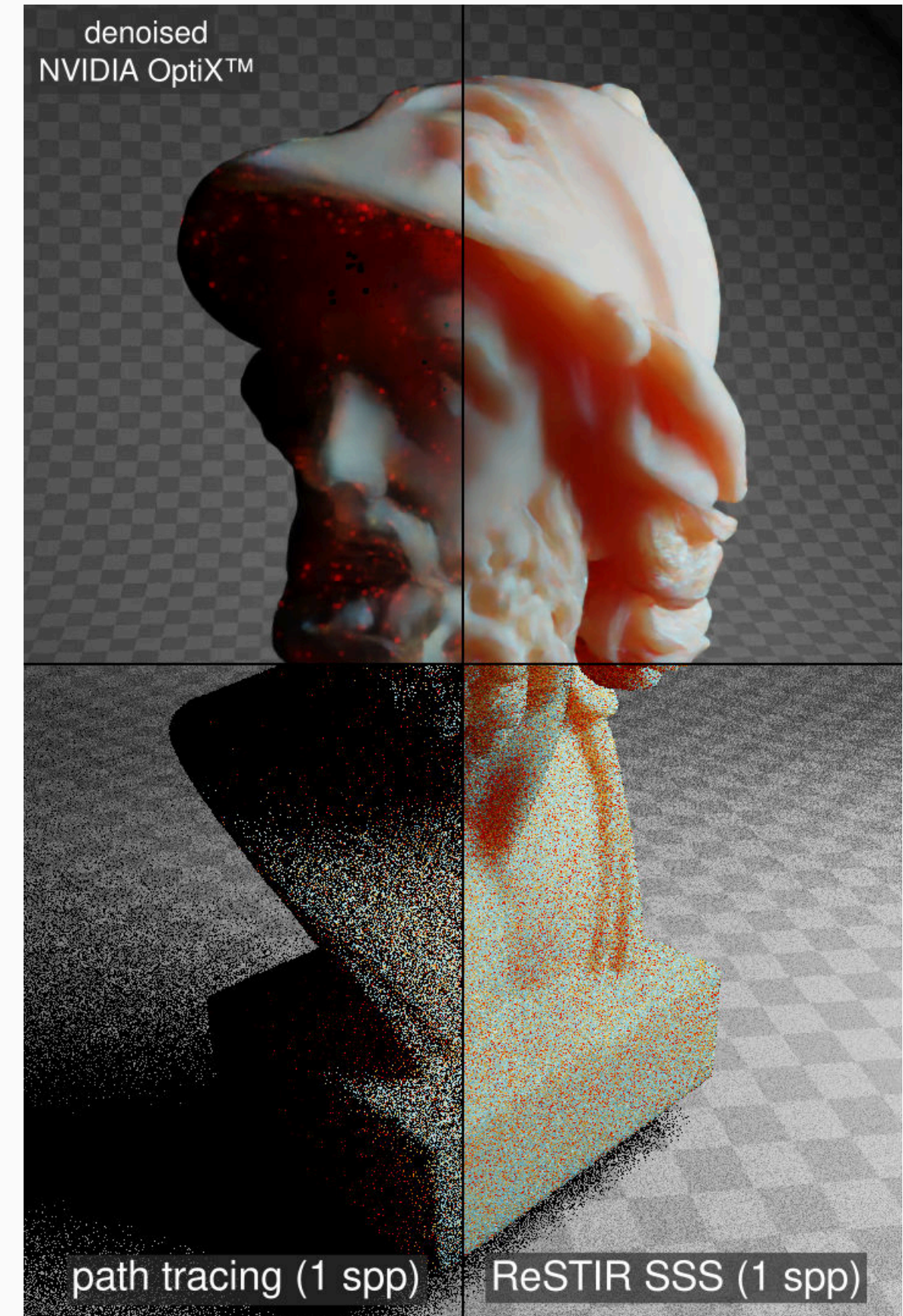
- ReSTIR SSS significantly reduces noise...
  - by using our hybrid





# Conclusion

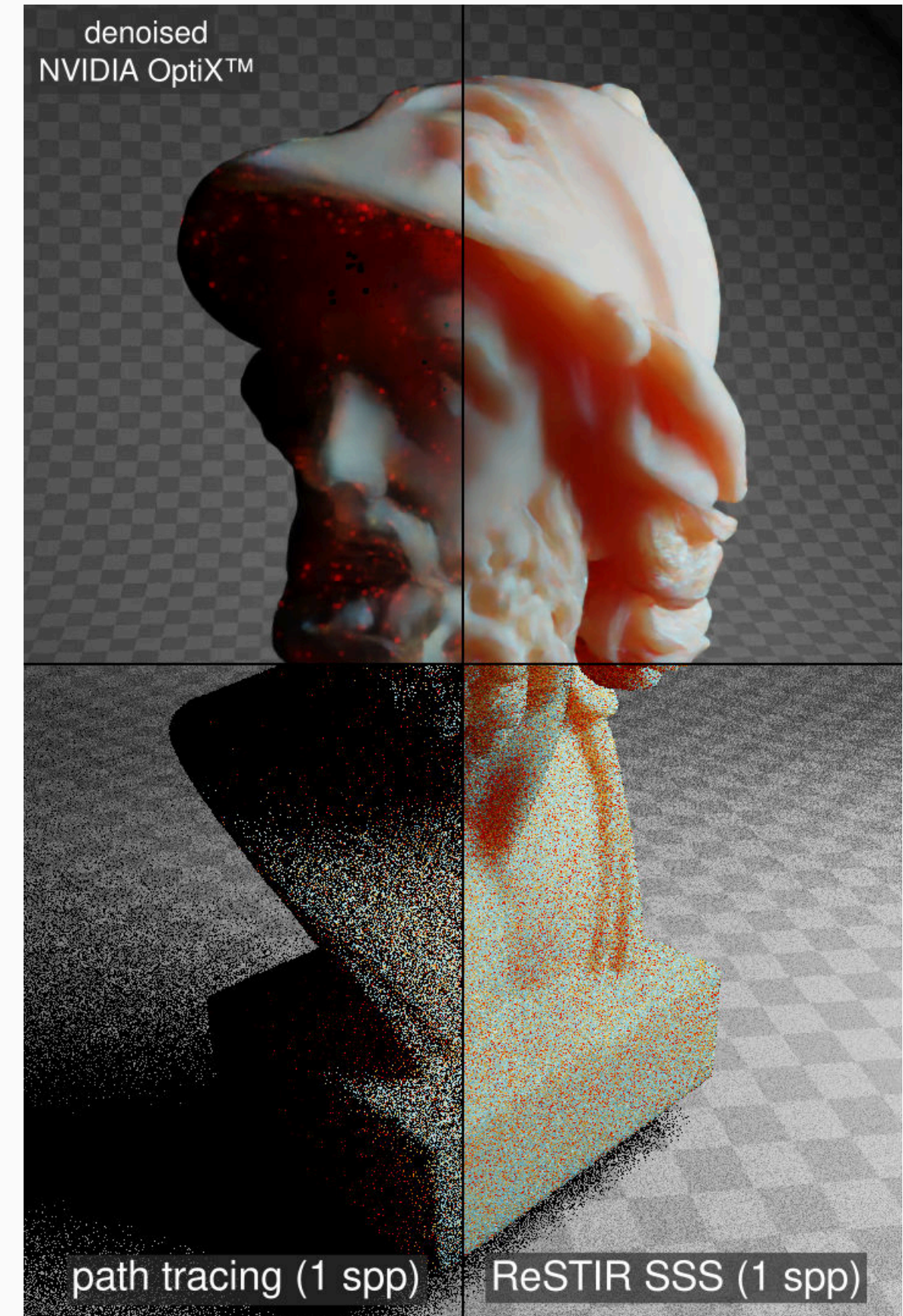
- ReSTIR SSS significantly reduces noise...
  - by using our hybrid
  - or sequential shifting strategies





# Conclusion

- ReSTIR SSS significantly reduces noise...
  - by using our hybrid
  - or sequential shifting strategies
- better quality after denoising...
  - due to better sampling

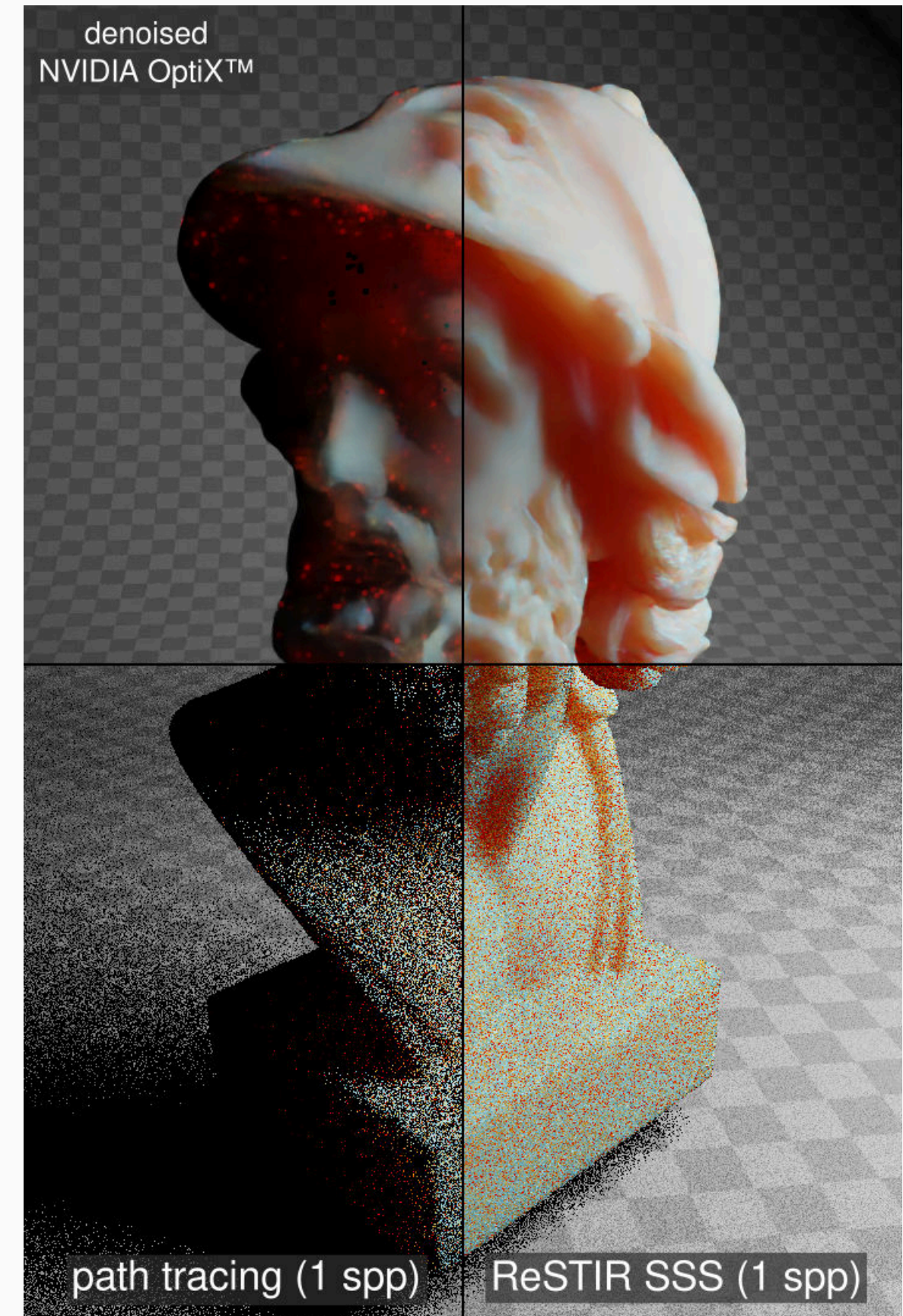




# Conclusion

- ReSTIR SSS significantly reduces noise...
  - by using our hybrid
  - or sequential shifting strategies
- better quality after denoising...
  - due to better sampling

Thank you!



# References

- [King et al. 2013]
  - Alan King, Christopher Kulla, Alejandro Conty, and Marcos Fajardo. 2013. BSSRDF importance sampling. In SIGGRAPH Talks. ACM, Article 48. <https://doi.org/10.1145/2504459.2504520>
- [Bitterli et al. 2020]
  - Benedikt Bitterli, Chris Wyman, Matt Pharr, Peter Shirley, Aaron Lefohn, and Wojciech Jarosz. 2020. Spatiotemporal reservoir resampling for real-time ray tracing with dynamic direct lighting. ACM Trans. Graph. 39, 4, Article 148 (Aug. 2020). <https://doi.org/10.1145/3386569.3392481>
- [Ouyang et al. 2021]
  - Y. Ouyang, S. Liu, M. Kettunen, M. Pharr, and J. Pantaleoni. 2021. ReSTIR GI: Path Resampling for Real-Time Path Tracing. Computer Graphics Forum 40, 8 (2021), 17-29. <https://doi.org/10.1111/cgf.14378>
- [Lin et al. 2022]
  - Daqi Lin, Markus Kettunen, Benedikt Bitterli, Jacopo Pantaleoni, Cem Yuksel, and Chris Wyman. 2022. Generalized resampled importance sampling: foundations of ReSTIR. ACM Trans. Graph. 41, 4, Article 75 (July 2022). <https://doi.org/10.1145/3528223.3530158>
- [Bitterli et al. 2017]
  - Benedikt Bitterli, Wenzel Jakob, Jan Novák, and Wojciech Jarosz. 2017. Reversible Jump Metropolis Light Transport Using Inverse Mappings. ACM Trans. Graph. 37, 1, Article 1 (Oct. 2017). <https://doi.org/10.1145/3132704>