Volume Scattering Probability Guiding

Kehan Xu[1],
Sebastian Herholz[2],
Marco Manzi[3],
Marios Papas[3],
Markus Gross[1,3]

[1]ETH Zürich, [2]Intel Corporation, [3]DisneyResearch|Studios

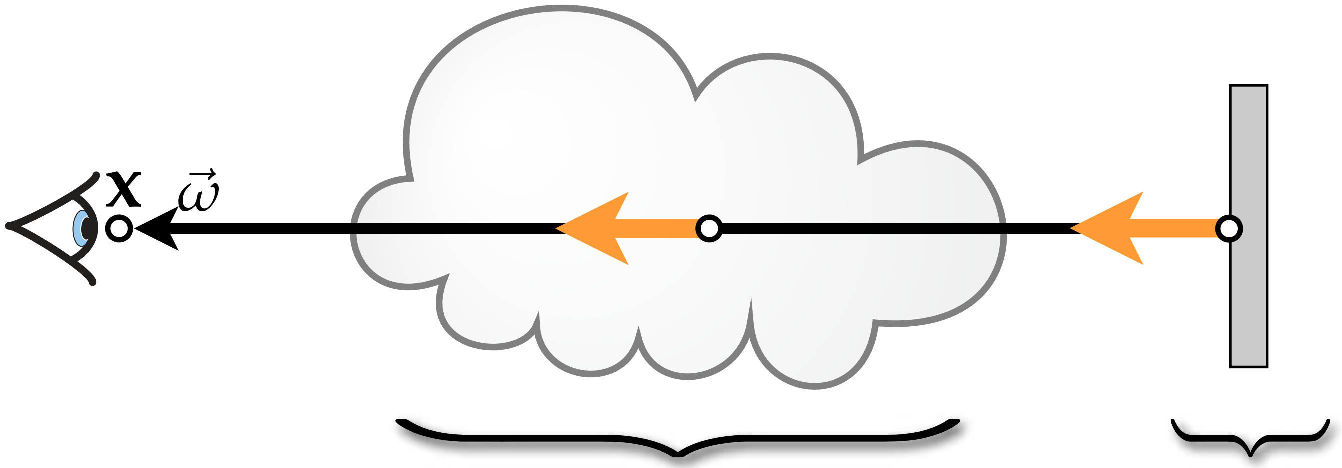
Finn Semmer - Pexels (CC0)

Jungle, 32 SPP, Surface Only

Jungle, 32 SPP, With Volume

Monte Carlo Estimator of the VRE

Volume Rendering Equation



$$L(\mathbf{x},\omega) = L_v(\mathbf{x},\omega) + L_s(\mathbf{x},\omega)$$

Figure source: Wojciech Jarosz

Monte Carlo Estimator

$$\langle L(\mathbf{x},\omega)\rangle = \begin{cases} \frac{1}{P_{\mathrm{vol}}}\langle L_v(\mathbf{x},\omega)\rangle \\ \frac{1}{1-P_{\mathrm{vol}}}\langle L_s(\mathbf{x},\omega)\rangle \end{cases}$$

P_{vol}: the binary sample distribution between surface and volume

Many Works on Optimizing $\langle L_v \rangle$ and $\langle L_s \rangle$

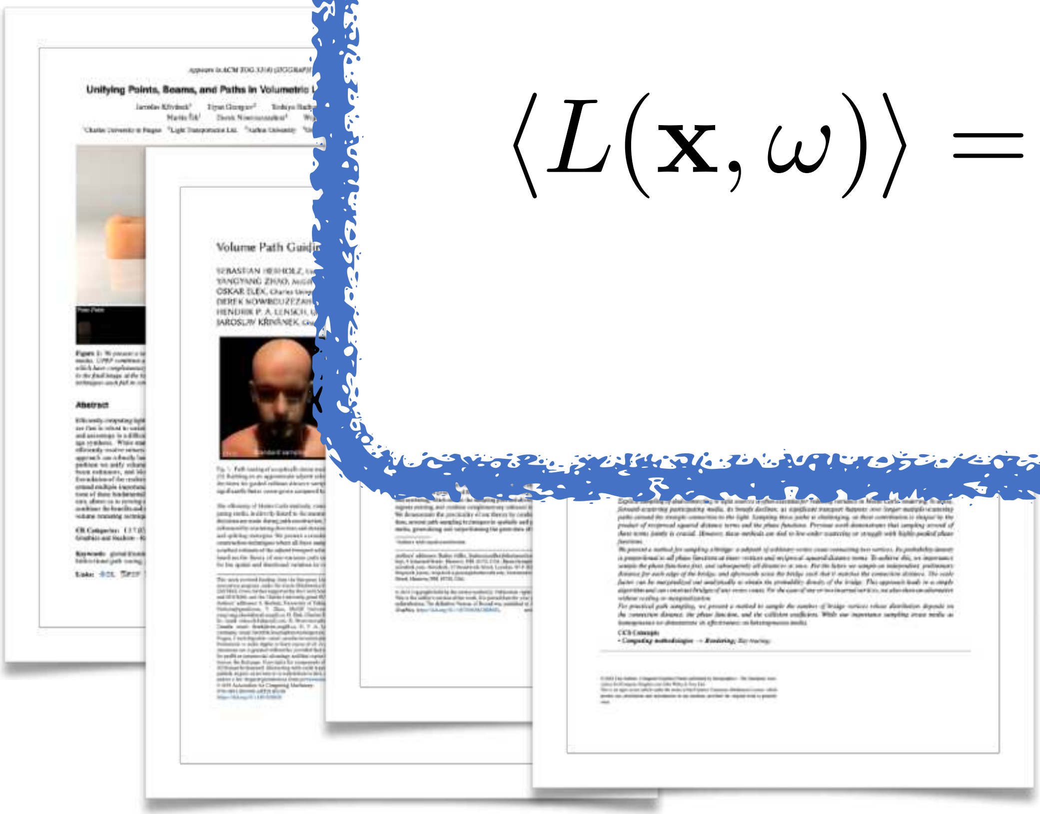
$$\langle L_v(\mathbf{x}, \omega) \rangle$$

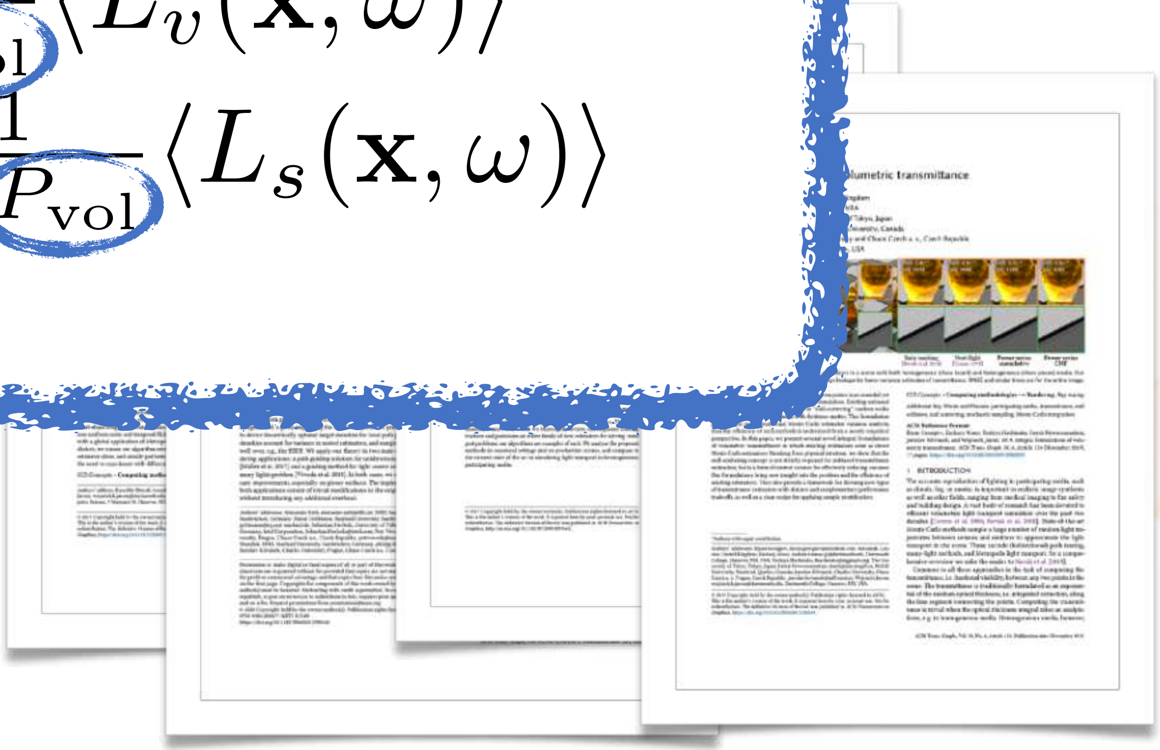
$$\langle L_s(\mathbf{x}, \omega) \rangle = \langle T_r(\mathbf{x}, \mathbf{x}_z) \rangle \langle L_o(\mathbf{x}, \omega) \rangle$$

Many Works on Optimizing $\langle L_v \rangle$ and $\langle L_s \rangle$

$\langle L_o(\mathbf{x}, \omega) \rangle$

$$\langle L(\mathbf{x}, \omega) \rangle = \begin{cases} \frac{1}{P_{\mathrm{vol}}} \langle L_v(\mathbf{x}, \omega) \rangle \\ \frac{1}{1-P_{\mathrm{vol}}} \langle L_s(\mathbf{x}, \omega) \rangle \end{cases}$$





Many Works on Optimizing $\langle L_v \rangle$ and $\langle L_s \rangle$

$$\langle L(\mathbf{x},\omega)\rangle = \begin{cases} \frac{1}{P_{\text{vol}}}\langle L_v(\mathbf{x},\omega)\rangle \\ \frac{1}{\cdots}\langle L_s(\mathbf{x},\omega)\rangle \end{cases}$$

$\langle L_o(\mathbf{x},\omega)\rangle$

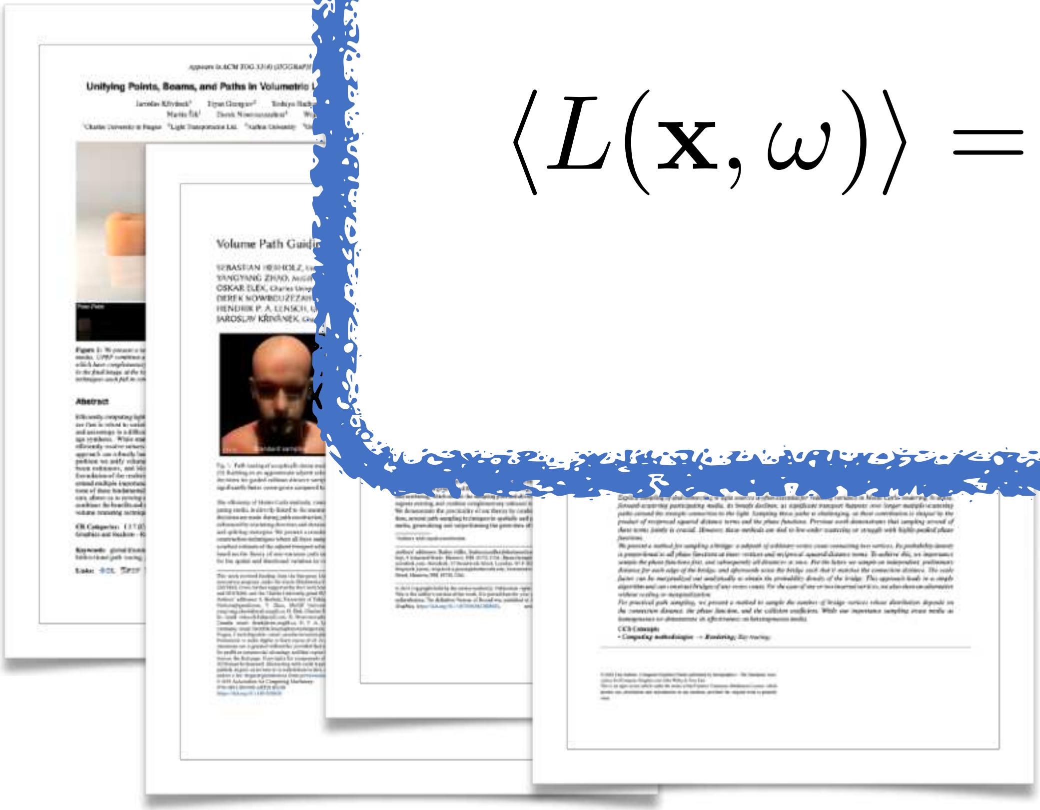
Delta Tracking

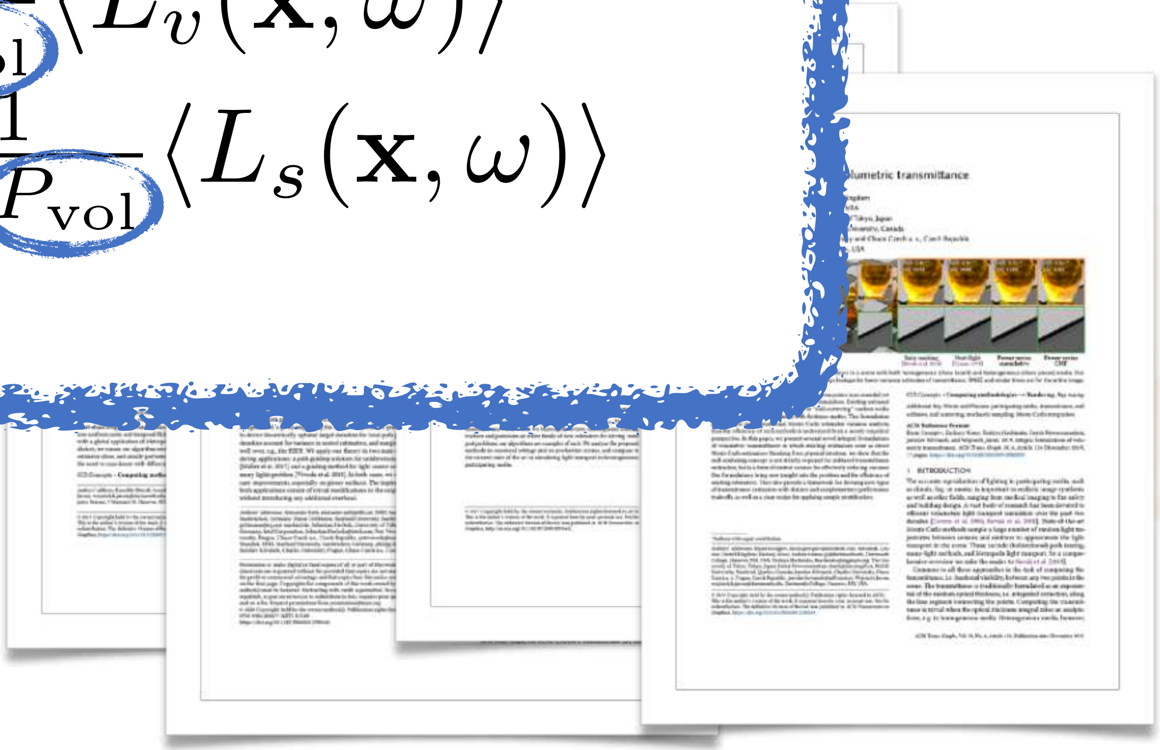
$$P_{\text{vol}}^{\Delta} = 1 - T_r(x, x_z)$$

Many Works on Optimizing $\langle L_v \rangle$ and $\langle L_s \rangle$

$\langle L_o(\mathbf{x}, \omega) \rangle$

$$\langle L(\mathbf{x}, \omega) \rangle = \begin{cases} \frac{1}{P_{\text{vol}}} \langle L_v(\mathbf{x}, \omega) \rangle \\ \frac{1}{1 - P_{\text{vol}}} \langle L_s(\mathbf{x}, \omega) \rangle \end{cases}$$

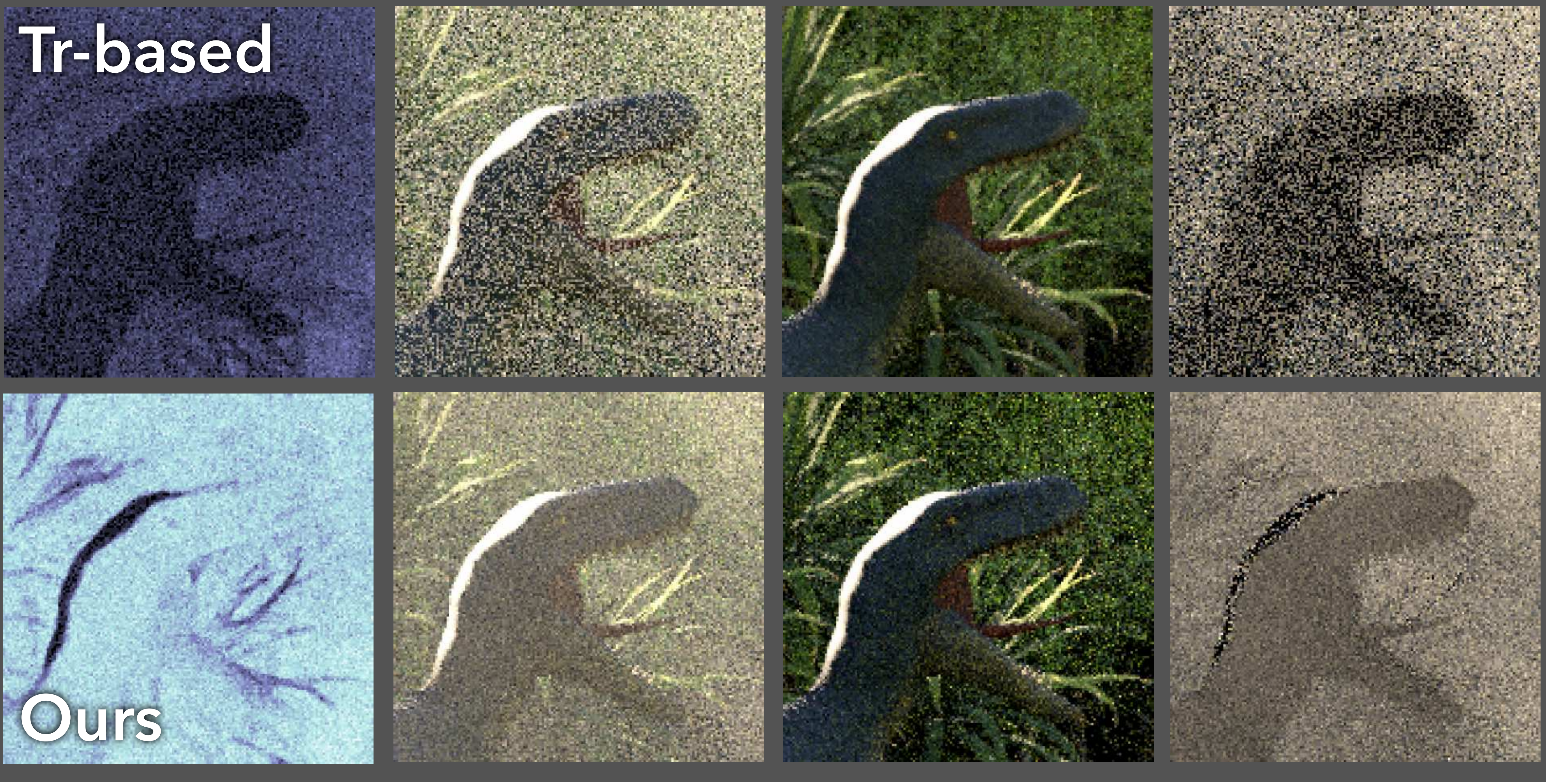




Jungle, 32 SPP, Tr-based

Jungle, 32 SPP, VSP Guiding (Ours)

$P_{\text{vol}} \quad L(\mathbf{x},\omega) = L_s(\mathbf{x},\omega) + L_v(\mathbf{x},\omega)$



$$P_{\text{vol}} \qquad L(\mathbf{x},\omega) = L_s(\mathbf{x},\omega) + L_v(\mathbf{x},\omega)$$

The Optimal VSP: Two Types

- Zero-Variance-based [Herholz et al. 2019]:
  - Assuming the nested estimators have **no variance**

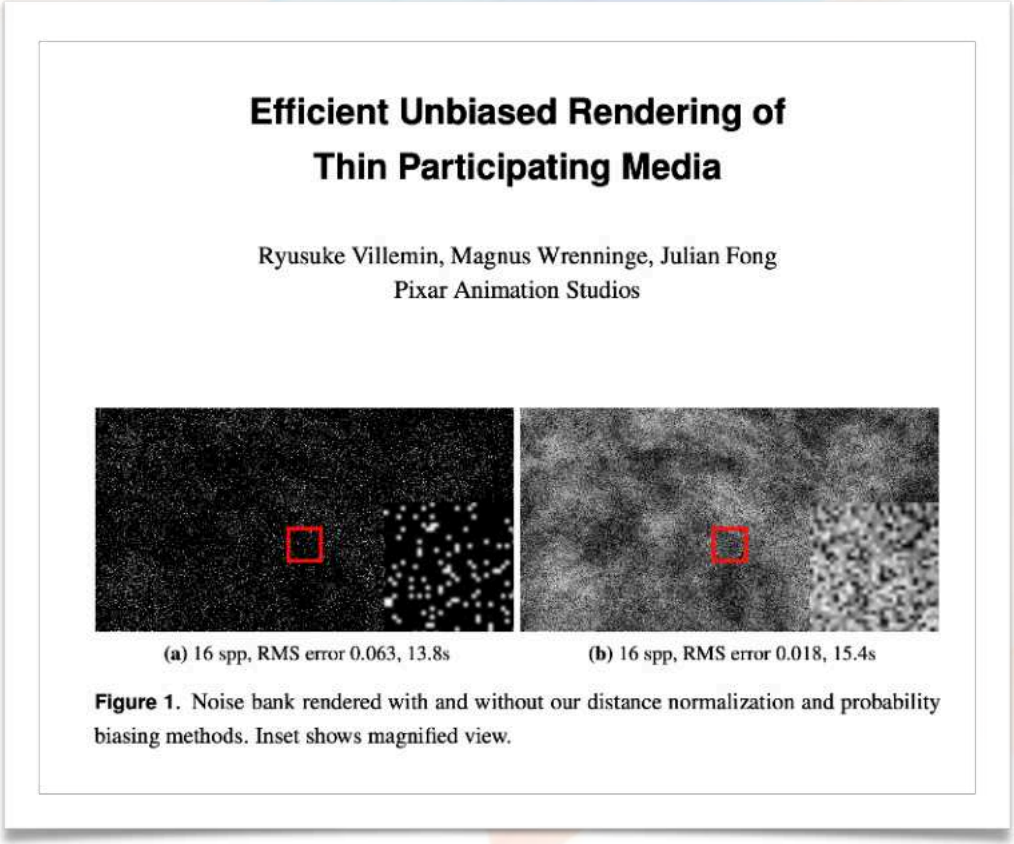
$$P_{\mathrm{vol}}^{\mathrm{1st}} = \frac{\mathbb{E}[\langle L_v(x,\omega)\rangle]}{\mathbb{E}[\langle L_v(x,\omega)\rangle] + \mathbb{E}[\langle L_s(x,\omega)\rangle]}$$

- Variance-based [Rath et al. 2020]:
  - Considering the **variance** of the nested estimators

$$P_{\mathrm{vol}}^{\mathrm{2nd}} = \frac{\mathbb{E}[\langle L_v(x,\omega)\rangle^2]}{\mathbb{E}[\langle L_v(x,\omega)\rangle^2] + \mathbb{E}[\langle L_s(x,\omega)\rangle^2]}$$

Normalized Distance Sampling

- Need to **manually** set the VSP per scene / per volume
- Only support increasing the VSP
- Not reaching the target VSP



Efficient Unbiased Rendering of Thin Participating Media

Ryusuke Villemin, Magnus Wrenninge, Julian Fong
Pixar Animation Studios

(a) 16 spp, RMS error 0.063, 13.8s

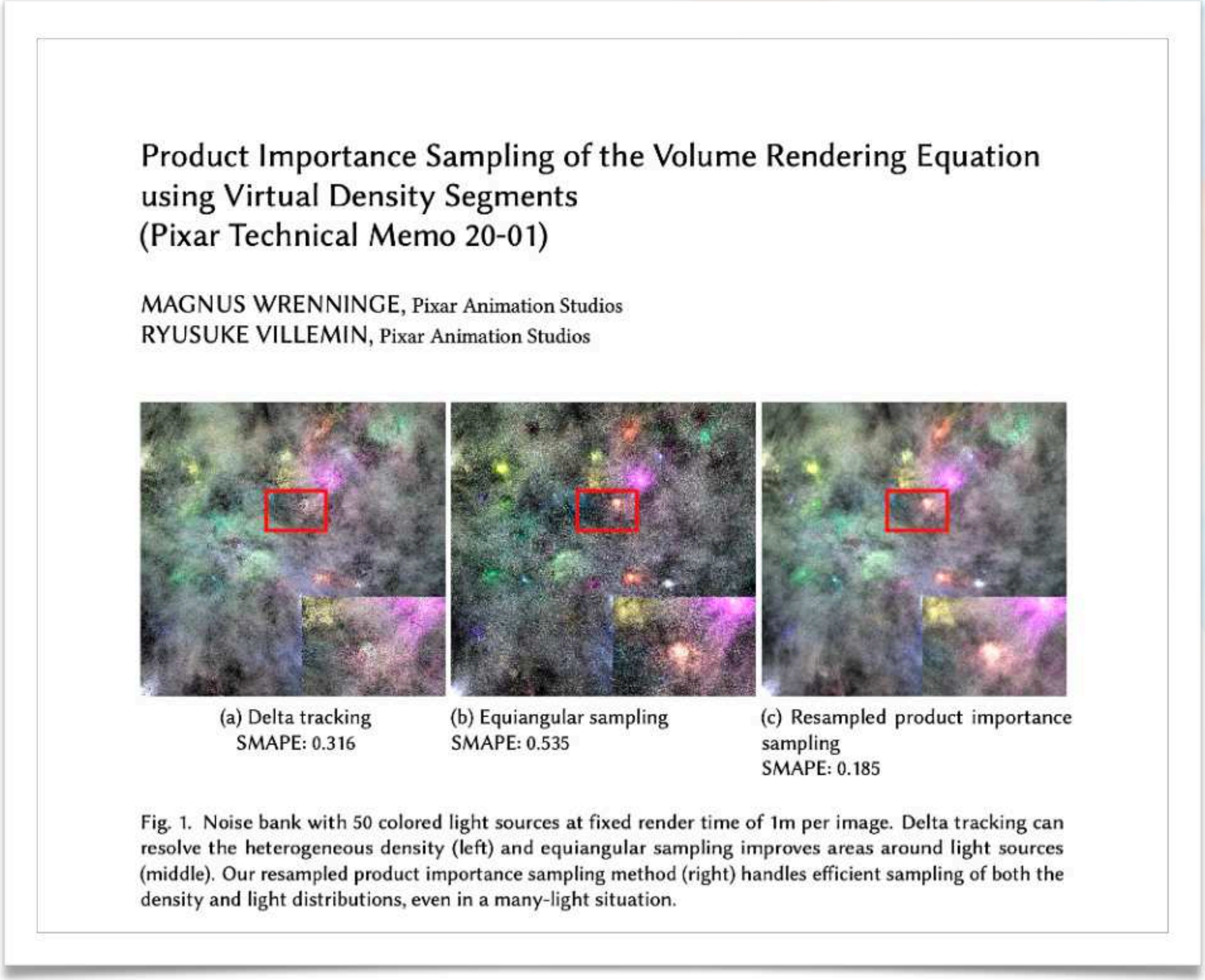
(b) 16 spp, RMS error 0.018, 15.4s

Figure 1. Noise bank rendered with and without our distance normalization and probability biasing methods. Inset shows magnified view.

- Homogeneous
- Heterogeneous

Our Distance Sampling Method

Delta Tracking as Resampling



Product Importance Sampling of the Volume Rendering Equation using Virtual Density Segments (Pixar Technical Memo 20-01)

MAGNUS WRENNINGE, Pixar Animation Studios
RYUSUKE VILLEMIN, Pixar Animation Studios

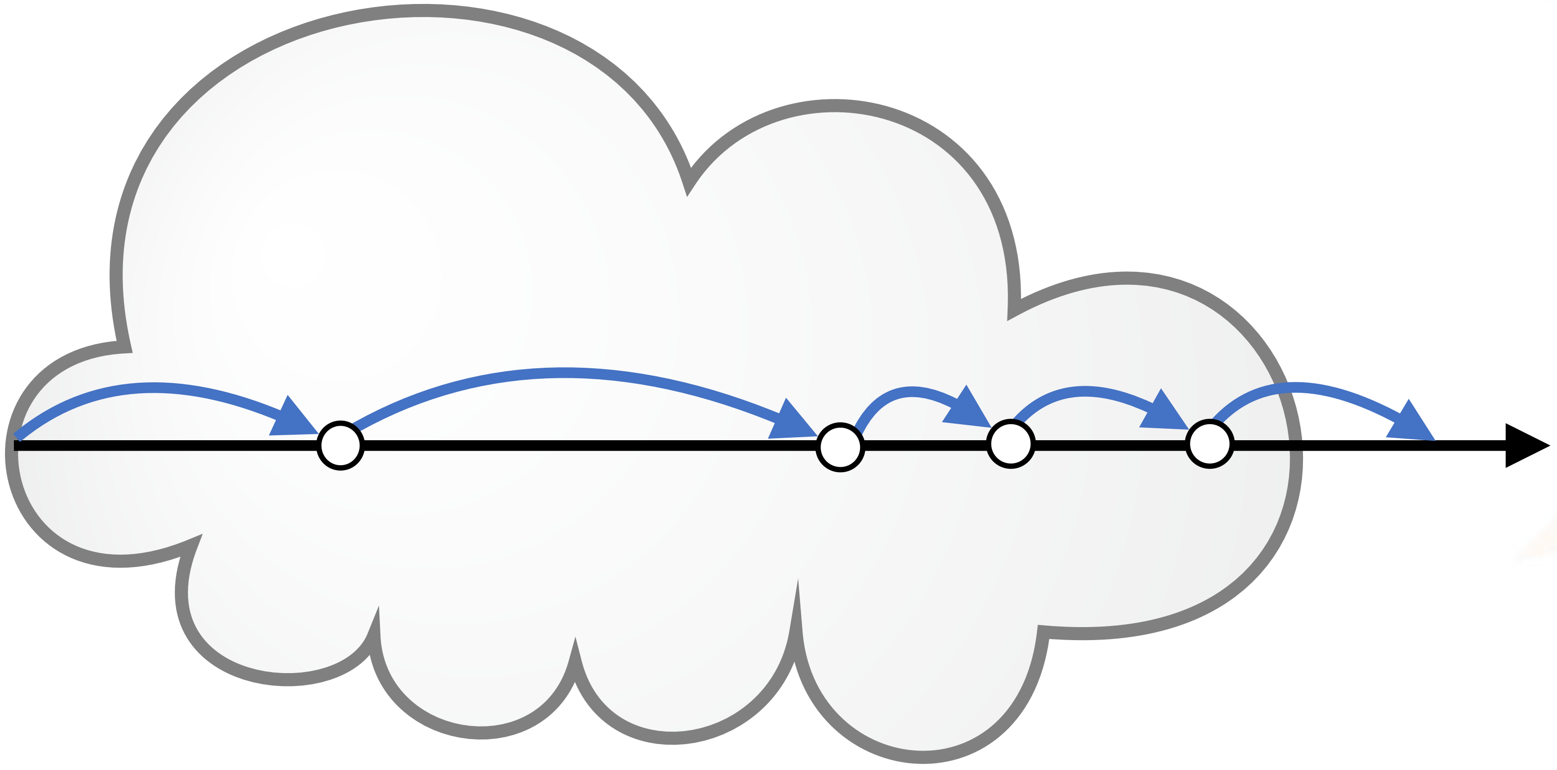
(a) Delta tracking
SMAPE: 0.316

(b) Equiangular sampling
SMAPE: 0.535

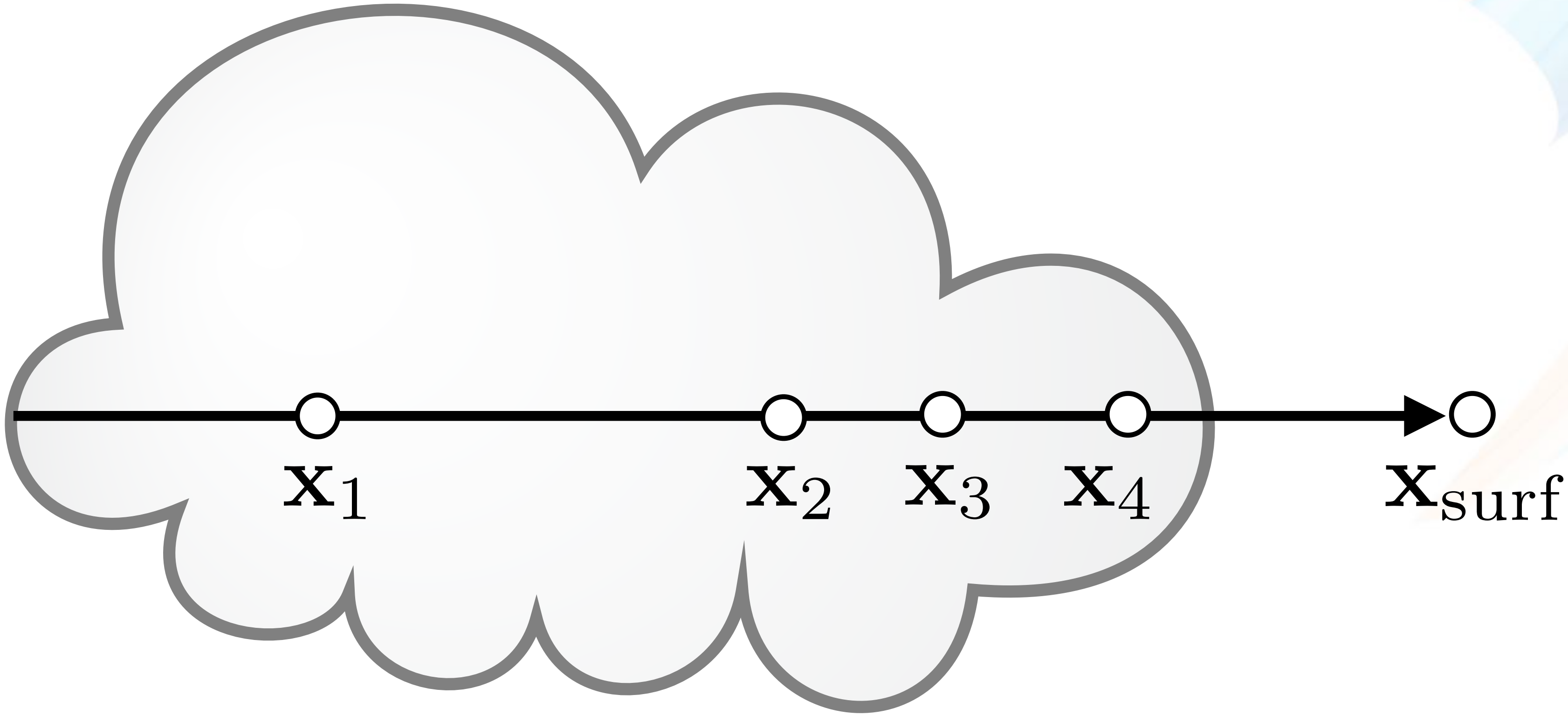
(c) Resampled product importance sampling
SMAPE: 0.185

Fig. 1. Noise bank with 50 colored light sources at fixed render time of 1m per image. Delta tracking can resolve the heterogeneous density (left) and equiangular sampling improves areas around light sources (middle). Our resampled product importance sampling method (right) handles efficient sampling of both the density and light distributions, even in a many-light situation.

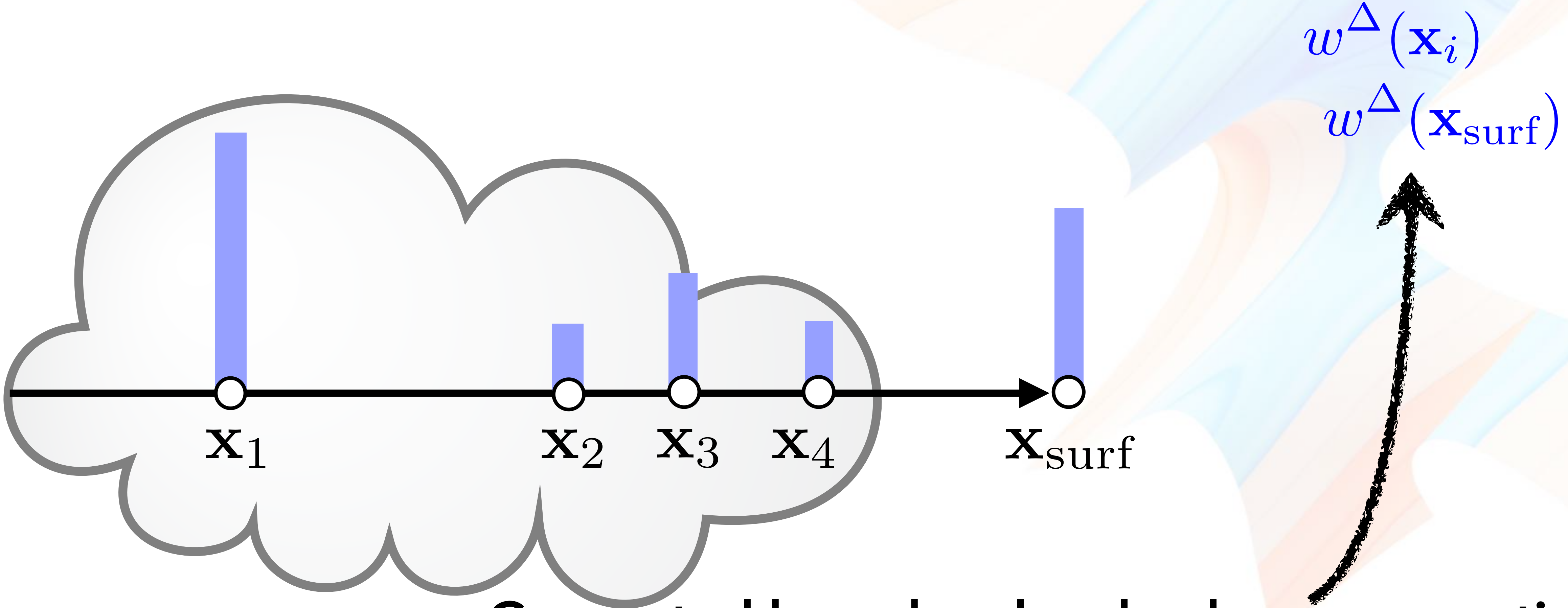
Delta Tracking as Resampling



Candidate Samples

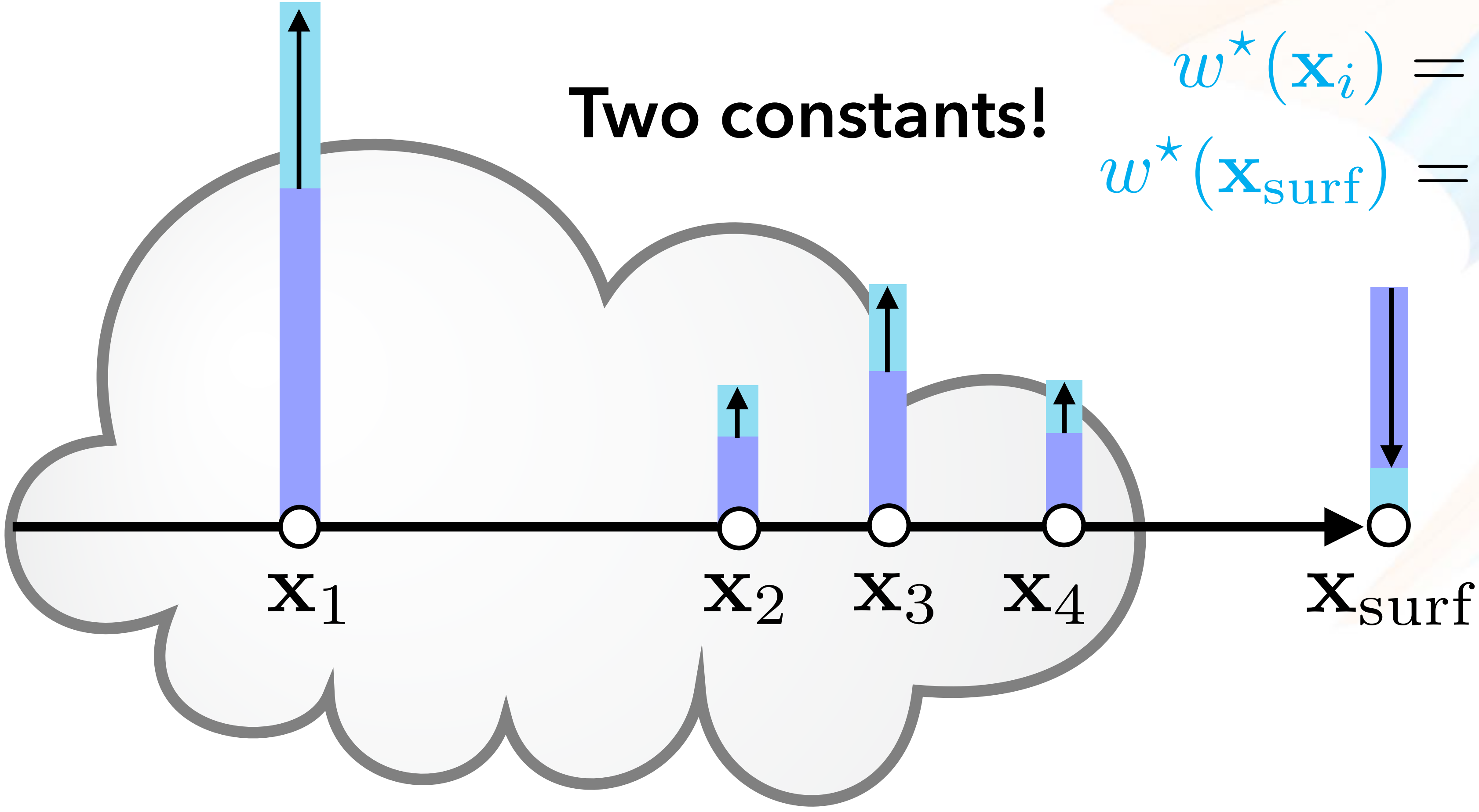


Resampling Weights for Delta Tracking



Computed based on local volume properties
(details in the paper)

Resampling Weights for Our Method

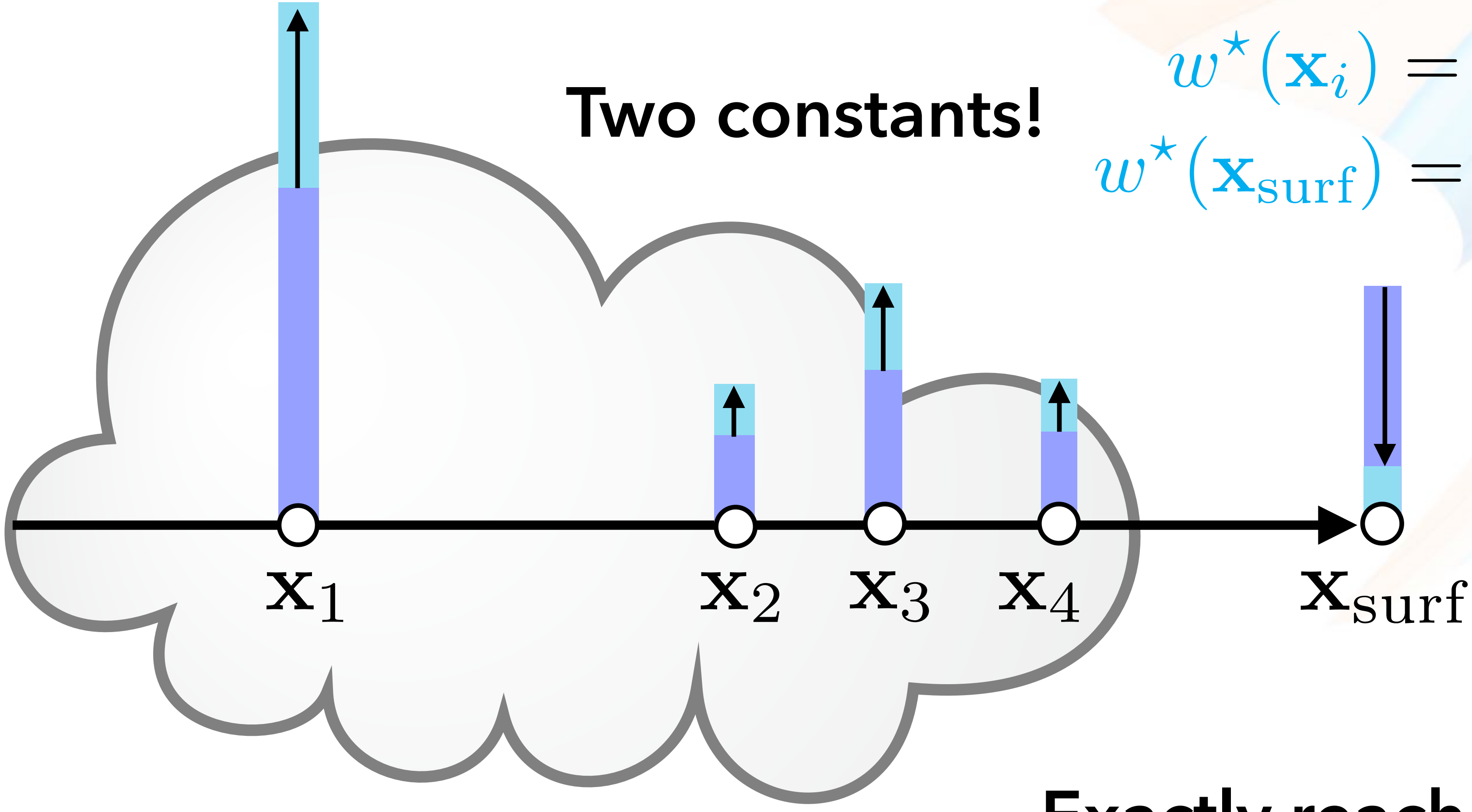


Two constants!

$$w^\star(\mathbf{x}_i) = C_{\text{vol}} * w^\Delta(\mathbf{x}_i)$$

$$w^\star(\mathbf{x}_{\text{surf}}) = C_{\text{surf}} * w^\Delta(\mathbf{x}_{\text{surf}})$$

Resampling Weights for Our Method



Two constants!

$$w^\star(\mathbf{x}_i) = C_{\text{vol}} * w^\Delta(\mathbf{x}_i)$$
$$w^\star(\mathbf{x}_{\text{surf}}) = C_{\text{surf}} * w^\Delta(\mathbf{x}_{\text{surf}})$$

$$C_{\text{vol}} \approx \frac{P^\star_{\text{vol}}}{P^\Delta_{\text{vol}}}$$

$$C_{\text{surf}} \approx \frac{1 - P^\star_{\text{vol}}}{1 - P^\Delta_{\text{vol}}}$$

Exactly reaches the target P_{vol}

Resampling Weights for Our Method

Reservoir Sampling :)

$$\mathbb{P}[\text{Accept } i] = \frac{w_i}{\sum_{j \leq i} w_j}$$

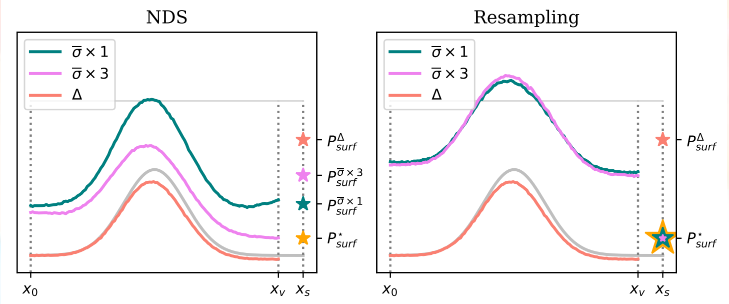
$$\frac{P^\star_{\text{vol}}}{P^\Delta_{\text{vol}}}$$

$$\frac{1 - P^\star_{\text{vol}}}{1 - P^\Delta_{\text{vol}}}$$

Exactly reaches the target P_{vol}

Details in the Paper!

- Zero volume event candidate
- Defensive resampling
- Volume sample distribution analysis
- Increase majorant
- …



ALGORITHM 1: Volume Scattering Probability Guiding

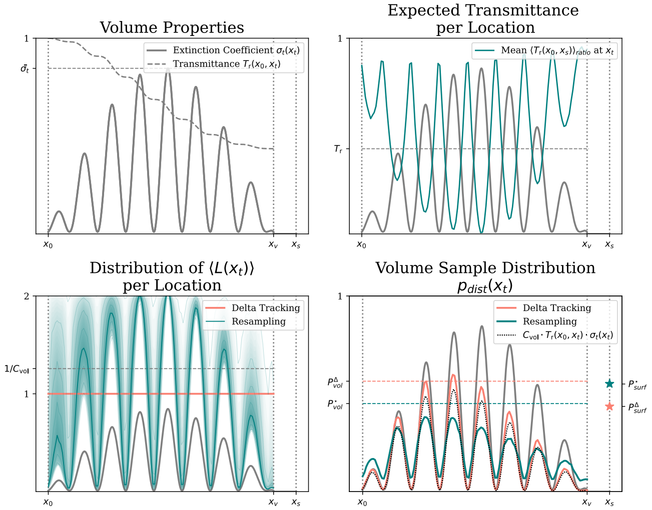
1. **Function** VSPG($\bar{\sigma}, t_v, P^{\star}_{vol}, \alpha$):
2. *Reservoir r*
3. $t \leftarrow 0, \langle T_r \rangle_{ratio} \leftarrow 1, w_{sum} \leftarrow 0$
4. $\bar{\sigma}', P^{\star\prime}_{vol} \leftarrow$ `ZeroVolumeCandidateCompensation`$(\bar{\sigma}, t_v, P^{\star}_{vol}, \alpha)$
5. **while** *true* **do**
6. $t \leftarrow t - \frac{\ln(1-\xi)}{\bar{\sigma}'}$ `// Distance sampling, Eq. 11`
7. $\mathbf{x}_i \leftarrow \mathbf{x} + t\omega$ `// Generate a volume candidate`
8. **if** $t \geq t_v$ **then**
9. **break**
10. $\langle T_r \rangle_{ratio} \leftarrow P_{null}(\mathbf{x}_i) \langle T_r \rangle_{ratio}$
11. $w_{sum} \leftarrow w_{sum} + w^{\Delta}(\mathbf{x}_i)$ `// Eq. 21`
12. r.update$(\mathbf{x}_i, \frac{w^{\Delta}(\mathbf{x}_i)}{w_{sum}})$
13. **end**
14. $\mathbf{x}_M \leftarrow \mathbf{x} + t_v\omega$ `// Generate the surface candidate`

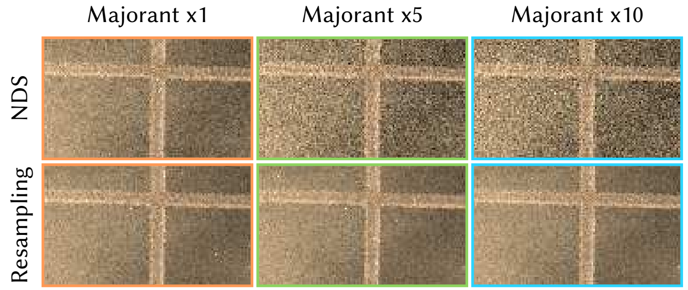
`/* Defensive resampling */`

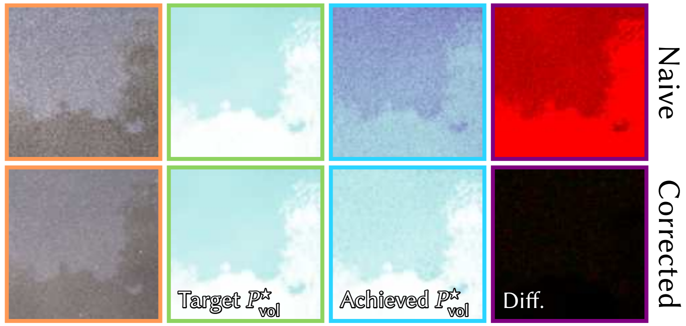
15. $w^{\alpha}_{sum} \leftarrow \alpha(1 - P^{\star\prime}_{vol}) + (1-\alpha)\, w_{sum}$ `// Eq. 27`
16. $w^{\alpha}(\mathbf{x}_M) \leftarrow \alpha P^{\star\prime}_{vol} + (1-\alpha)\, w^{\Delta}(\mathbf{x}_M)$ `// Eq. 22, Eq. 27`
17. $w^{\alpha}_{sum} \leftarrow w^{\alpha}_{sum} + w^{\alpha}(\mathbf{x}_M)$
18. r.update$(\mathbf{x}_M, \frac{w^{\alpha}(\mathbf{x}_M)}{w^{\alpha}_{sum}})$ `//` $w^{\alpha}_{sum} = 1$

`/* Set path segment throughput */`

19. $P^{\alpha}_{vol} \leftarrow \alpha P^{\star\prime}_{vol} + (1-\alpha)(1 - \langle T_r \rangle_{ratio})$ `// Resulting VSP`
20. $r.T_p \leftarrow \frac{1-\langle T_r \rangle_{ratio}}{P^{\alpha}_{vol}}$ or $\frac{\langle T_r \rangle_{ratio}}{1-P^{\alpha}_{vol}}$ `// Eq. 28`
21. **return** r
22. **Function** `ZeroVolumeCandidateCompensation`$(\bar{\sigma}, t_v, P^{\star}_{vol}, \alpha)$:
23. $\bar{\sigma}' \leftarrow \max\left(\bar{\sigma}, -\ln(1 - P^{\star}_{vol})/t_v\right)$ `// Eq. 25`
24. $P^{\star\prime}_{vol} \leftarrow P^{\star}_{vol}/(1 - \exp(-\bar{\sigma}' t_v))$ `// Eq. 26`
25. **return** $\bar{\sigma}', P^{\star\prime}_{vol}$







Our VSP Guiding Framework

- Structures to query the optimal P_{vol} for every path segment
- Incremental training during rendering

Figure source: Wojciech Jarosz

Our VSP Guiding Framework

Primary rays:

Auxiliary image space buffer

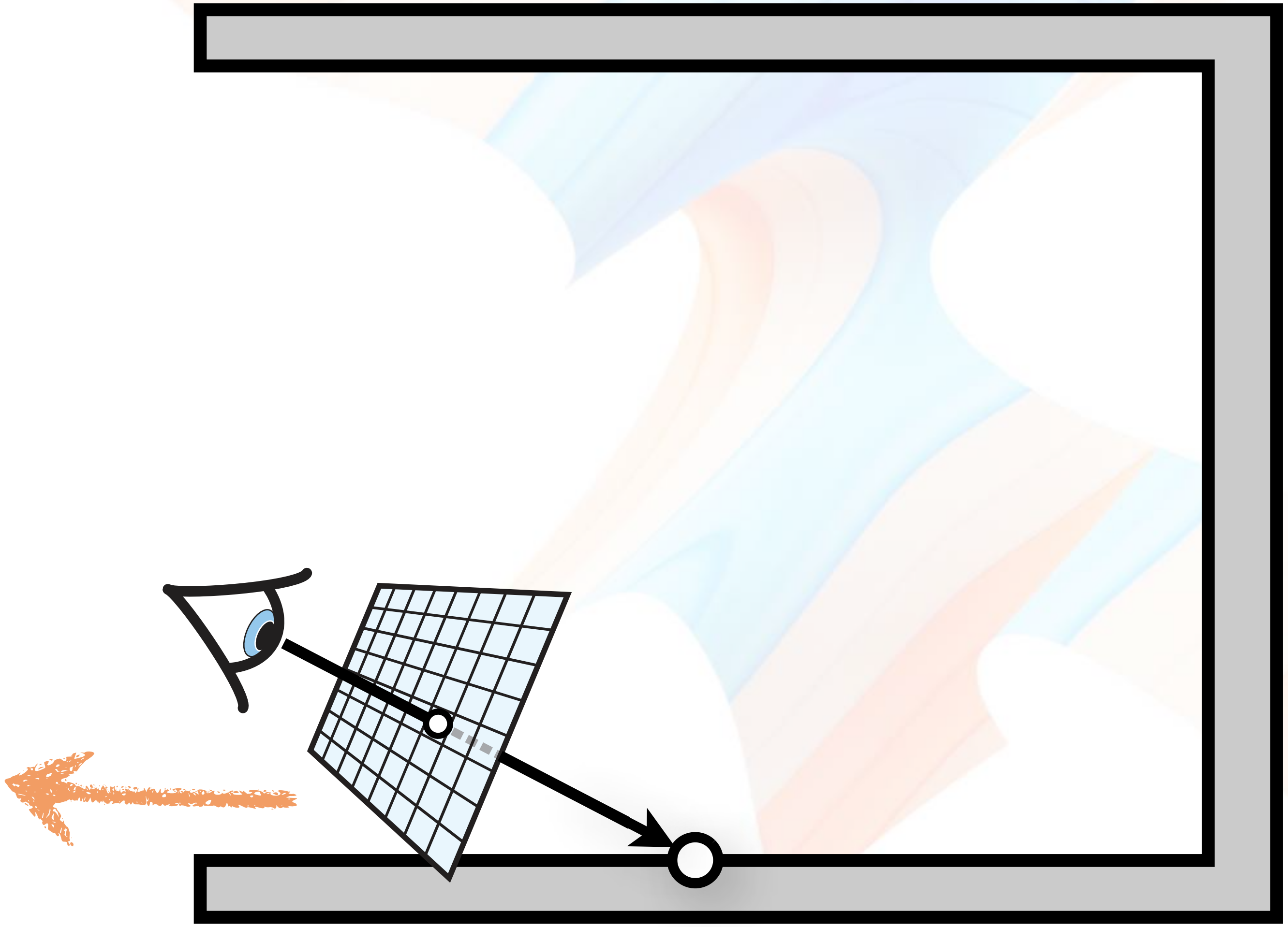


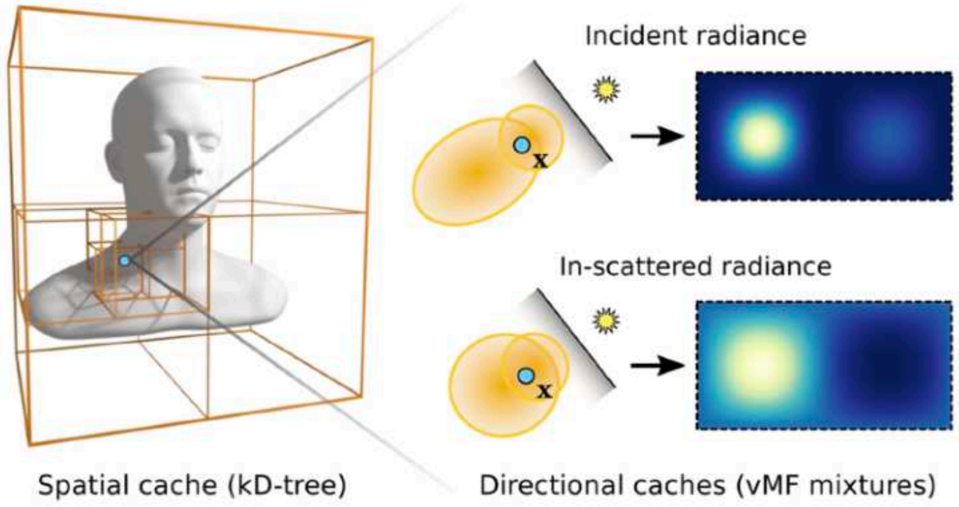
Figure source: Wojciech Jarosz

Our VSP Guiding Framework

Secondary rays:

5D spatial-directional data structure

Piggyback existing data structures (e.g., from path guiding)!



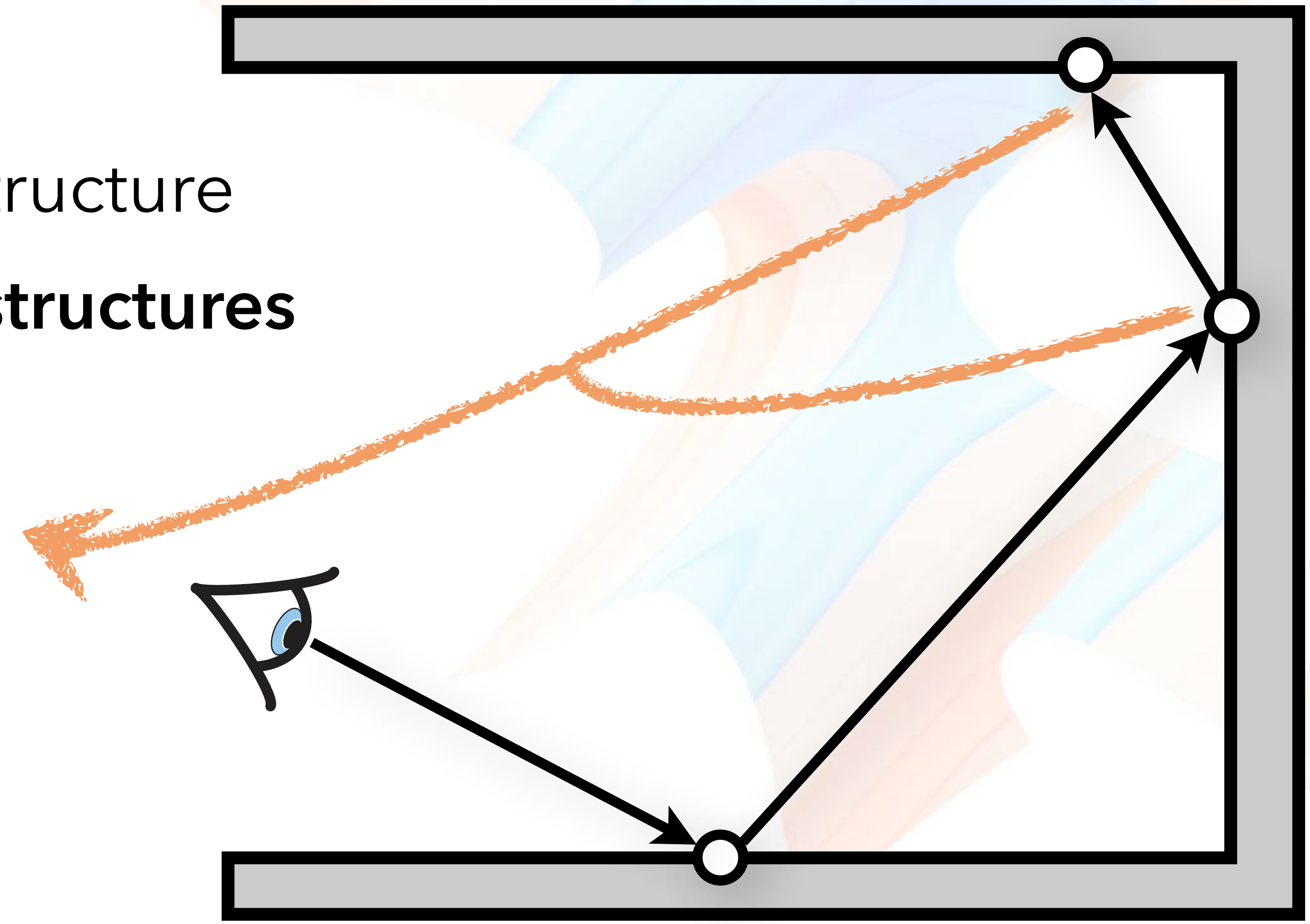


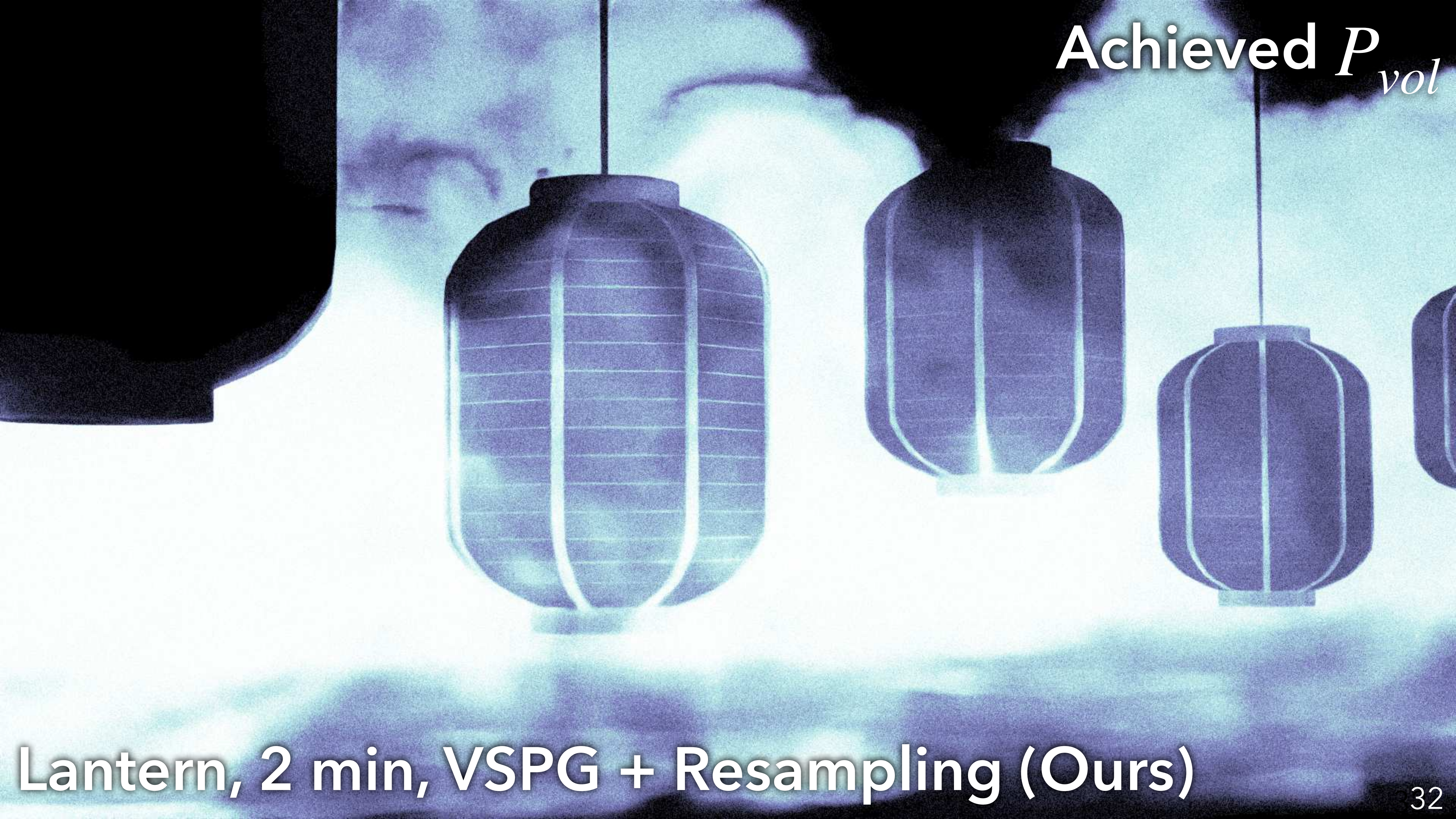
Figure source: Wojciech Jarosz

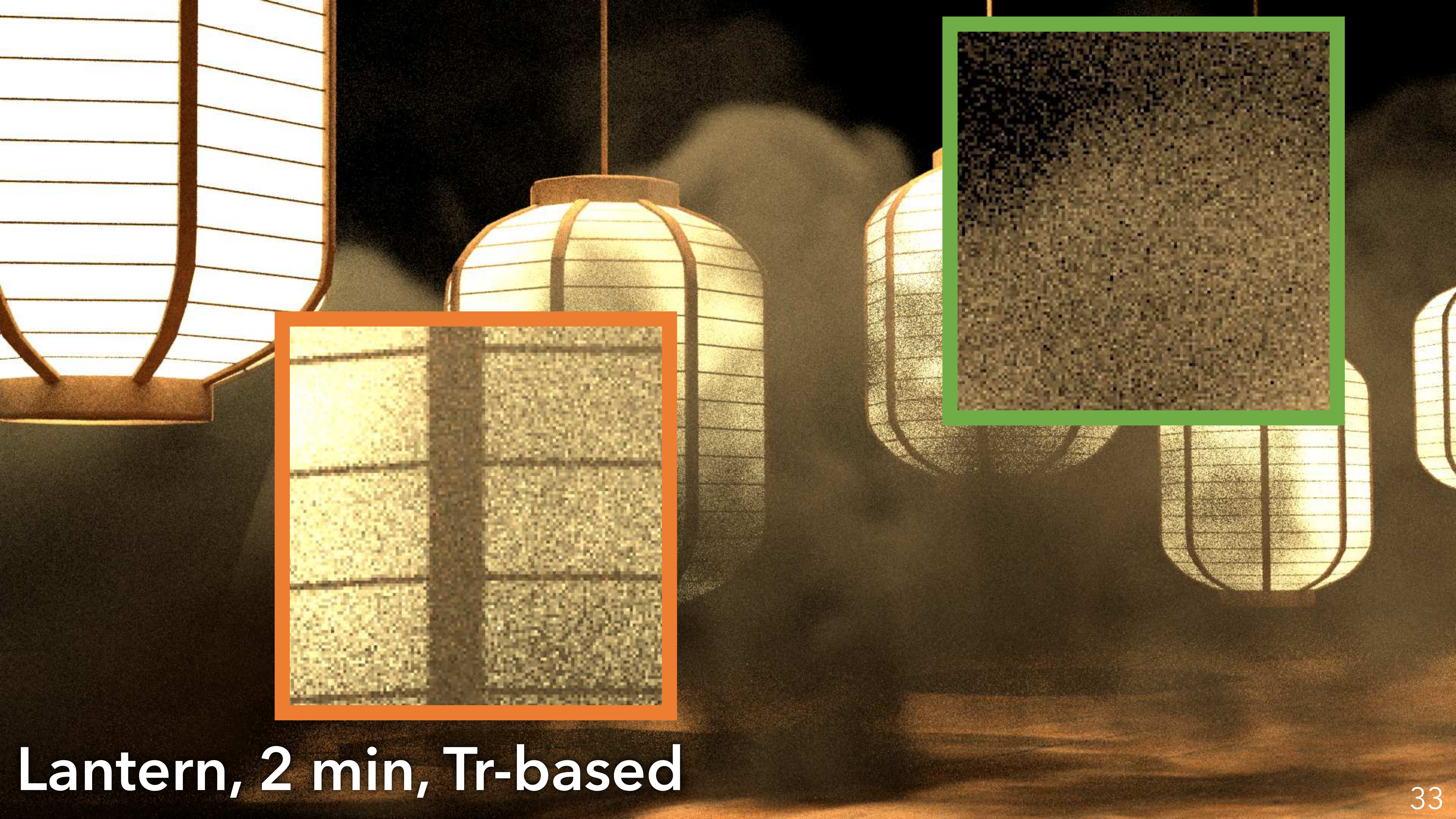
Evaluation

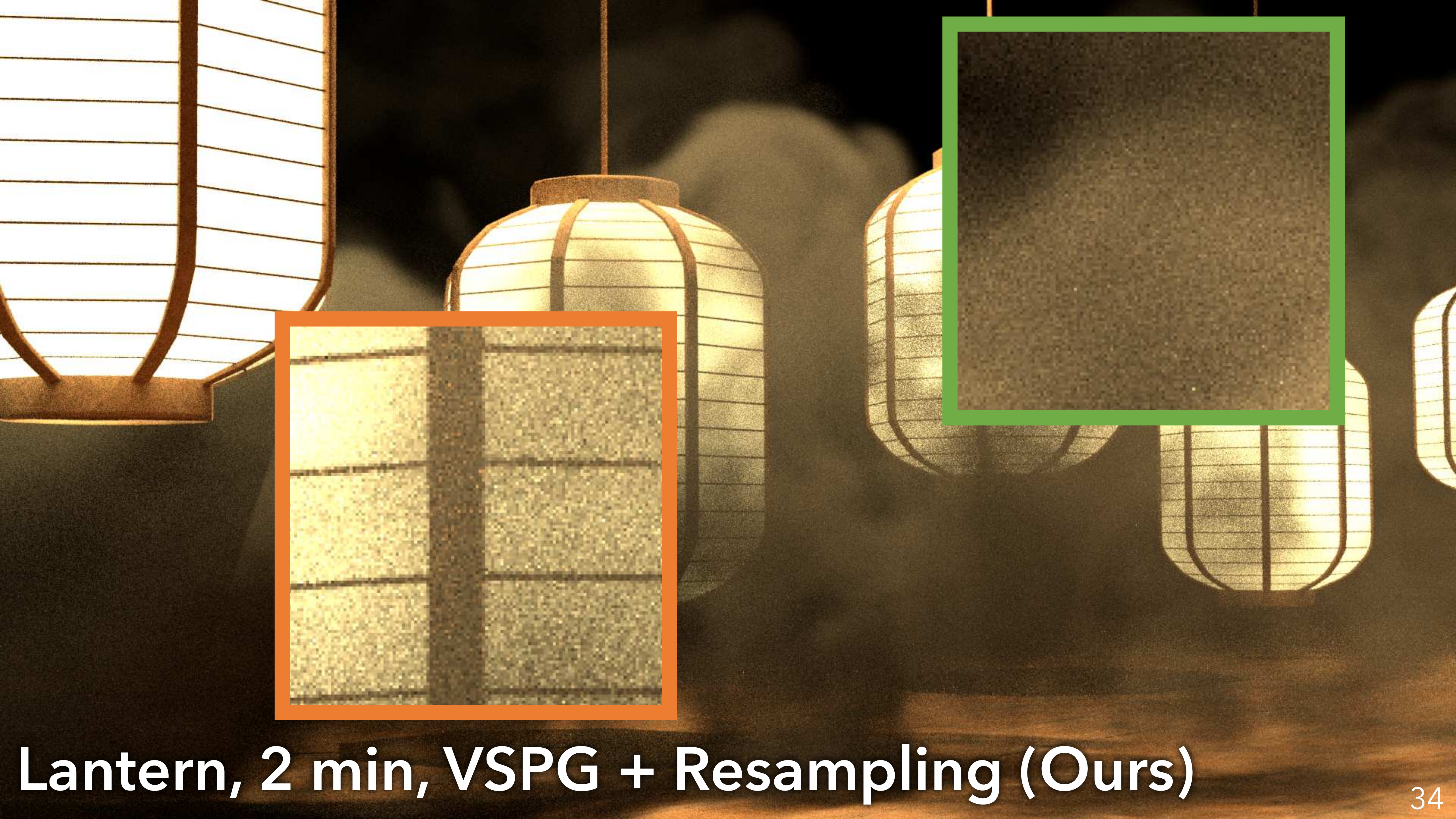
- Equal time
- Directional guiding enabled
- Distance sampling methods:
  - Transmittance-based P_{vol}
  - The VSP Guiding framework (**Ours**):
    - NDS
    - Resampling (**Ours**)
      - MIS: 0.75

Lantern, 2 min, Tr-based

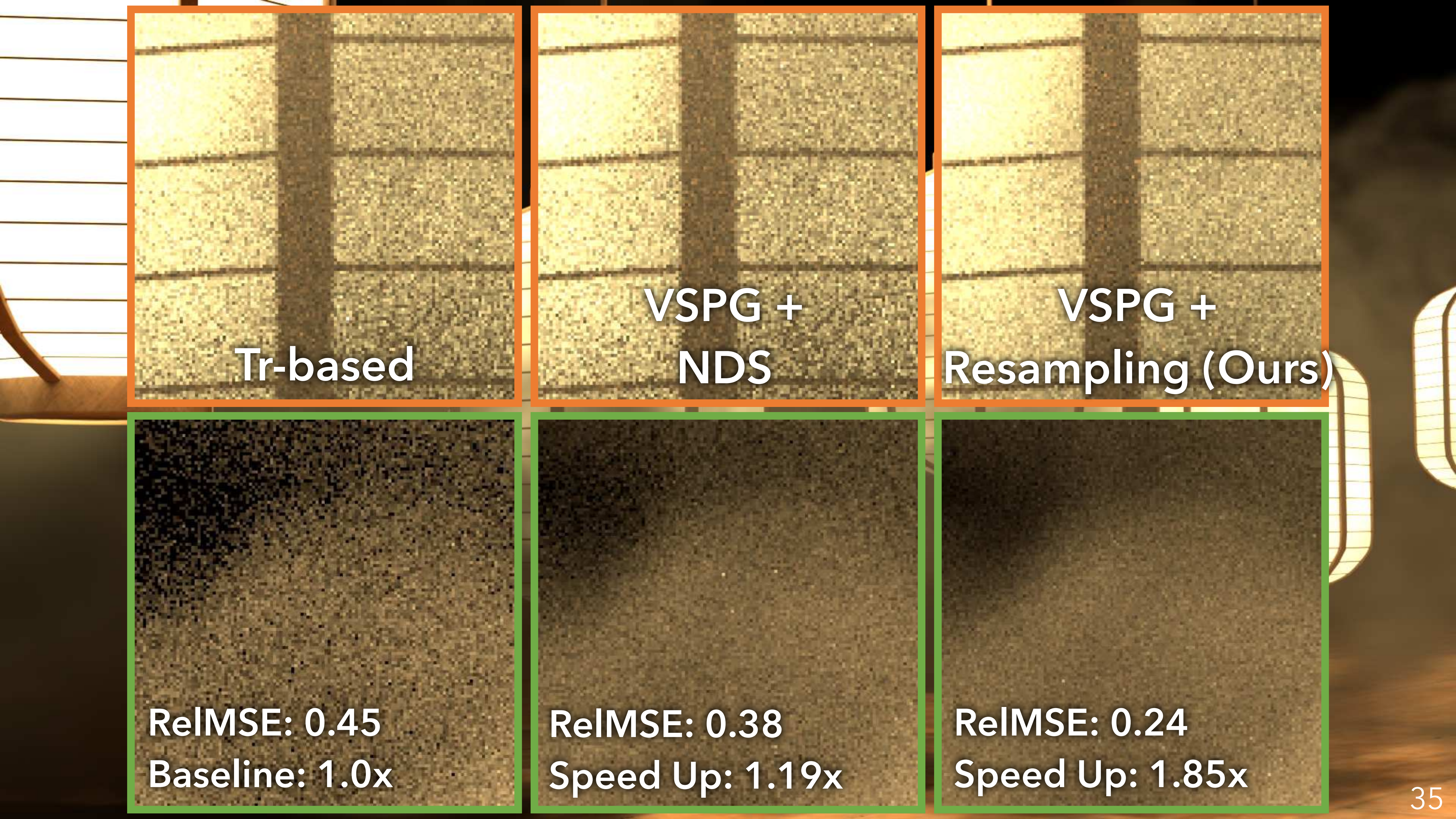
Achieved P_{vol}

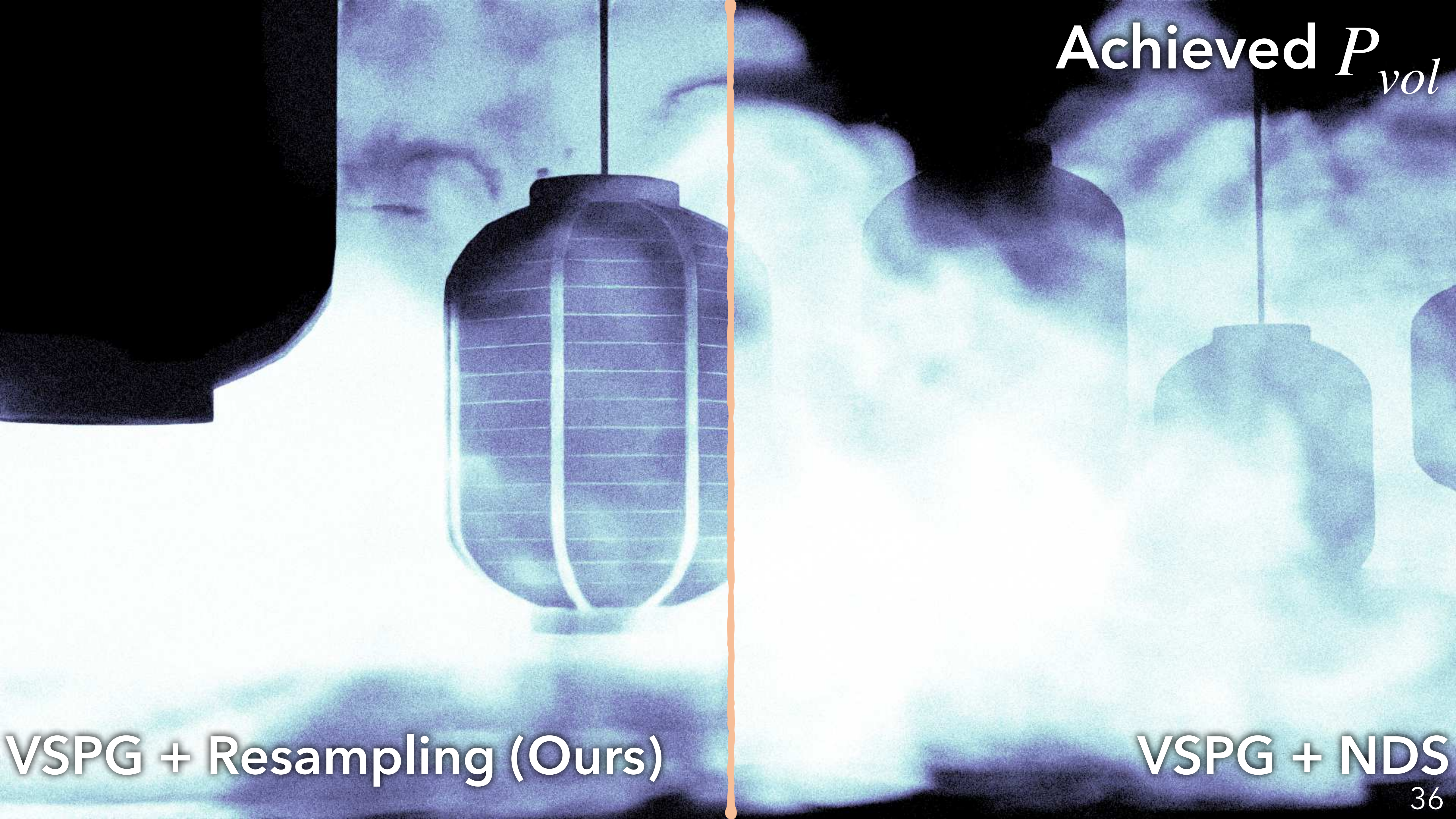

Achieved P_{vol}
Lantern, 2 min, VSPG + Resampling (Ours)


Lantern, 2 min, Tr-based



Lantern, 2 min, VSPG + Resampling (Ours)


Tr-based
VSPG +
NDS
VSPG +
Resampling (Ours)
RelMSE: 0.45
Baseline: 1.0x
RelMSE: 0.38
Speed Up: 1.19x
RelMSE: 0.24
Speed Up: 1.85x


Achieved P_{vol}
VSPG + Resampling (Ours)
VSPG + NDS

Average speedup: **2.25x**

Summary

- **Key insight:** explicitly controlling P_{vol} can improve efficiency
- A practical framework for:
  1. Computing the optimal P_{vol}
  2. Achieve precise control over P_{vol}

✓ Unbiased

✓ Easy to implement & minimal overhead

- Perfect combination with directional guiding!

✓ Fully automatic, no user parameter

Volume Scattering Probability Guiding

Thank you!

intel OPENPGL

- VSPG framework included in upcoming **v0.8.0** release

Project Page

- Interactive viewer
- PBRT source code (**soon**)

Backup Slides

Related Work

	Efficient in heterogenous volumes	Automatic Target VSP	Arbitrary Target VSP function	Reach Target VSP	Increase VSP	Decrease VSP
Zero-Variance Volume Path Guiding	✗	✓	✗	✓	✓	✓
Normalized Distance Sampling (NDS)	✓	✗	✓	✗	✓	✗
Resampling (Ours)	✓	✓	✓	✓	✓	✓

Contribution vs Variance VSP

The Jungle Scene

Contribution-based (1st Moment)

Variance-based (2nd Moment)

Landscape, 32 spp, Contribution-based VSPG

Landscape, 32 spp, Variance-based VSPG

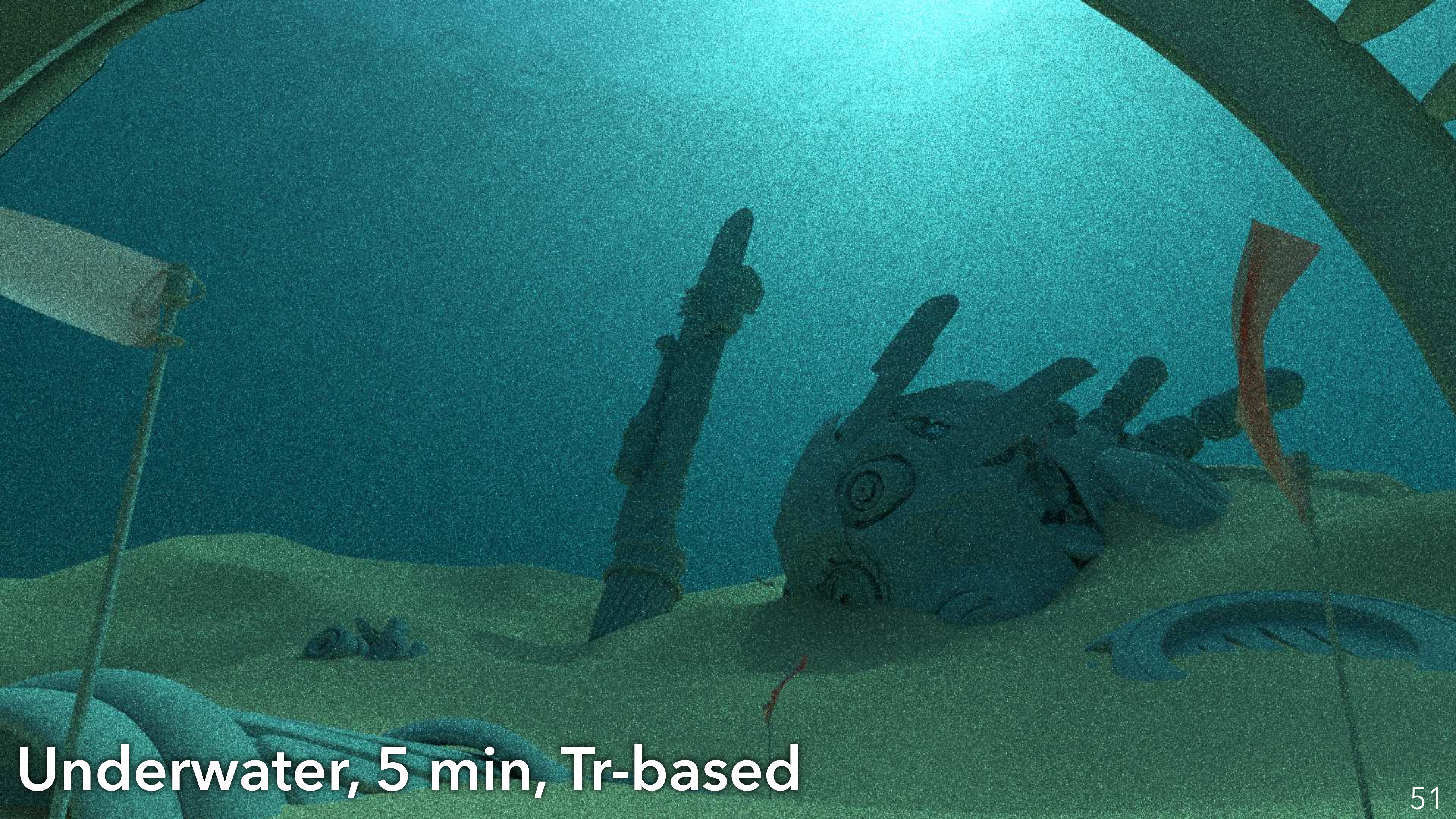
Target (contribution-based) P^{1st}_{vol}


Target (variance-based) P^{2nd}_{vol}

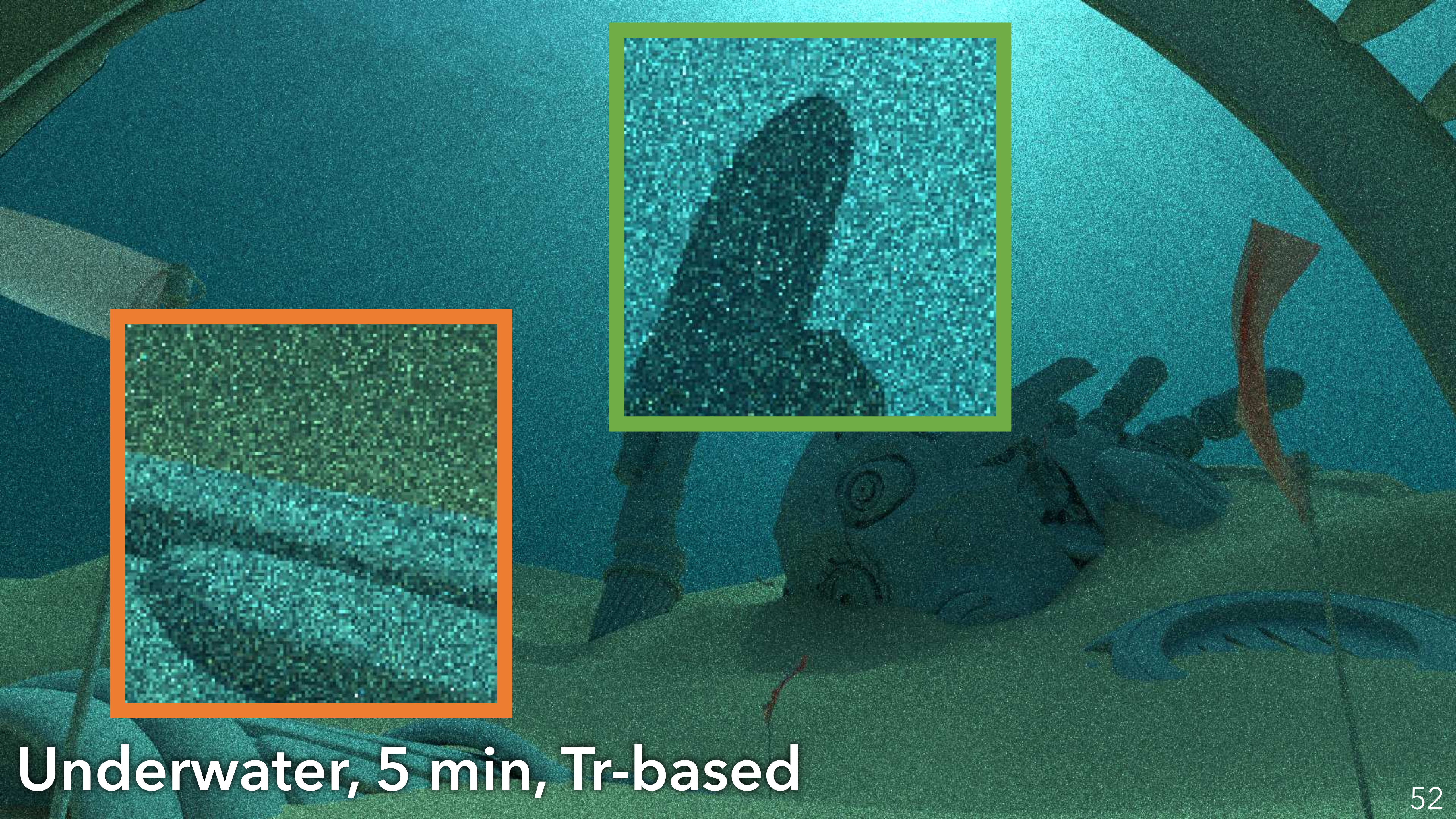

Achieved (contribution-based) P^{1st}_{vol}


Achieved (variance-based) P^{2nd}_{vol}

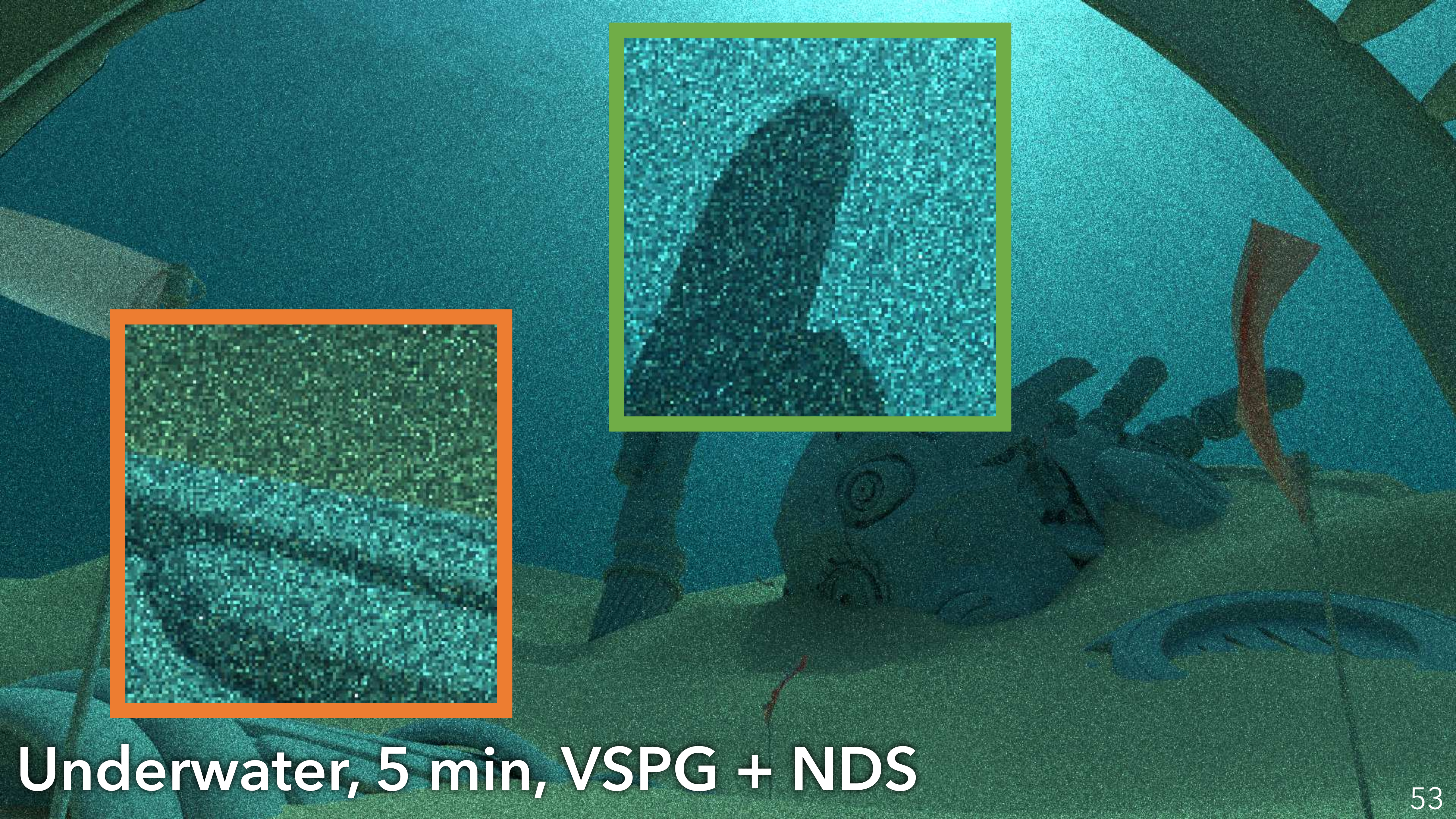
Additional Evaluation

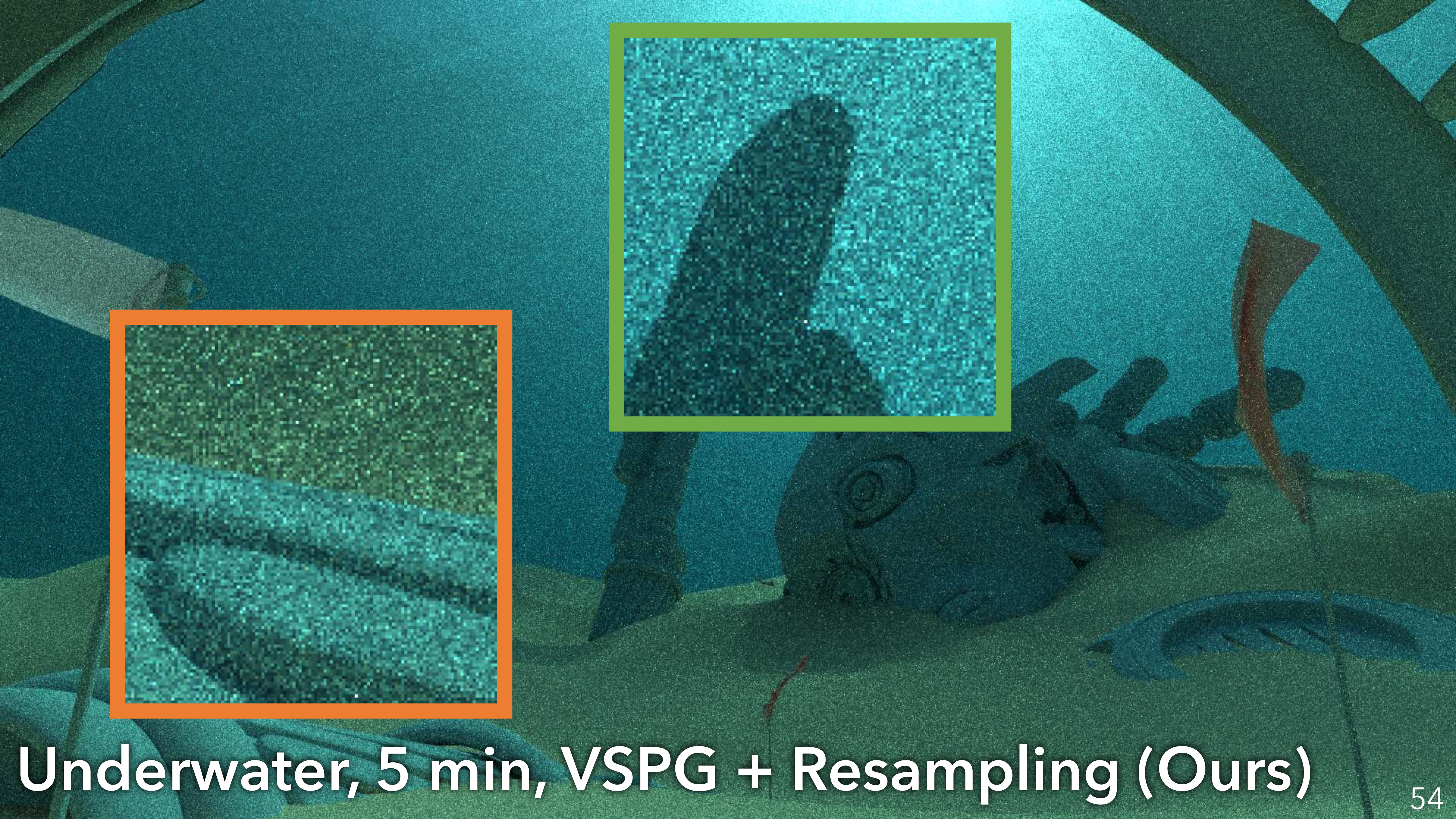


Underwater, 5 min, Tr-based

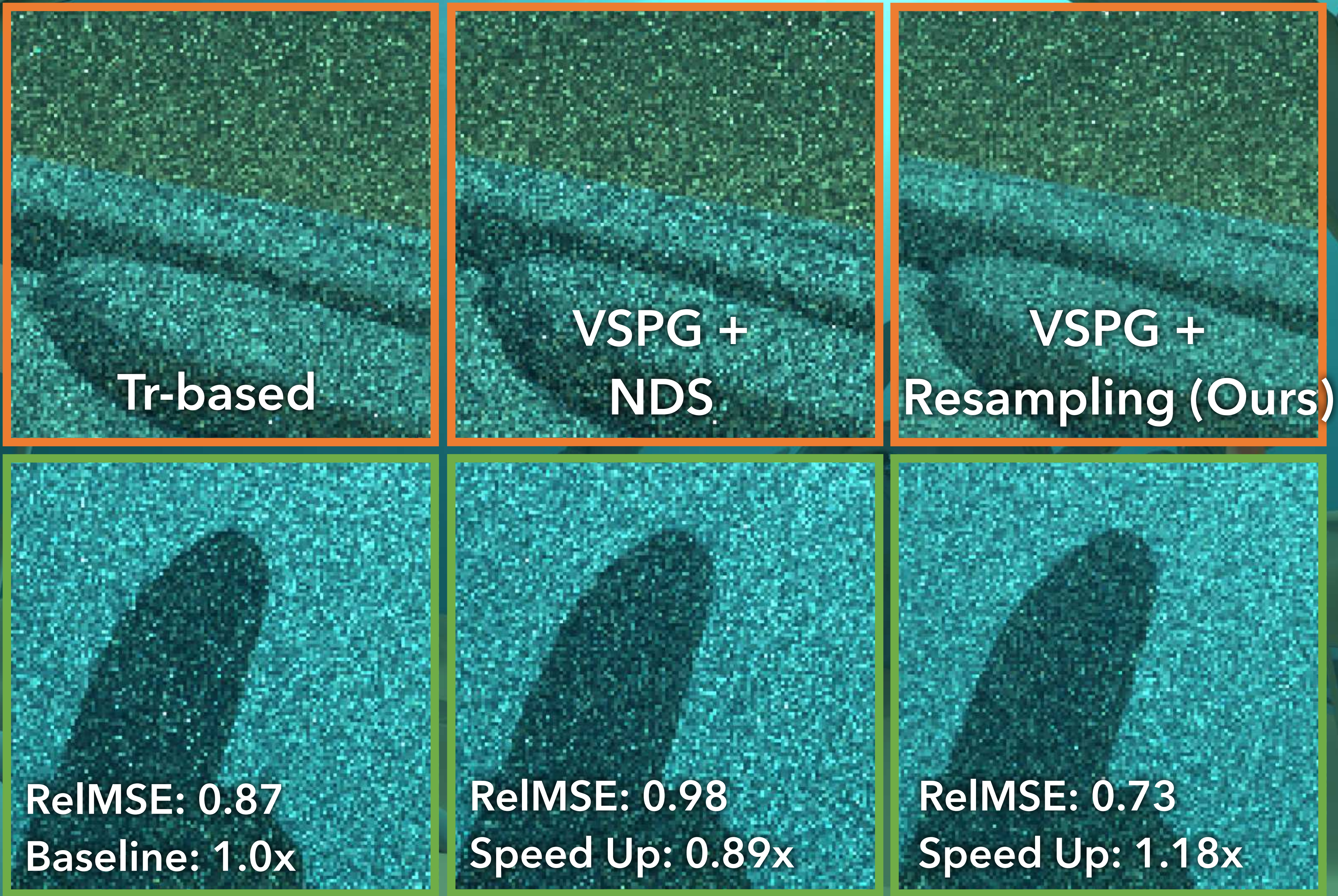


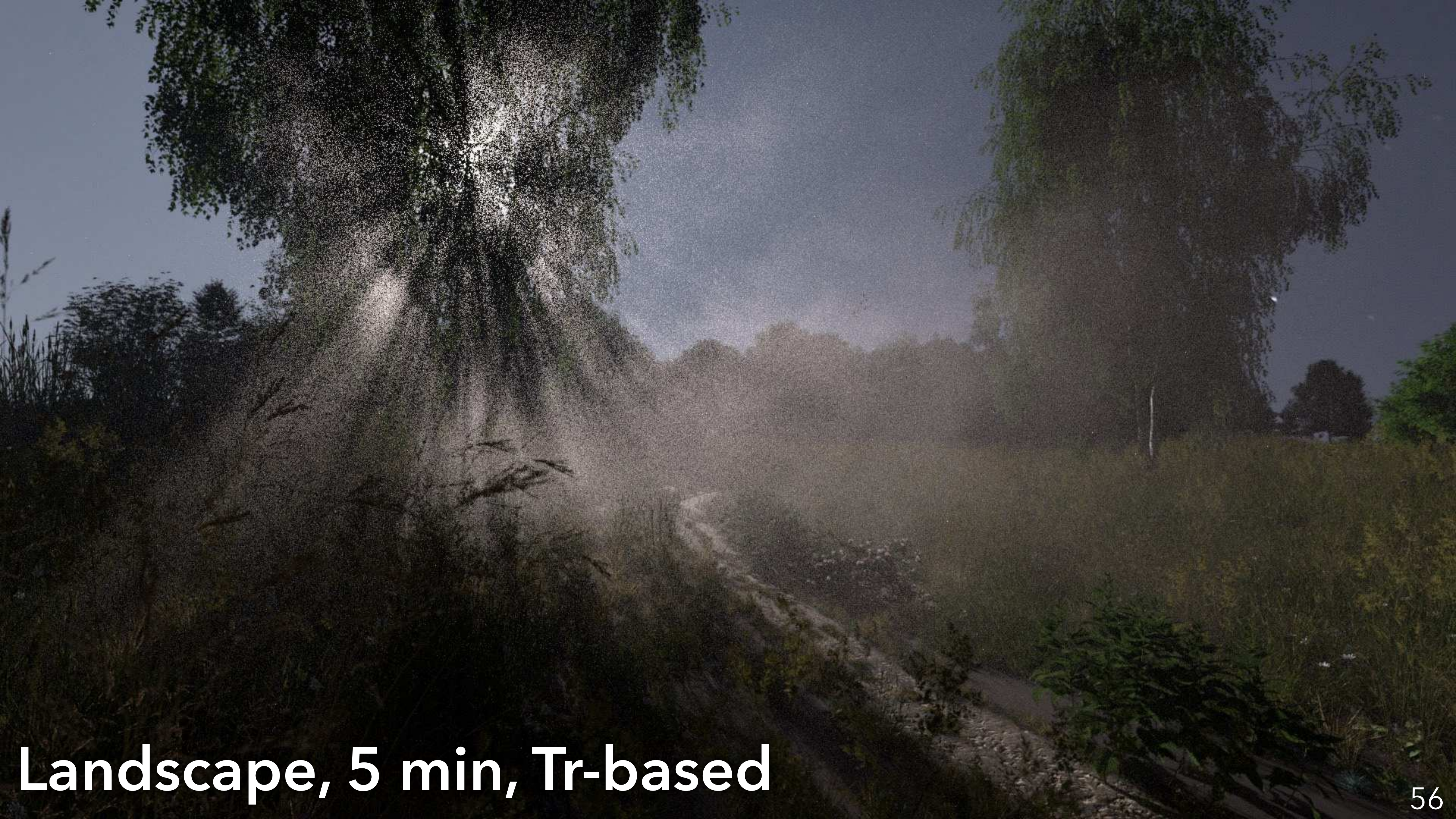
Underwater, 5 min, Tr-based


Underwater, 5 min, VSPG + NDS



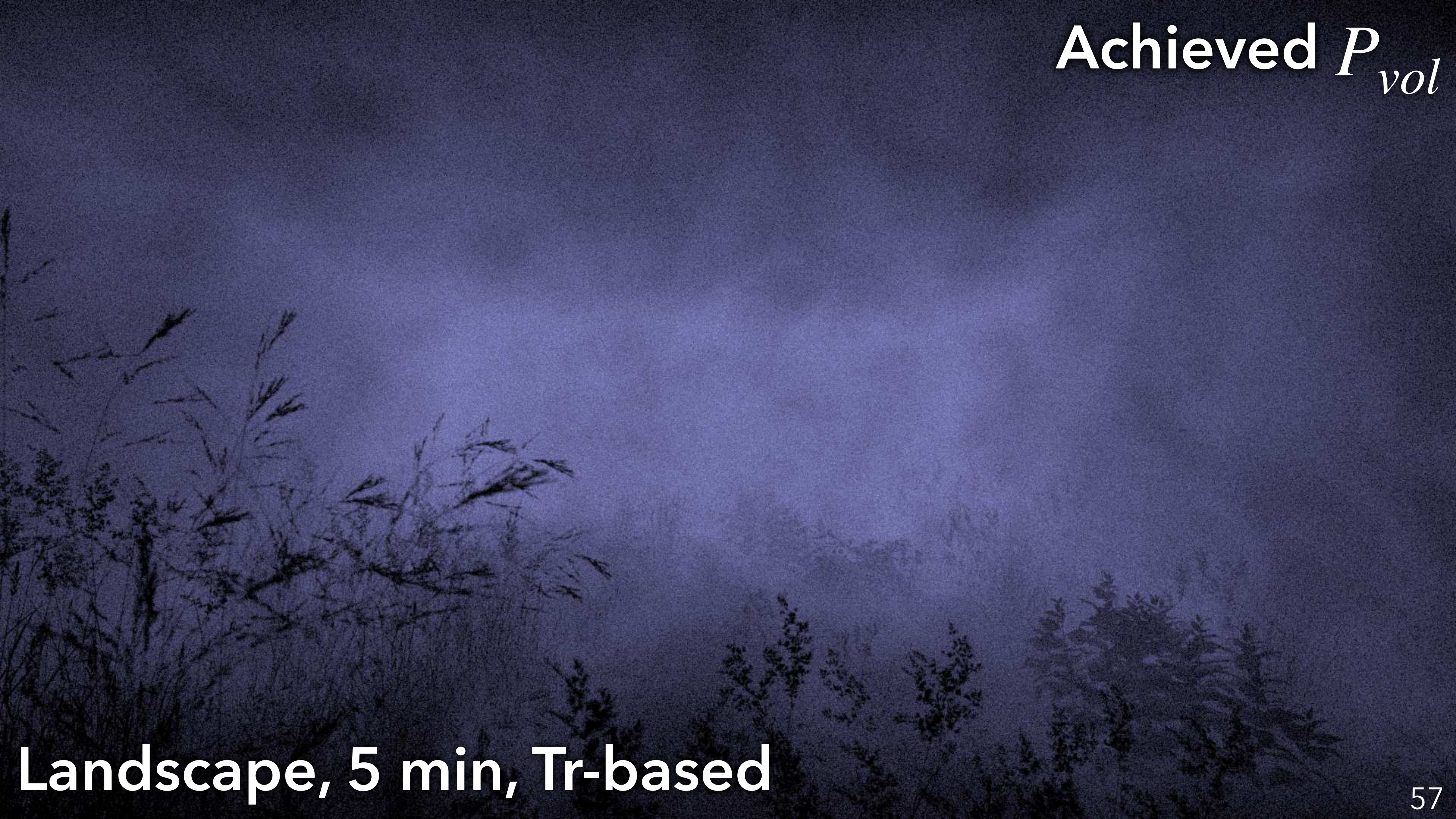
Underwater, 5 min, VSPG + Resampling (Ours)


Tr-based
VSPG +
NDS
VSPG +
Resampling (Ours)
RelMSE: 0.87
Baseline: 1.0x
RelMSE: 0.98
Speed Up: 0.89x
RelMSE: 0.73
Speed Up: 1.18x

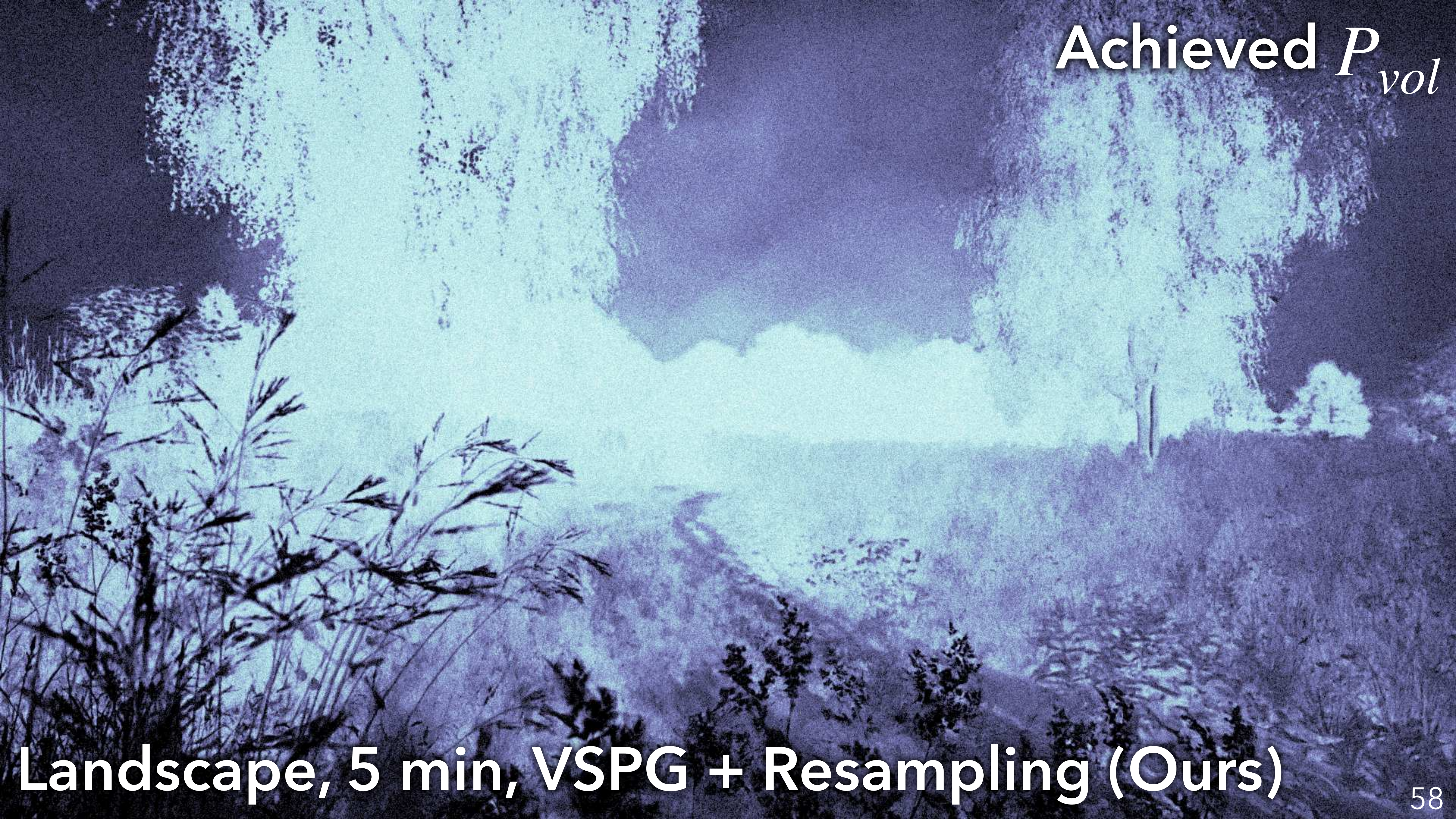


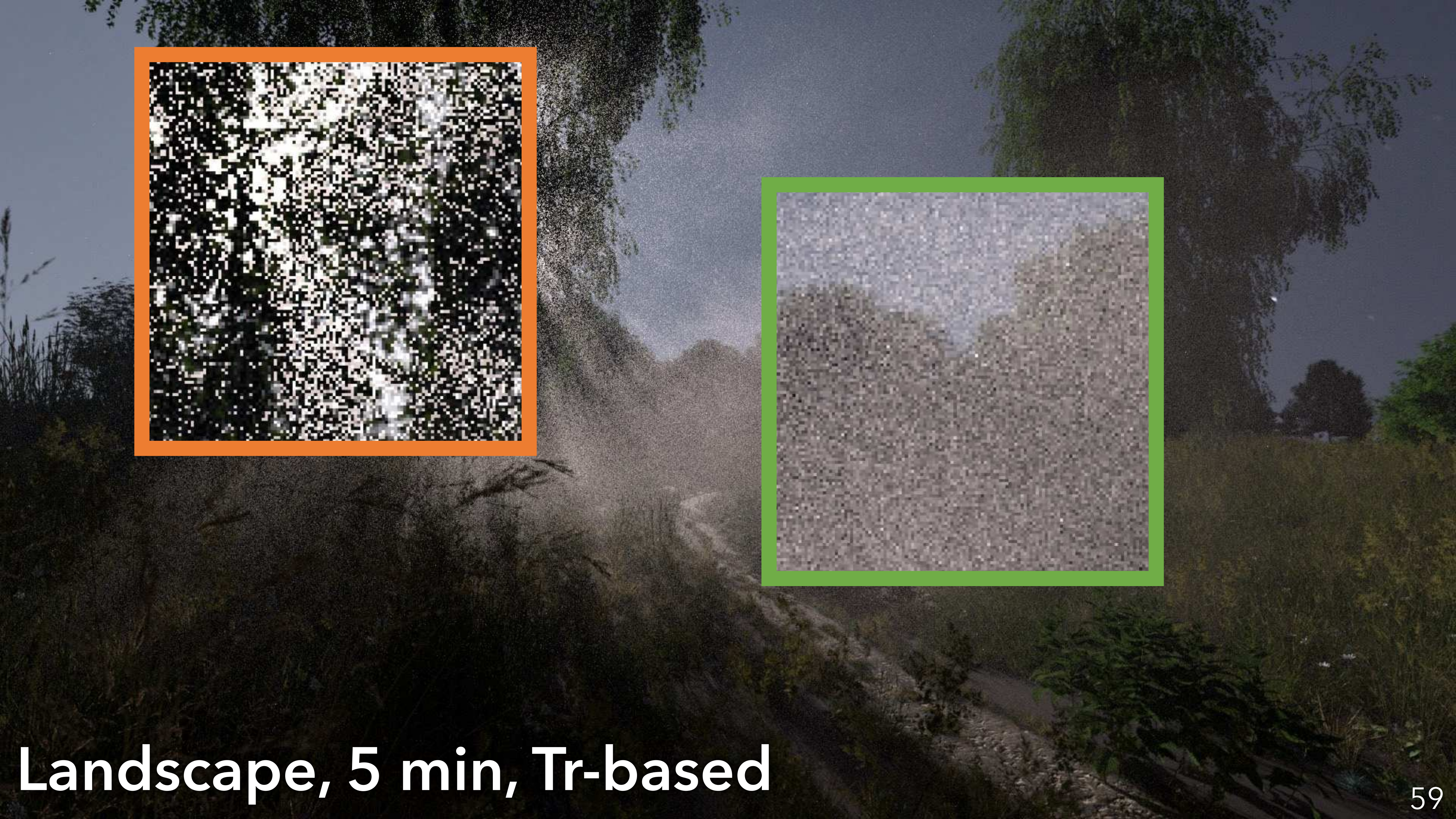
Landscape, 5 min, Tr-based

Achieved P_{vol}



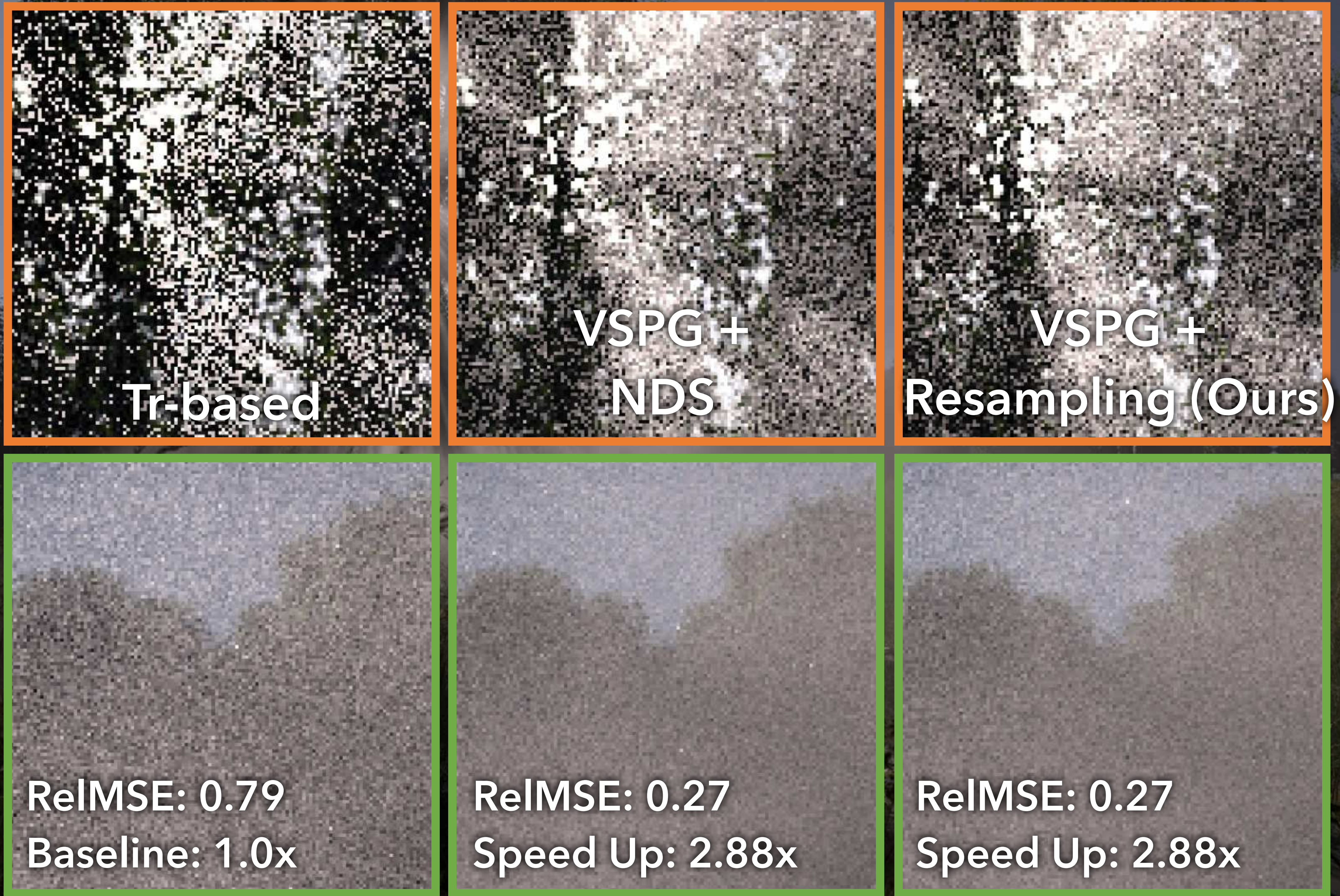
Landscape, 5 min, Tr-based

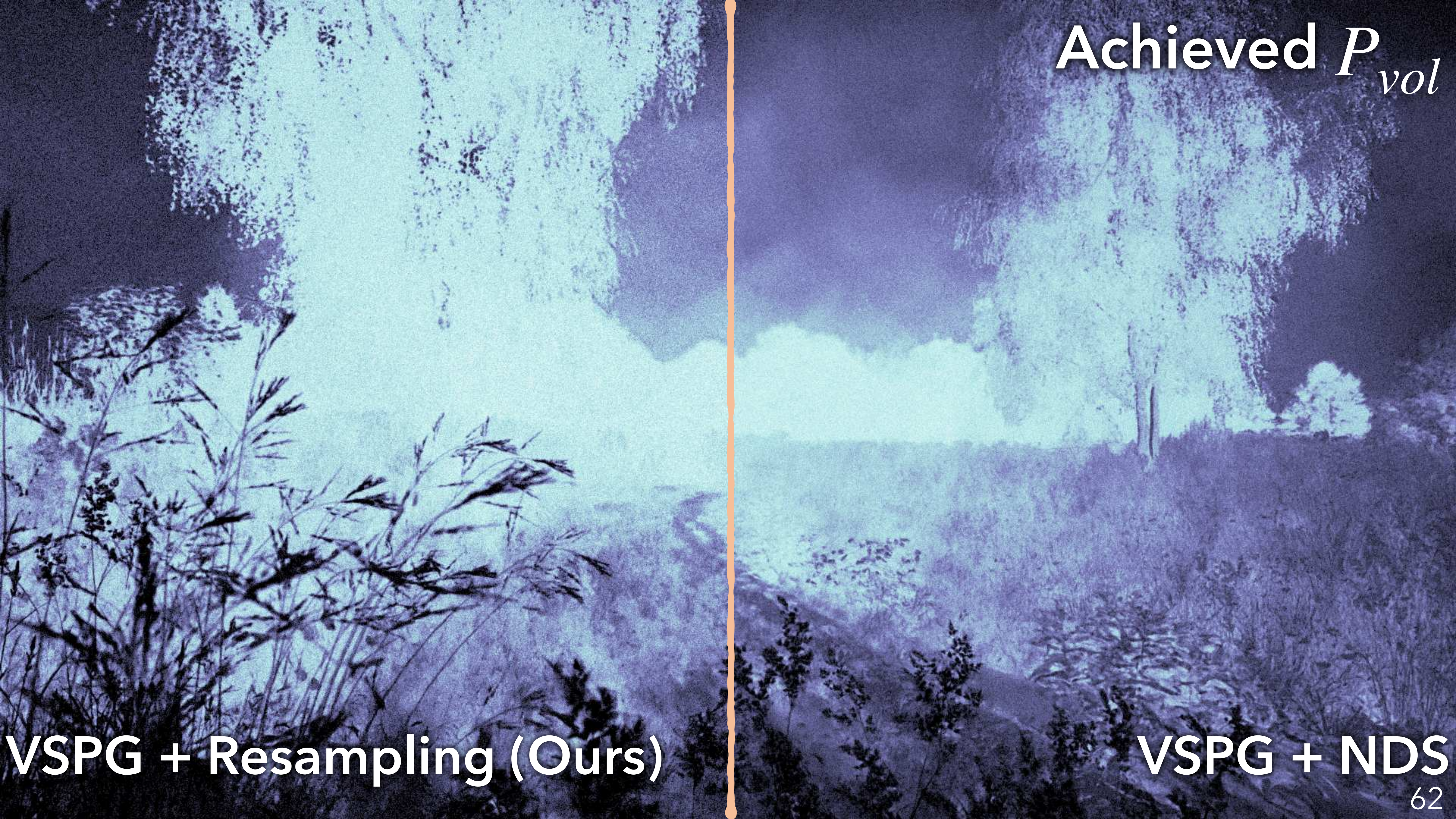

Achieved P_{vol}
Landscape, 5 min, VSPG + Resampling (Ours)



Landscape, 5 min, Tr-based

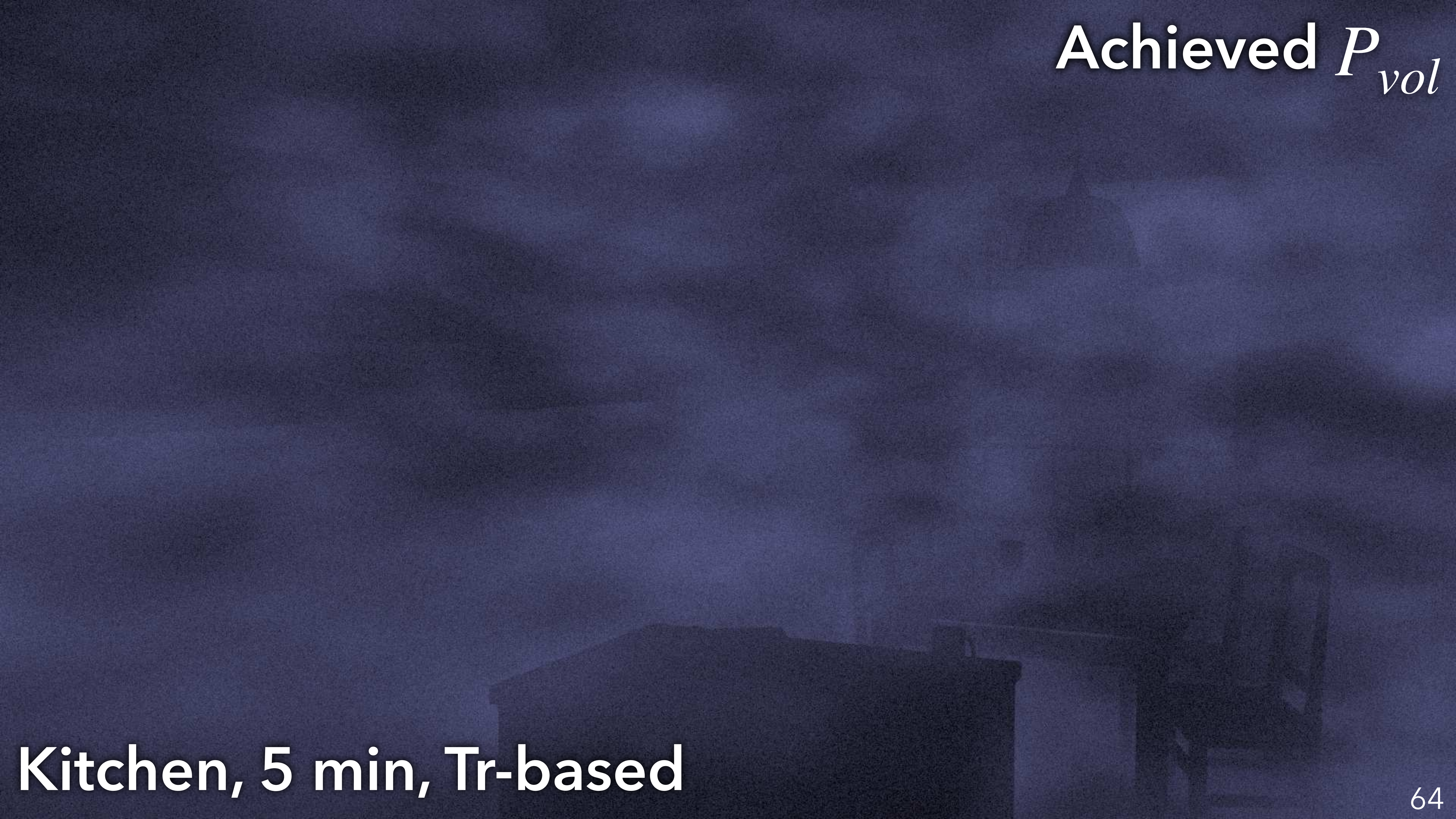
Landscape, 5 min, VSPG + Resampling (Ours)


Tr-based
VSPG +
NDS
VSPG +
Resampling (Ours)
RelMSE: 0.79
Baseline: 1.0x
RelMSE: 0.27
Speed Up: 2.88x
RelMSE: 0.27
Speed Up: 2.88x


Achieved P_{vol}
VSPG + Resampling (Ours)
VSPG + NDS

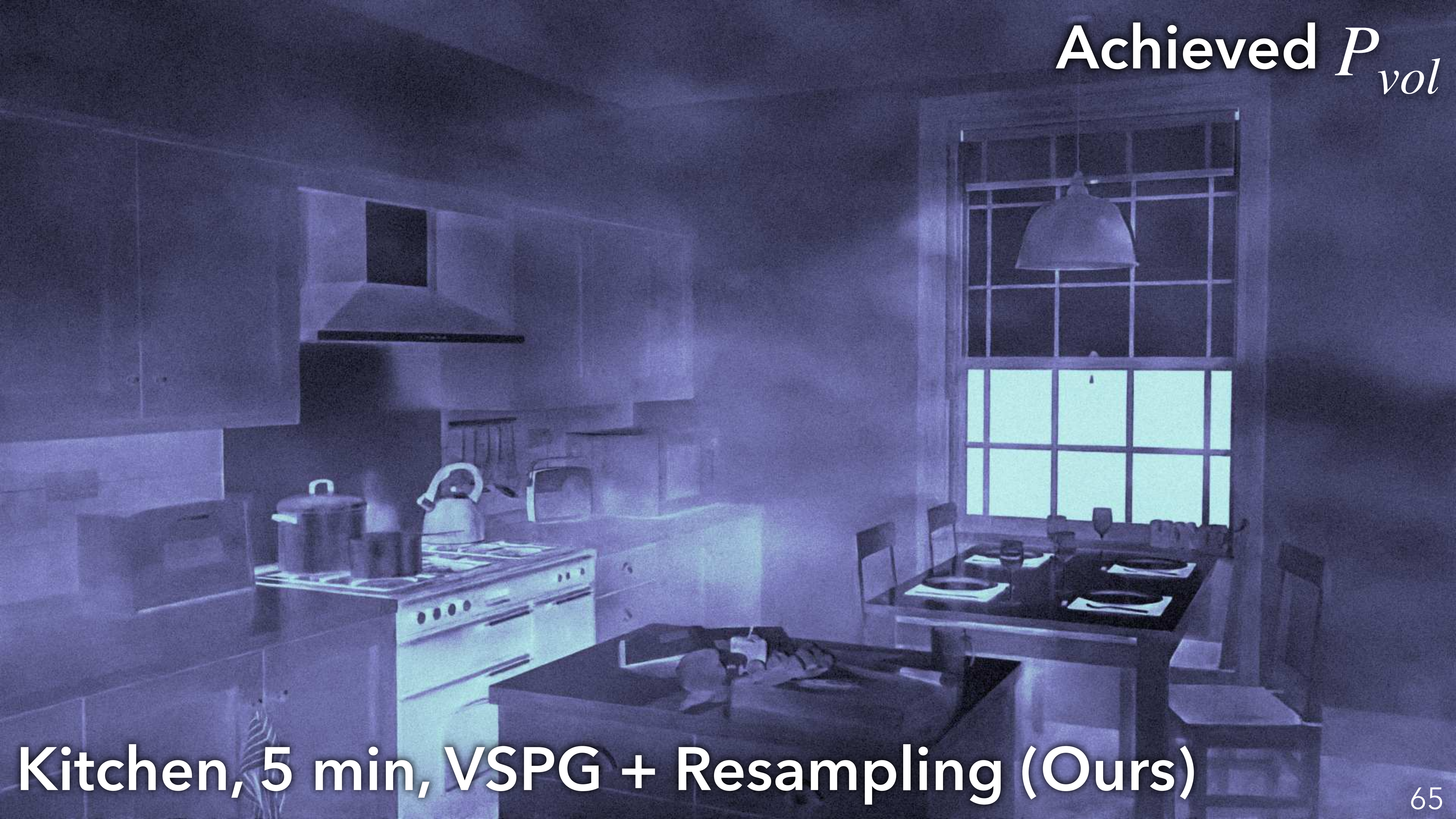
Kitchen, 5 min, Tr-based

Achieved P_{vol}

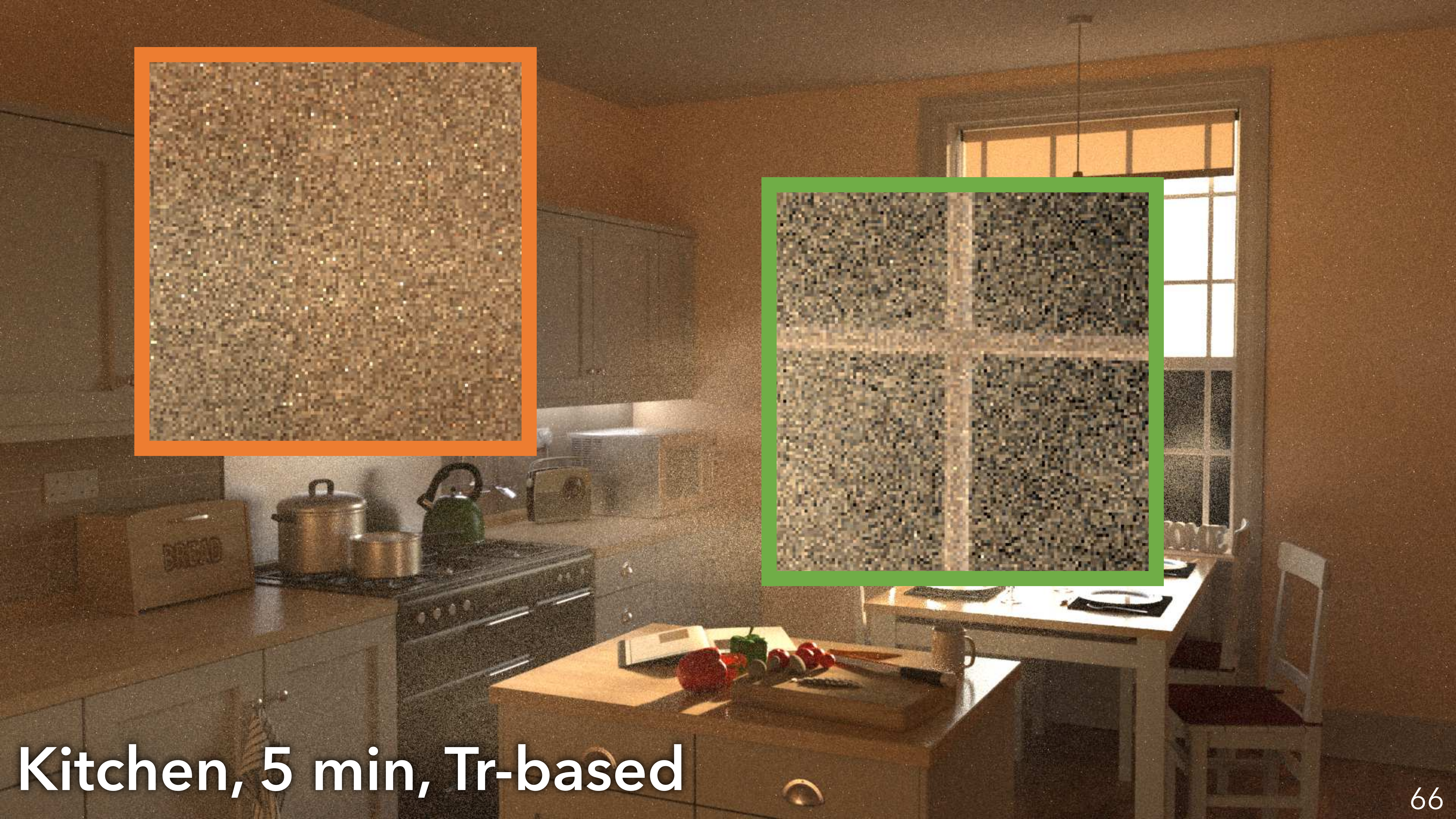


Kitchen, 5 min, Tr-based

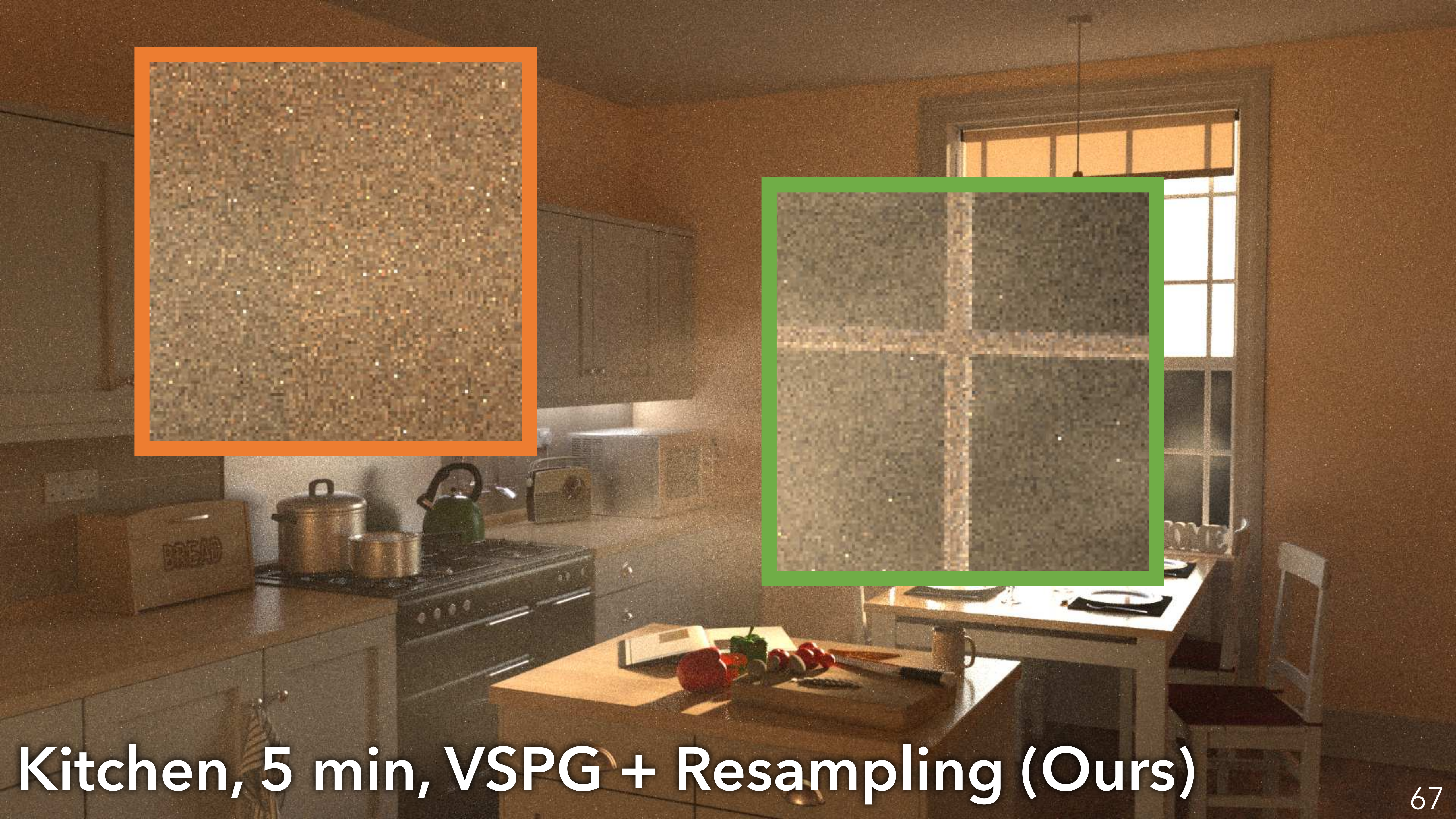
Achieved P_{vol}



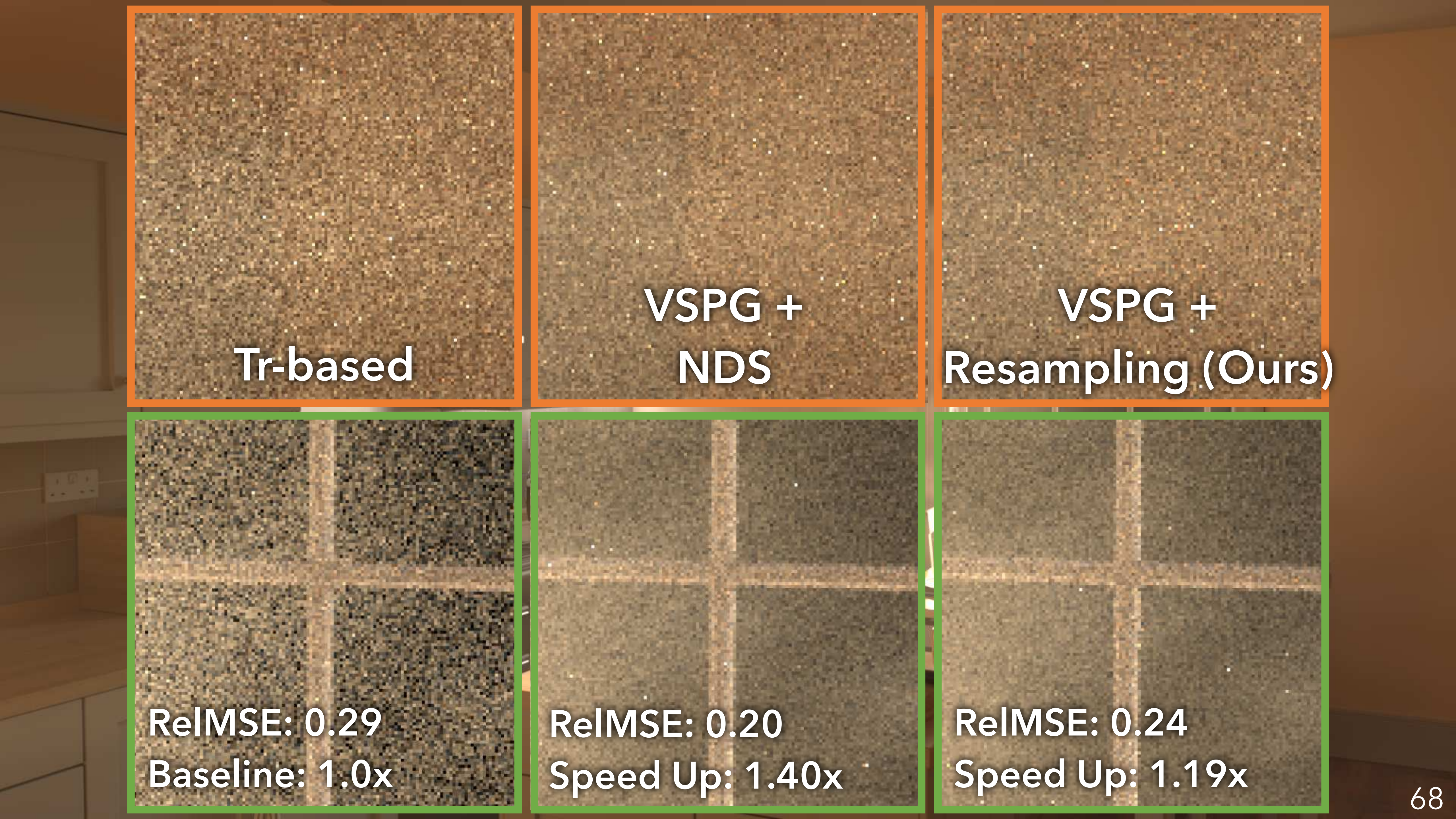
Kitchen, 5 min, VSPG + Resampling (Ours)



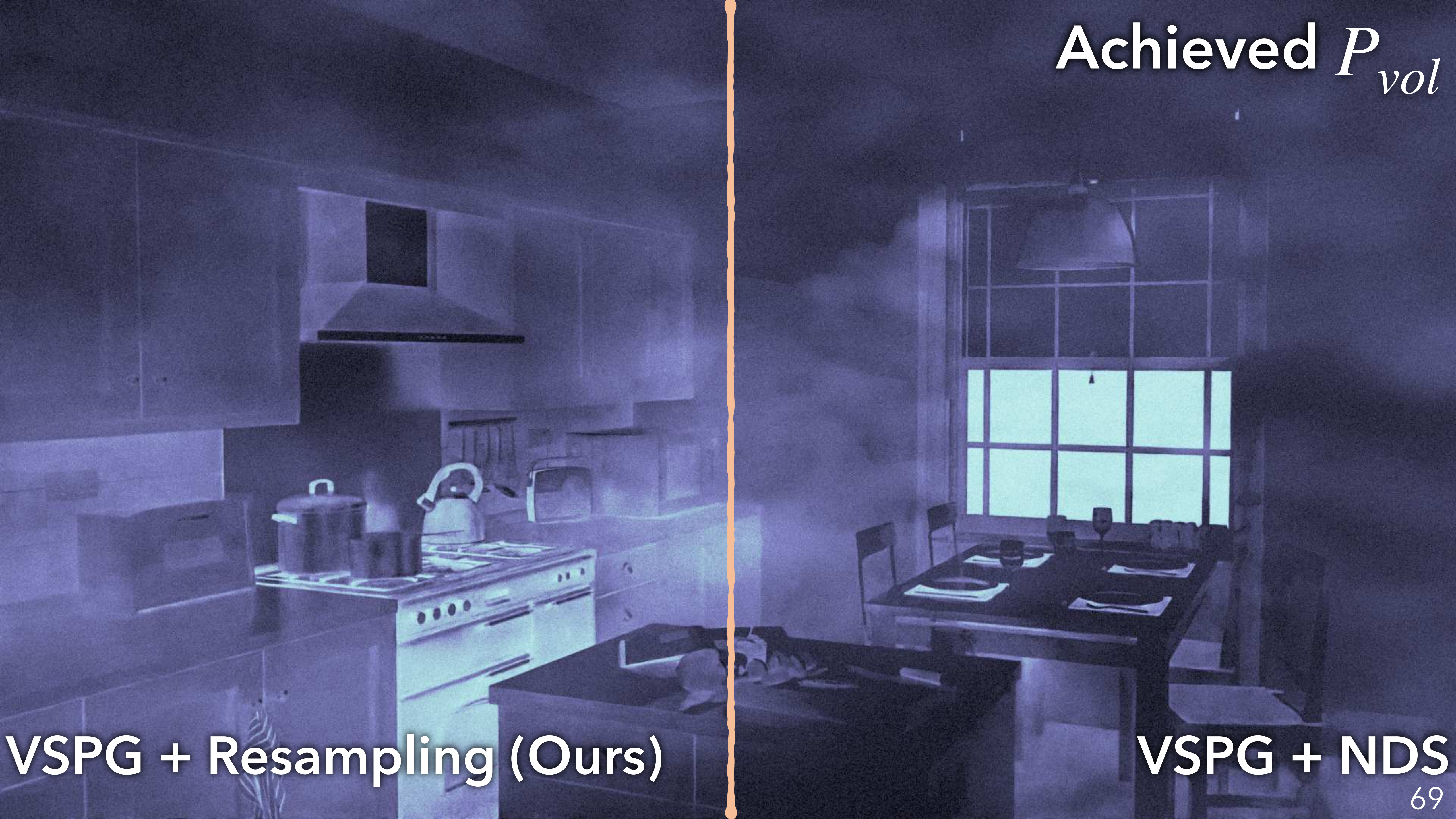
Kitchen, 5 min, Tr-based



Kitchen, 5 min, VSPG + Resampling (Ours)

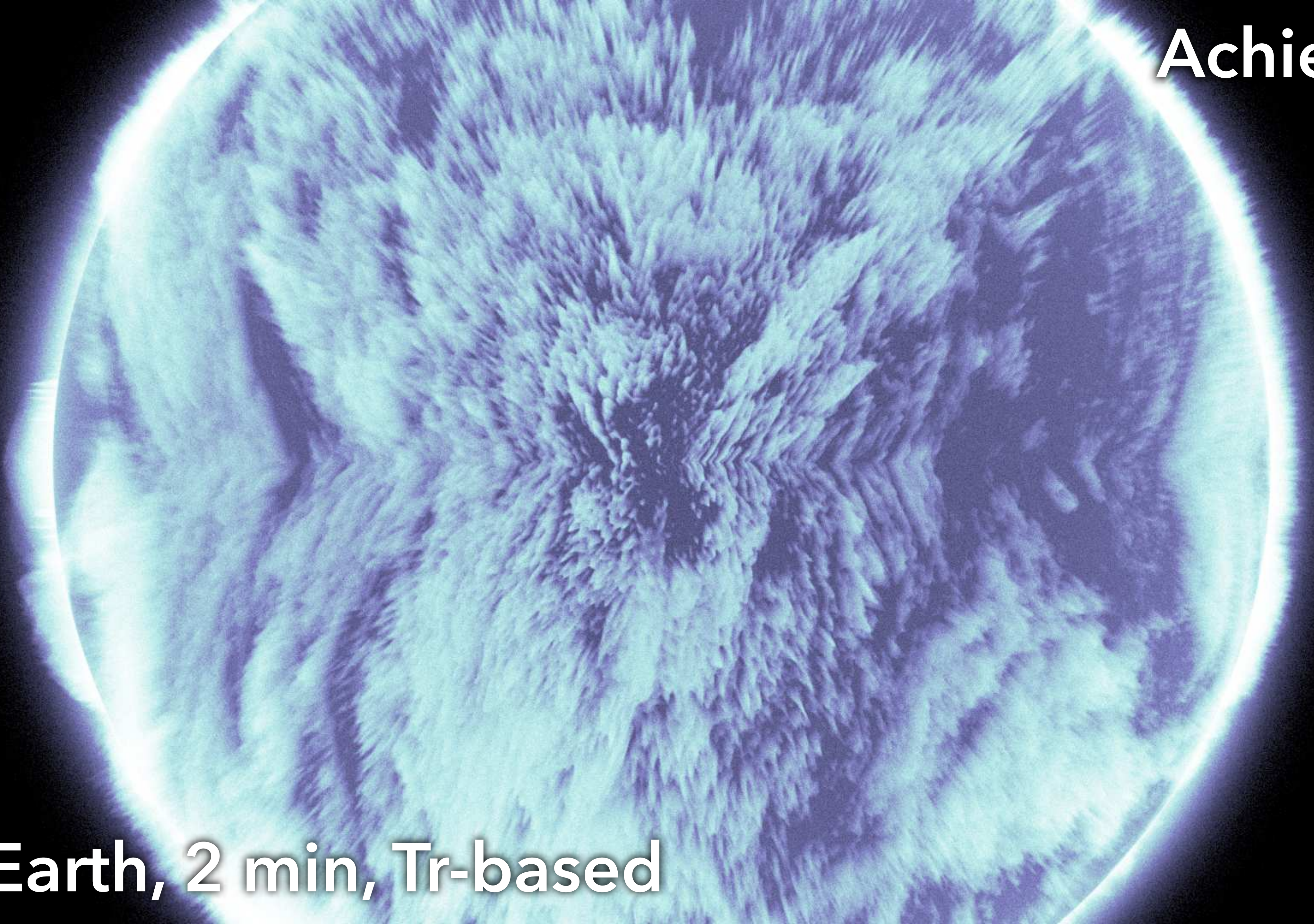


Tr-based
VSPG +
NDS
VSPG +
Resampling (Ours)
RelMSE: 0.29
Baseline: 1.0x
RelMSE: 0.20
Speed Up: 1.40x
RelMSE: 0.24
Speed Up: 1.19x

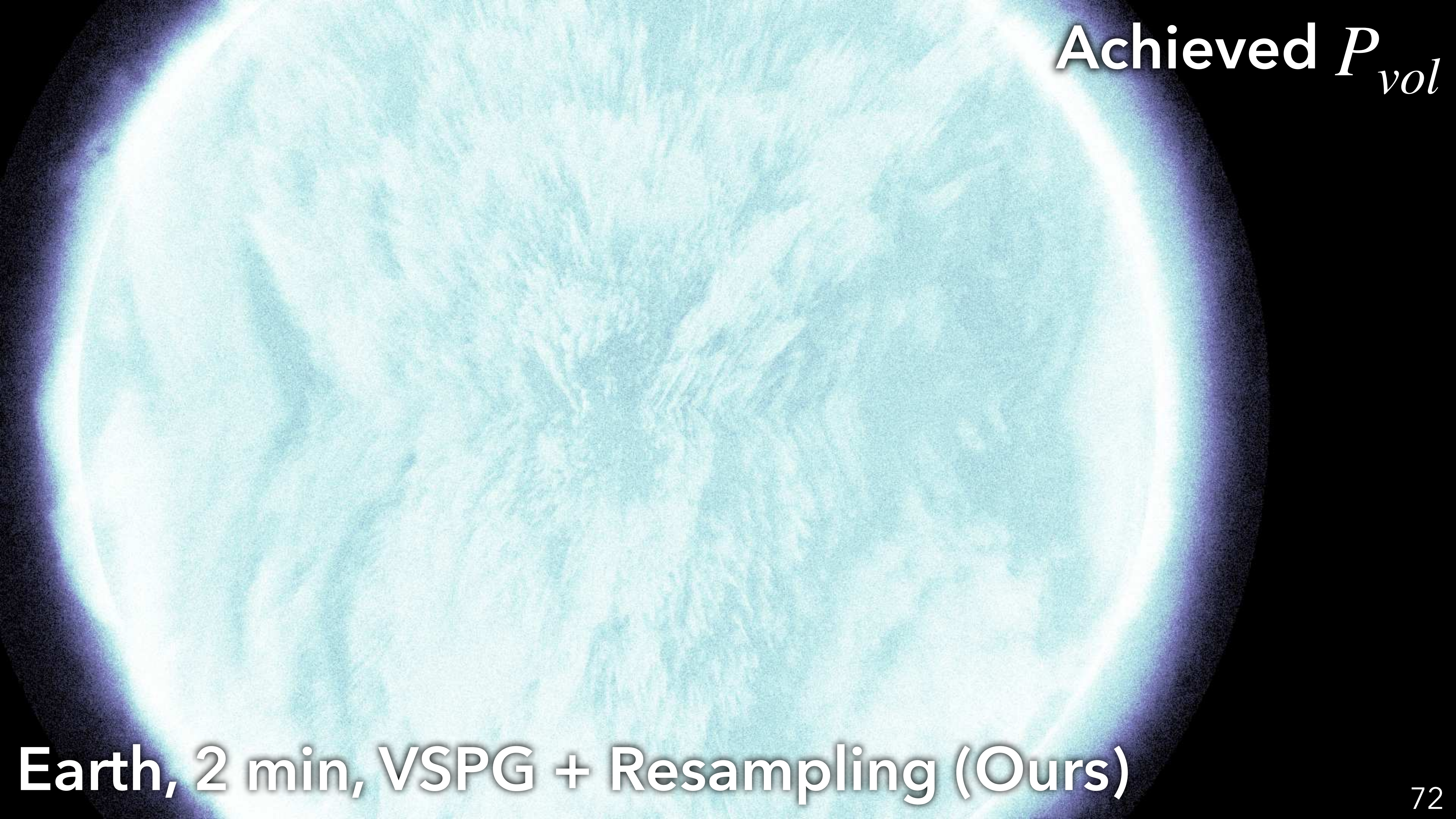

Achieved P_{vol}
VSPG + Resampling (Ours)
VSPG + NDS

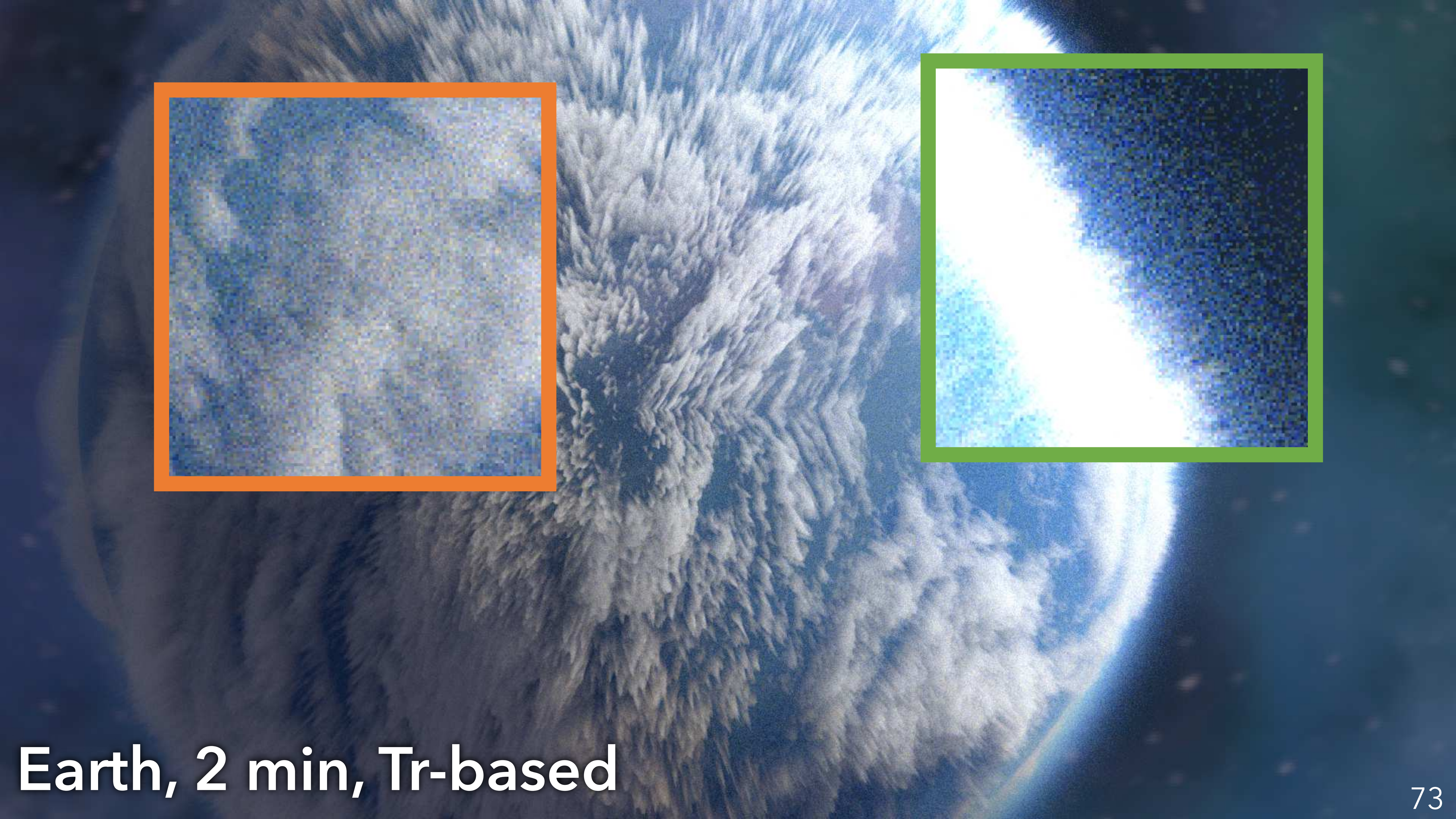

Earth, 2 min, Tr-based

Achieved P_{vol}

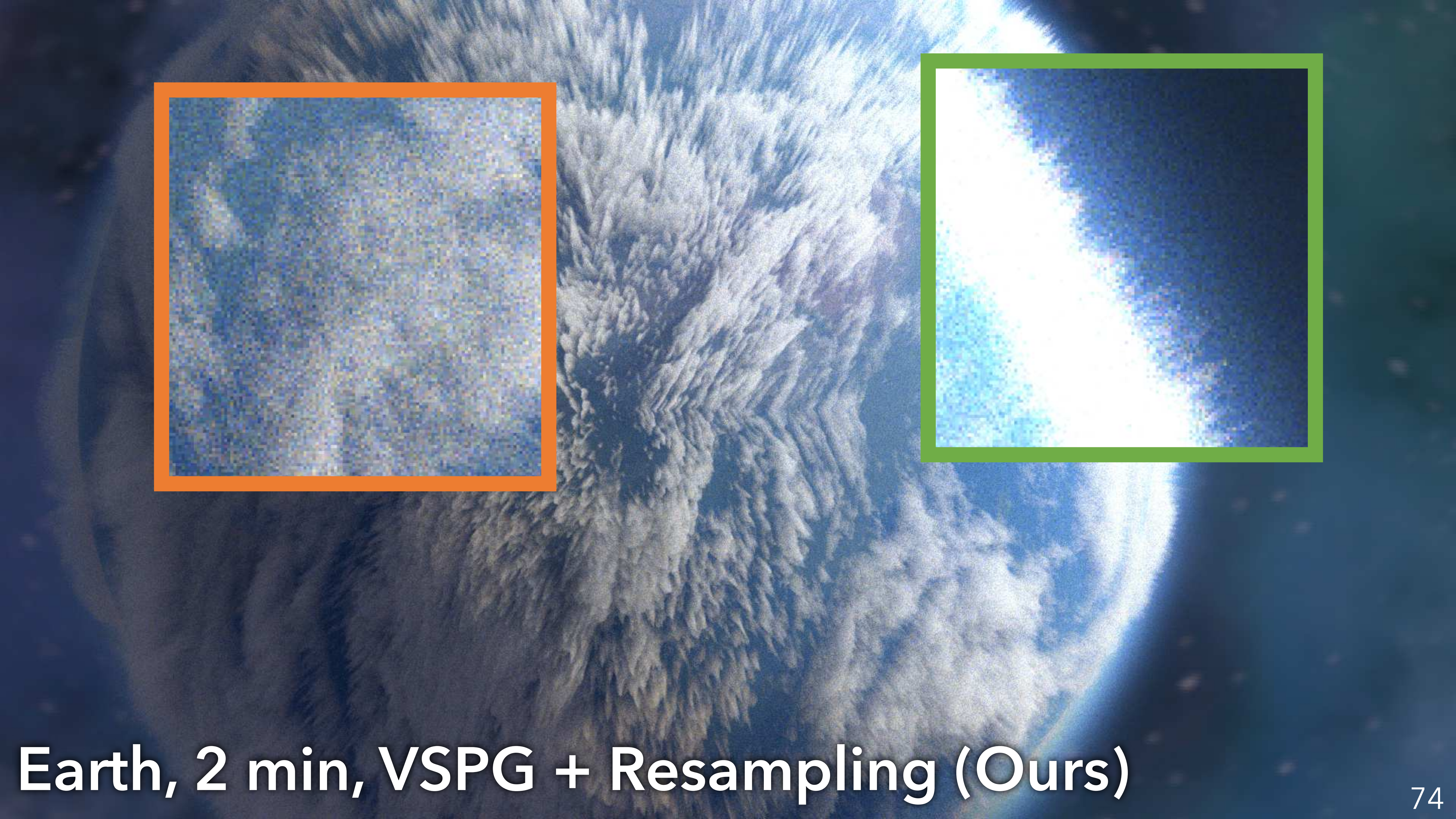


Earth, 2 min, Tr-based

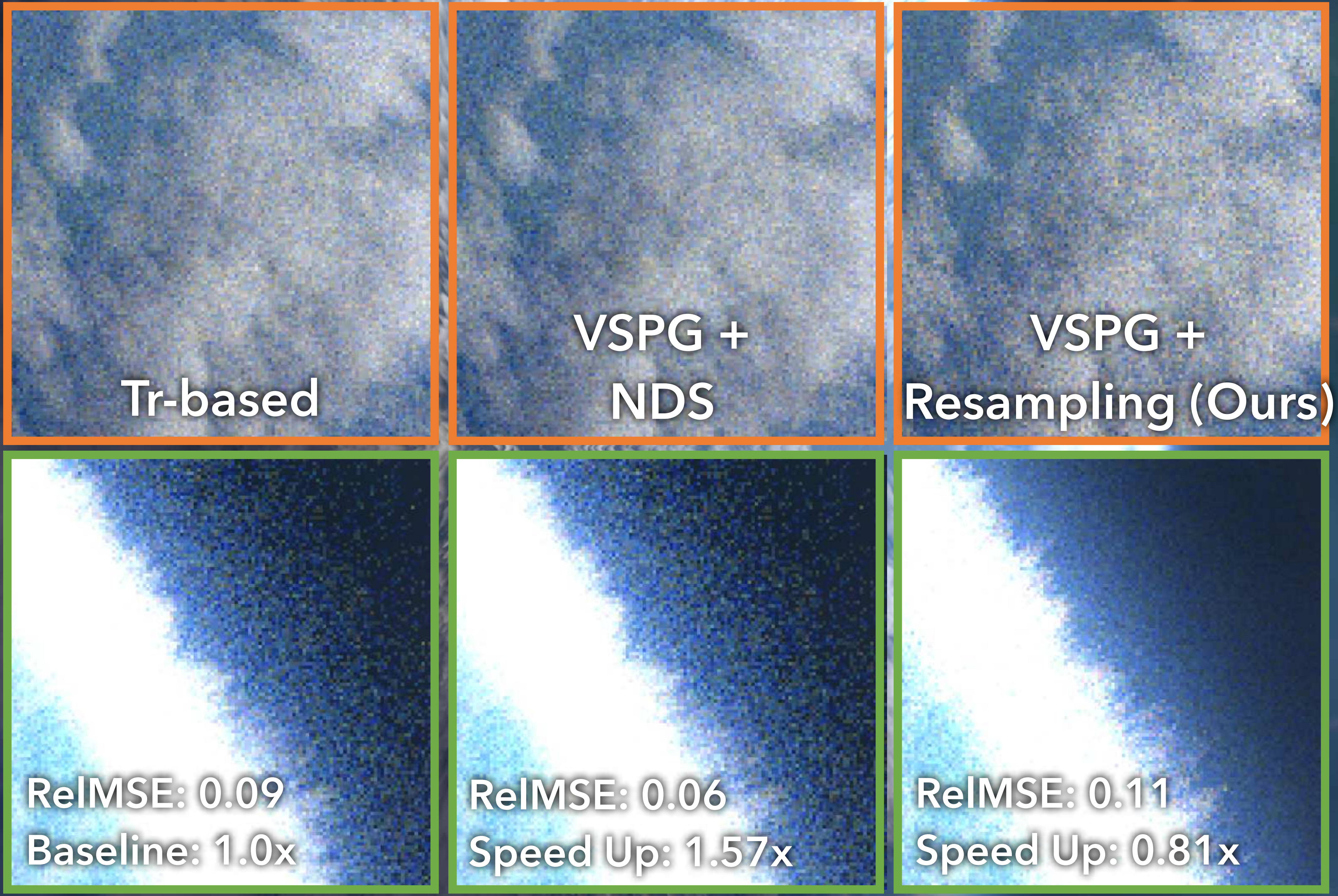

Achieved P_{vol}
Earth, 2 min, VSPG + Resampling (Ours)

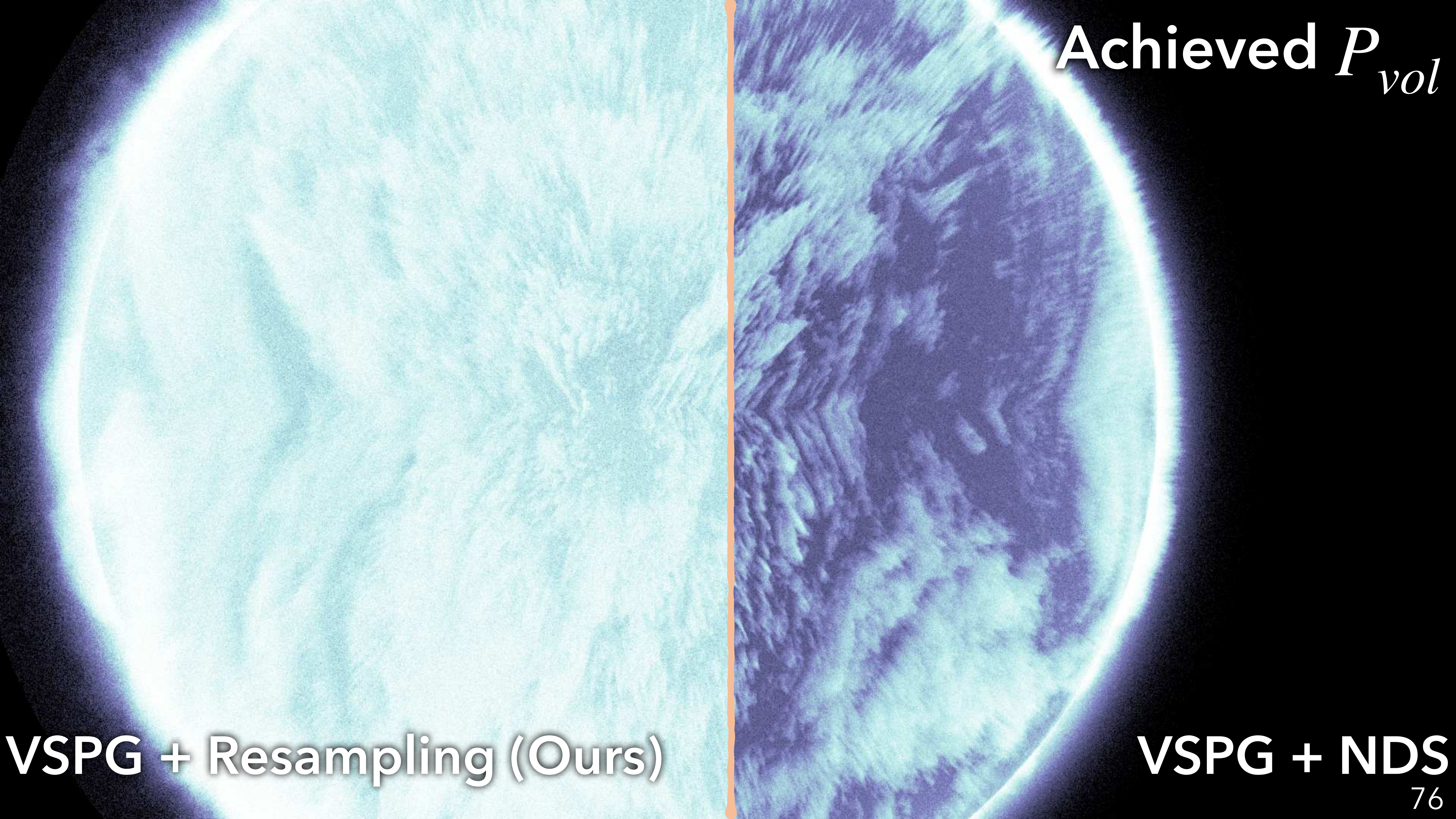


Earth, 2 min, Tr-based



Earth, 2 min, VSPG + Resampling (Ours)


Tr-based
VSPG + NDS
VSPG + Resampling (Ours)
RelMSE: 0.09
Baseline: 1.0x
RelMSE: 0.06
Speed Up: 1.57x
RelMSE: 0.11
Speed Up: 0.81x



Achieved P_{vol}
VSPG + Resampling (Ours)
VSPG + NDS

Jungle, 5 min, Tr-based

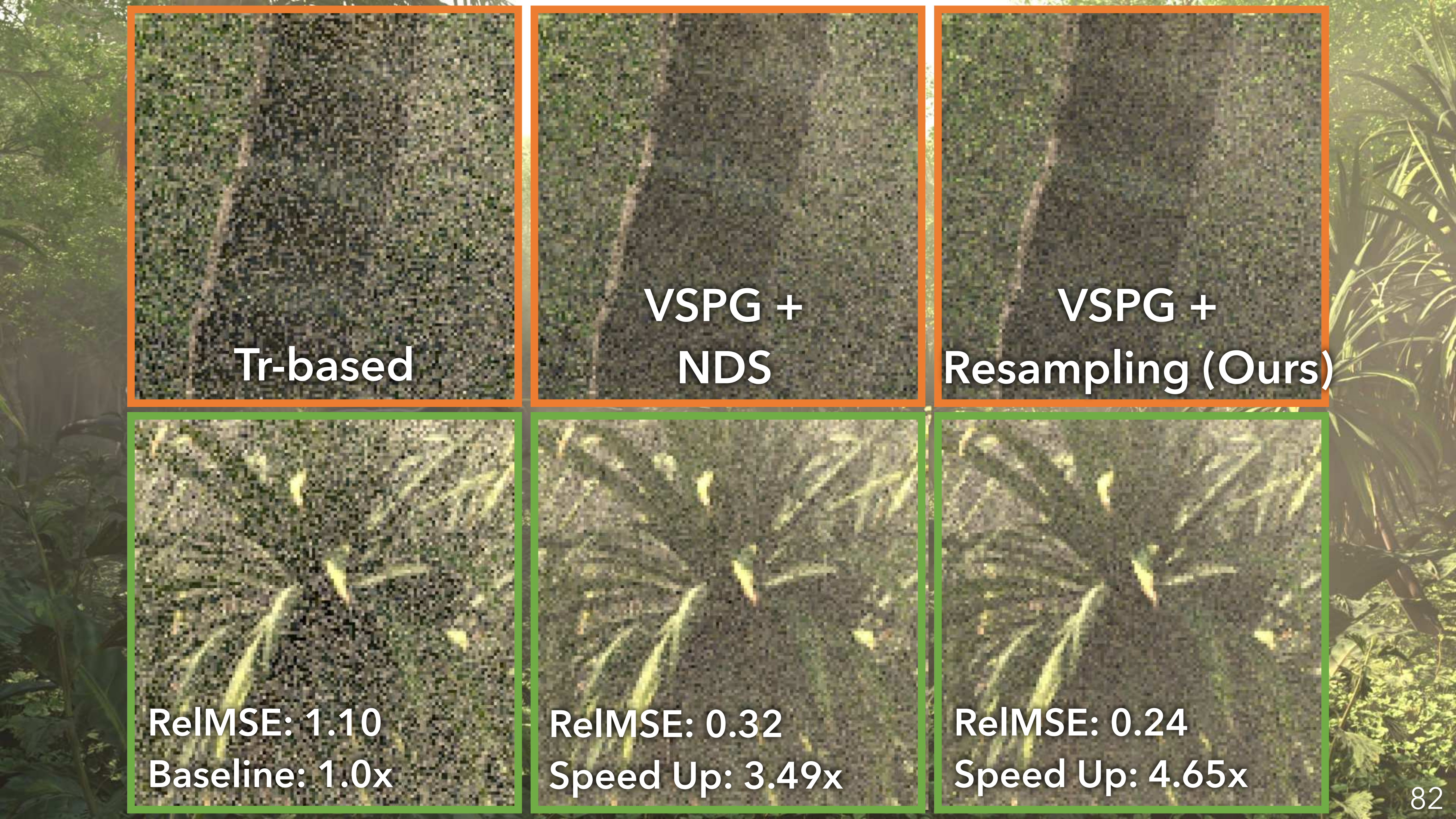
Achieved P_{vol}

Jungle, 5 min, Tr-based


Achieved P_{vol}
Jungle, 5 min, VSPG + Resampling (Ours)

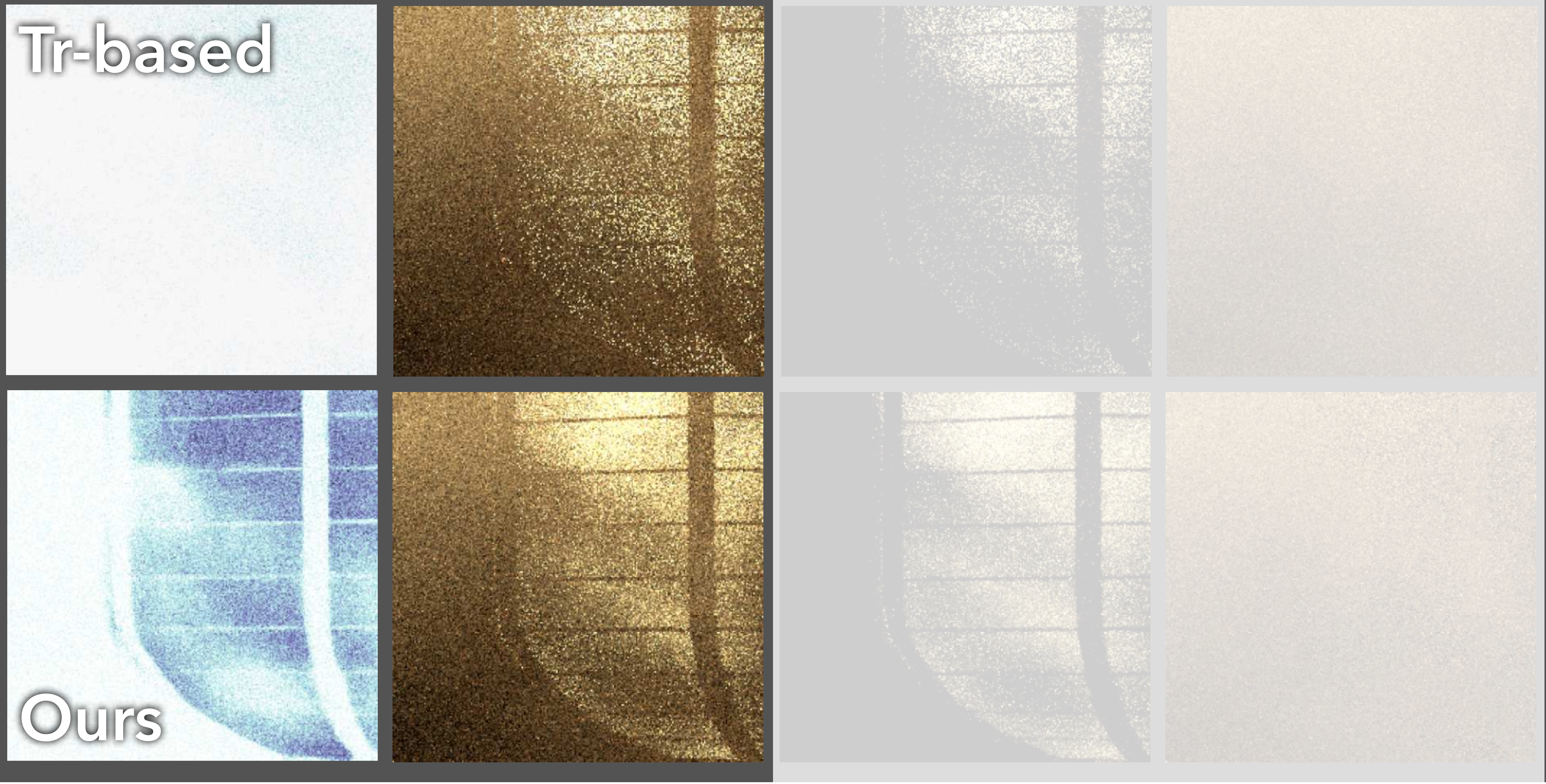
Jungle, 5 min, Tr-based

Jungle, 5 min, VSPG + Resampling (Ours)

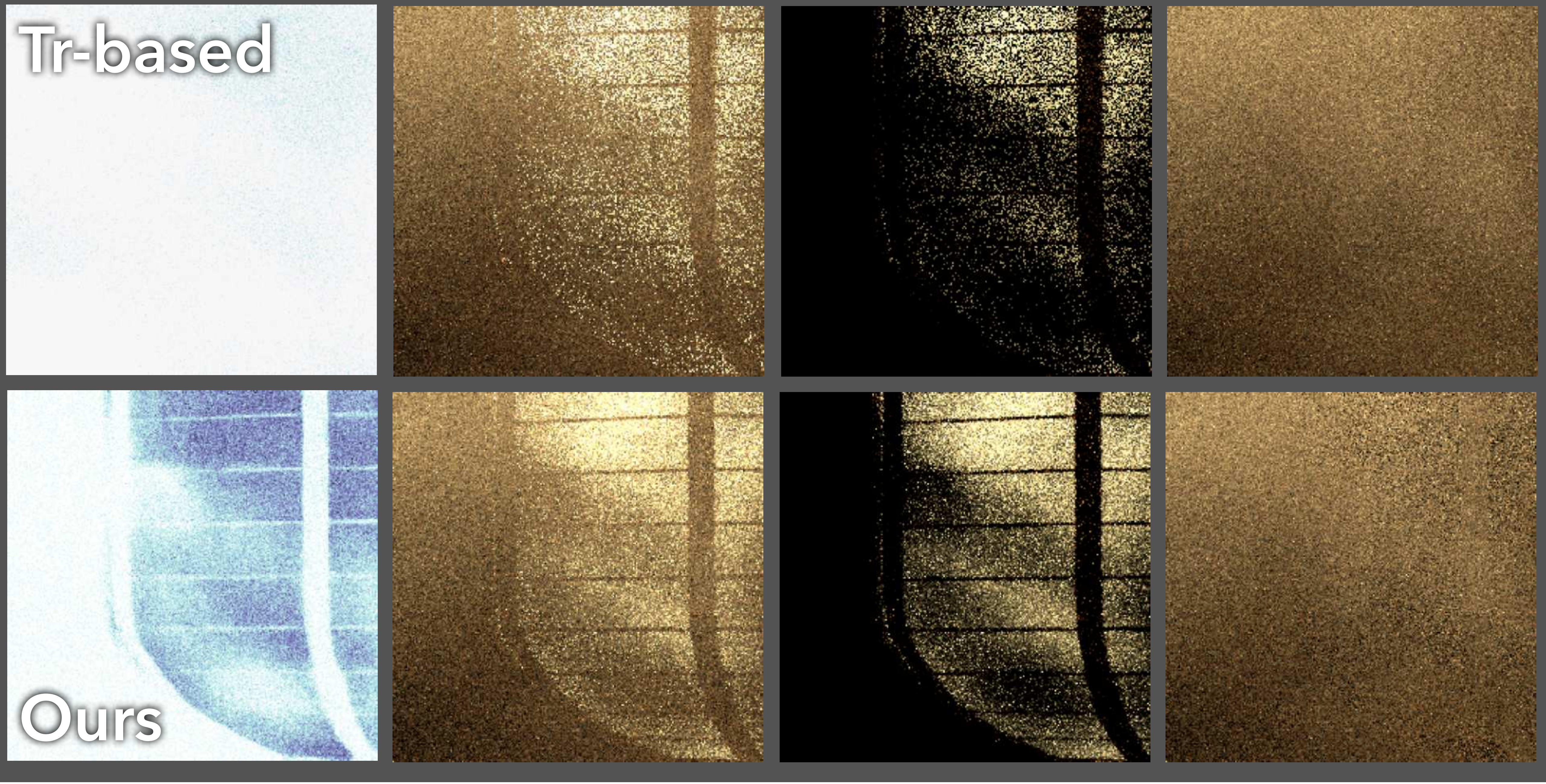

Tr-based
VSPG +
NDS
VSPG +
Resampling (Ours)
RelMSE: 1.10
Baseline: 1.0x
RelMSE: 0.32
Speed Up: 3.49x
RelMSE: 0.24
Speed Up: 4.65x


Achieved P_{vol}
VSPG + Resampling (Ours)
VSPG + NDS

Lantern

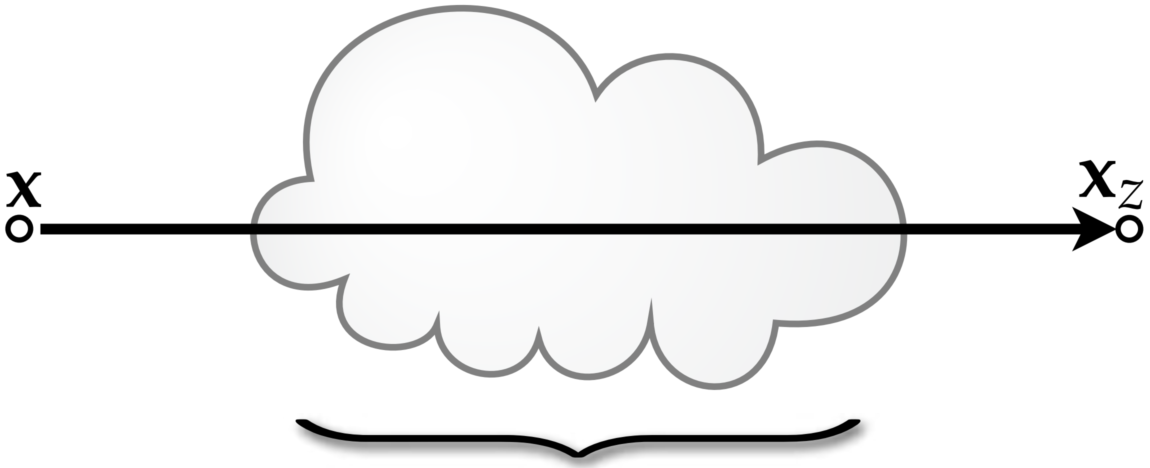


P_{vol} $L(\mathbf{x}, \omega) = L_s(\mathbf{x}, \omega) + L_v(\mathbf{x}, \omega)$



P_{vol} $L(\mathbf{x}, \omega) = L_s(\mathbf{x}, \omega) + L_v(\mathbf{x}, \omega)$

Transmittance-based Distance Sampling



$$T_r(x, x_z) = e^{-\int_0^z \sigma_t(x)dx}$$

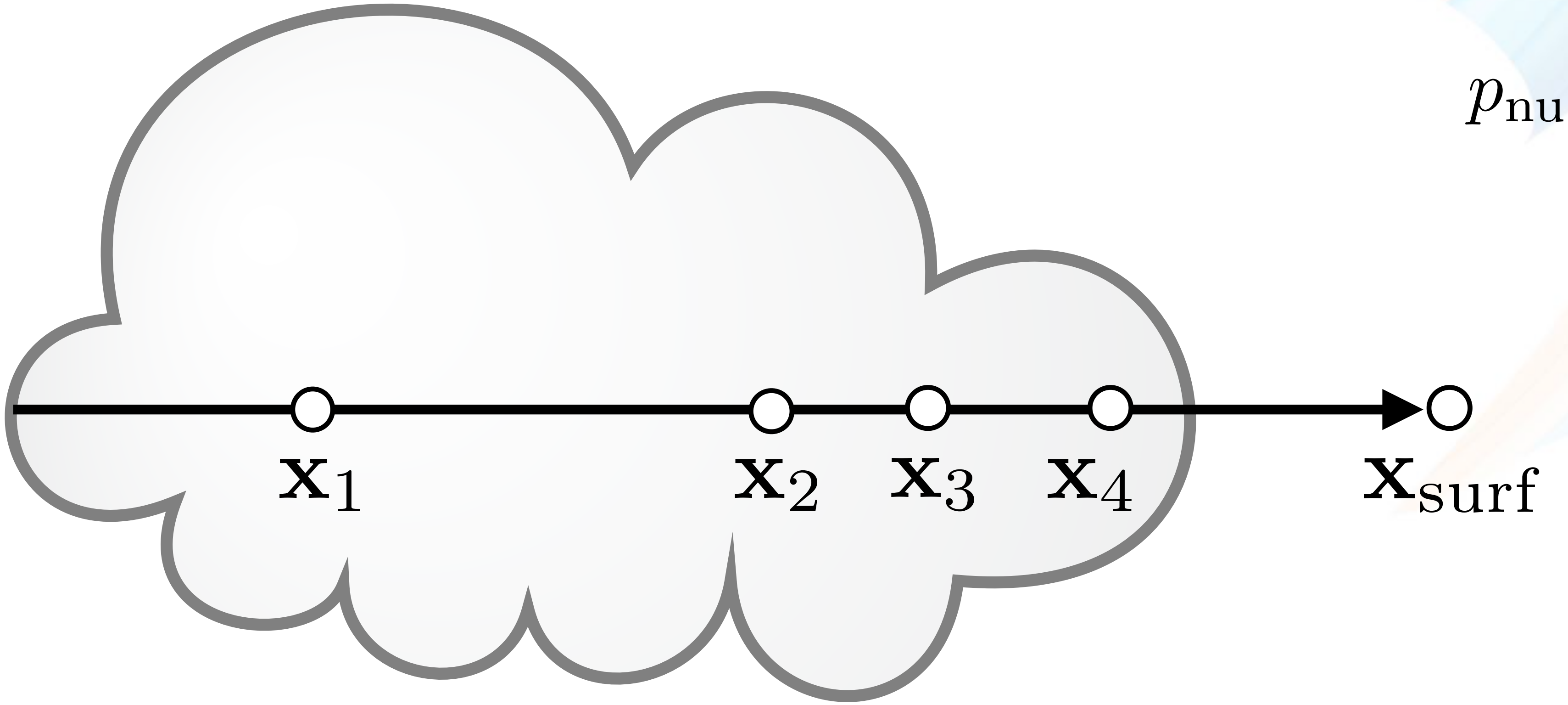
- P_{vol}^{Δ} : an implicit decision based on local volume properties (i.e., transmittance)

$$P_{\text{vol}}^{\Delta} = 1 - T_r(x, x_z)$$

Candidate Samples

$$p_{\text{real}}(x_i) = \frac{\sigma_t(x_i)}{\overline{\sigma}}$$

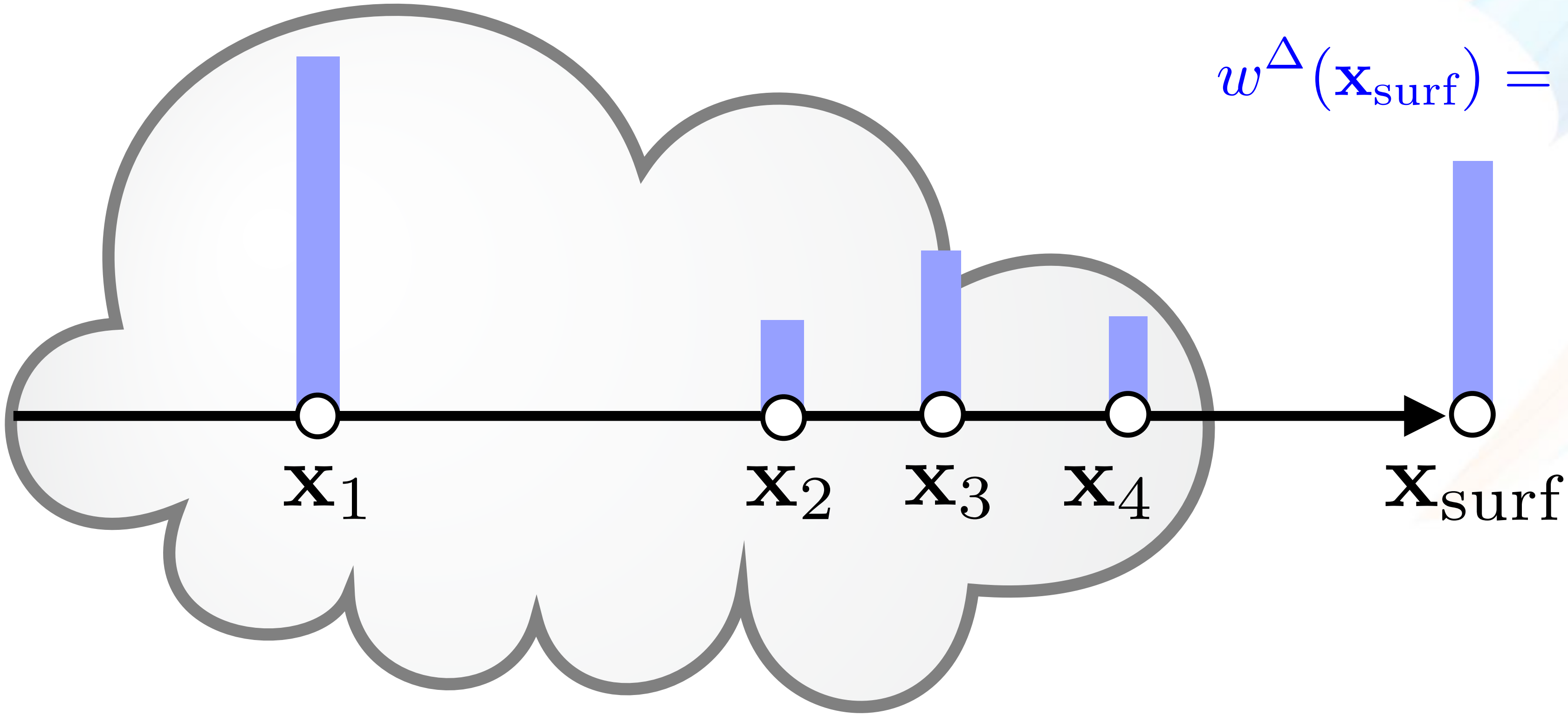
$$p_{\text{null}}(x_i) = 1 - p_{\text{real}}(x_i) = \frac{\sigma_n(x_i)}{\overline{\sigma}}$$



Resampling Weights for Delta Tracking

$$w^{\Delta}(\mathbf{x}_i) = \sigma_t(\mathbf{x}_i)\, p_{\text{real}}(\mathbf{x}_i) \prod_{j \le i} p_{\text{null}}(\mathbf{x}_j)$$

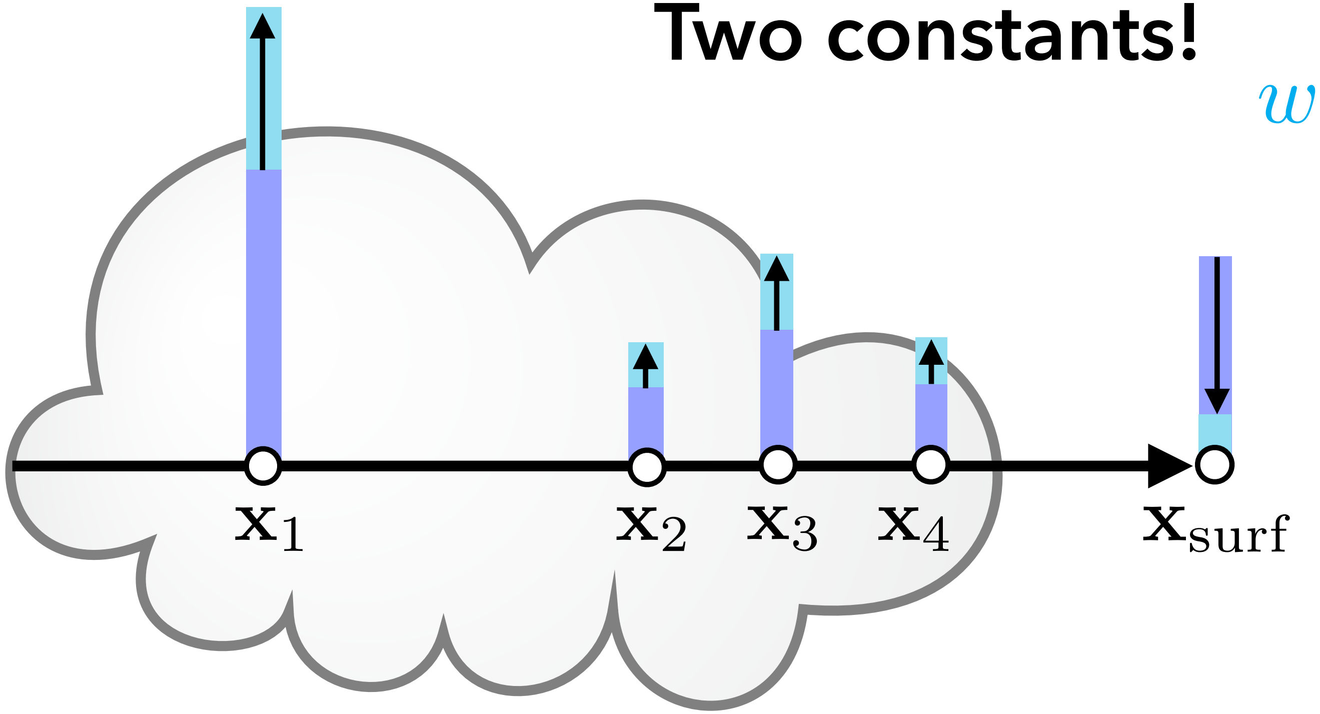
$$w^{\Delta}(\mathbf{x}_{\text{surf}}) = \prod_{i} p_{\text{null}}(\mathbf{x}_i)$$



Computed based on local volume properties

Resampling Weights for Our Method

Two constants!



$$w^\star(\mathbf{x}_i) = C_{\text{vol}} * w^\Delta(\mathbf{x}_i)$$

$$w^\star(\mathbf{x}_{\text{surf}}) = C_{\text{surf}} * w^\Delta(\mathbf{x}_{\text{surf}})$$

$$C_{\text{vol}} = \frac{P^\star_{\text{vol}}}{1 - \prod_i p_{\text{null}}(x_i)} \approx \frac{P^\star_{\text{vol}}}{P^\Delta_{\text{vol}}}$$

$$C_{\text{surf}} = \frac{1 - P^\star_{\text{vol}}}{\prod_i p_{\text{null}}(x_i)} \approx \frac{1 - P^\star_{\text{vol}}}{1 - P^\Delta_{\text{vol}}}$$

Exactly reaches the target P_{vol}