Eurographics 2015
The 36th Annual Conference of the
European Association for Computer Graphics

General and Robust Error Estimation and Reconstruction for Monte Carlo Rendering

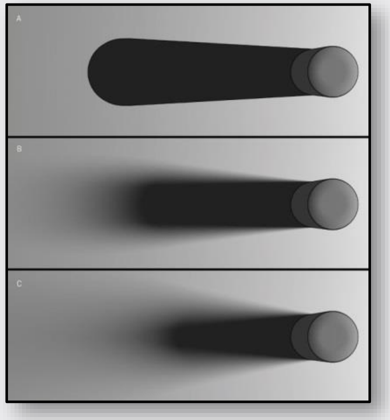
Pablo Bauszat[1], Martin Eisemann[1,2], Elmar Eisemann[2], Marcus Magnor[1]

[1] Computer Graphics Lab, TU Braunschweig, Germany
[2] Delft University of Technology, Netherlands

Monte Carlo Rendering

- Today's industry standard
- General and unbiased
- Covers variety of natural phenomena

- Requires extensive sampling
  - Pixel (2D integral)
  - Camera lens (2D integral)
  - Time (1D integral)
  - Global illumination (2D integral per bounce)
  - … and more …

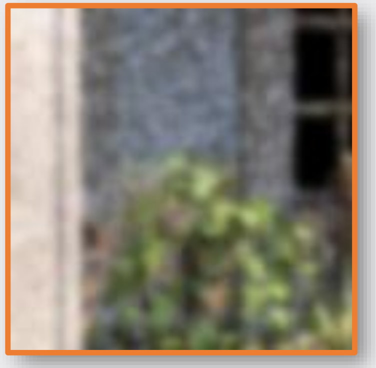


Noise

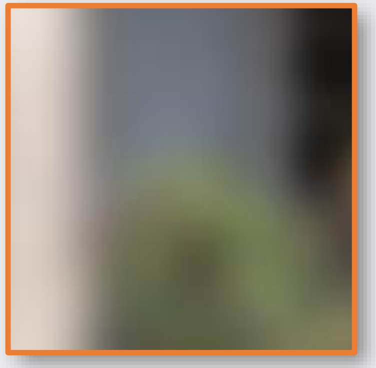
Filtering

Noisy

Reference



Uniform filter (small)



Uniform filter (large)

Adaptive filtering

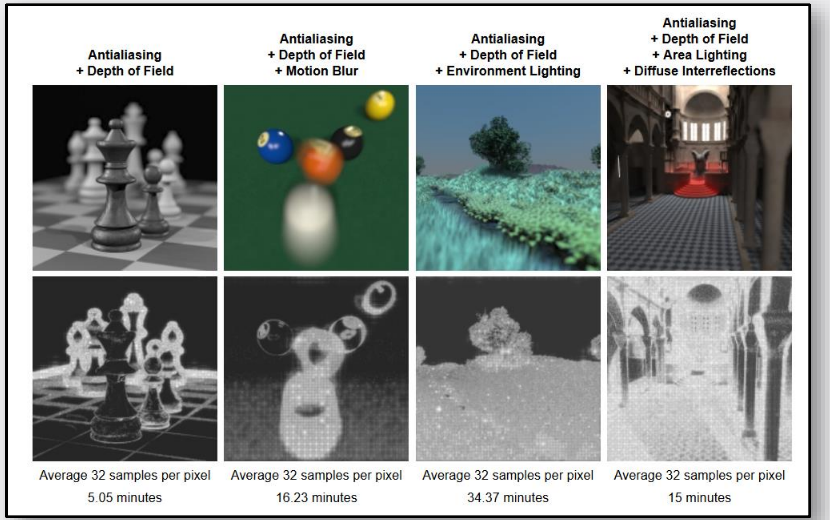
Adaptive Reconstruction

- **Filter bank**
  - Set of filters with different properties
  - Select best filter on a per-pixel level

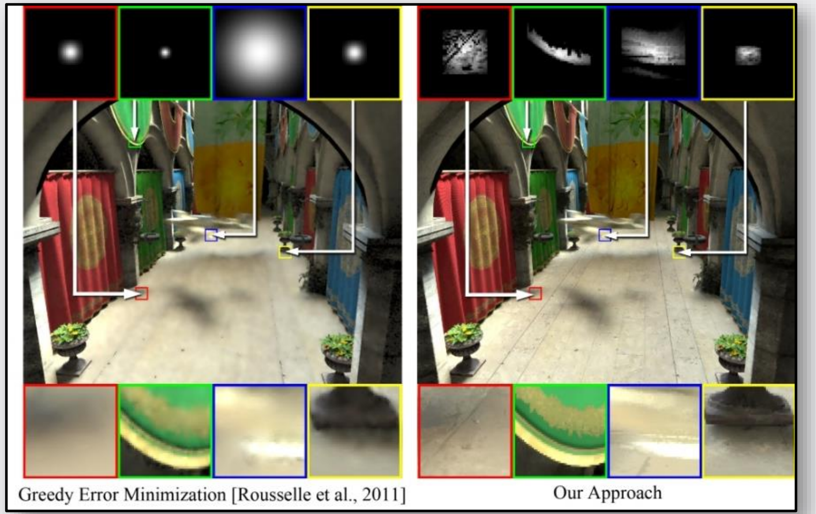
Problem statement

How to choose the best filter from the set for a pixel?

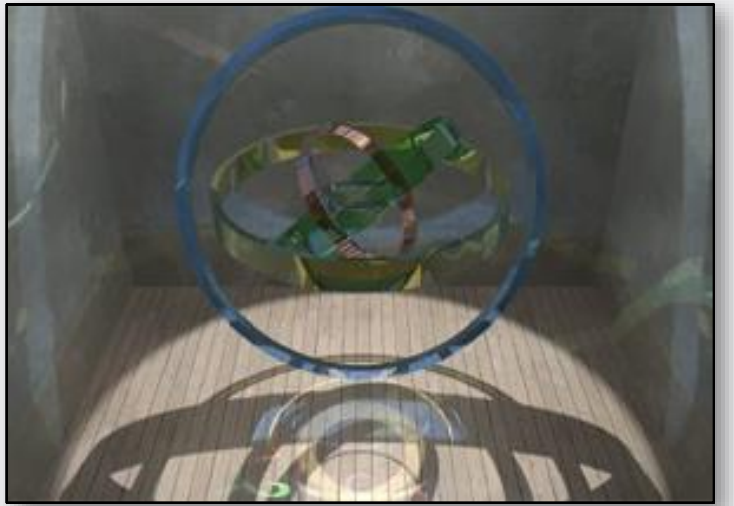
Previous work



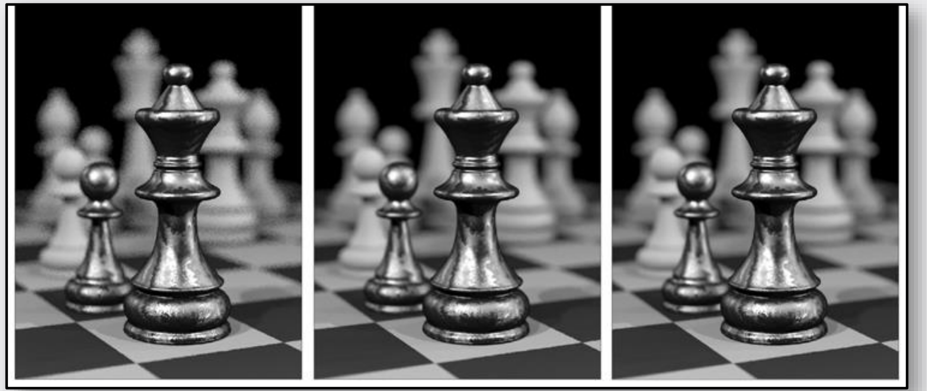
Overbeck et al. 2009



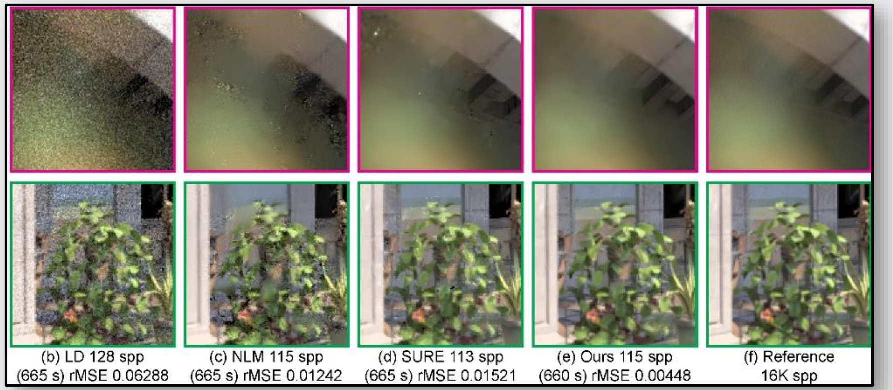
Li et al. 2012



Rousselle et al. 2011/2012/2013



Kalantari et al. 2013



Moon et al. 2014

Limitations of previous work

- Filter selection based on noisy image
- Often tailored for specific filters
- Switching filters may cause seams

Local selection

Our method

Insights

Our method is based on three key insights:

1. Filter selection is often more crucial than sampling rate
2. Filter error is locally smooth for most image regions
3. Often multiple filters are close-to-optimal choices

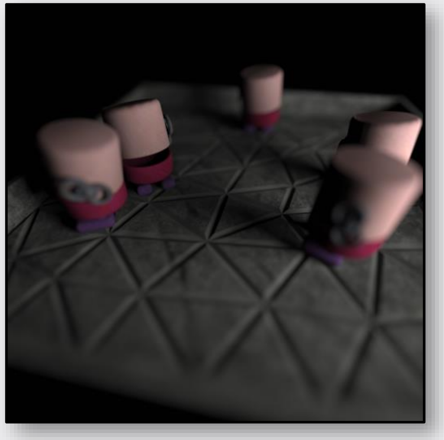
1. Filter selection is often more crucial than sampling rate

Filter bank of 4 Gaussian and
4 Joint Bilateral filters

32 spp SURE	12.3 MSE^{-3}	
32 spp Best choice	1.6 MSE^{-3}	**(x 7.7)**
16 spp Best choice	2.3 MSE^{-3}	**(x 5.3)**

Recently employed by [Li2012] and [Rousselle2013]

1. Filter selection is often more crucial than sampling rate



Scene	SURE 32 spp	Best choice 32 spp	Best choice 16 spp
Conference	12.327	1.605 **(x 7.7)**	2.344 **(x 5.3)**
Sibenik	0.758	0.157 **(x 4.8)**	0.258 **(x 2.9)**
Toasters	0.187	0.096 **(x 1.9)**	0.156 **(x 1.2)**
San Miguel	16.880	6.419 **(x 2.6)**	9.831 **(x 1.7)**

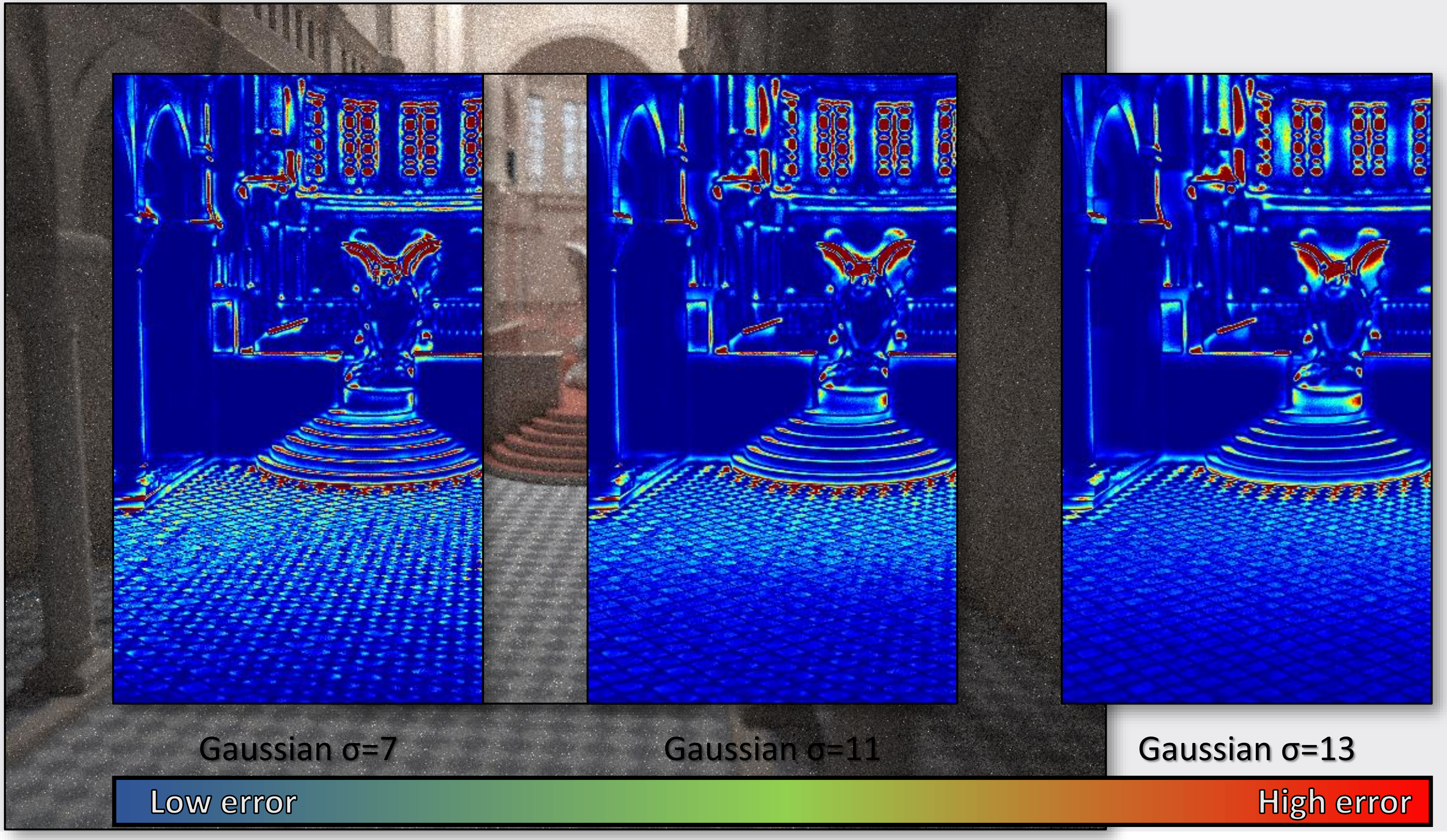
Mean squared error (MSE) * 10^{-3} – Same filter bank

Insights

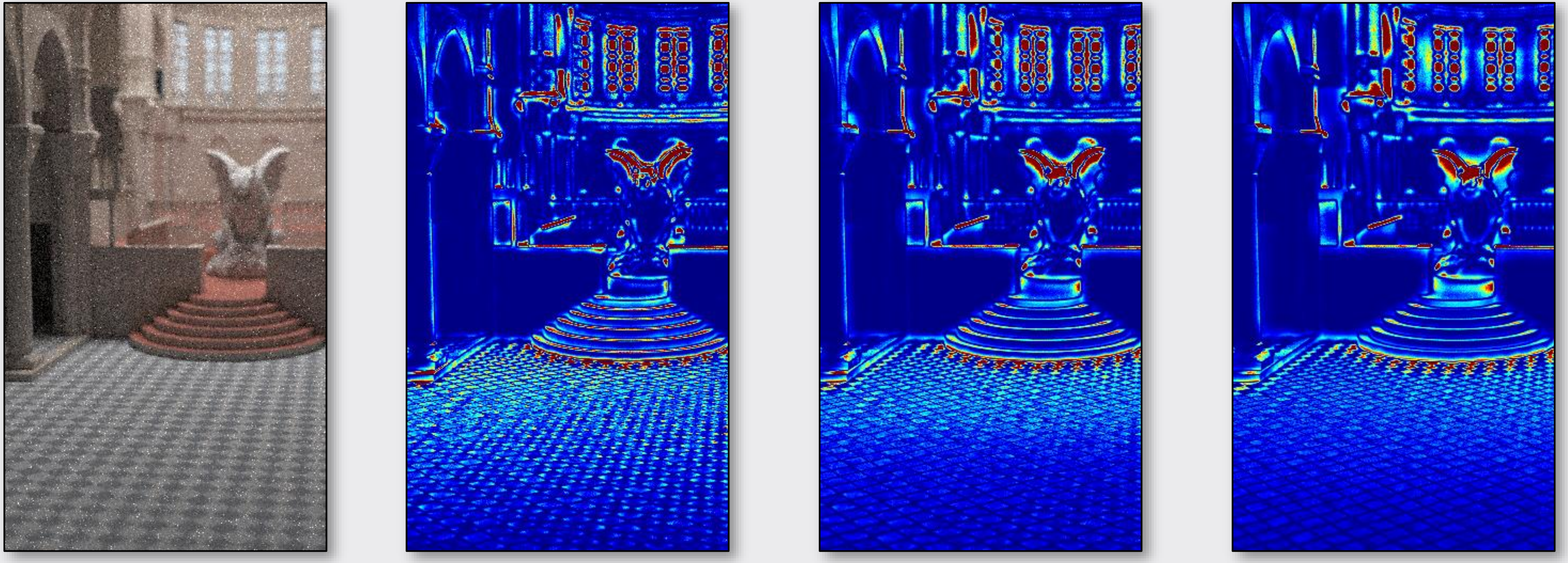
Our method is based on three key insights:

1. Filter selection is often more crucial than sampling rate
2. Filter error is locally smooth for most image regions
3. Often multiple filters are close-to-optimal choices

2. Error smoothness – Gaussian filters



2. Error smoothness – Guided Image Filtering [He2010]



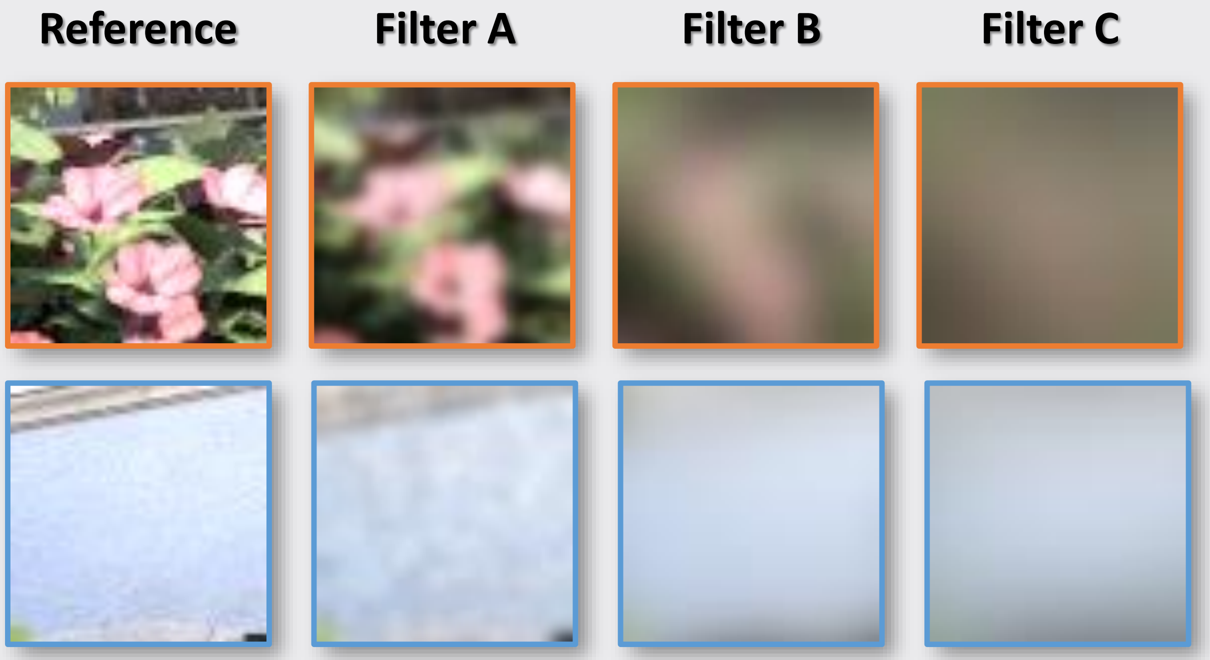
Guided radius=4 Guided radius=8 Guided radius=16

Insights

Our method is based on three key insights:

1. Filter selection is often more crucial than sampling rate
2. Filter error is locally smooth for most image regions
3. Often multiple filters are close-to-optimal choices

3. Often multiple filters are close-to-optimal choices



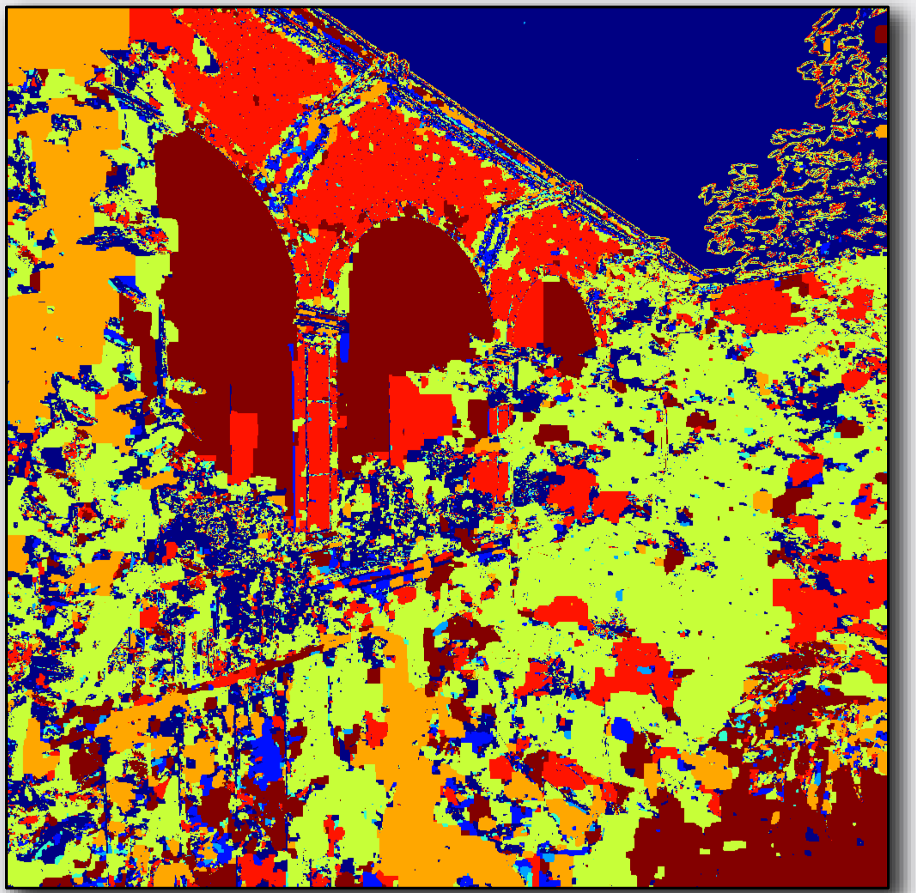
3. Often multiple filters are close-to-optimal choices

Optimal selection via ground truth

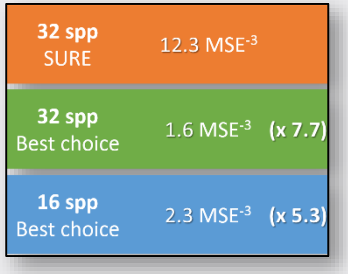
- MSE down to **8.0%** from noisy image

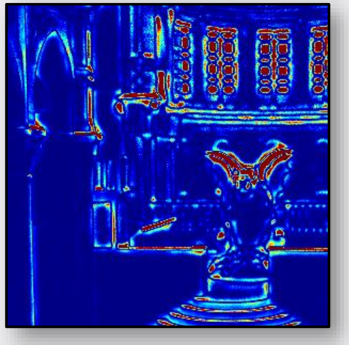
Regularized selection

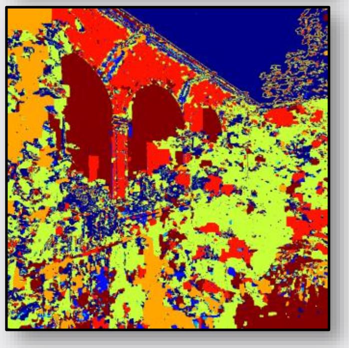
- MSE down to **9.1%** from noisy image
- Variations in selection are penalized



What do we learn from the insights?

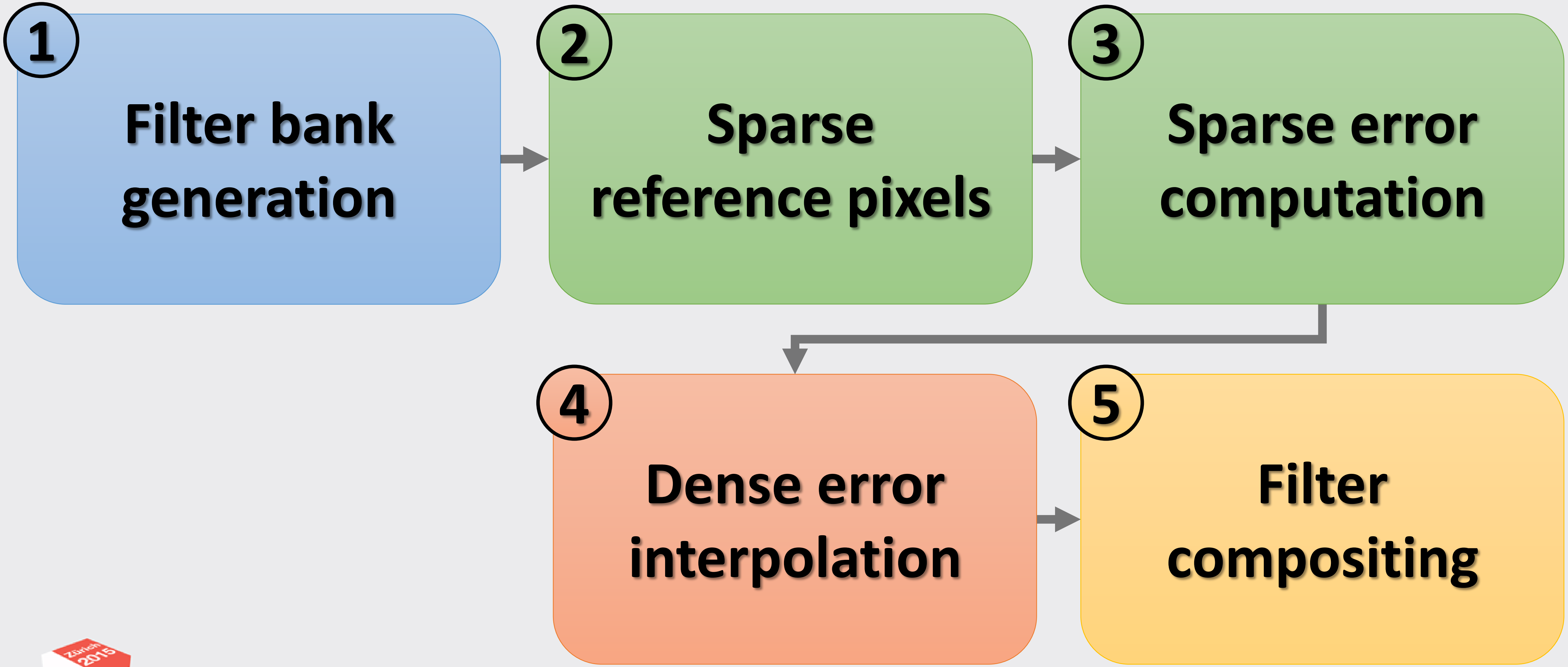






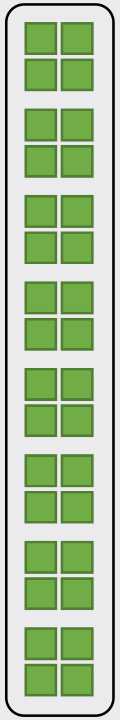
- Filter selection is crucial
- Filter error is piece-wise smooth
- Non-optimal filter selection does not imply large error

Our Method

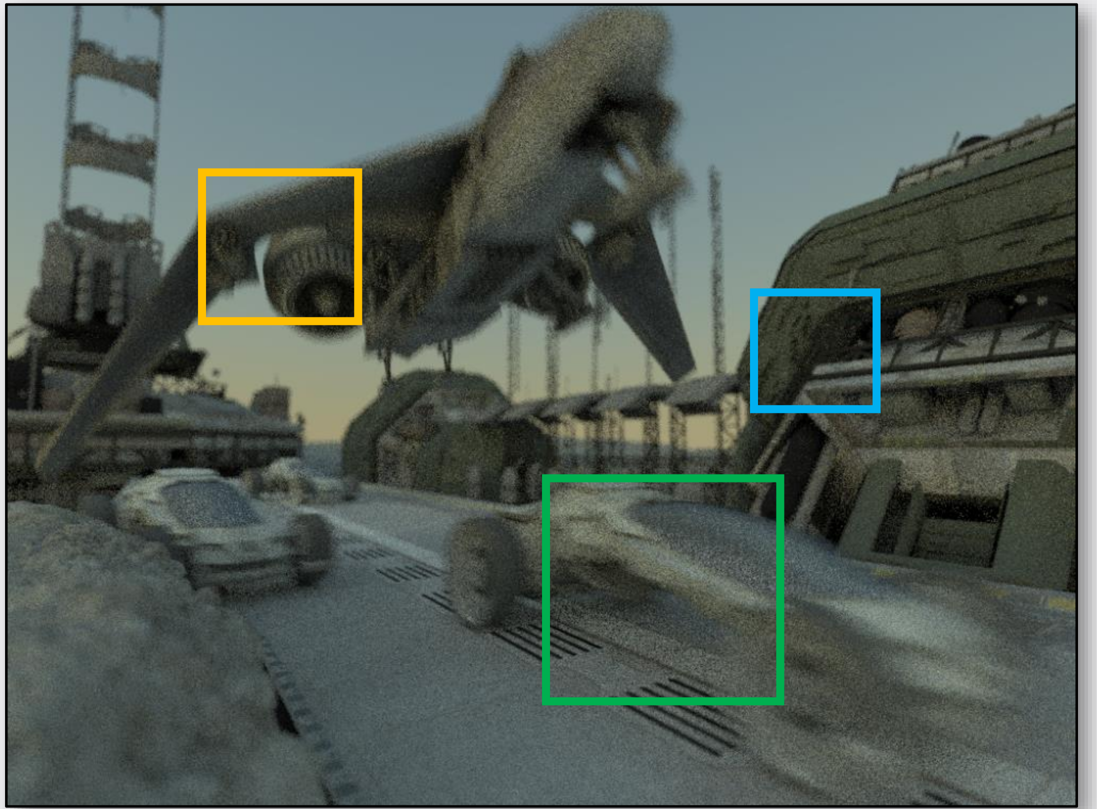

1
Filter bank generation
2
Sparse reference pixels
3
Sparse error computation
4
Dense error interpolation
5
Filter compositing

1. Filter bank generation

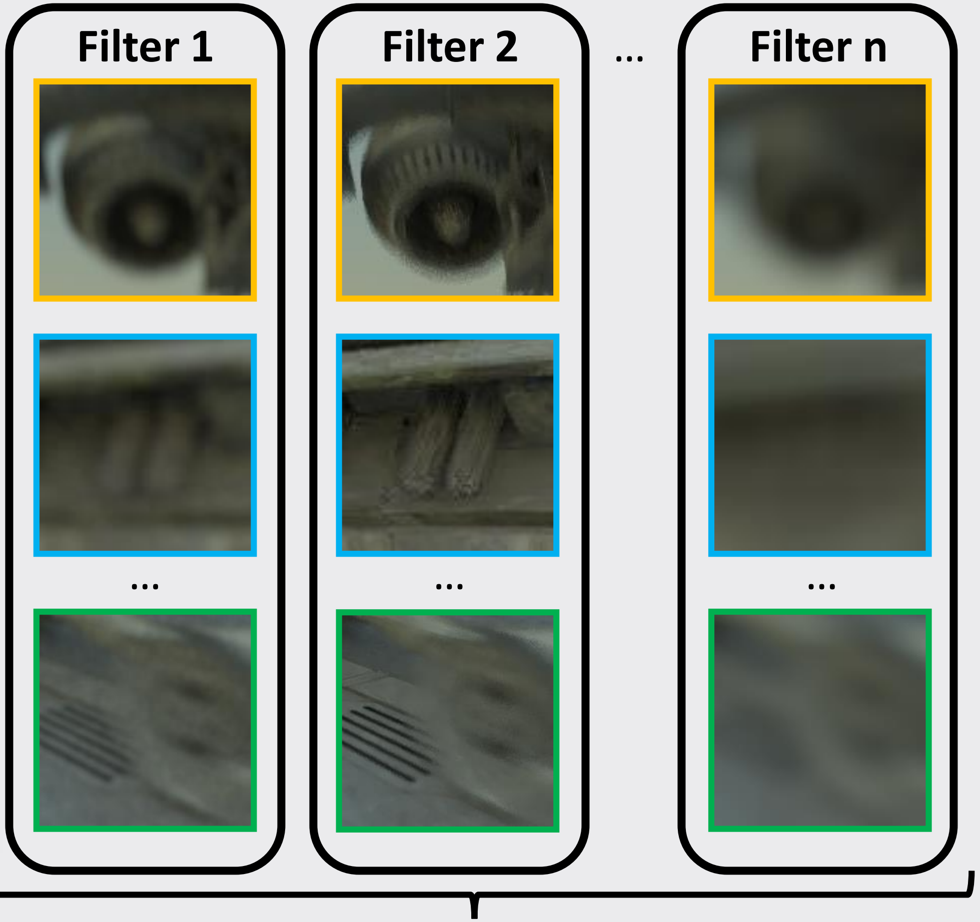
Sample Budget



32 spp

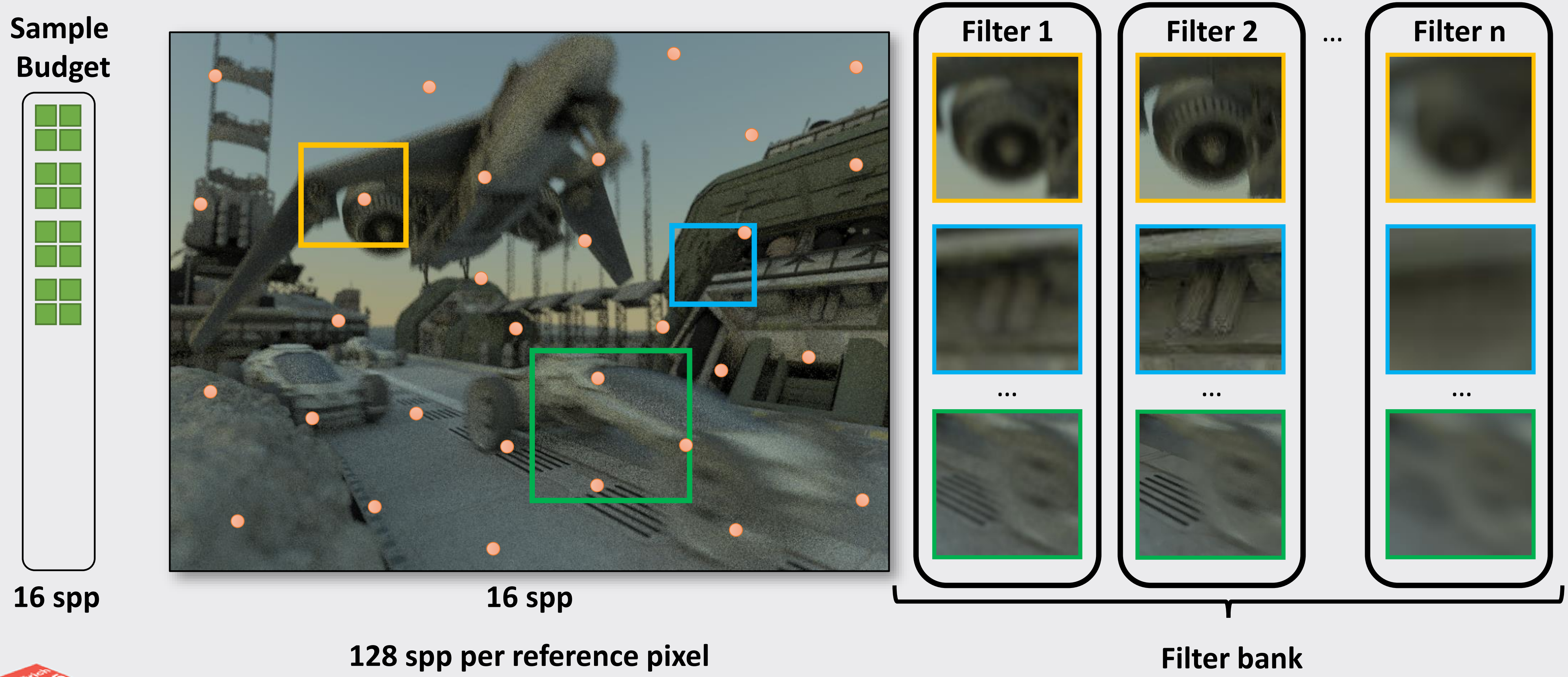


16 spp

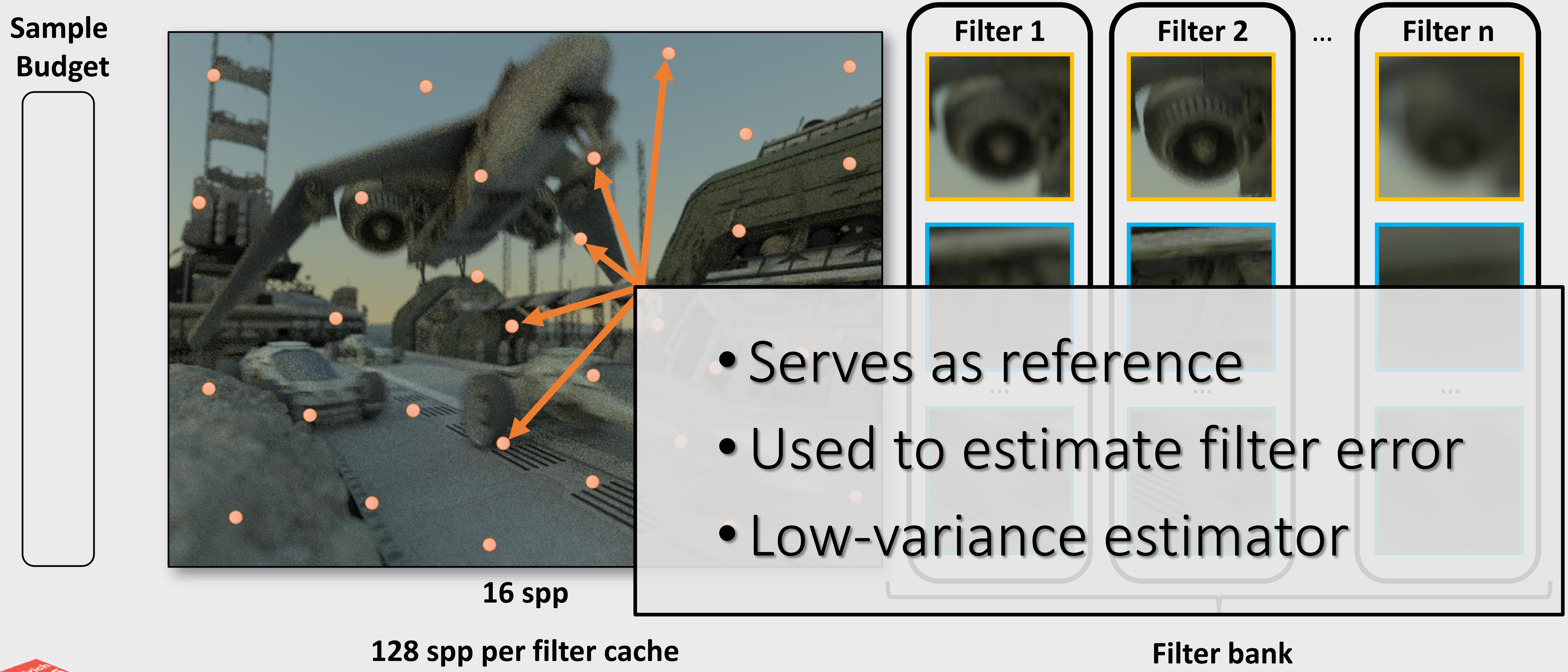


Filter bank

2. Sparse reference pixels



3. Sparse error computation



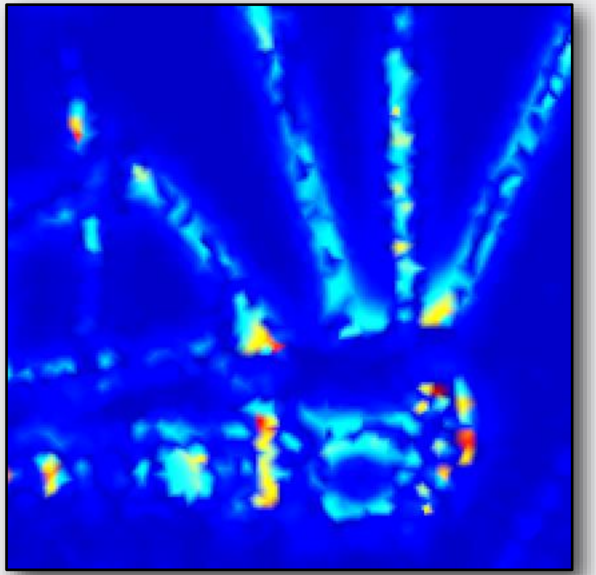
- Serves as reference
- Used to estimate filter error
- Low-variance estimator

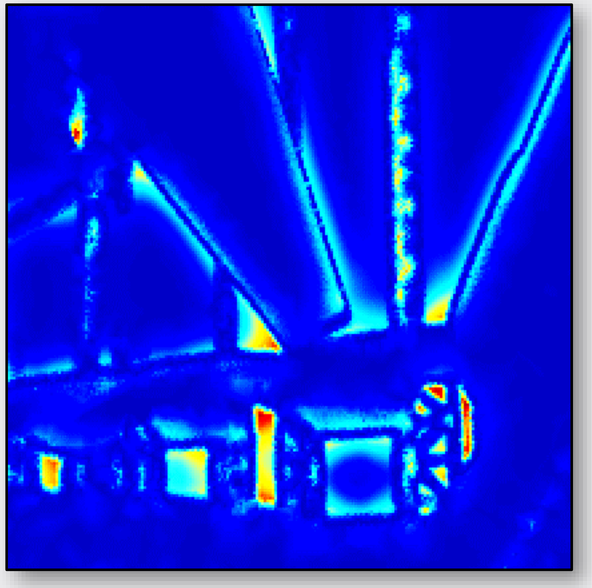
4. Dense error interpolation

- Interpolation of sparse error estimate (per filter)

Sparse error
(zoom-in)

Interpolated error
(zoom-in)



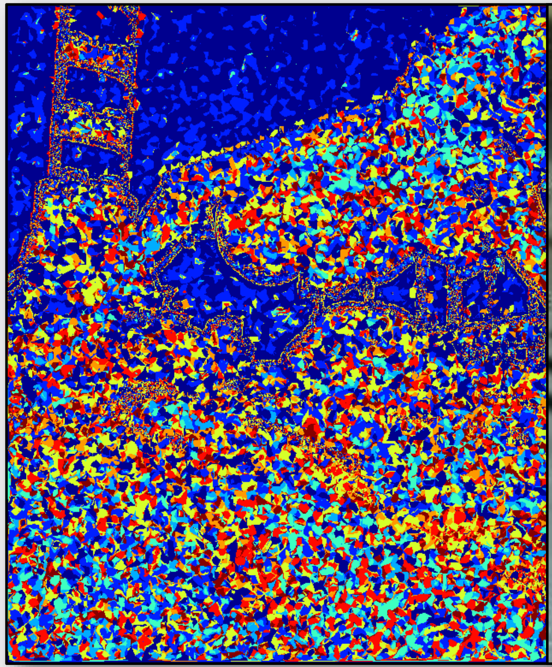


Filter error using reference
(zoom-in)

4. Dense error interpolation

- Best selection from interpolated error leads to seams

Optimal selection
(per-pixel)



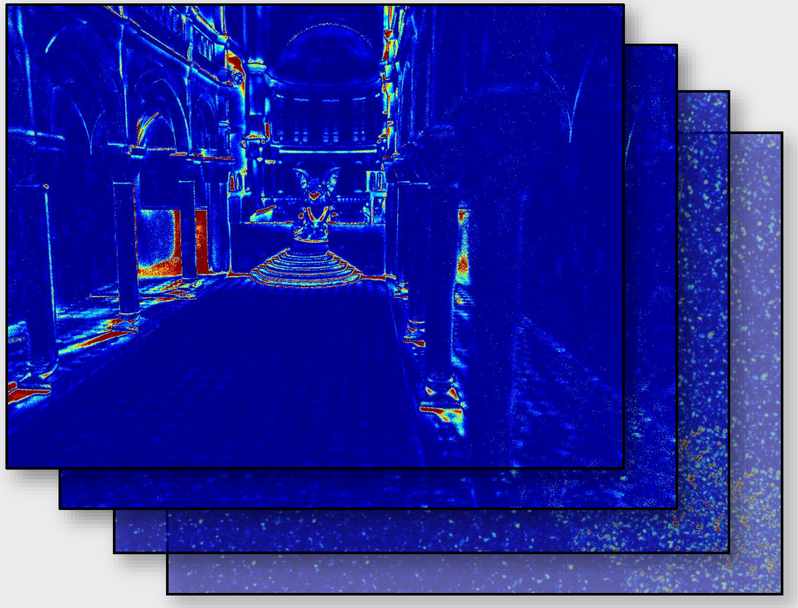
Seams
(closeup)

5. Filter compositing

Globally optimize filter selection (seek labeling L)

$$\underset{L}{\operatorname{argmin}}\, E(L) = E_{Data}(L) + \lambda \cdot E_{regularizer}(L)$$

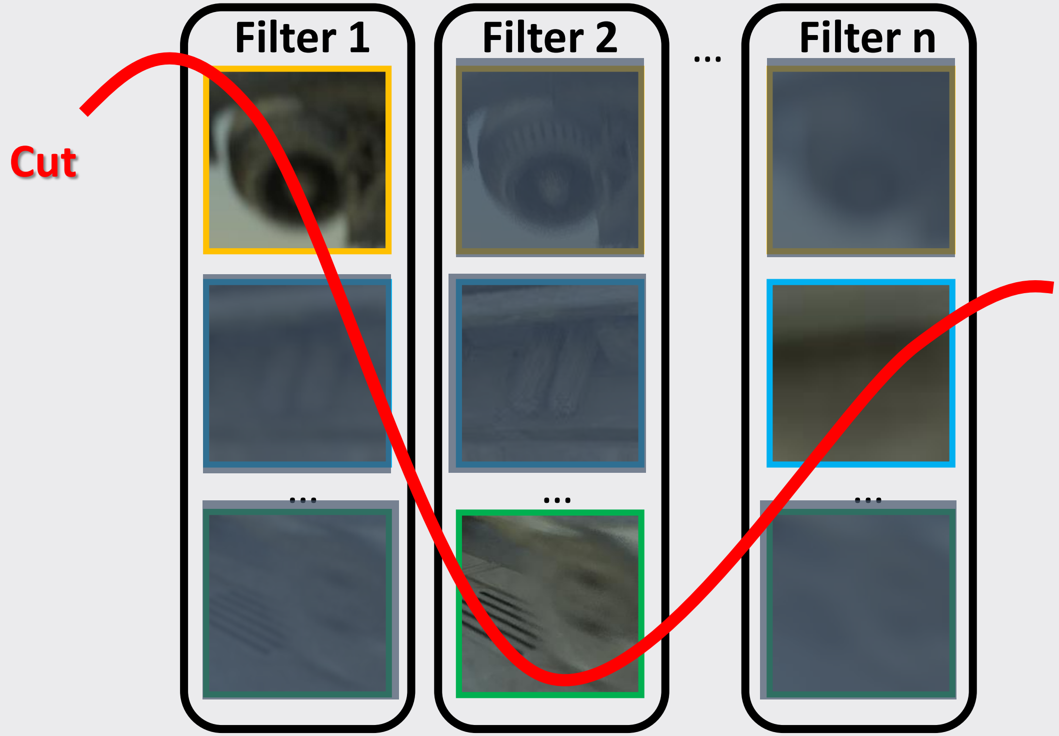
Data term
Local errormaps
Minimize MSE

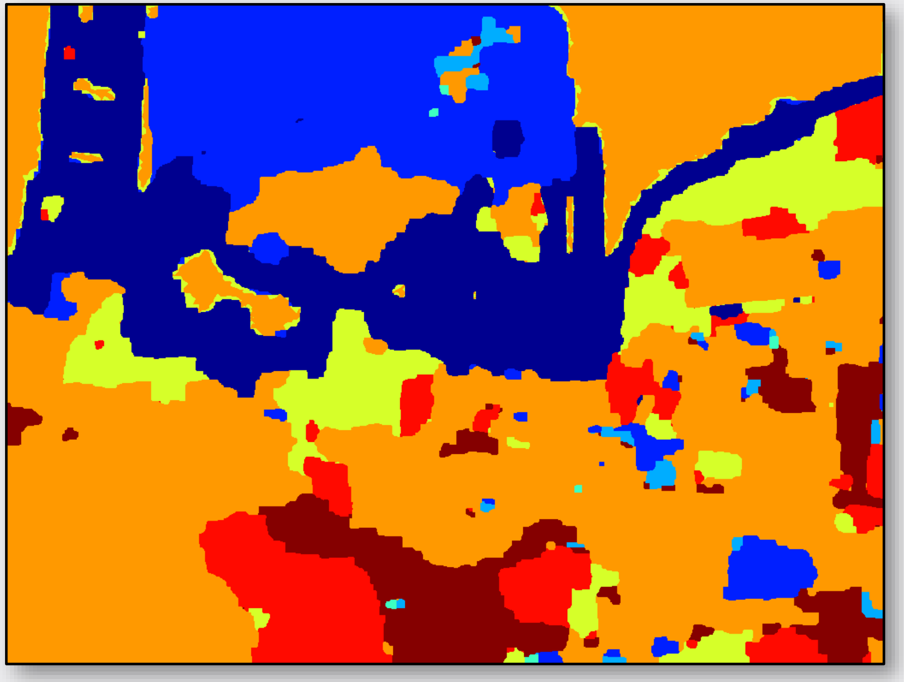


Regularization term
Solution image gradients
Avoid seams

5. Filter compositing

- Solve by **graph-cuts**
  „Fast approximate energy minimization via graph cuts", Boykov et al. 2001





Globally optimized label map

5. Filter compositing

- Solve by **graph-cuts**

  „Fast approximate energy minimization via graph cuts", Boykov et al. 2001

Local selection

Global selection

Our Method

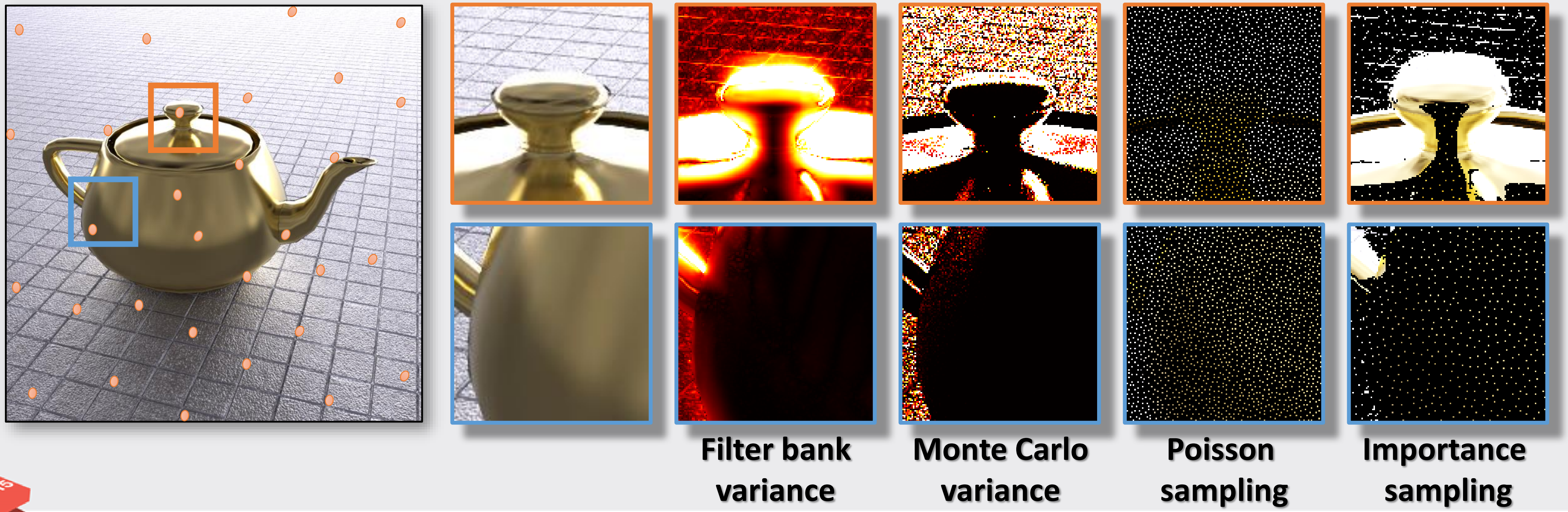
1. Filter bank generation
2. Sparse reference pixels
3. Sparse error computation
4. Dense error interpolation
5. Filter compositing

Bells & Whistles

- Choice of regularization in filter compositing
- Integration of high-quality radiance values (not included the filter bank)
- **Select „best“ pixels for sparse error estimate**

Adaptive placement of sparse estimates

- Required for highly variant error regions
- Reduces residual variance in radiance estimate

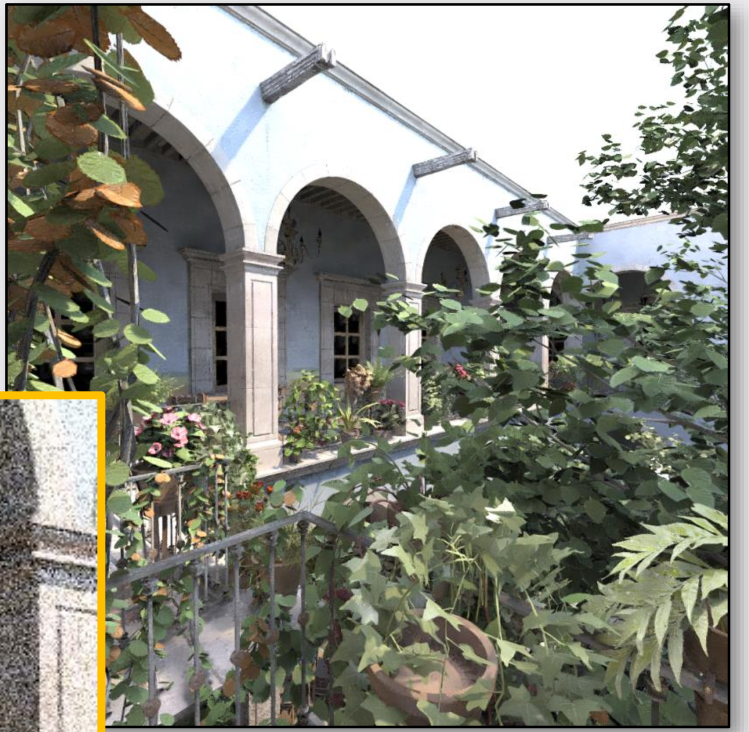


Results

Results – San Miguel

MC 4096 spp
15,449 sec

MC 32 spp
146 sec



Our result 32 spp
146 + 13 sec

Global illumination

Results - Chess

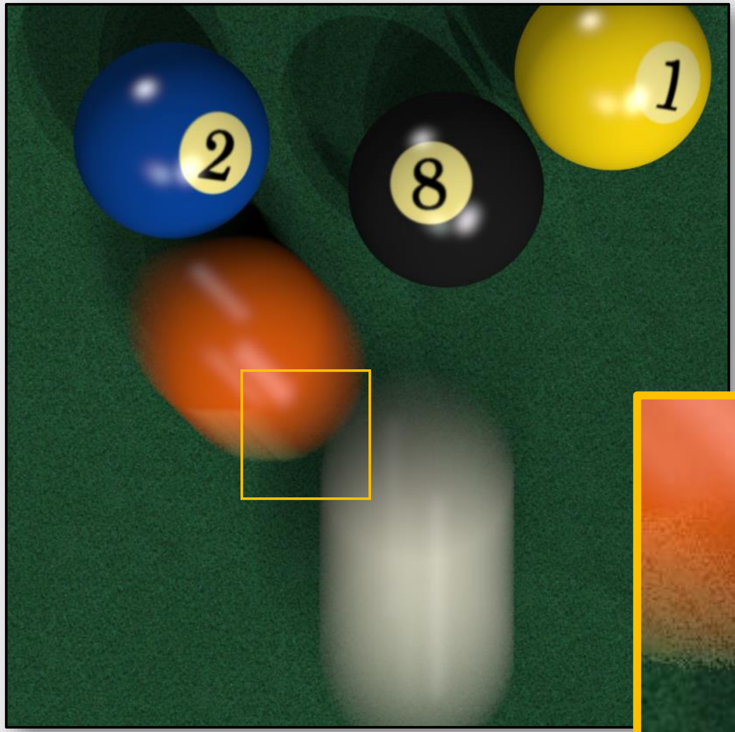
MC 4096 spp
1,492 sec

MC 8 spp
9 sec

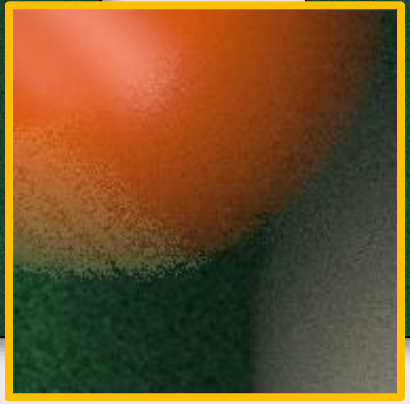
Our result 8 spp
9 + 29 sec

Depth-of-field

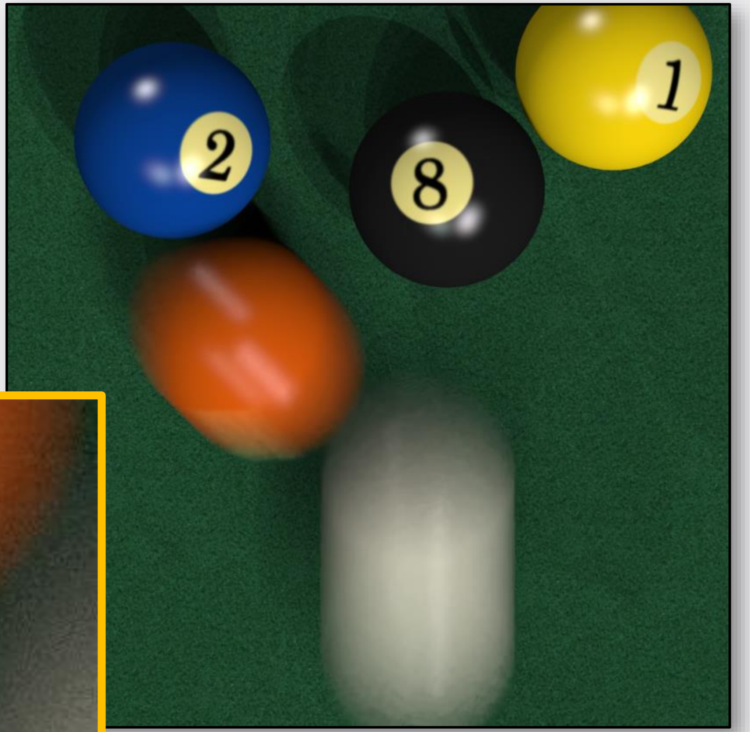
Results - Poolball



MC 4096 spp
10,989 sec



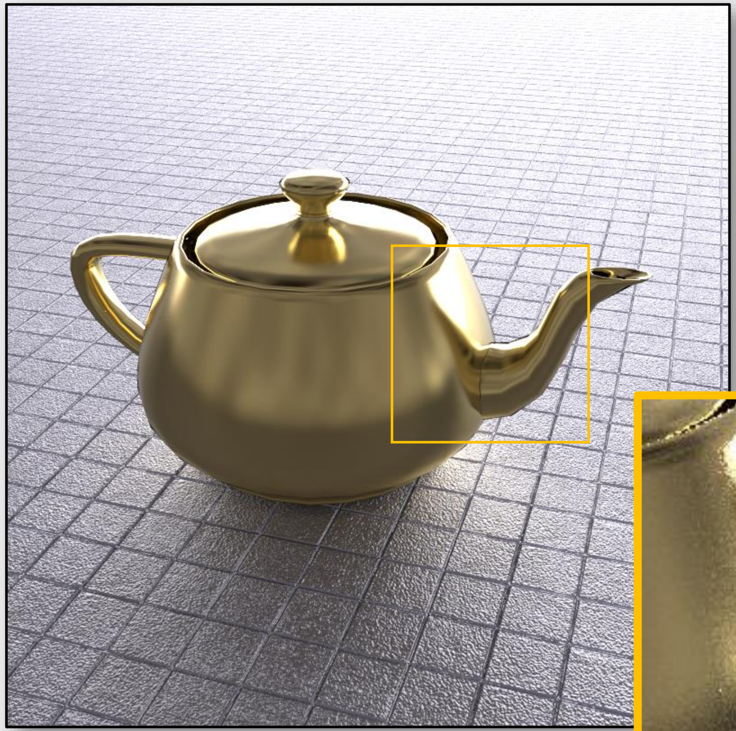
MC 8 spp
25 sec



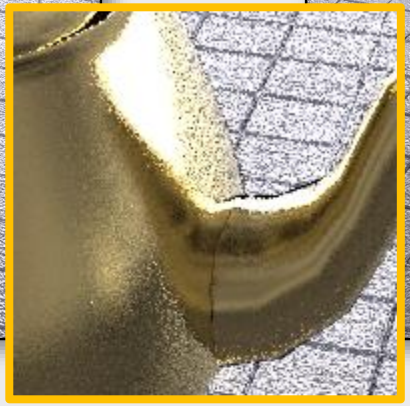
Our result 8 spp
25 + 25 sec

Motion blur

Results - Teapot



MC 4096 spp
3,619 sec

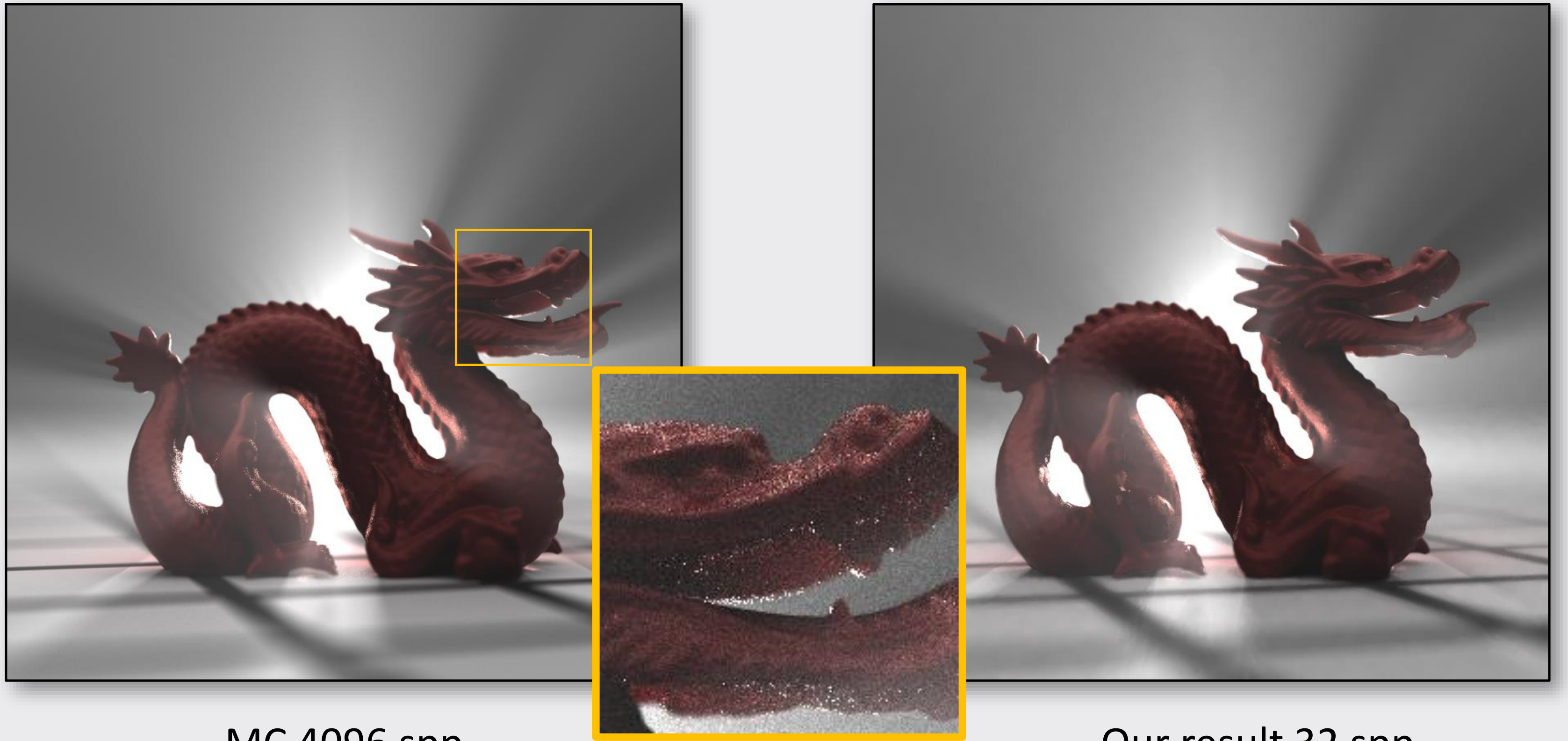


MC 16 spp
14 sec

Our result 16 spp
14 + 8 sec

Glossy materials

Results - Dragon



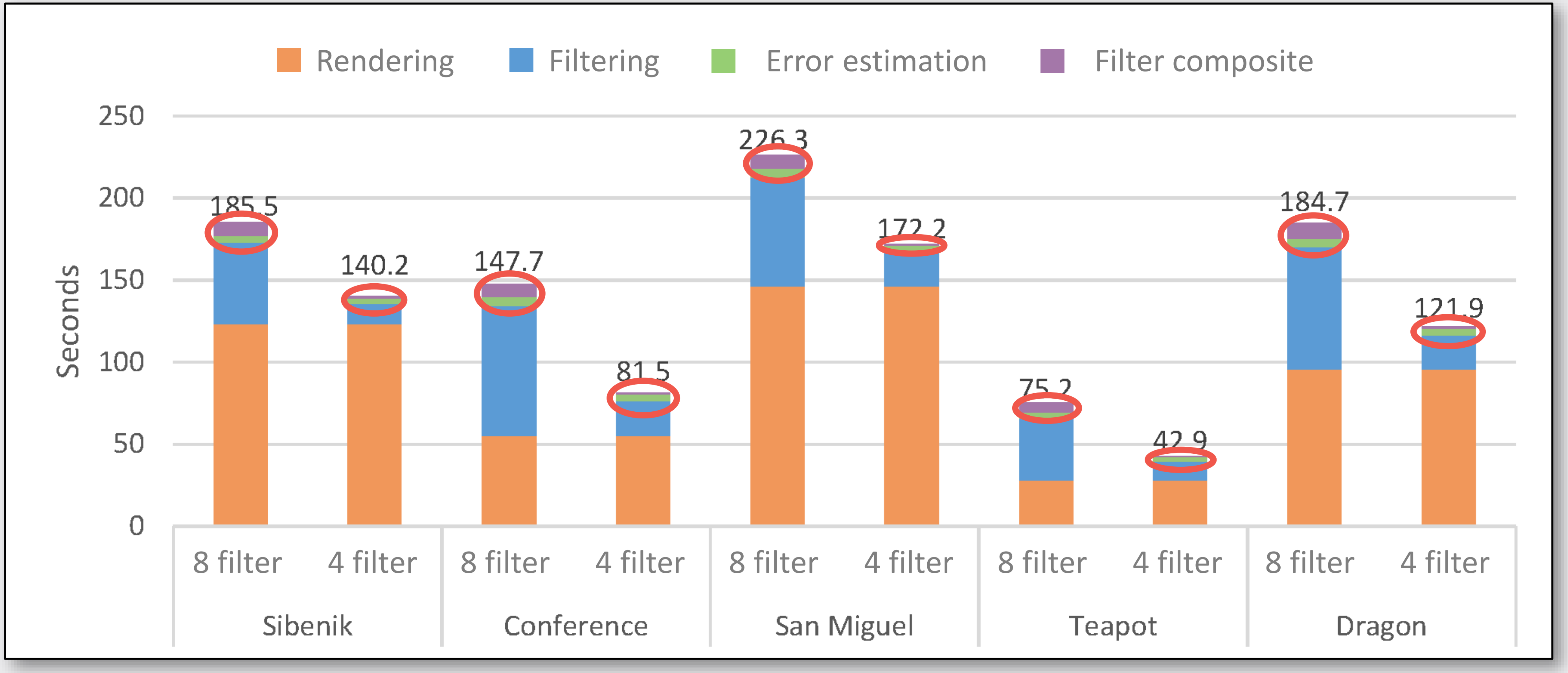
MC 4096 spp
12,464 sec

MC 32 spp
95 sec

Our result 32 spp
95 + 12 sec

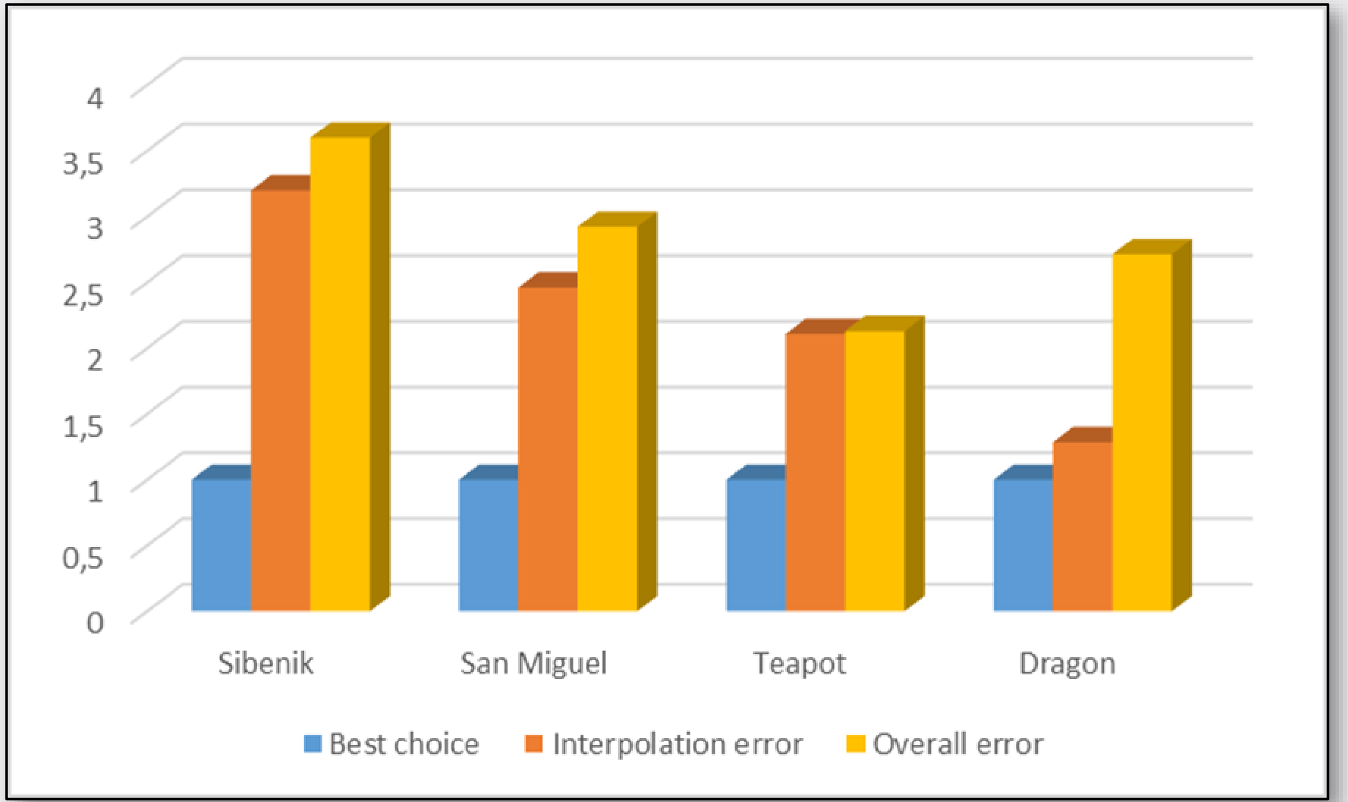
Participating media

Results - Timings



Intel Core i7-2600, 3.40 GHz, 16 GB RAM, NVIDIA GeForce 780 GTX, Windows 7 64-bit
Rendered with PBRT 2 path tracing.

Error analysis



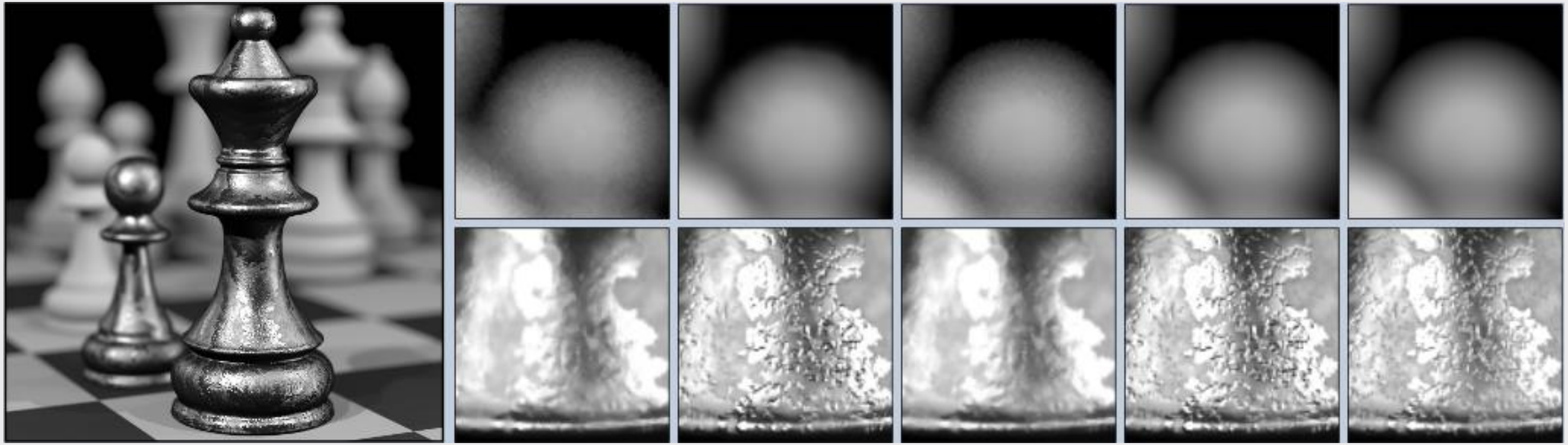
Two error sources

Interpolation error

Residual variance in radiance

Results – GID

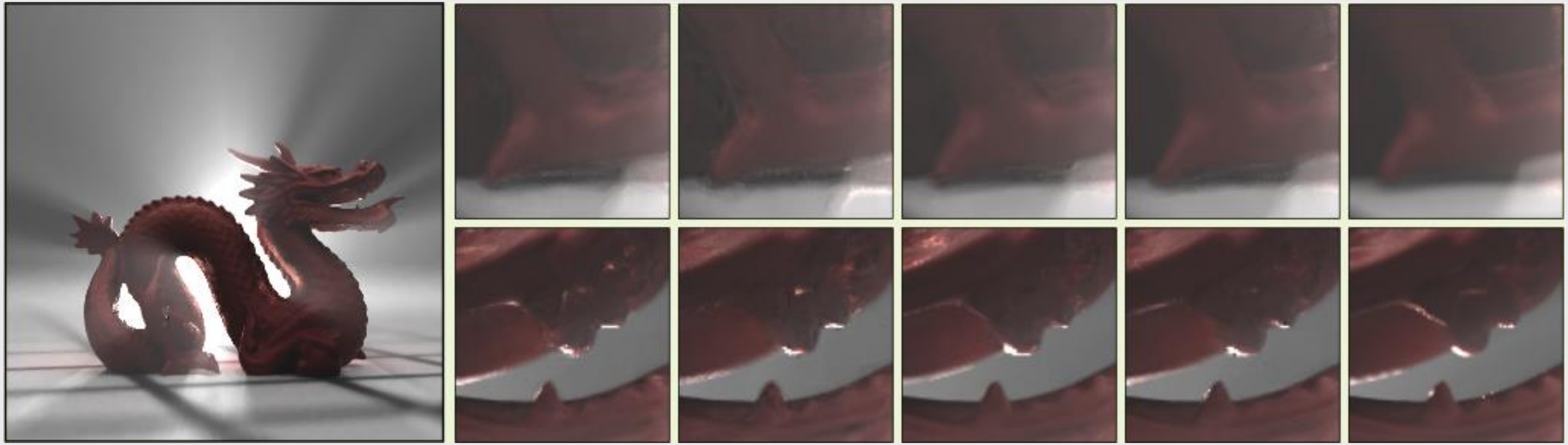
(„Removing the Noise in Monte Carlo Rendering with General Image Denoising Algorithms”, Kalantari et al. 2013)



	GID (8 spp)	**Ours** (8 spp)	**GID** (32 spp)	**Ours** (32 spp)	**Reference**
Chess scene	MSE=2.6491 SSIM=0.9516	MSE=**1.38179** SSIM=**0.9874**	MSE=2.4006 SSIM=0.9558	MSE=**0.8962** SSIM=**0.9948**	

Results – RD

(„Robust Denoising using Feature and Color Information”, Rousselle et al. 2013)



	RD (16 spp)	**Ours** (16 spp)	**RD** (32 spp)	**Ours** (32 spp)	**Reference**
Dragon scene	MSE=13.6693 SSIM=**0.9654**	MSE=**10.1914** SSIM=0.9599	MSE=9.3887 SSIM=**0.9781**	MSE=**7.8838** SSIM=0.9768	

Error sparsity

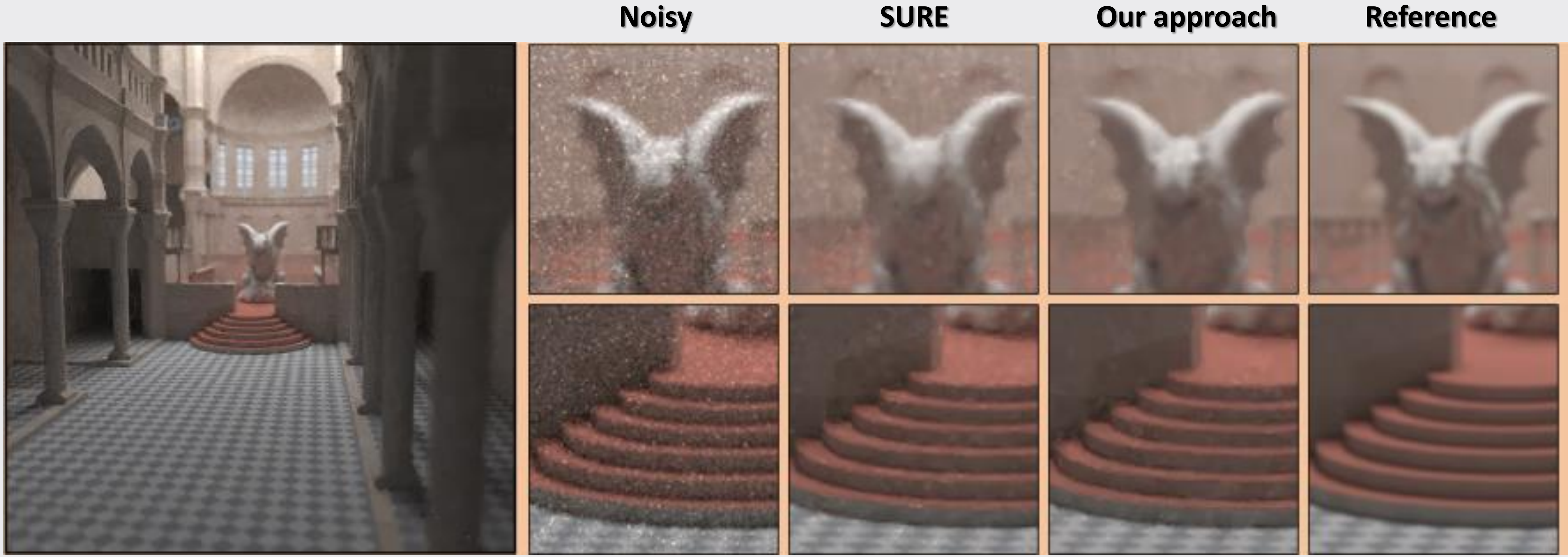
- Sparsity of error maps in transform domain (CDF 9/7 wavelets)
- Redundant information

Gaussian σ=7	Gaussian σ=11	Gaussian σ=13
86.46%	88.58%	89.86%

Guided radius=4	Guided radius=8	Guided radius=16
81.34%	87.07%	89.43%

NLM	BM3D	BLS-GSM
60.06%	67.35%	73.62%

Results – SURE [Stein1981]



Sibenik scene

	Noisy	SURE	Our approach
MSE	6.0644	0.7681	**0.3556**
SSIM	0.9066	0.9643	**0.9829**

Conclusion

- Summary
  - Redistributing samples can improve filter selection
  - Global filter selection removes image seams

- Benefits
  - Works with arbitrary filters
  - No assumptions regarding scene and image content
  - Easy integration into existing rendering frameworks

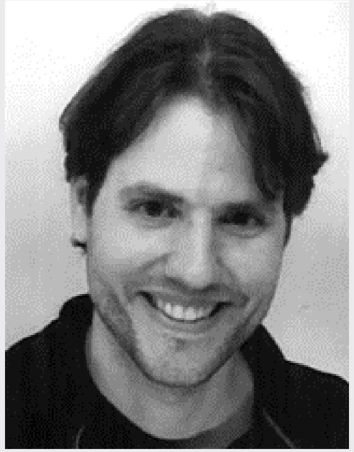
Outlook

- Investigate other interpolation schemes
- Adaptive sampling feedback loop
- Temporal coherence

Thank you for your attention!

Pablo Bauszat

Martin Eisemann



Elmar Eisemann

Marcus Magnor

Technische Universität Braunschweig

graphics.tu-bs.de

TU Delft
Delft University of Technology

graphics.tudelft.nl