EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

FACULTY OF
SCIENCE
Computer Graphics

Machine Learning in Graphics and Vision

- Global Illumination & ML -

SoSe 2018

Hendrik Lensch

Previous Work

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

	Non-linear Playback	Appearance Changes	Temporal Alignment	Handling Detours
a Evangelidis et al. (2011) 	✓	✗	✓	✗
b Torii et al. (2015) 	✗	✓	✗	✗
c Douze et al. (2016) 	✗	✓	✓	✗
d 	✓	✓	✓	✓

Efficiency: Sparse Feature-Extraction (260h Data)

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

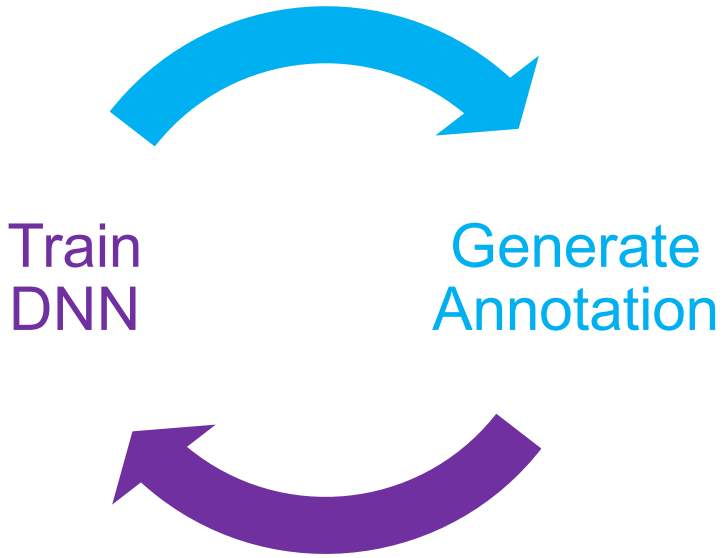
Approach	FPS	Total Time
Evangelidis et al. (2011)	0.11	0.81 years (~42 weeks)
Evangelidis et al. (2013)	0.11	0.81 years (~42 weeks)
Wang et al. (2014)	3.77	8.6 days
Ours	140	5.57 hours
Ours (12 GPUS)	12 * 140	28 minutes

x1272 speed up

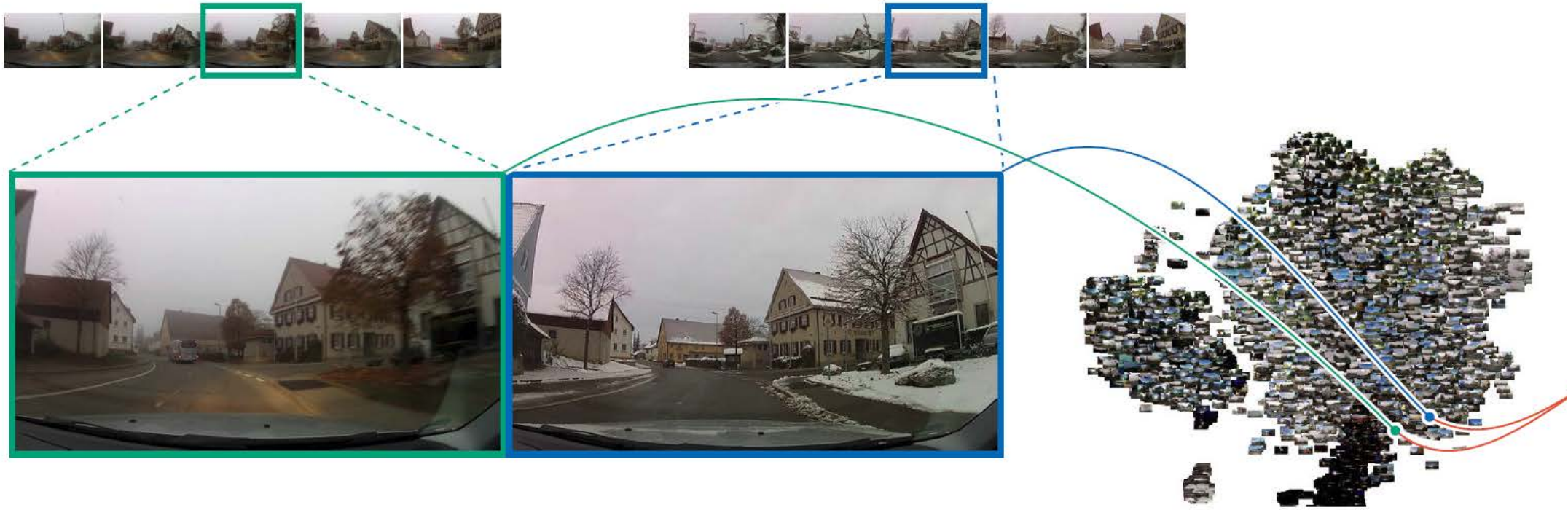
Data-Driven Approaches without Annotations

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Chicken-Egg-Problem



Encoding each Frame

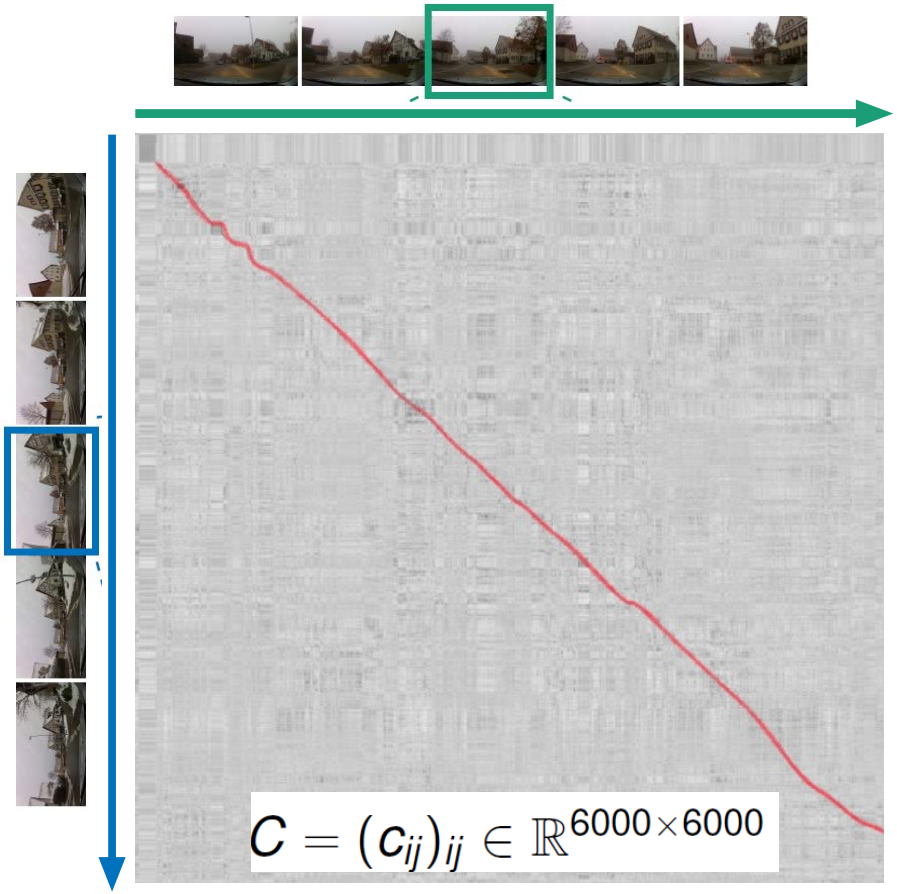


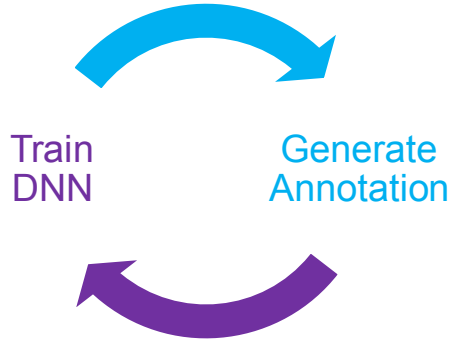
- Visualization (t-SNE): mapping 1024 dimensions to 2 dimensions
- Euclidean distance between two frames in high-dimensional encoding-space represents their similarity.

Phase: Generate Annotation

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- Assume trained Deep Neural Network is given



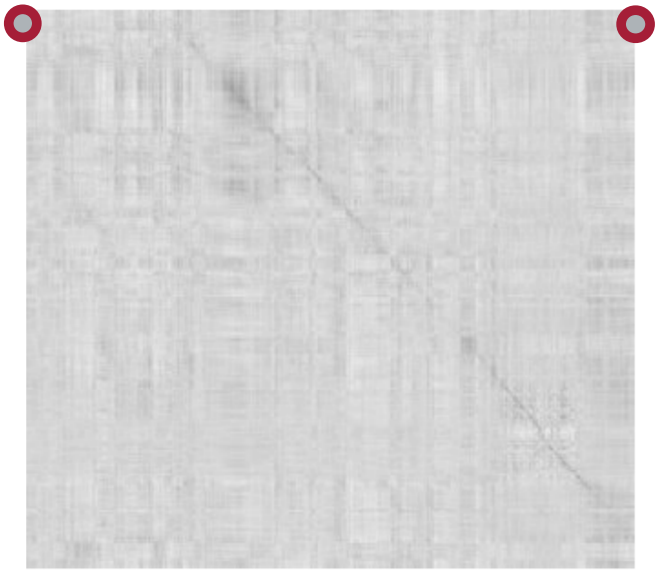


1. pair-wise distances C‘
2. low-rank approximation
3. de-correlated distances C
4. find syncronization path

Finding a Synchronization Path

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

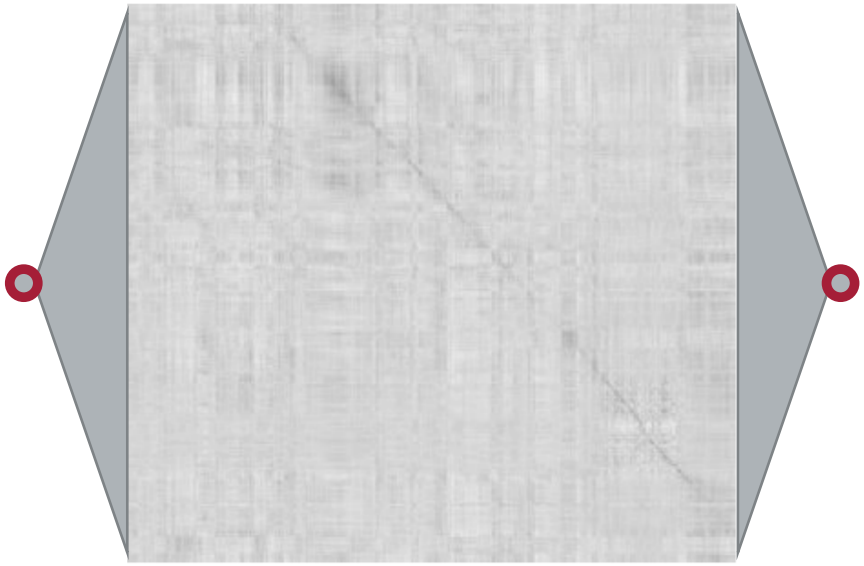
- Dijkstra's algorithm finds shortest path between two given nodes.
- We do not know the start or end node!



Finding a Synchronization Path

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

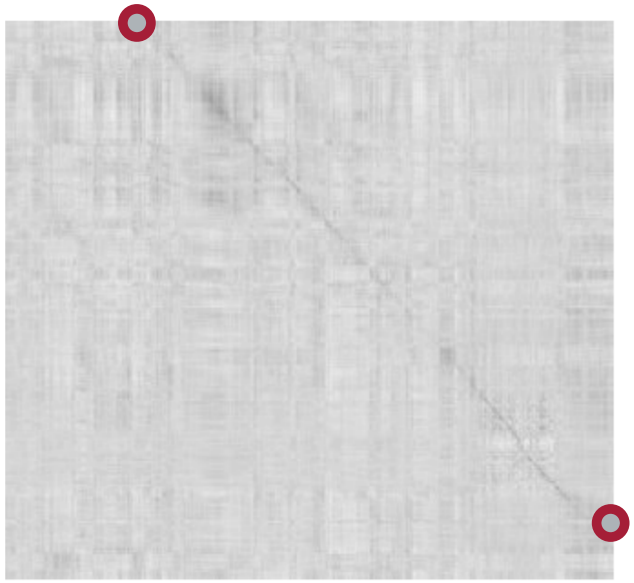
- Dijkstra's algorithm finds shortest path between two given nodes.
- We do not know the start or end node!



Finding a Synchronization Path

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

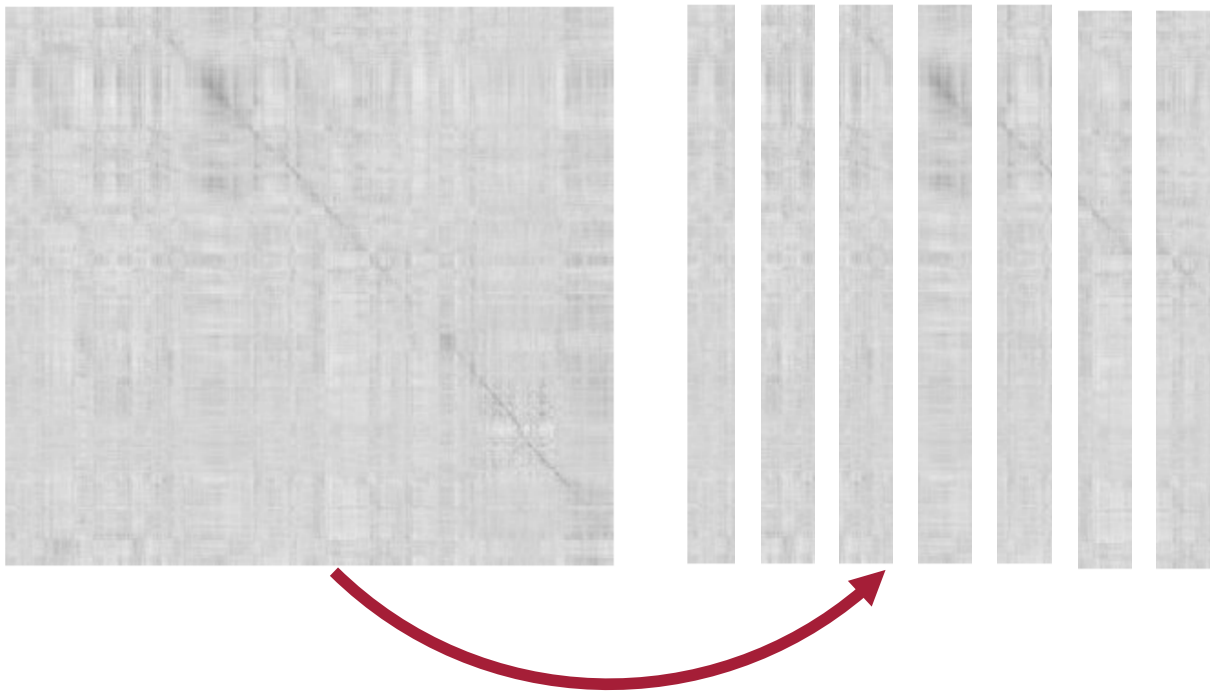
- Dijkstra's algorithm finds shortest path between two given nodes.
- We do not know the start or end node!
- We even do not know if there is any path!



Finding a Synchronization Path

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

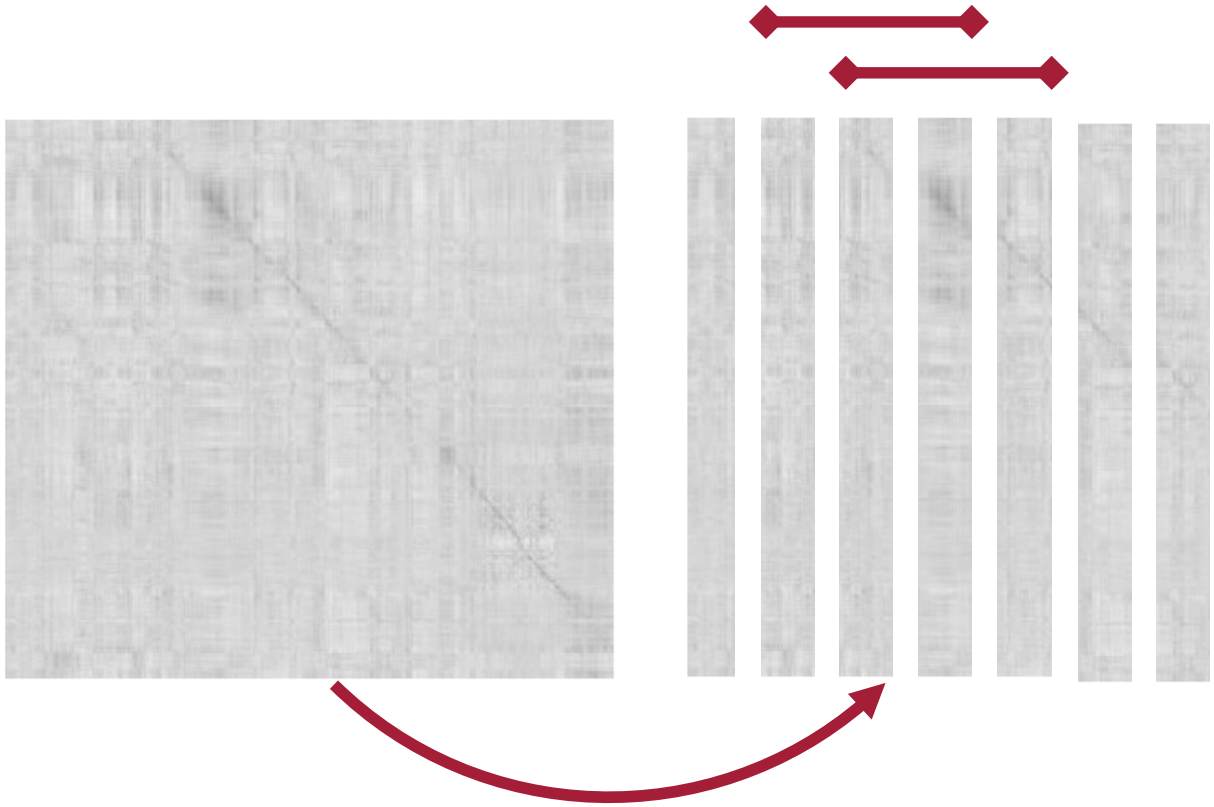
- Dijkstra's algorithm finds shortest path
- Partition into strips



Finding a Synchronization Path

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

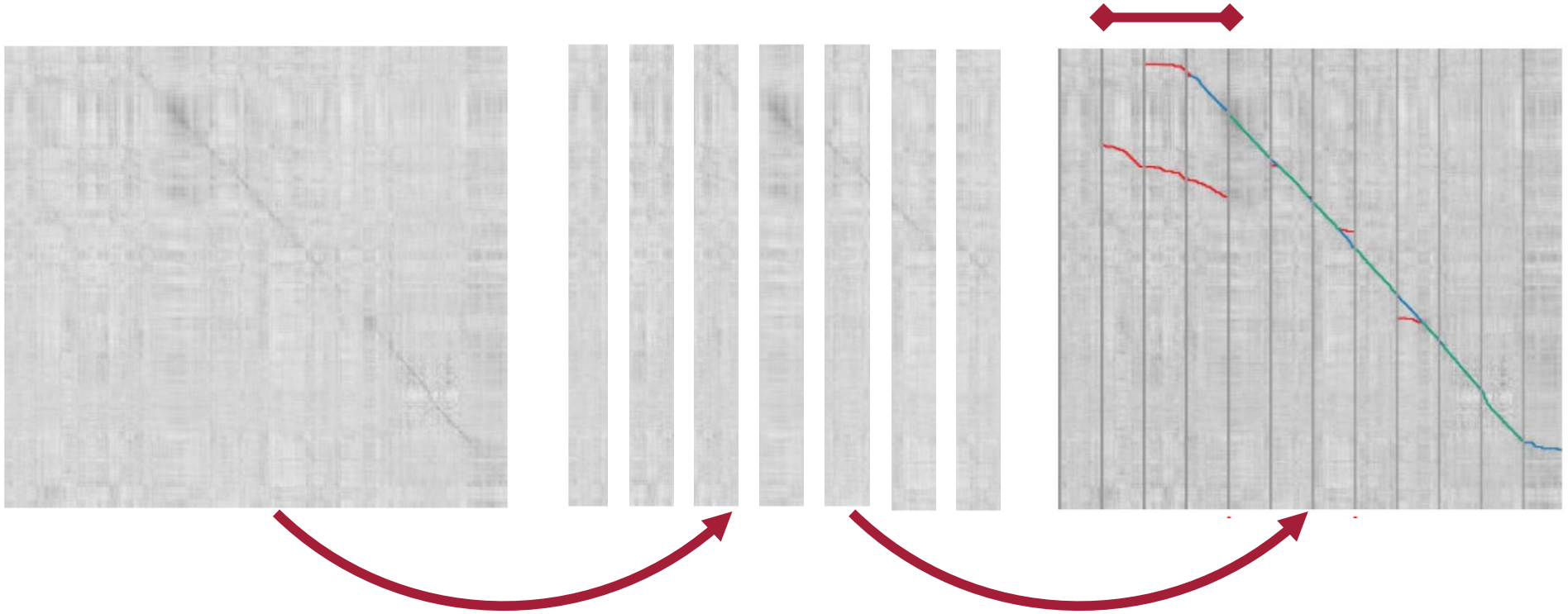
- Dijkstra's algorithm finds shortest path
- Partition into strips
- Find shortest path for overlapping strips



Finding a Synchronization Path

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

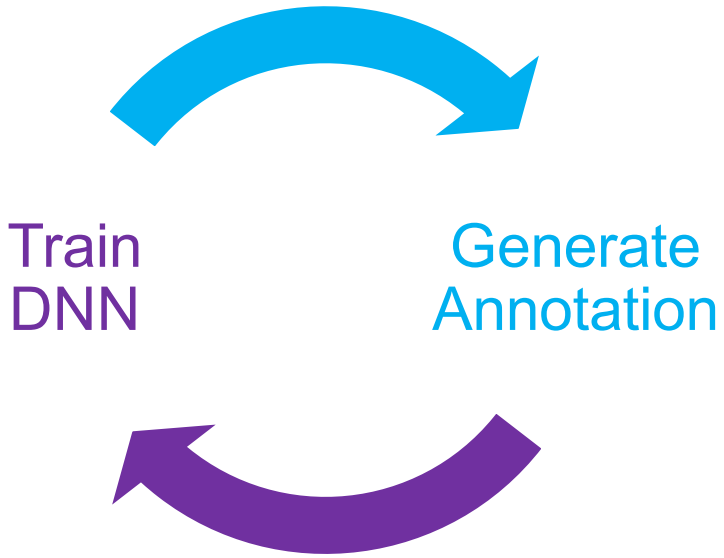
- Dijkstra's algorithm finds shortest path
- Partition into strips
- Find shortest path for overlapping strips
- Judge consistency in overlap regions



Data-Driven Approaches without Annotations

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

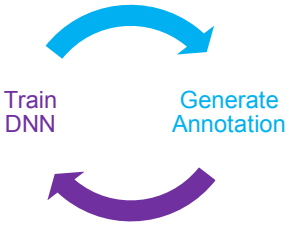
- Chicken-Egg-Problem

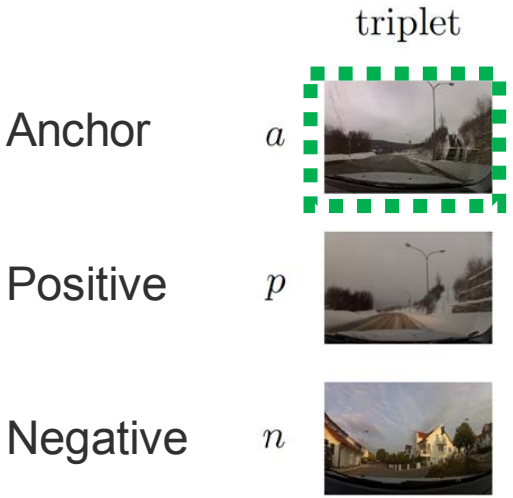


Phase: Train DNN

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

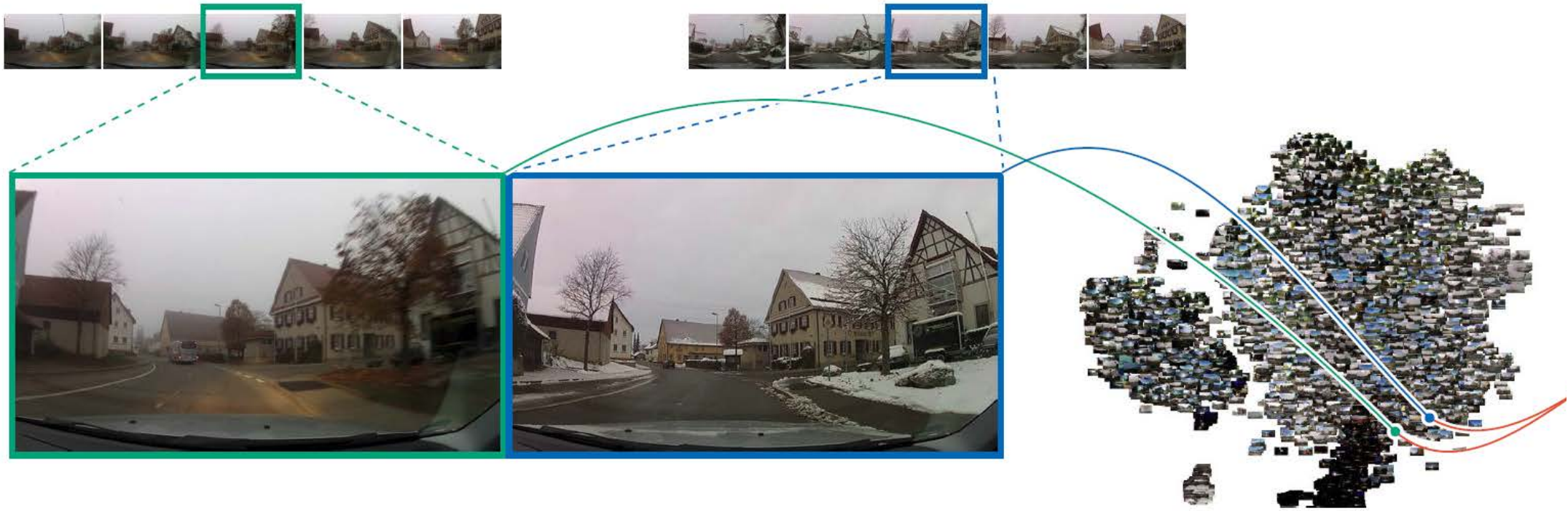
- Assume ground-truth annotations are given
- Optimize triplet loss





$$\max\{0, m + \|\Phi_a - \Phi_p\|_2^2 - \|\Phi_a - \Phi_n\|_2^2\}$$

Encoding each Frame

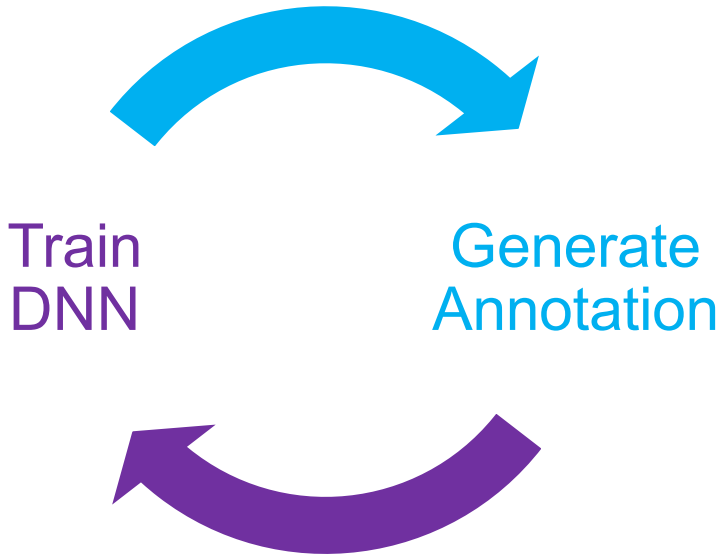


- Encoding clearly is not perfect
- Interplay of CNN encoding and shortest path on cleaned matrix render it a stable process

Data-Driven Approaches without Annotations

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

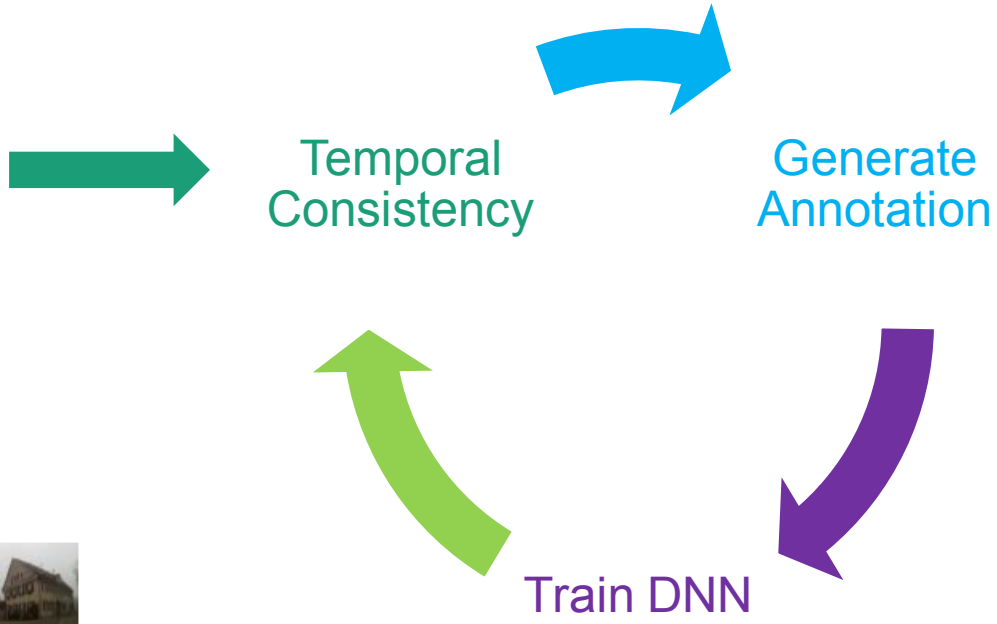
- Chicken-Egg-Problem



Data-Driven Approaches without Annotations

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Solve the Chicken-Egg-Problem

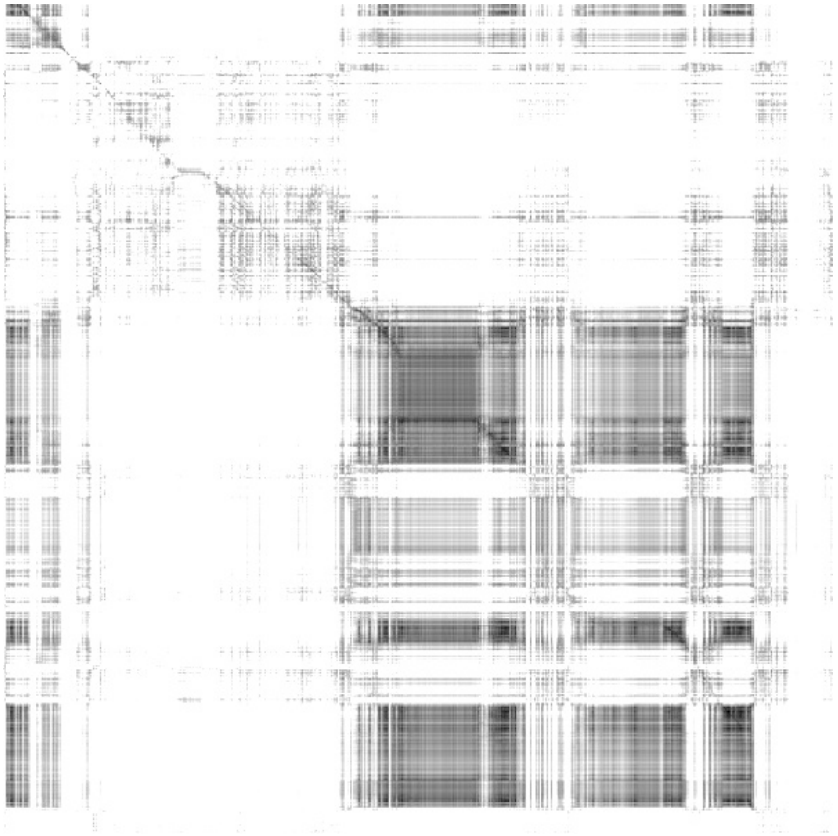


- Causality of Time

Curriculum Learning

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Start with examples which are not too difficult and not too obvious:



1. Intra-Video Sampling
   consecutive frames are similar
2. Inter-Video Sampling
   ensure temporal consistency to reject false positives
3. Transitive Inter-Video Sampling
   $v/_{\sim}$ is equivalence relation $a \sim b \sim c$

Important Features

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Why are local features not useful?
- What is the network looking for?

HARRIS

used in Evangelidis et al. 2011, 2013

SIFT

used in Wang et al. 2014

Learned Features (Saliency Maps)

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Input | ResNet (ImageNet) | Ours

Saliency Maps

Robustness

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

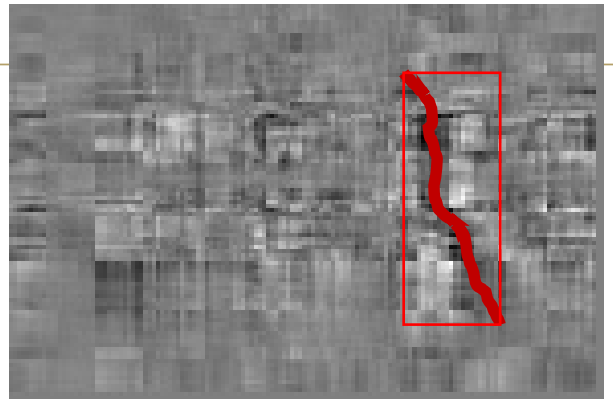
after tree-felling

before tree-felling

Application to other Events

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Golf swing – same neural network
- Event boundaries visible



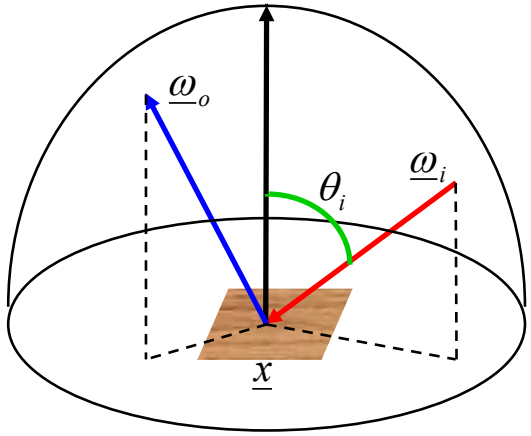
Global Illumination

Surface Radiance

$$L(\underline{x},\underline{\omega}_o) = L_e(\underline{x},\underline{\omega}_o) + L_r(\underline{x},\underline{\omega}_o)$$

$$= L_e(\underline{x},\underline{\omega}_o) + \int_{\Omega_+} f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o) L(\underline{x},\underline{\omega}_i) \cos\theta_i \, d\underline{\omega}_i$$

- Visible surface radiance $L(\underline{x},\underline{\omega}_o)$
  - Surface position $\underline{x}$
  - Outgoing direction $\underline{\omega}_o$
  - Incoming illumination direction $\underline{\omega}_i$
- Self-emission $L_e(\underline{x},\underline{\omega}_o)$
- Reflected light
  - Incoming radiance from all directions $L_i(\underline{x},\underline{\omega}_i)$
  - Direction-dependent reflectance (BRDF: bidirectional reflectance distribution function) $f_r(\underline{\omega}_i,\underline{x},\underline{\omega}_o)$



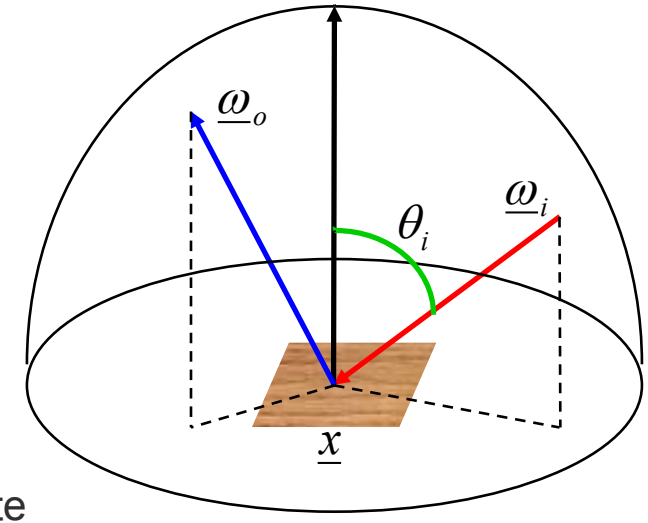
(Surface) Rendering Equation

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- In Physics: Radiative Transport Equation
- Expresses energy equilibrium in scene

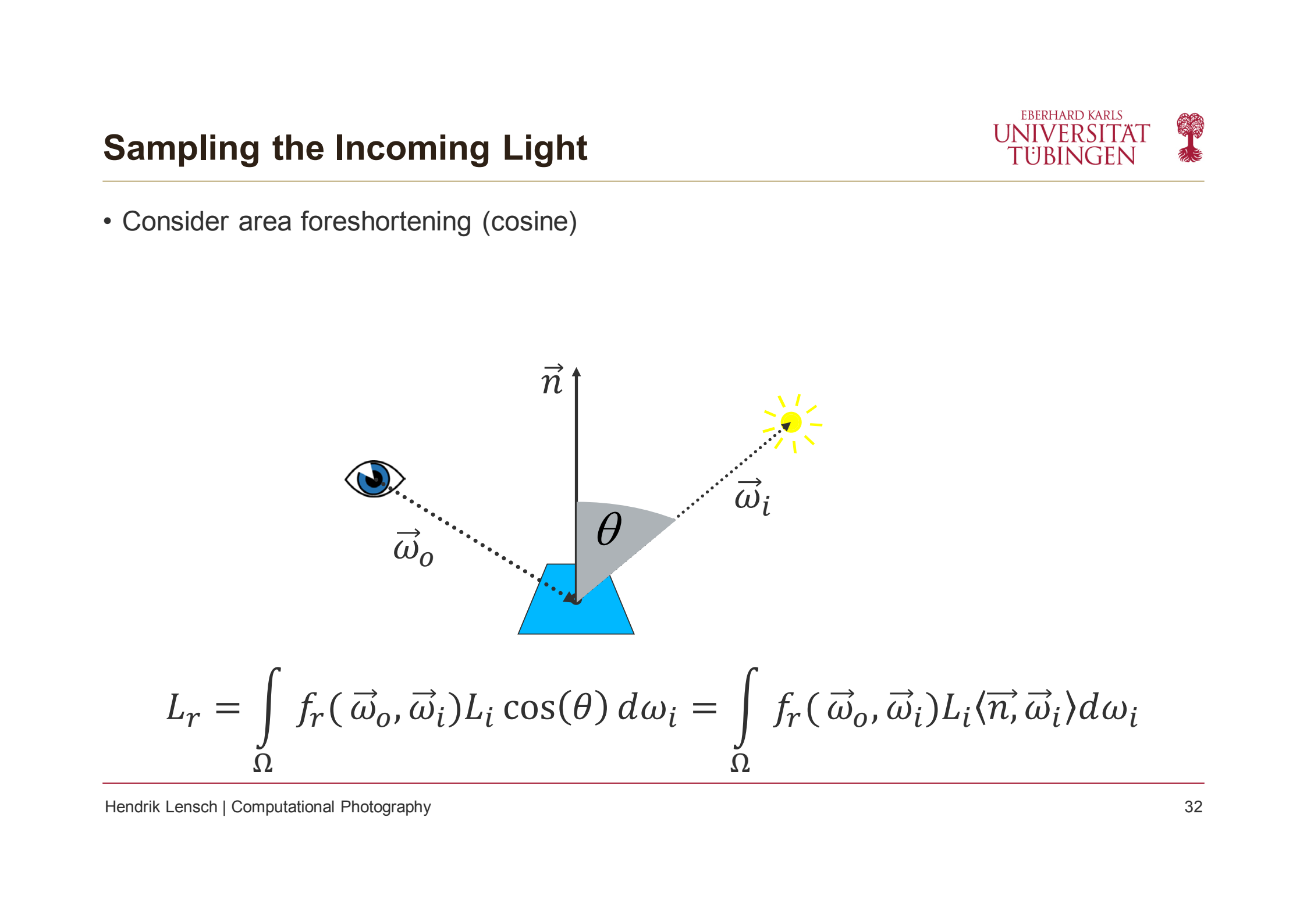
$$L(\underline{x}, \underline{\omega}_o) = L_e(\underline{x}, \underline{\omega}_o) + \int_{\Omega} f_r(\underline{\omega}_i, \underline{x}, \underline{\omega}_o)\, L(\underline{x}, \underline{\omega}_i) \cos\theta_i \, d\underline{\omega}_i$$

- total radiance = e[illegible]d radiance + reflected radiance
- First term: emissivity of th[illegible]rface
  - non-zero only for light sources
- Second term: reflected radiance
  - integral over all possible incoming directions of irradiance times angle-dependent surface reflection function
- Fredholm integral equation of 2nd kind
  - unknown radiance appears on lhs and inside the integral
  - Numerical methods necessary to compute approximate solution



Sampling the Incoming Light

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

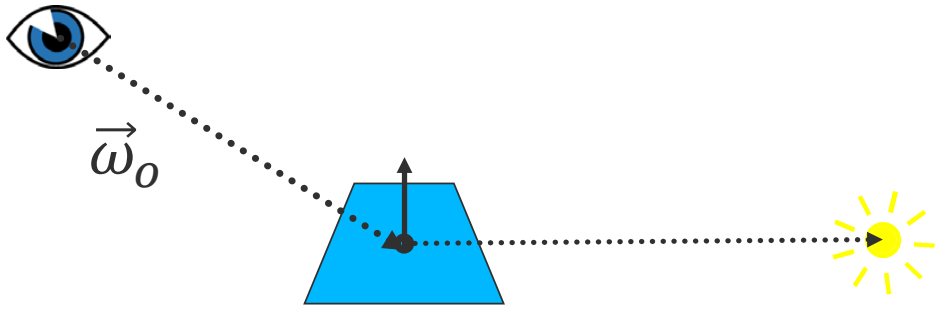


- Consider area foreshortening (cosine)

$$L_r = \int_\Omega f_r(\vec{\omega}_o, \vec{\omega}_i) L_i \cos(\theta)\, d\omega_i = \int_\Omega f_r(\vec{\omega}_o, \vec{\omega}_i) L_i \langle \vec{n}, \vec{\omega}_i \rangle d\omega_i$$

Sampling the Incoming Light

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

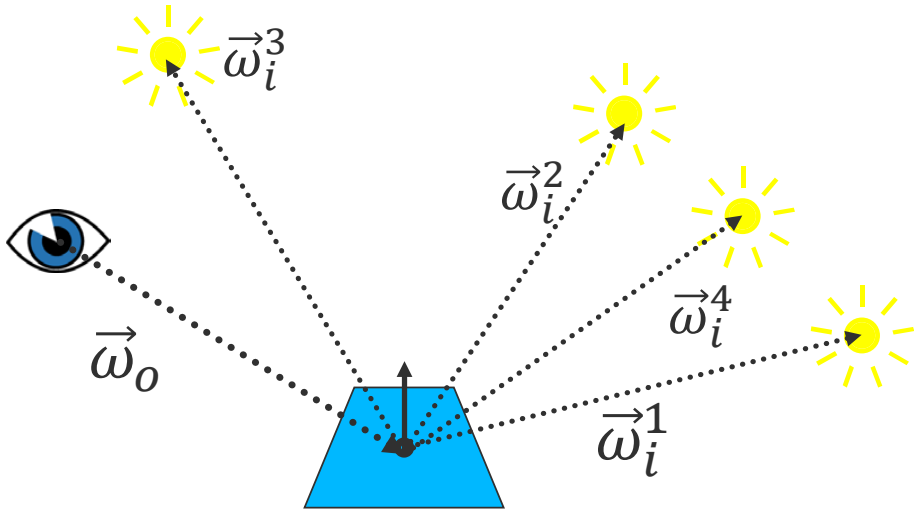


- Continuous integral

$$L_r = \int_{\Omega} f_r(\vec{\omega}_o, \vec{\omega}_i) L_i \cos(\theta)\, d\omega_i$$

Sampling Light Sources

- Consider discrete samples of the path space
- Multiple Light Sources:
  - add their contributions



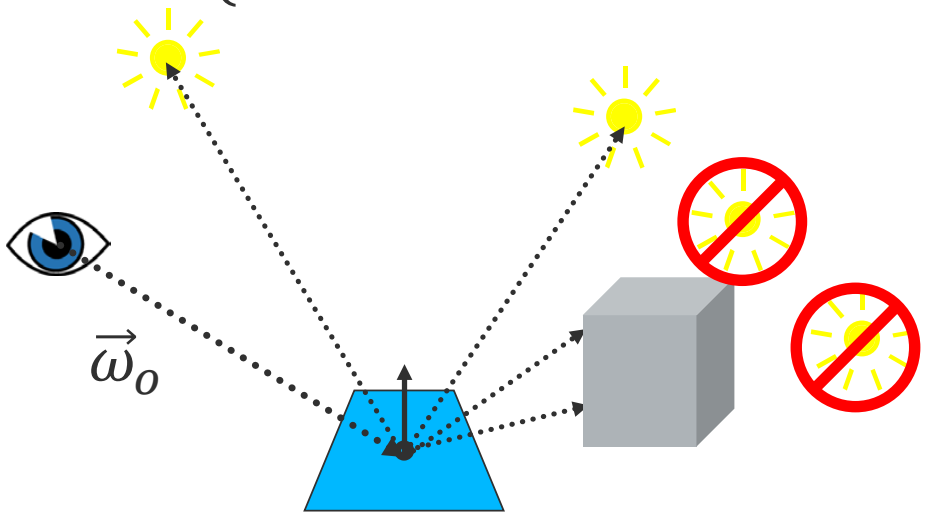
$$L_r = \sum_N f_r(\vec{\omega}_o, \vec{\omega}_i) L_i^k \cos(\theta^k) \approx \int_\Omega f_r(\vec{\omega}_o, \vec{\omega}_i) L_i \cos(\theta)\, d\omega_i$$

Sampling Light Sources w/ Occlusions

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Visibility function

$$v(\vec{\omega}_i) = \begin{cases} 0 & \text{source is occluded} \\ 1 & \text{source is visible} \end{cases}$$



$$L_r = \sum_N f_r(\vec{\omega}_o, \vec{\omega}_i) v(\vec{\omega}_i) L_i^k \cos\left(\theta^k\right)$$

Approximation – Ambient Occlusion

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- Aproximation for global illumination effects
- Only consider visibility in a uniformly lit room
- AO = fraction of the hemisphere that is not occluded

$$L_r = \int_\Omega f_r(\vec{\omega}_o, \vec{\omega}_i) v(\vec{\omega}_i) L_i \cos(\theta)\, d\omega_i$$

$$L_r \approx L_{AO} = \rho \cdot AO \cdot \int_\Omega L_i \cos(\theta)\, d\omega_i = \rho \cdot AO \cdot E$$

$$AO = \frac{1}{2\pi} \int_\Omega v'(\vec{\omega}_i) d\omega_i \omega$$

$$v'(\vec{\omega}_i) = v(\vec{\omega}_i)$$

up to some max. distance

Example: Lighting

$$\mathrm{L_r} \approx \mathrm{L_{AO}} = \rho \cdot AO \cdot \int_{\Omega} L_i \cos(\theta) d\omega_i = \rho \cdot AO \cdot E$$

Example: Lighting & AO

$$\mathrm{L_r} \approx \mathrm{L_{AO}} = \rho \cdot AO \cdot \int_{\Omega} L_i \cos(\theta) d\omega_i = \rho \cdot AO \cdot E$$

Example: Ligting & Shading & AO

$$\mathrm{L_r} \approx \mathrm{L_{AO}} = \rho \cdot AO \cdot \int_{\Omega} L_i \cos(\theta) d\omega_i = \rho \cdot AO \cdot E$$

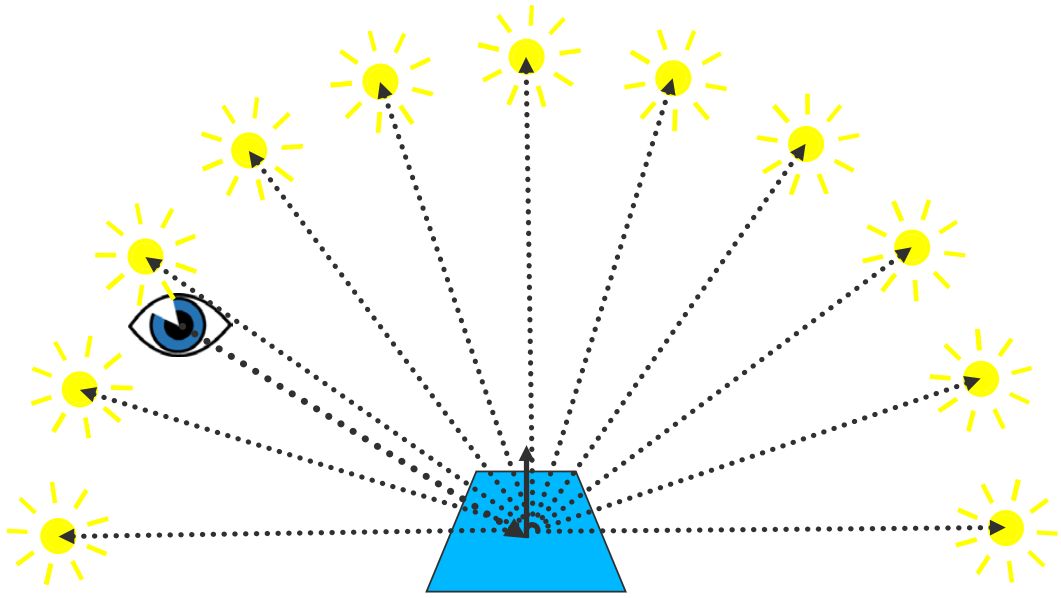
Example: Ligting & Shading

$$\mathrm{L_r} \approx \mathrm{L_{AO}} = \rho \cdot AO \cdot \int_{\Omega} L_i \cos(\theta) d\omega_i = \rho \cdot AO \cdot E$$

Ambient Occlusion

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

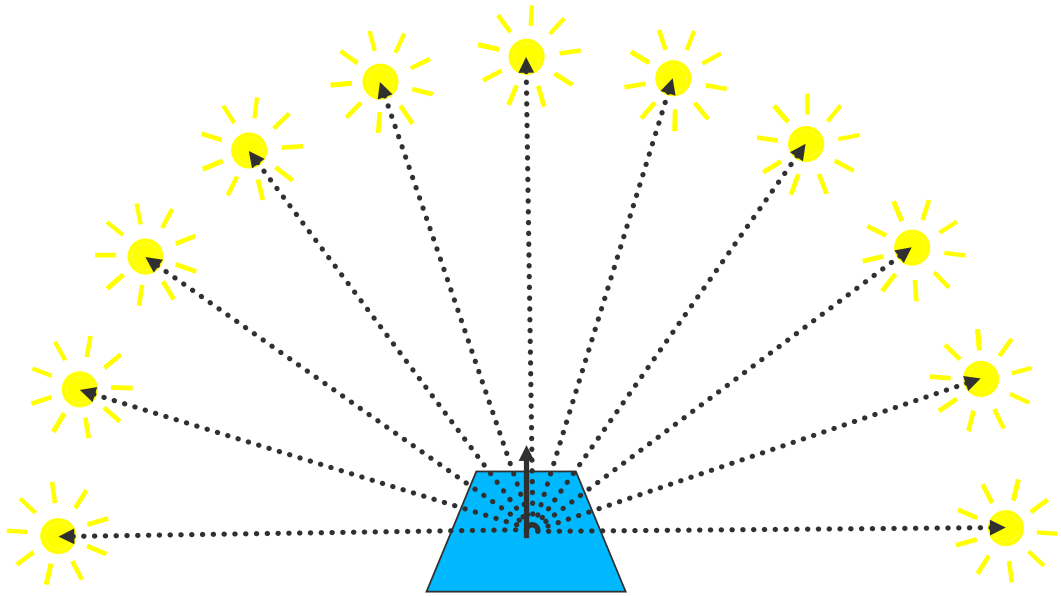
- Only consider visibility in a uniformly lit room
- AO = fraction of the hemisphere that is not occluded



$$AO = \frac{1}{N} \sum_{N} v(\vec{\omega}_i^k)$$

Ambient Occlusion

- Fraction of the hemisphere that is not occluded
- Independent on viewing direction

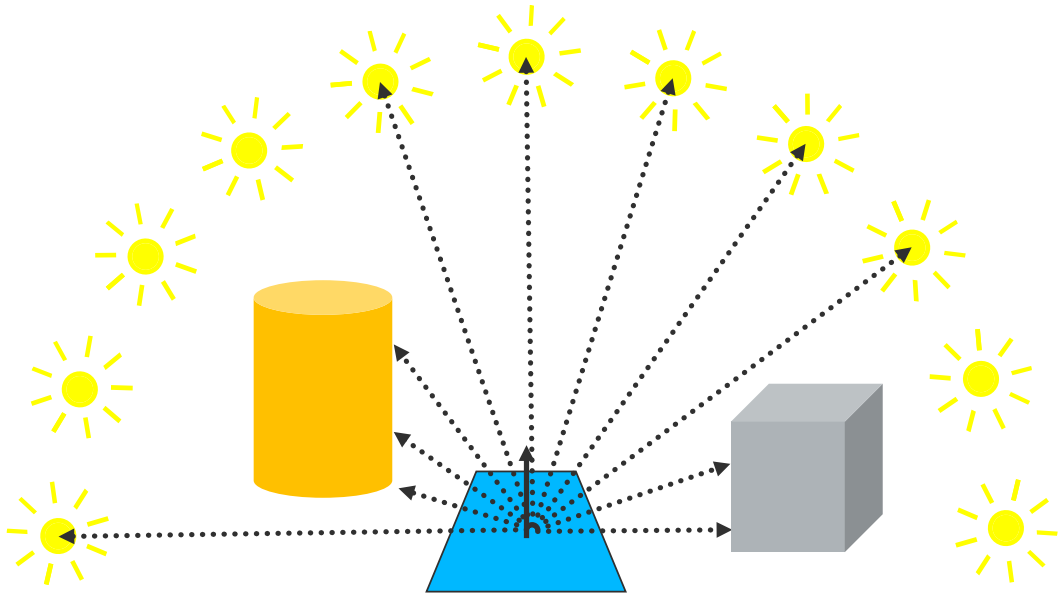


$$AO = \frac{1}{N}\sum_{N} v(\vec{\omega}_i^k) = 1$$ completely unoccluded

Ambient Occlusion

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Fraction of the hemisphere that is not occluded

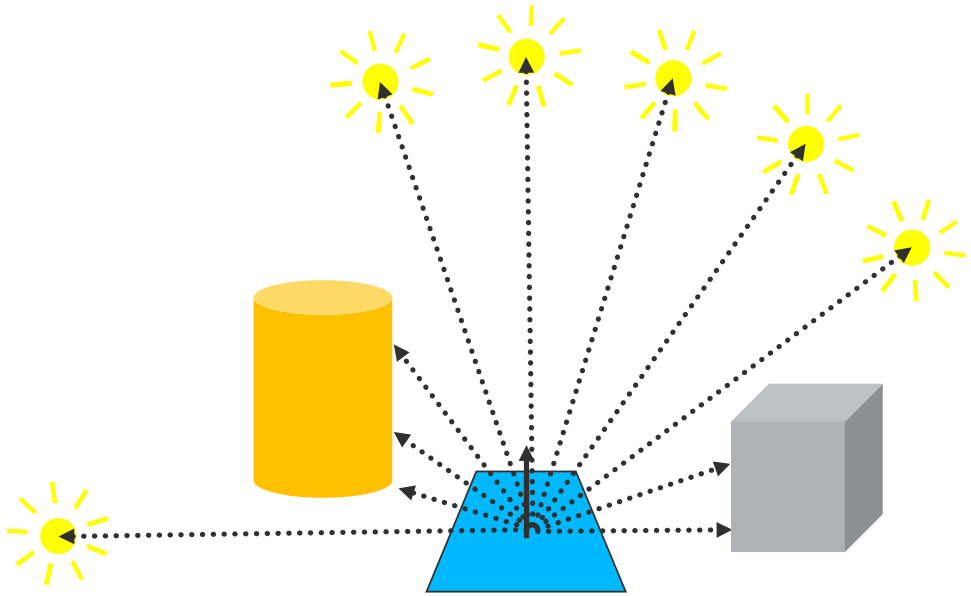


$$AO = \frac{1}{N}\sum_{N} v(\vec{\omega}_i^k)$$

Ambient Occlusion

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Fraction of the hemisphere that is not occluded

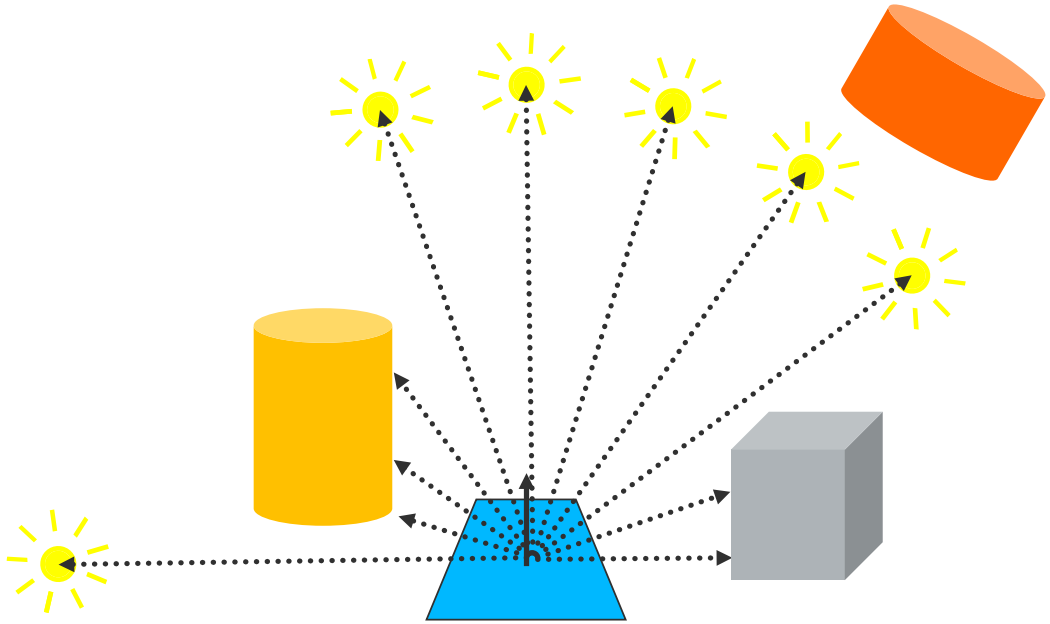


$$AO = \frac{1}{N} \sum_{N} v(\vec{\omega}_i^k)$$

Ambient Occlusion

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Fraction of the hemisphere that is not occluded
- Only consider close by objects



$$AO = \frac{1}{N} \sum_{N} v(\vec{\omega}_i^k)$$

Ambient Occlusion

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Almost looks like global light transport

The problem

- Variance appears as noise at low sample rates

scene by Wojciech Jarosz

General Framework for Monte Carlo Filtering

scene by Herminio Nieves and tf3dm.com user ysup12

Observations

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- After only a few minutes, we have a lot of information about the scene…

scene rendered in a few minutes (8 samples/pixel)

scene by M. Leal Llaguno

Observations

- Can we filter this to get a result as good as a reference image?

scene rendered in over 24 hrs (8,192 samples/pixel)

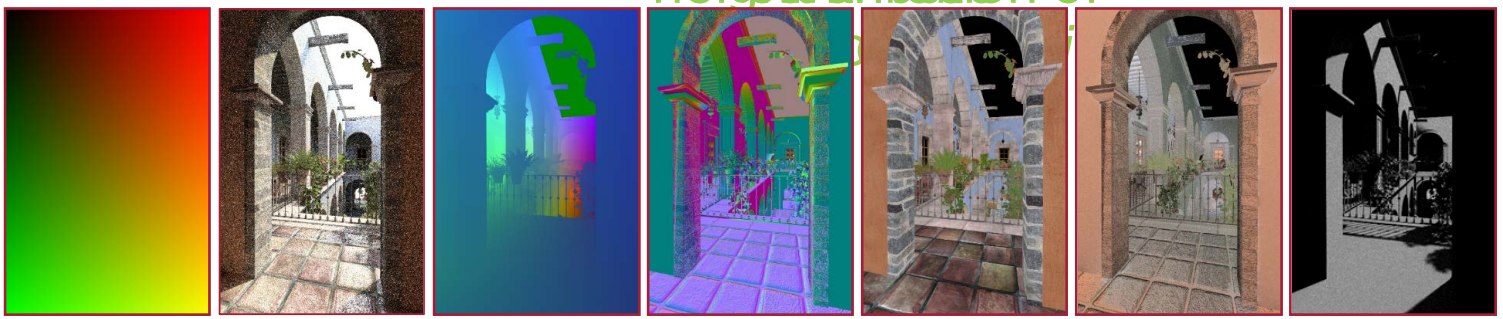
scene by M. Leal Llaguno

Filter

$$\hat{\mathbf{c}}_i = \frac{\sum_{j \in \mathcal{N}(i)} d_{i,j} \bar{\mathbf{c}}_j}{\sum_{j \in \mathcal{N}(i)} d_{i,j}}$$

← noisy pixel

neighborhood of



screen position | pixel color | world positions | shading normals | texture 1st bounce | texture 2nd bounce | visibility

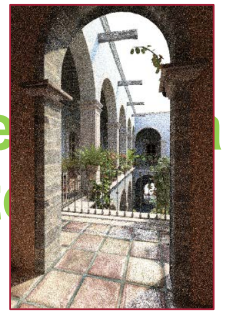
Cross-bilateral filter

$$\hat{\mathbf{c}}_i = \frac{\sum_{j \in \mathcal{N}(i)} d_{i,j} \bar{\mathbf{c}}_j}{\sum_{j \in \mathcal{N}(i)} d_{i,j}}$$

$$d_{i,j} = \exp\left[- \frac{\|\bar{\mathbf{p}}_i - \bar{\mathbf{p}}_j\|^2}{2\alpha_i^2} \right] \qquad \beta_i^2 \qquad \gamma_{k,i}^2$$

- α_i^2, β_i^2, $\gamma_{k,i}^2$ are [illegible]riance of ea[illegible]
  term[illegible] parameter[illegible]

screen position

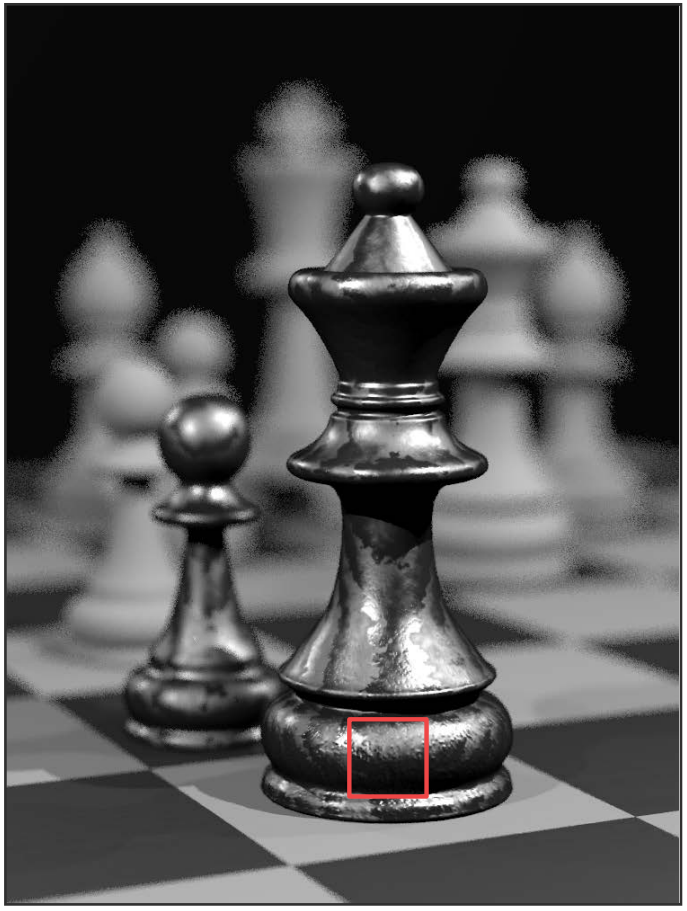


pixel color

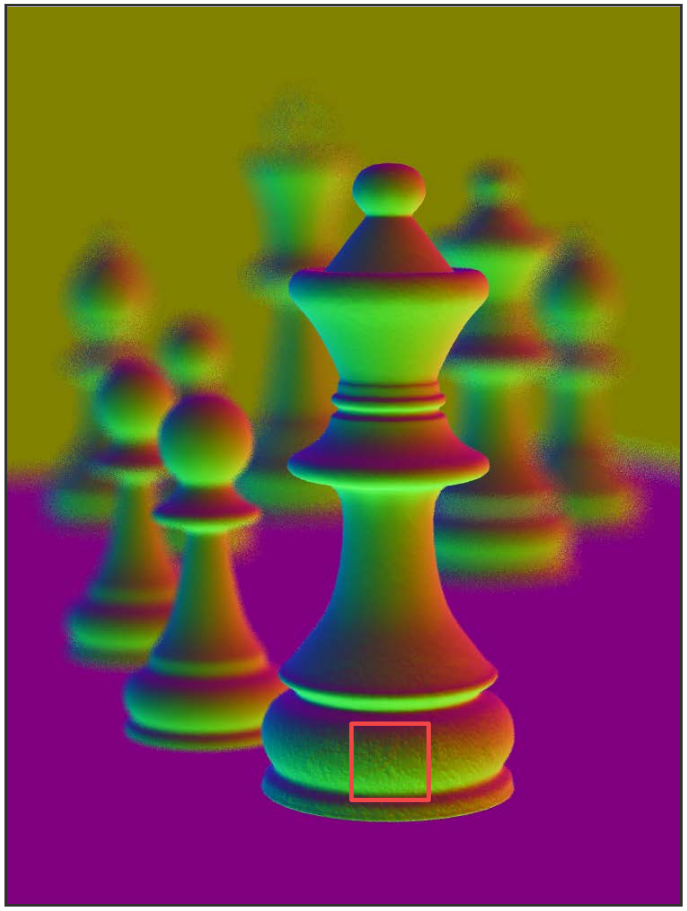
additional features

Why per pixel parameters?

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

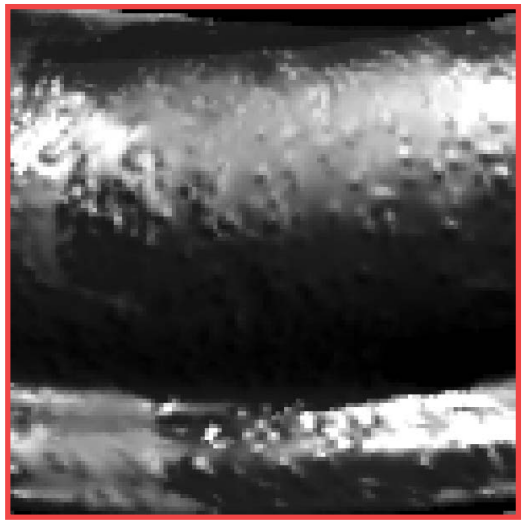


input MC 8 samples/pixel

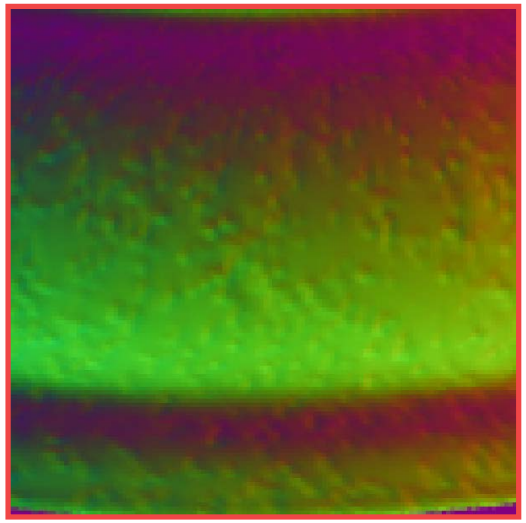


shading normals

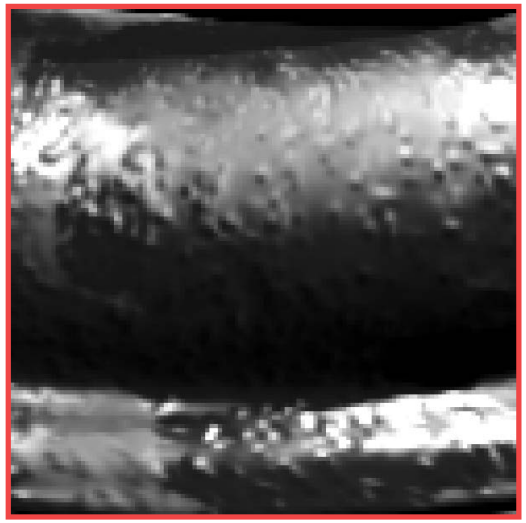
Why per pixel parameters?



input MC
8 samples/pixel



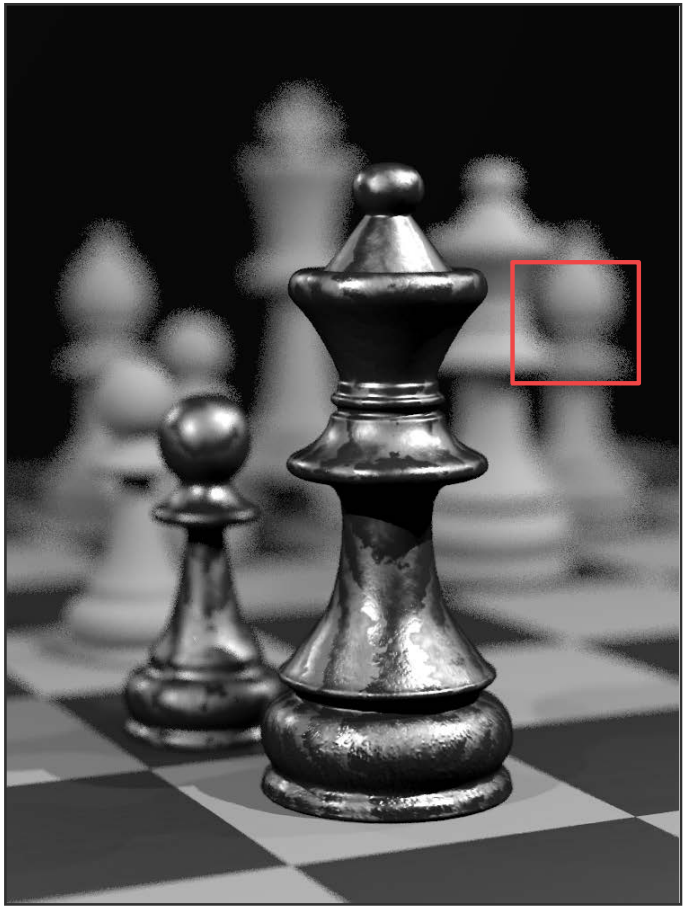
shading normals



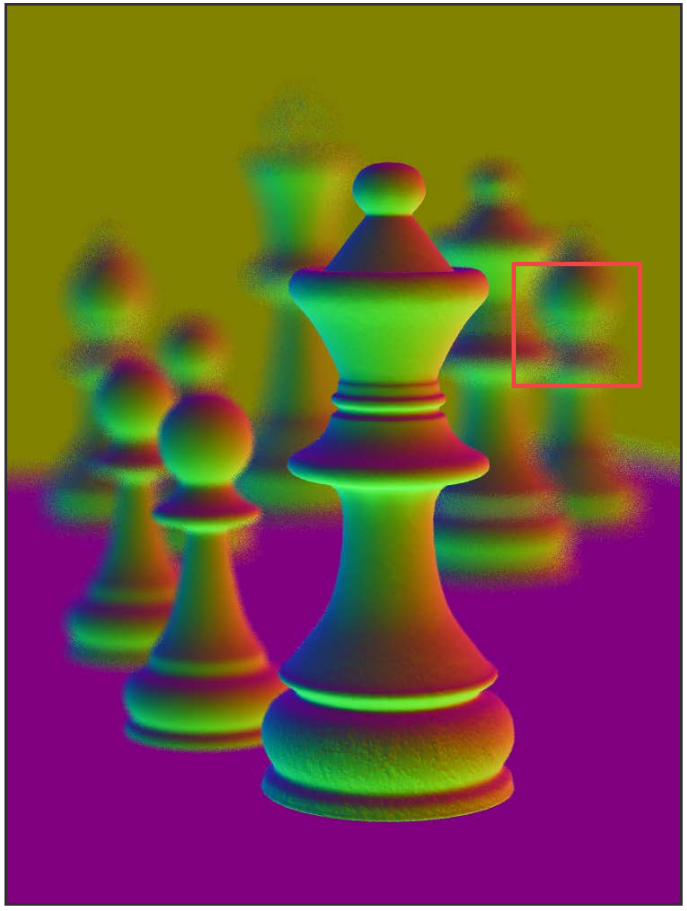
ground truth

Why per pixel parameters?

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN



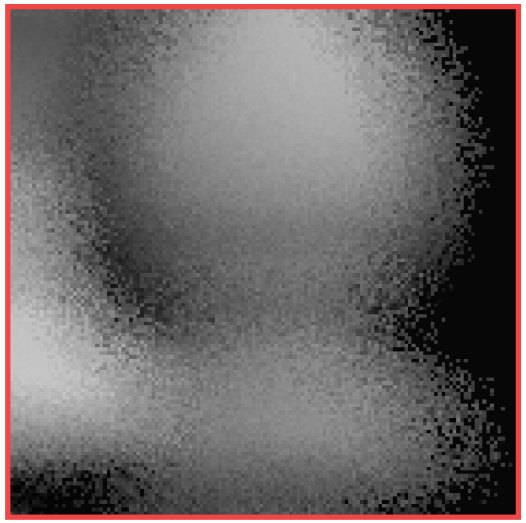
input MC 8 samples/pixel



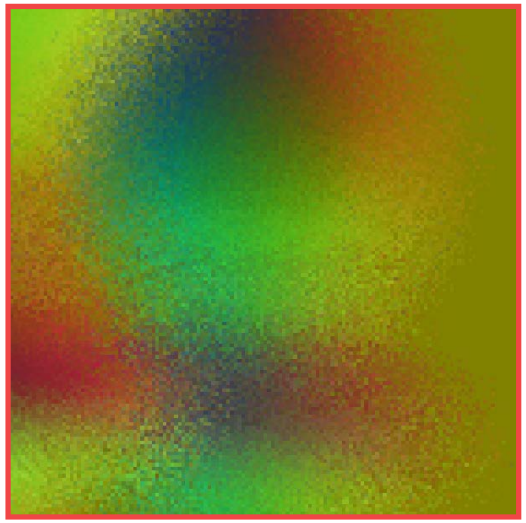
shading normals

Why per pixel parameters?

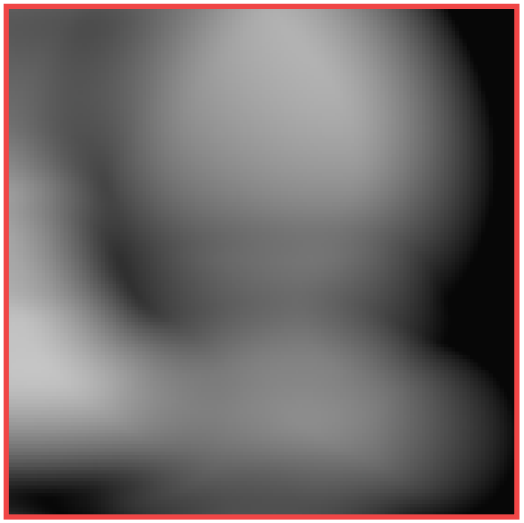
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN



input MC
8 samples/pixel

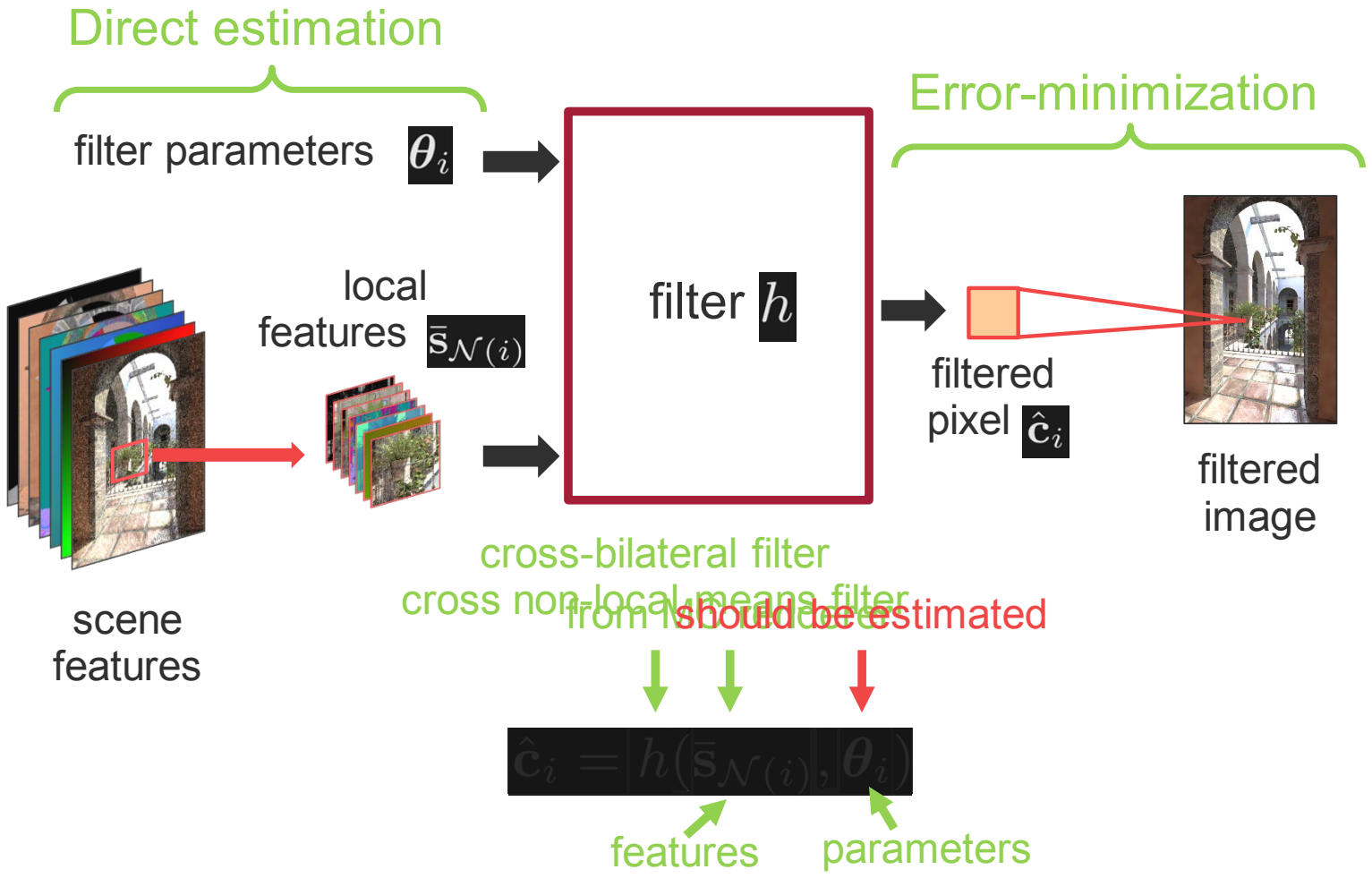


shading normals



ground truth

Problem formulation



Our result with a cross-bilateral filter (4 spp)

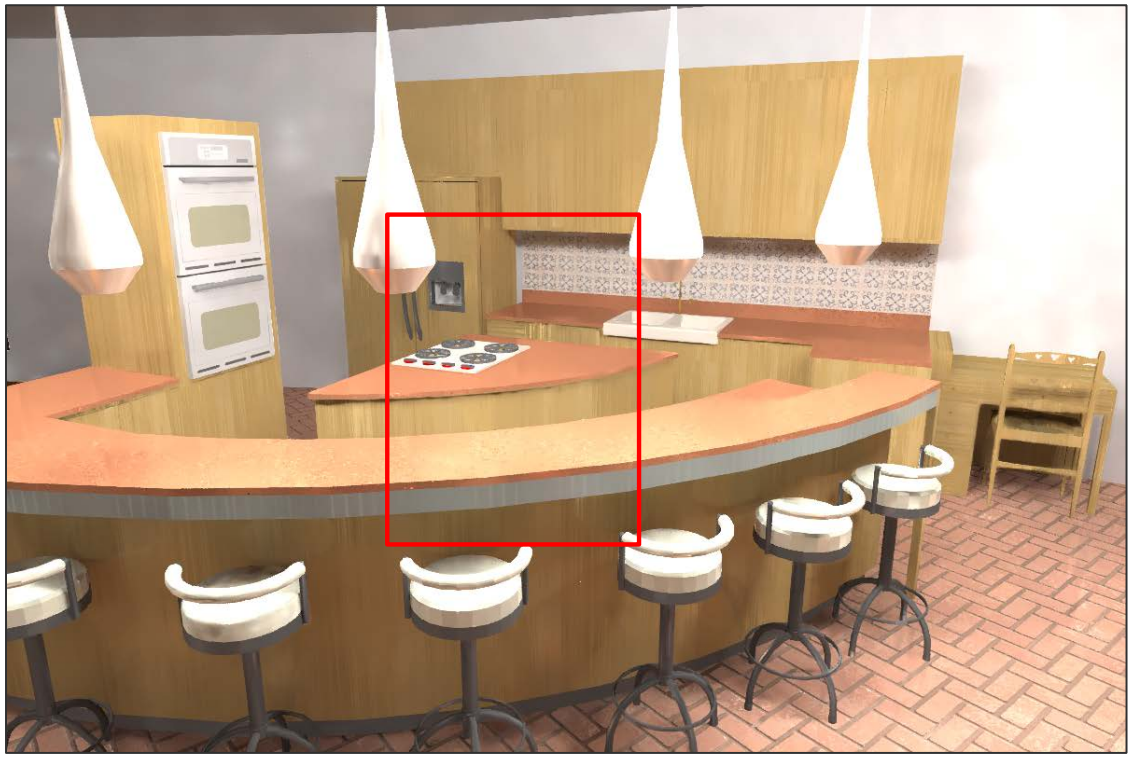
Our result with a non-local means filter (4 spp)

A Machine Learning Approach for Filtering Monte Carlo Noise

Kalantari et al. SIGGRAPH 2015

Sample result

[Kalantari et al. 2015]

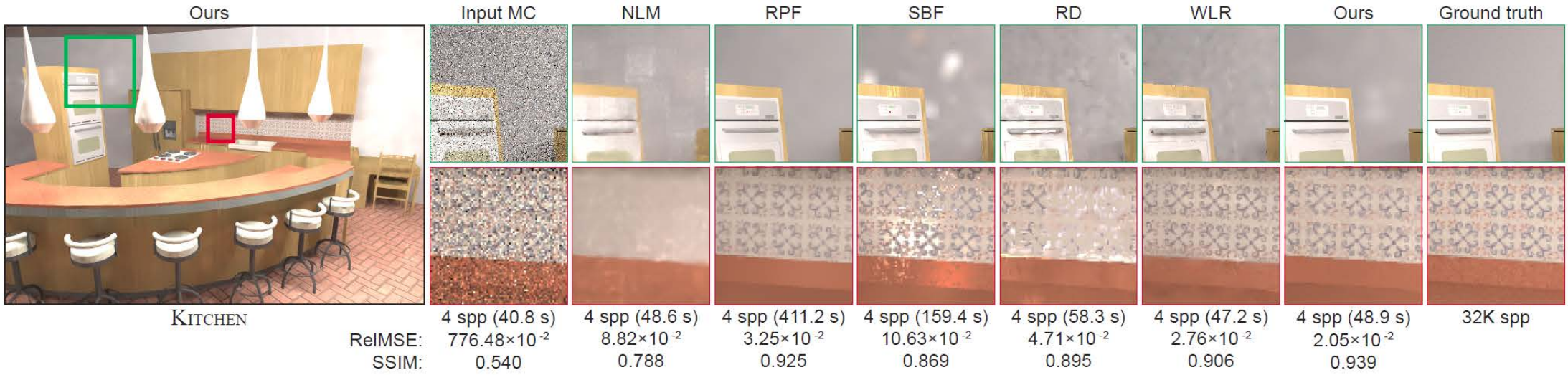


scene by Jo Ann Elliott

4 samples/pixel
(48.9 sec)

using only post-process filter!

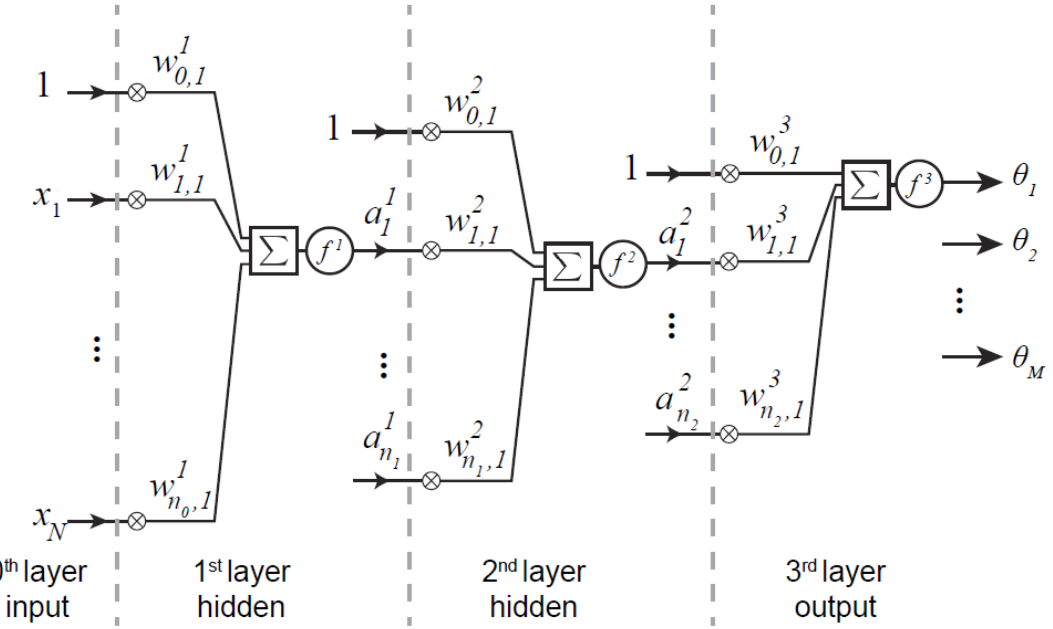
Results


Ours
Input MC
NLM
RPF
SBF
RD
WLR
Ours
Ground truth
KITCHEN
RelMSE:
SSIM:
4 spp (40.8 s)
776.48×10 -2
0.540
4 spp (48.6 s)
8.82×10 -2
0.788
4 spp (411.2 s)
3.25×10 -2
0.925
4 spp (159.4 s)
10.63×10 -2
0.869
4 spp (58.3 s)
4.71×10 -2
0.895
4 spp (47.2 s)
2.76×10 -2
0.906
4 spp (48.9 s)
2.05×10 -2
0.939
32K spp

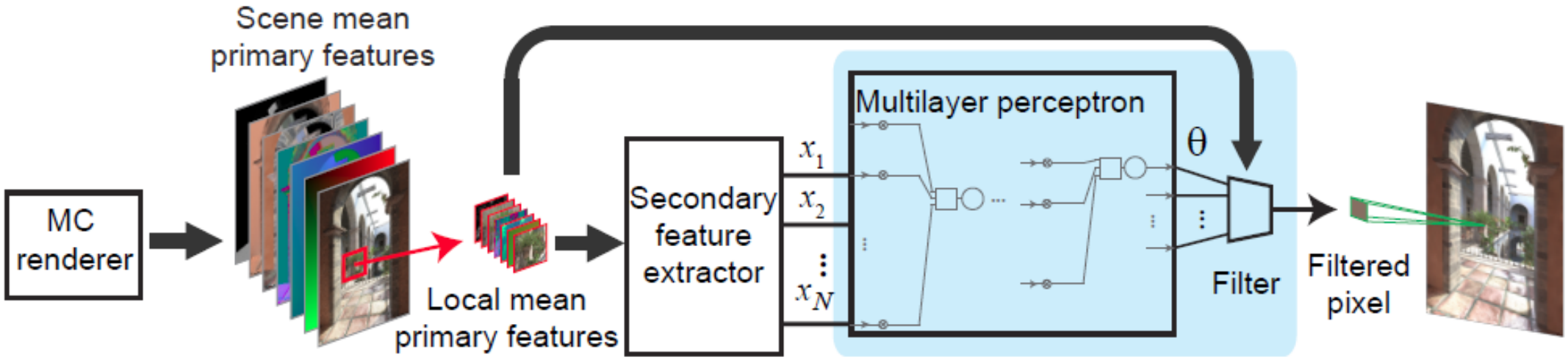
MLP for Denoising

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Simple 3-layer MLP



Incorporating primary Features



Denoising with Kernel Prediction and Asymmetric Loss Functions

Vogels et al. SIGGRAPH 2018

Kernel-predicting Convolutional Network (KPCN)

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- training to minimize the average distance between the ground-truth r and the denoised image $d = g(x;\ \theta)$ via a loss function l

$$\widehat{\boldsymbol{\theta}} = \underset{\boldsymbol{\theta}}{\operatorname{argmin}} \frac{1}{N} \sum_{n=1}^{N} \ell(\mathbf{r}^n, g(\mathbf{x}^n; \boldsymbol{\theta}))$$

- symmetric mean absolute percentage error

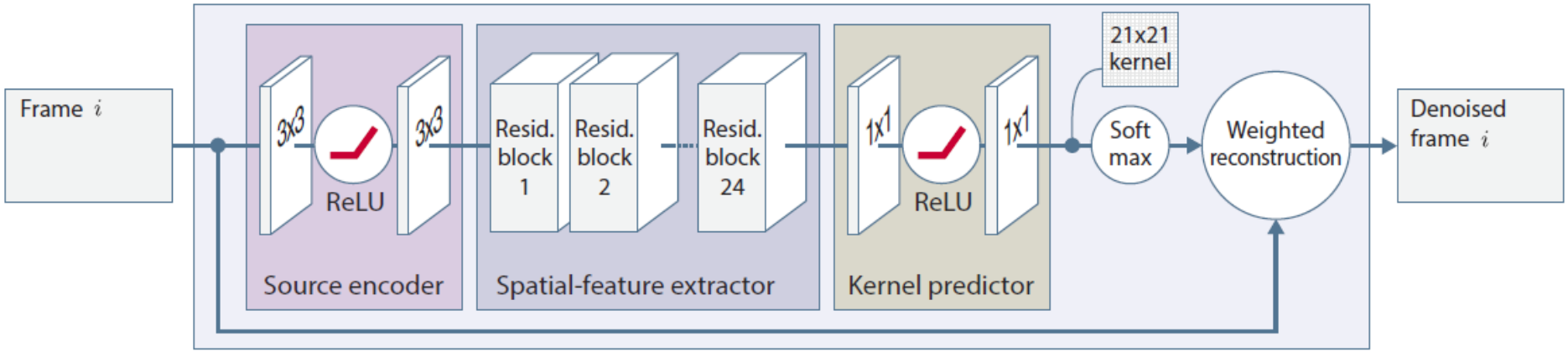
$$\ell(\mathbf{r}, \mathbf{d}) = \frac{1}{3|\mathcal{I}|} \sum_{p \in \mathcal{I}} \sum_{c \in \mathcal{C}} \frac{\left|\mathbf{d}_{p,c} - \mathbf{r}_{p,c}\right|}{\left|\mathbf{d}_{p,c}\right| + \left|\mathbf{r}_{i,c}\right| + \varepsilon}$$

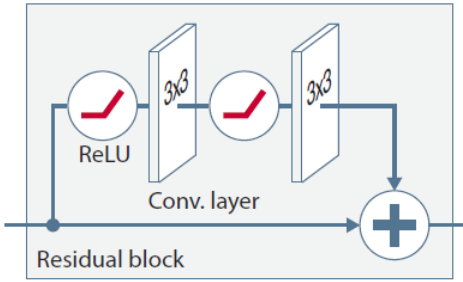
Kernel-predicting Convolutional Network (KPCN)

EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- Network predicts 21x21 filter kernel for each neighborhood followed by softmax.

$$\mathbf{d}_p = g_p(\mathbf{x};\theta) = \sum_{q \in \mathcal{N}(p)} w_{pq}\, \mathbf{c}_q \qquad w_{pq} = \frac{\exp([\mathbf{z}_p]_q)}{\sum_{q' \in \mathcal{N}(p)} \exp([\mathbf{z}_p]_{q'})}$$

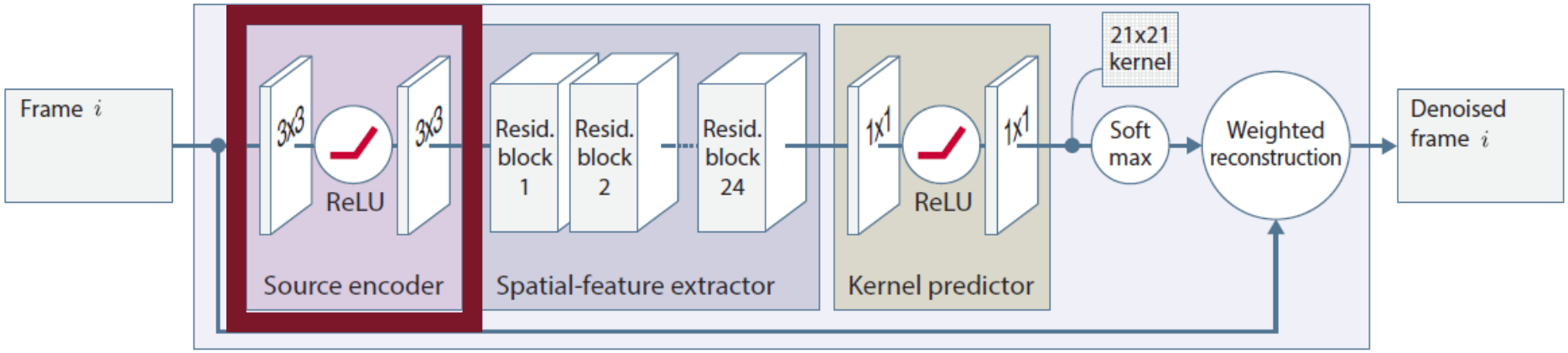




Source-aware Encoder

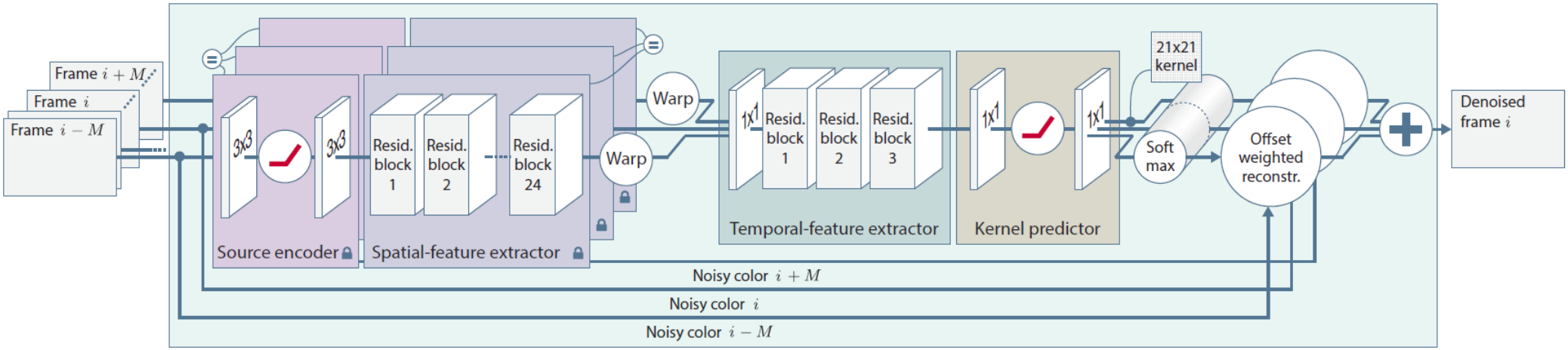
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Simple adaptaption to different renderer providing different input modalities
- Can incorporate different auxilliary buffers (color, depth, geometry, normal, texture, ….)
- Simple to retrain



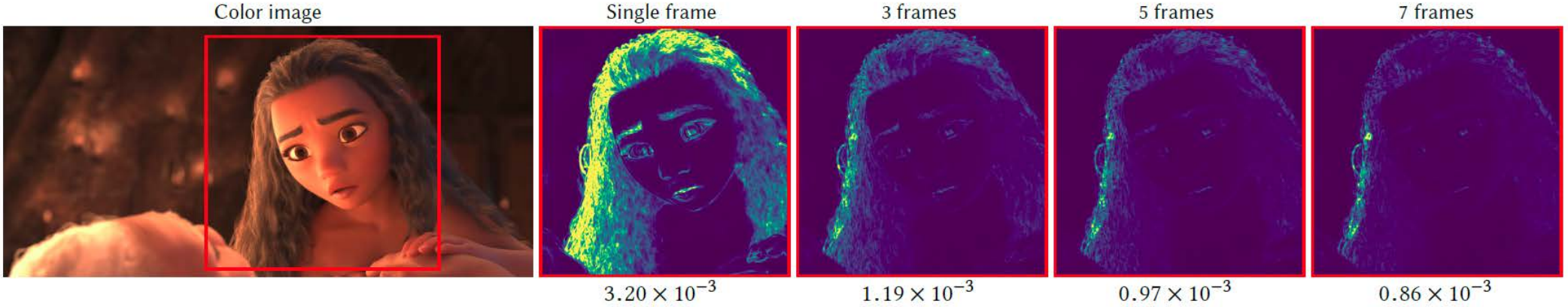
Temporal denosing

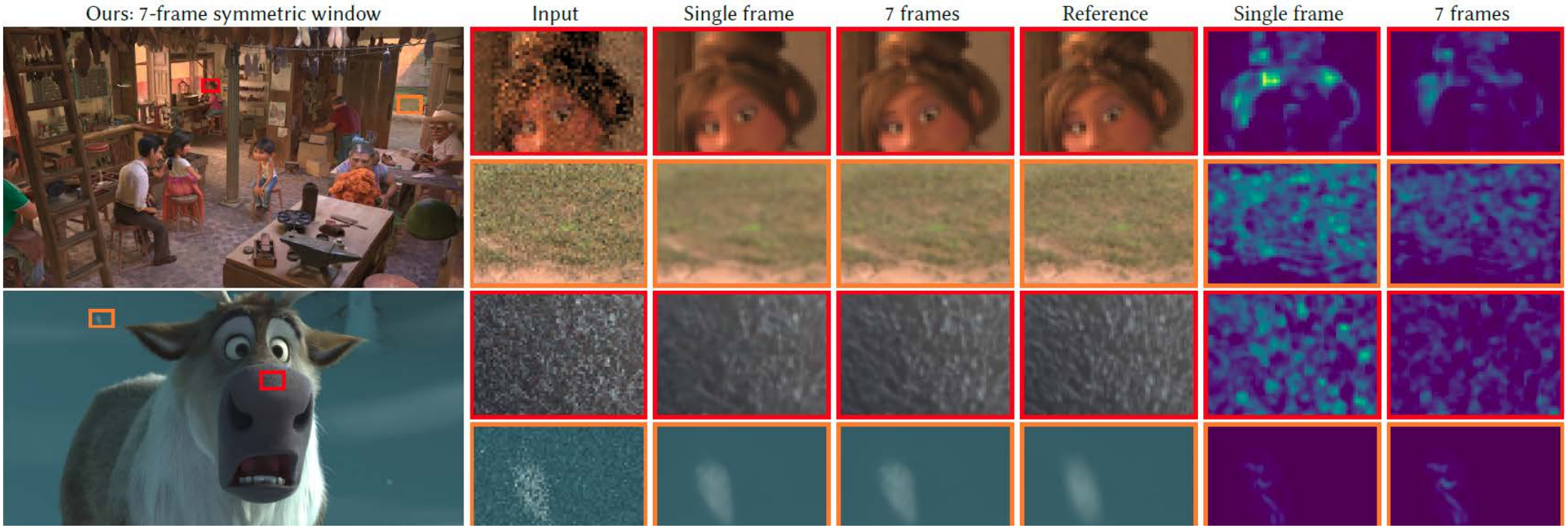
- Jointly predict kernels for eachs consecutive frame
- Warping based on known geometry (we are denosing images produced by a renderer) or by optical flow
- Train source encoder and feature extractor for individual frames first
- Then train temporal component



Temporal Window Size

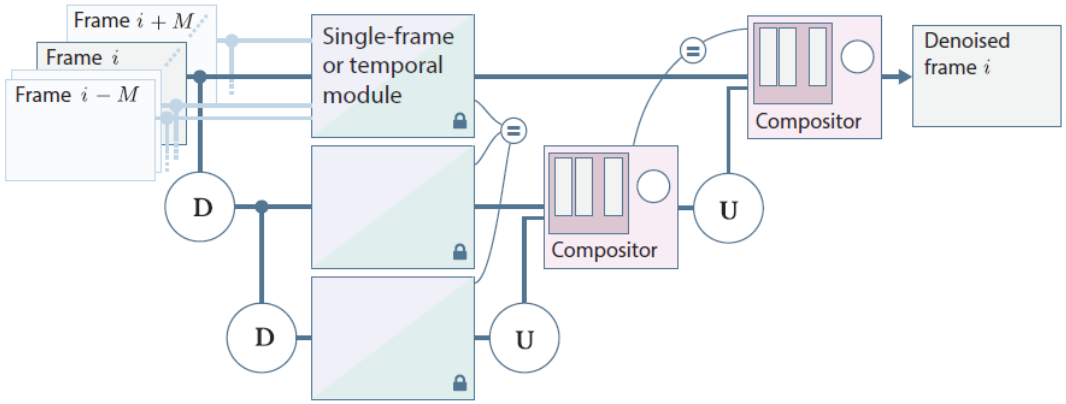
EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

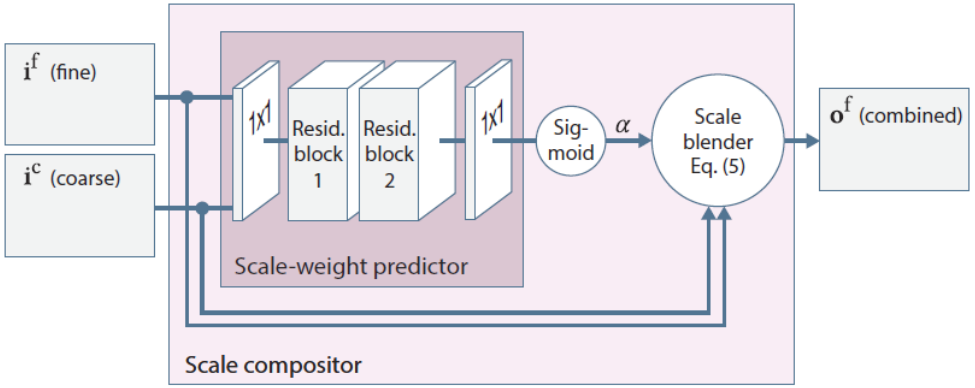


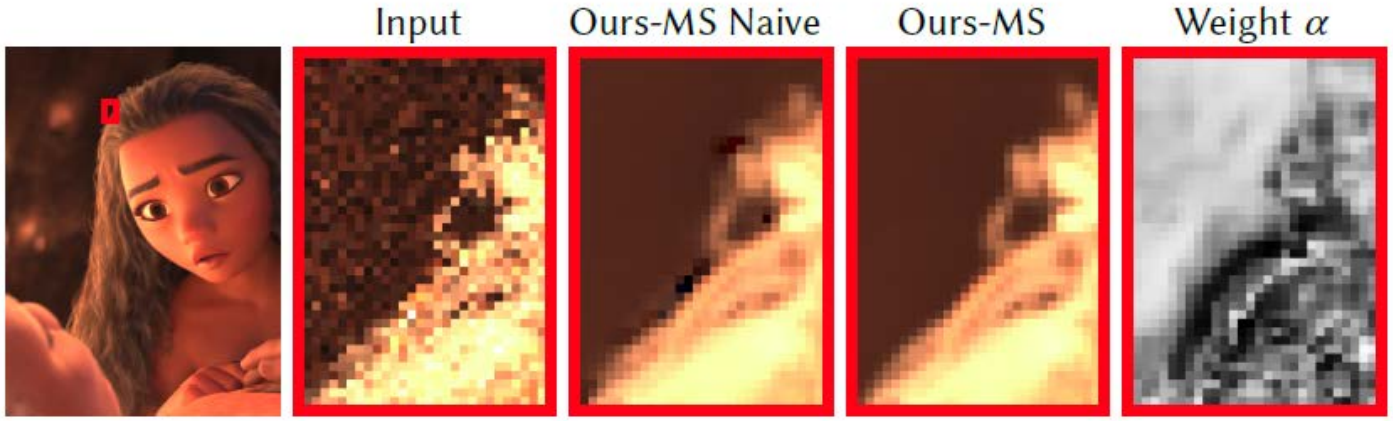


Multi-scale Prediction

- Single scale denoising might suffer from low-frequency noise
- Compose final result by weighted scale blending



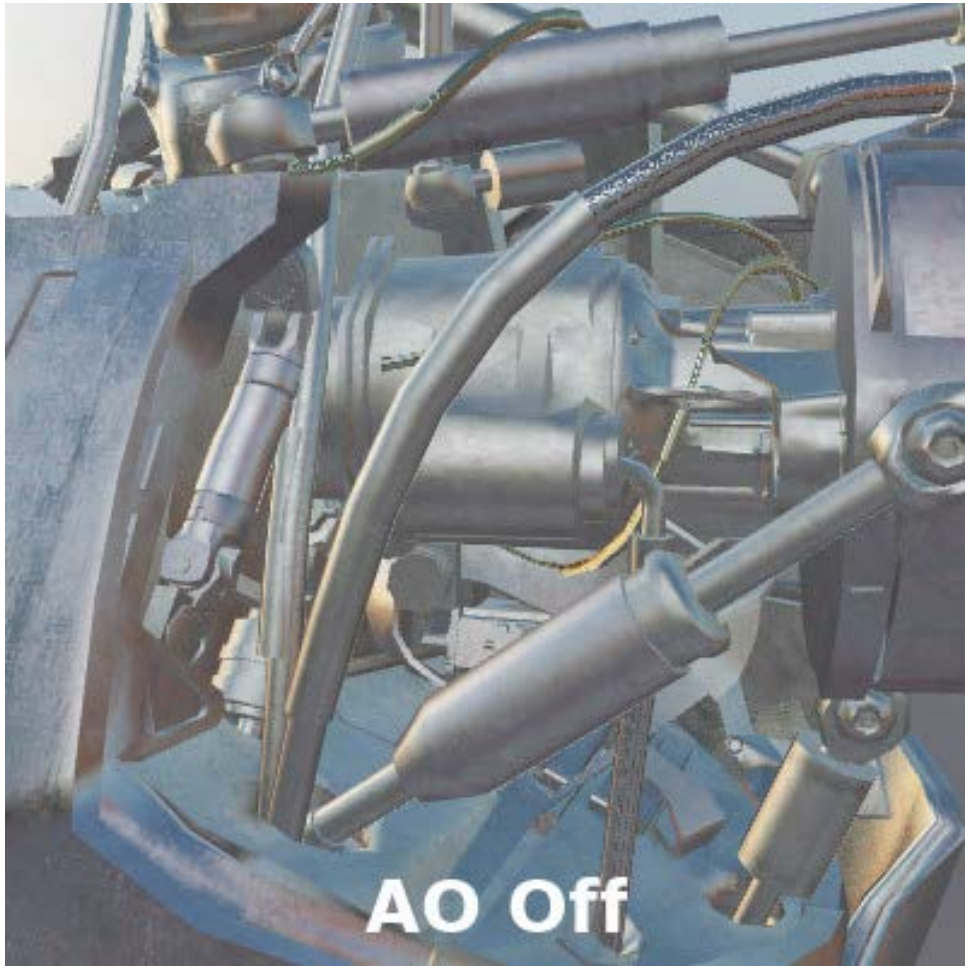




Neural Network Ambient Occlusion

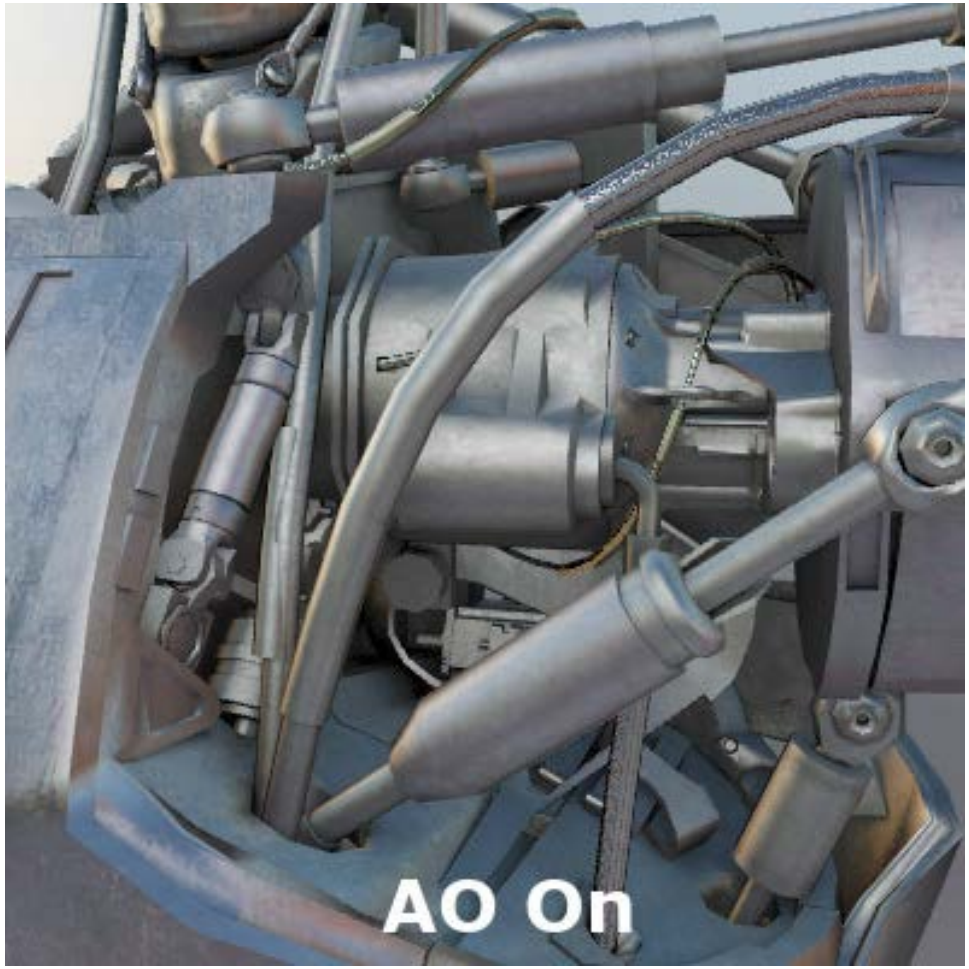
Holden et al. SIGGRAPH Asia 2016

AO approximates Global Illumination



Source: https://developer.playcanvas.com/en/user-manual/graphics/lighting/ambient-occlusion/

AO approximates Global Illumination



Source: https://developer.playcanvas.com/en/user-manual/graphics/lighting/ambient-occlusion/

Cost of AO evaluation

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Raytracing AO still slow
- (256x256 px, 1 sample per pixel, 1024 samples per point = 67 108 864 rays) 2.5 seconds

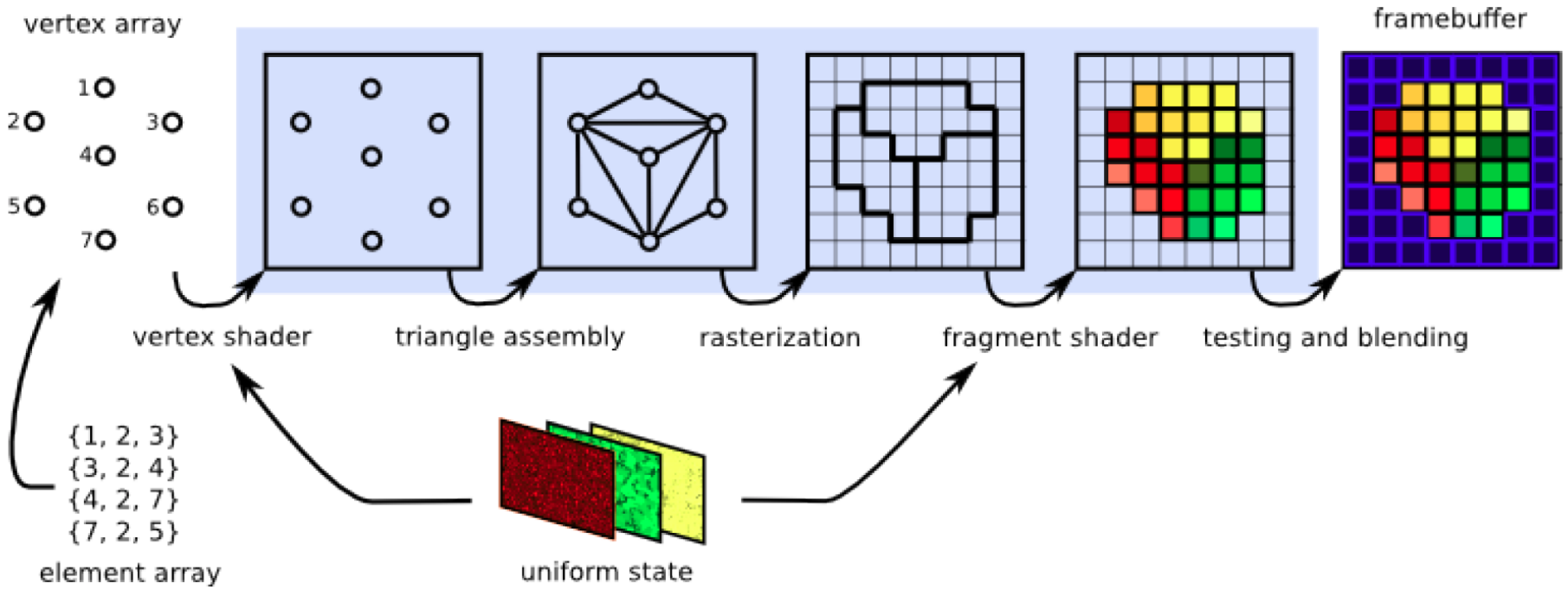
- Too slow for games

- Idea: Utilize information that is already available from the G-Buffer in the Deferred Shading pipeline

G-Buffer

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Directly supported by GPUs

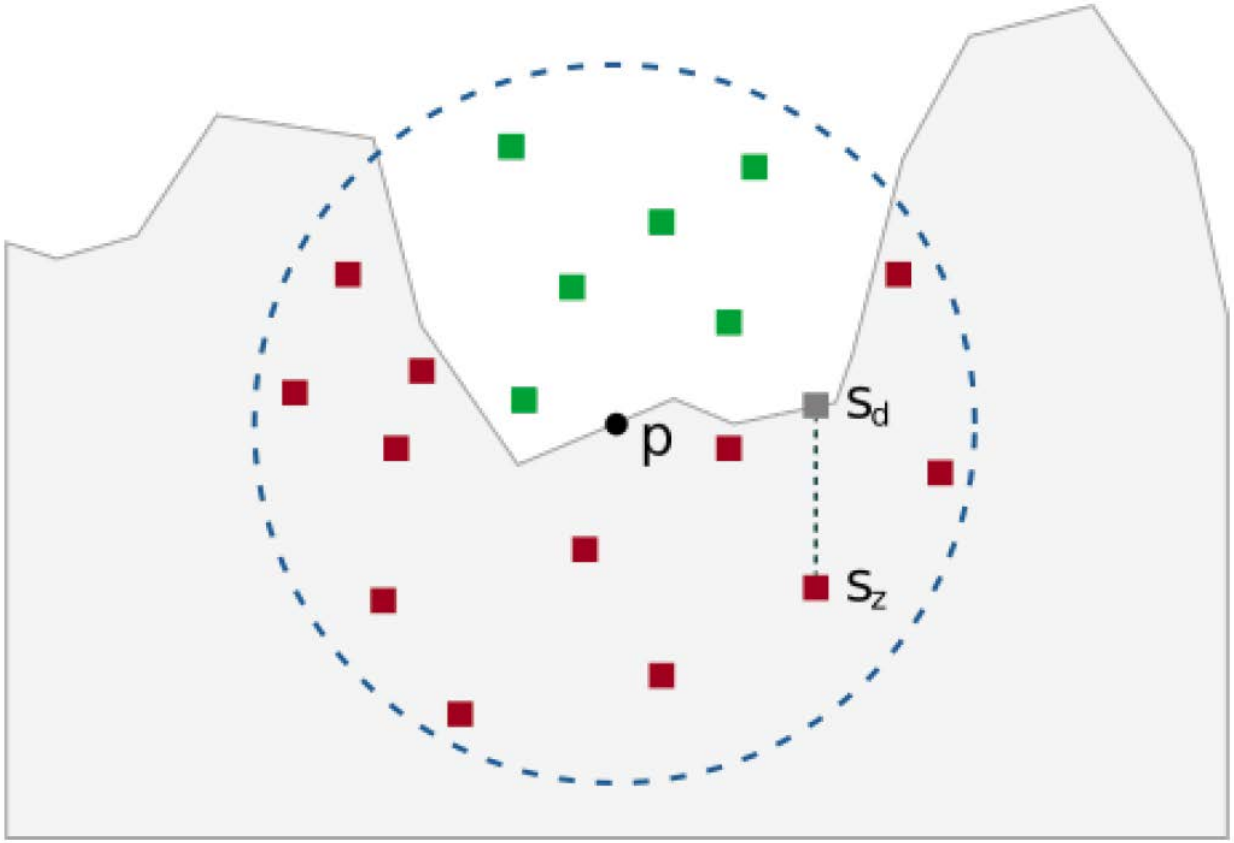


Source: https://www.enlightenment.org/playground/evas-gl.md

Screen-space Ambient Occlusion

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Idea: Percentage of samples placed behind the depth map.

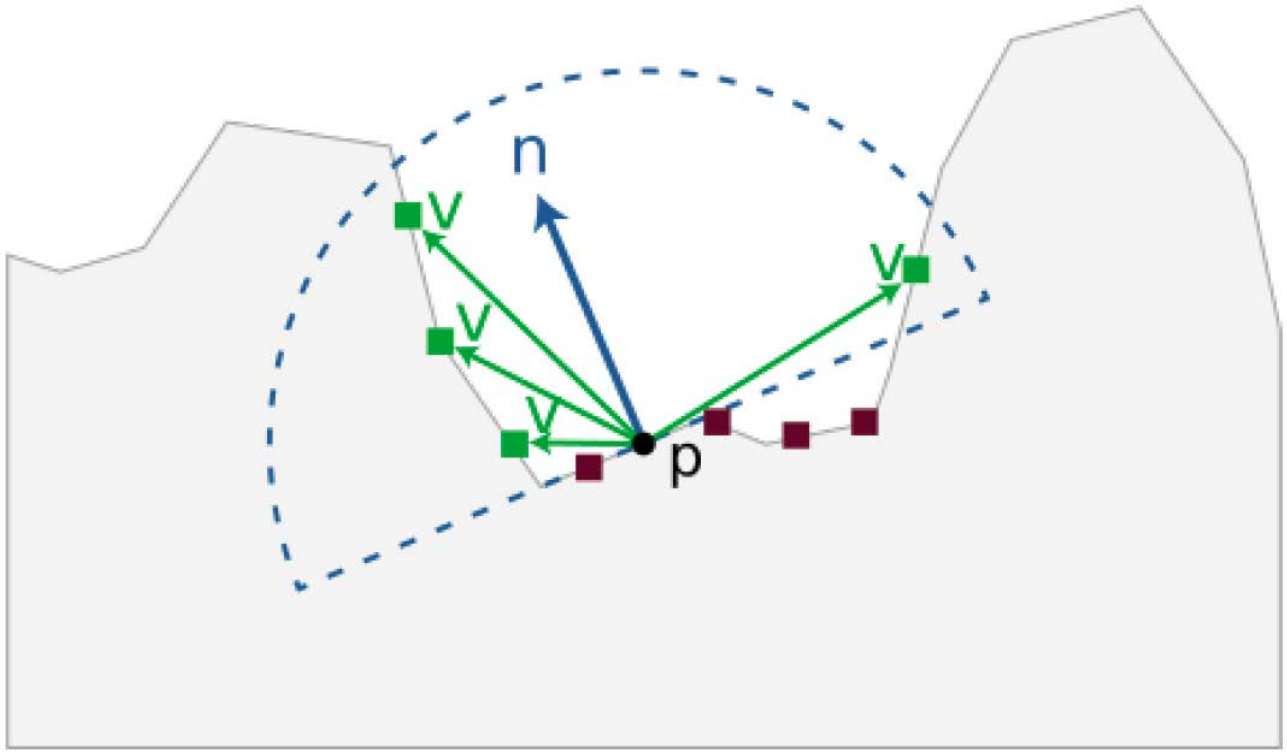


Source: F. Aalund and J.A. Bärentzen. A comparative study of screen-space ambient occlusion methods. Bachelor Thesis, Technical University of Denmark Informatics and Mathematical Modelling, 2013

Alchemy Ambient Occlusion

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Idea: Sum of weighted sample point distances.

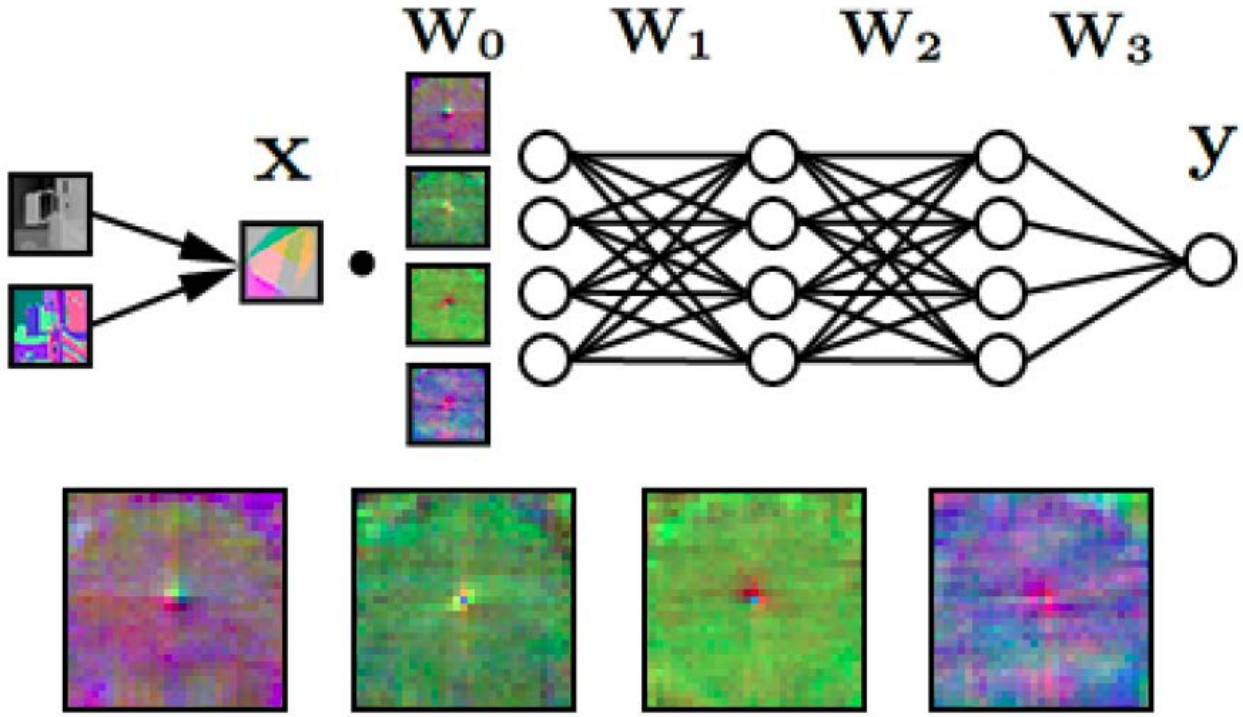


Source: F. Aalund and J.A. Bärentzen. A comparative study of screen-space ambient occlusion methods. Bachelor Thesis, Technical University of Denmark Informatics and Mathematical Modelling, 2013

NNAO: Simple MLP for Ambient Occlusion Prediction

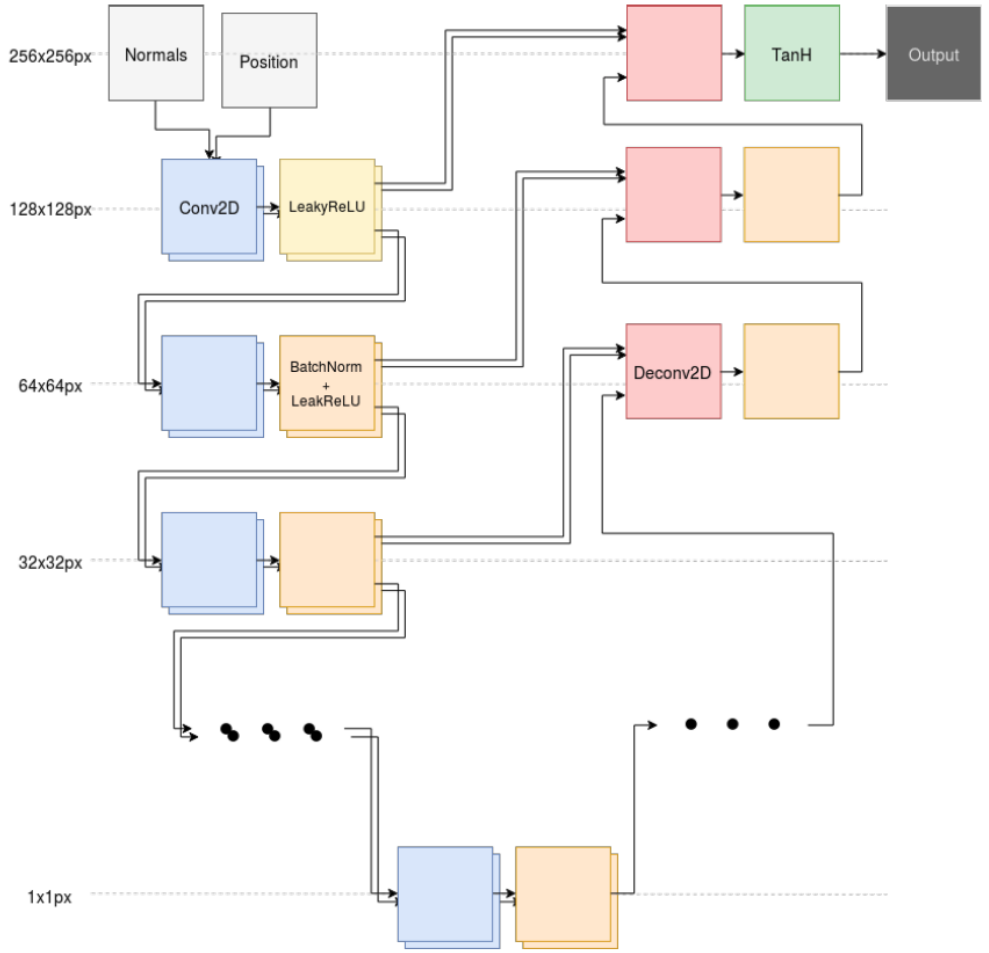
EBERHARD KARLS UNIVERSITÄT TÜBINGEN

- Predict the AO value of the central pixel of one region
- Input: prepared neighborhood from screen-space buffers
  - Absolute values and difference to central pixel of depth and normal



- Video

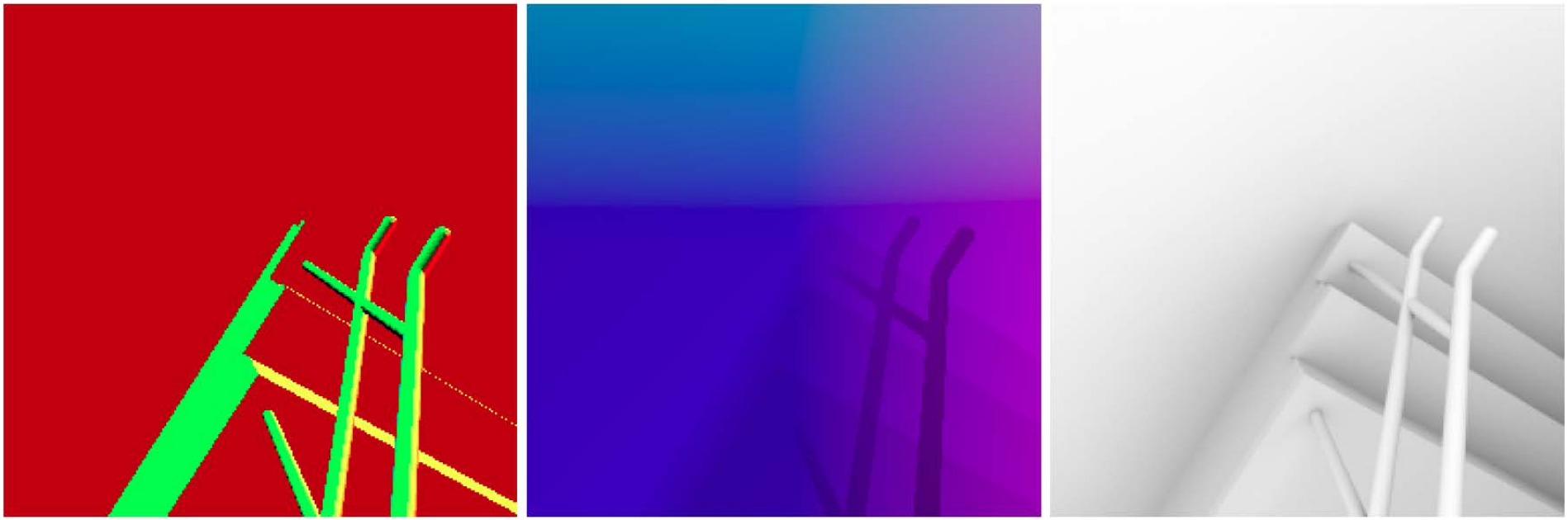
Reimplementation (Jens Schindel)



Training Data

EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

- Training Set 358650 image pairs
- Validation Set 72000 image pairs (20%)
- Test Set 27002 image pairs (7.5%)

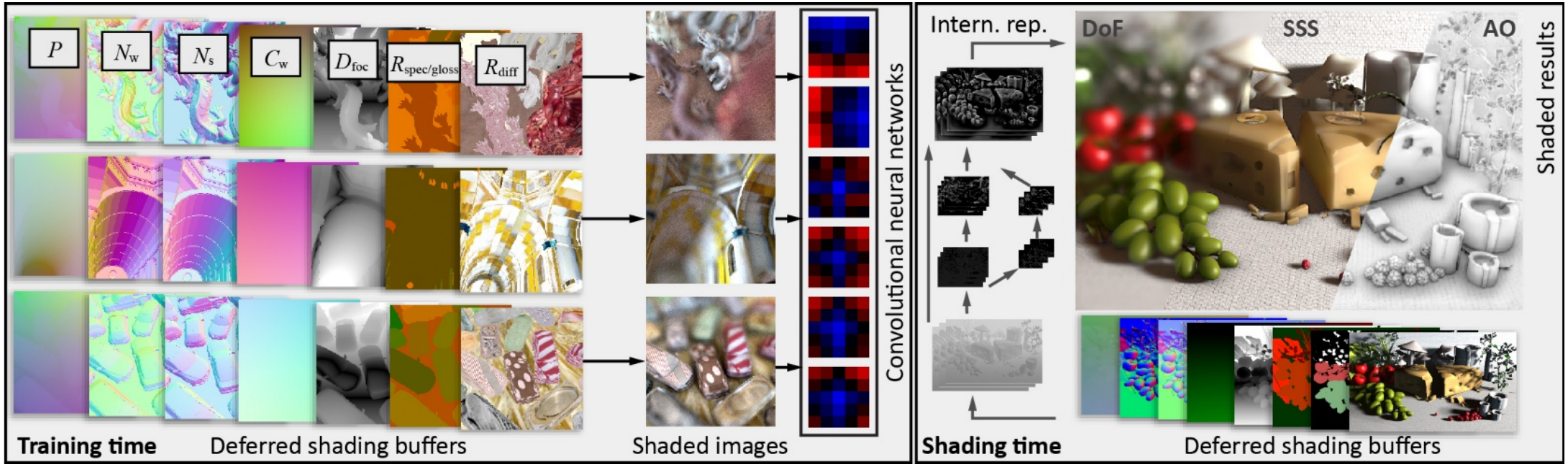


[Jens Schindel]

Video Results

Video Results

el]



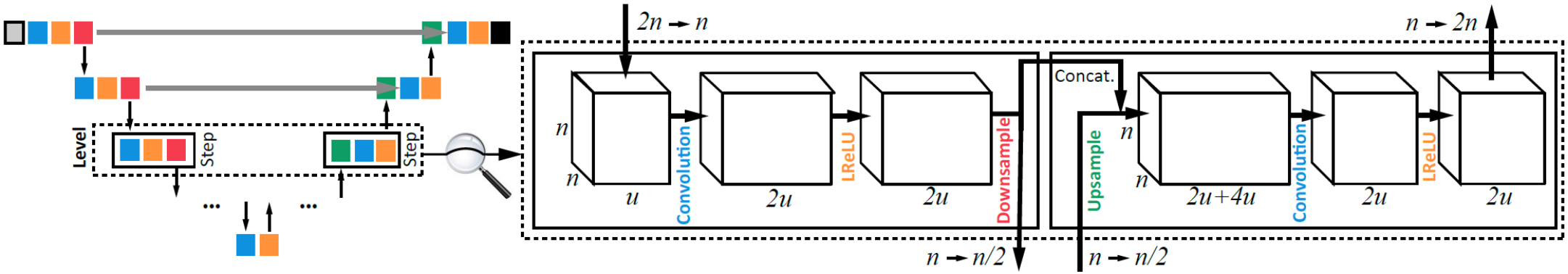
Deep Shading – Convolutional Neural Networks for Screen-Space Shading

Nalbach et al. EGSR 2017

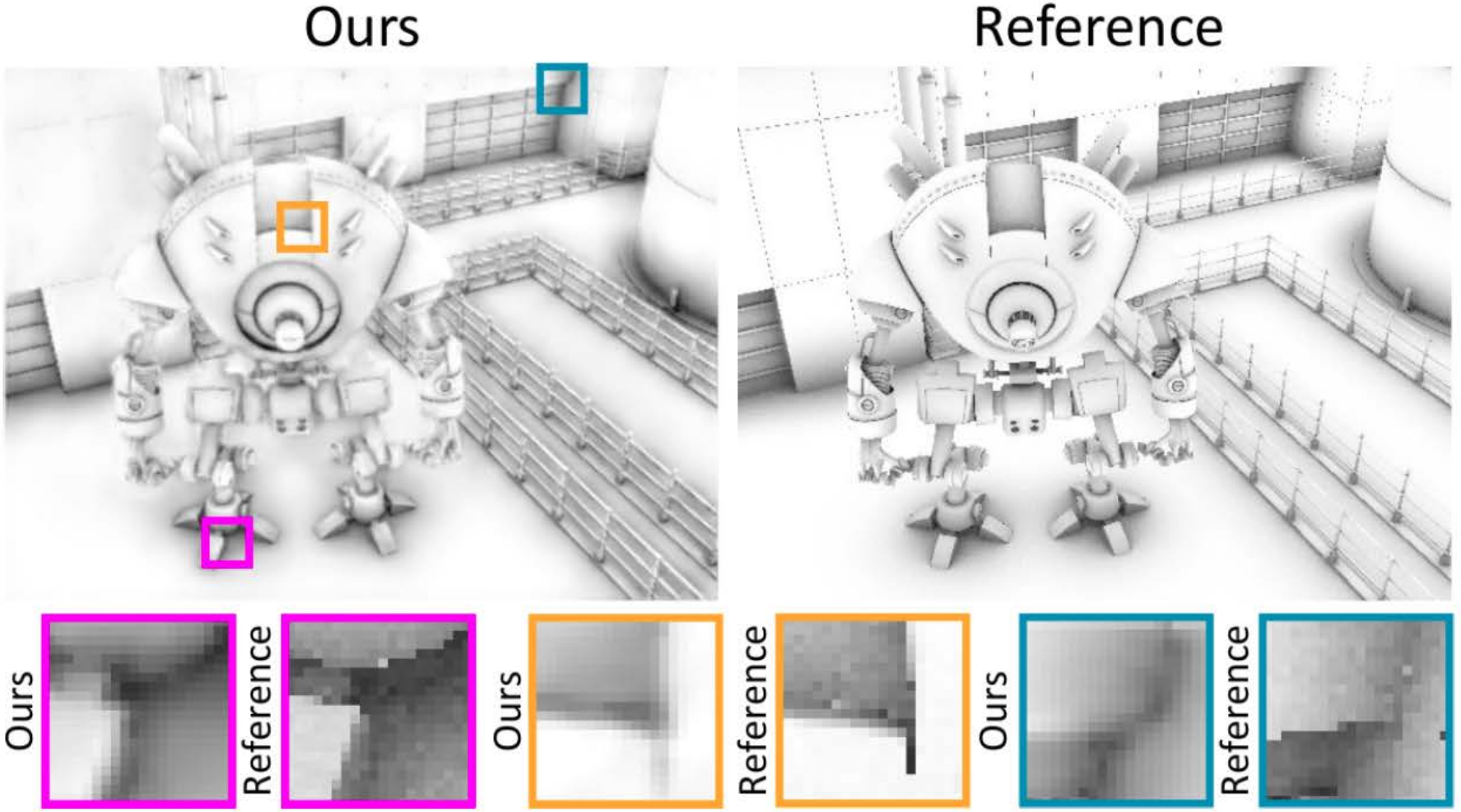
Network Architecture

- Trained for different rendering tasks (separate training set each)
  - Ambient occlusion
  - Depth of field
  - Subsurface scattering
  - Image-based lighting
  - AO + DoF

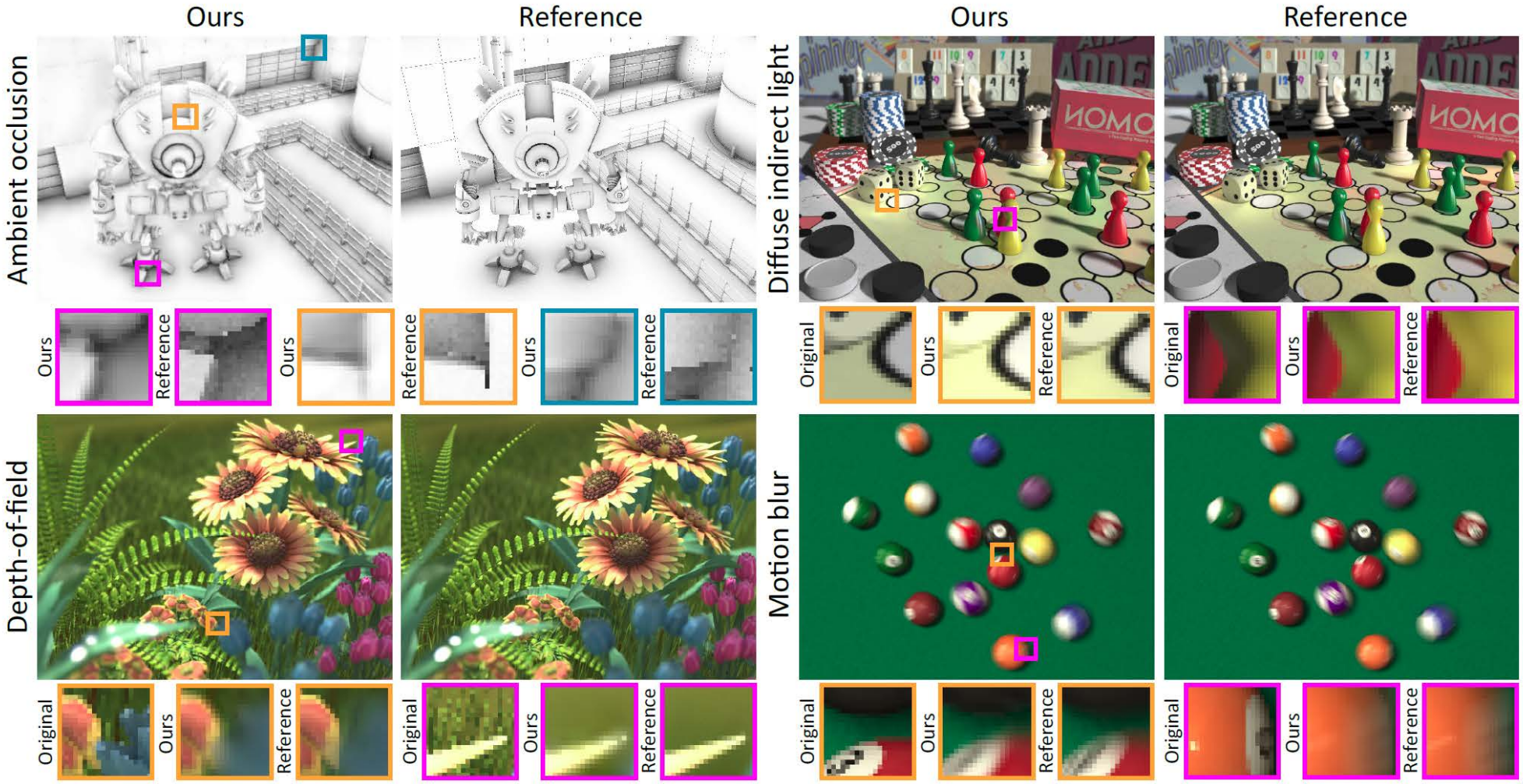
- Ambient occlusion
- Depth of field
- Subsurface scattering
- Image-based lighting
- AO + DoF
- …



Results


Ours
Reference
Ours
Reference
Ours
Reference
Ours
Reference

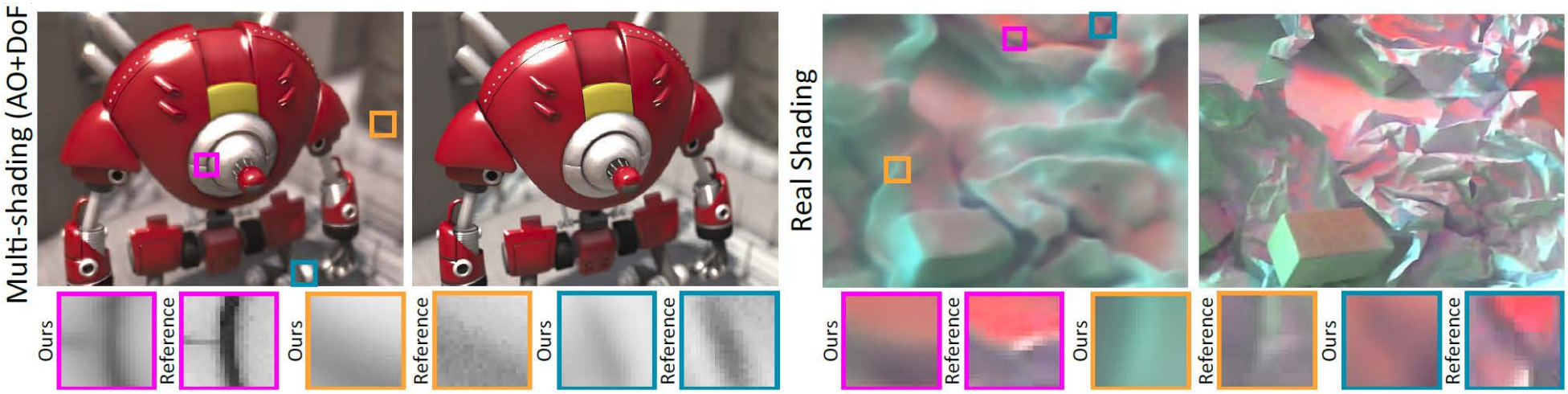
Results


Ours
Reference
Ambient occlusion
Ours
Reference
Ours
Reference
Ours
Reference
Diffuse indirect light
Original
Ours
Reference
Original
Ours
Reference
Depth-of-field
Original
Ours
Reference
Original
Ours
Reference
Motion blur
Original
Ours
Reference
Original
Ours
Reference

Results


Sub-surface scattering
Original
Ours
Reference
Original
Ours
Reference
Anti-aliasing
Original
Ours
Reference
Original
Ours
Reference
Image-based lighting
Original
Ours
Reference
Original
Ours
Reference
Directional occlusion
Ours
Reference
Ours
Reference
Ours
Reference

Results


Multi-shading (AO+DoF
Real Shading
Ours
Reference
Ours
Reference
Ours
Reference
Ours
Reference
Ours
Reference
Ours
Reference
Ours
Reference

References

- Patrick Wieschollek, Ido Freeman, Hendrik P.A. Lensch, Learning Robust Video Synchronization without Annotations, ICMLA 2017
- Nima Khademi Kalantari, Steve Bako, Pradeep Sen, A Machine Learning Approach for Filtering Monte Carlo Noise, ACM SIGGRAPH 2015
- Vogels et al., Denoising with Kernel Prediction and Asymmetric Loss Functions, ACM SIGGRAPH 2018.
- Daniel Holden, Jun Saitoy, Taku Komuraz, Neural Network Ambient Occlusion, SIGGRAPH Asia 2016
- Oliver Nalbach and Elena Arabadzhiyska and Dushyant Mehta, Hans-Peter Seidel and Tobias Ritschel, Deep Shading – Convolutional Neural Networks for Screen-Space Shading, EGSR 2017