Deep Shading:

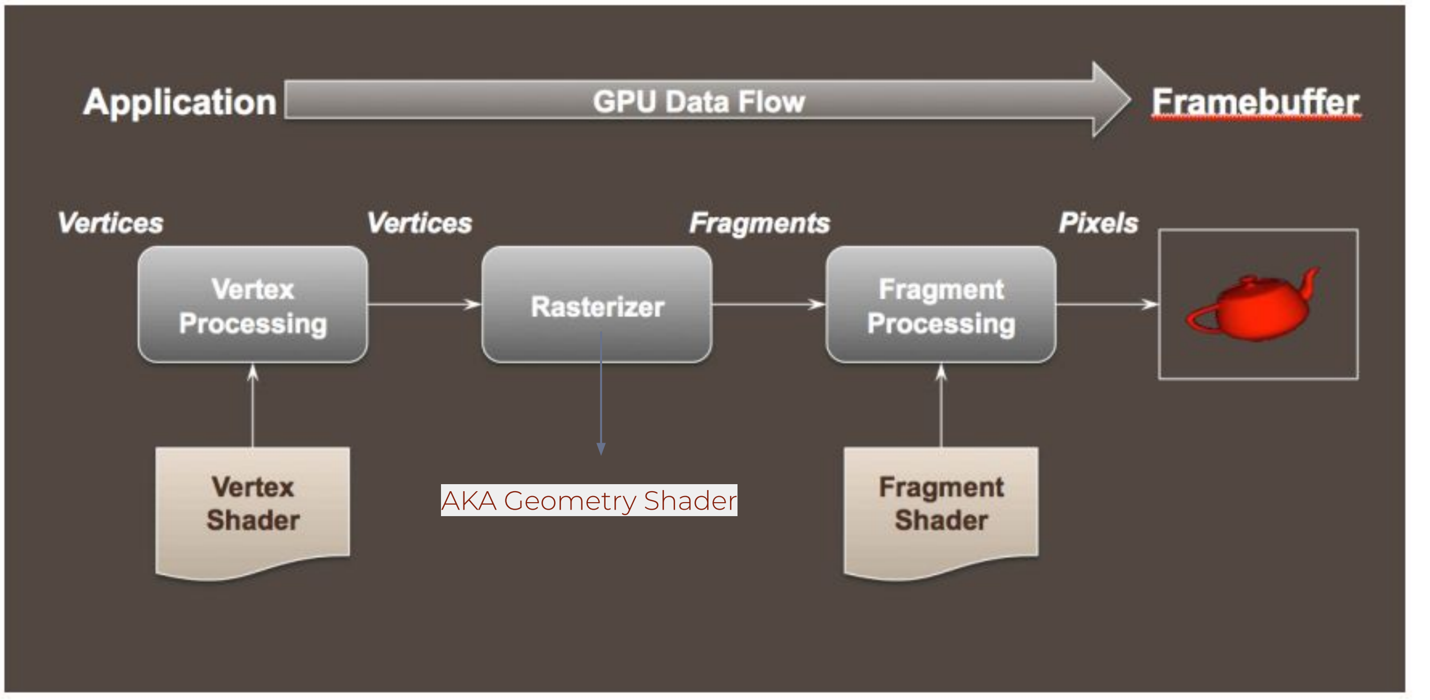
Convolutional Neural Networks for Screen Space Shading

Paper Review
Mirkamil Mijit

Outline

- ❑ Background For Graphics Pipeline
- ❑ Objective
- ❑ Previous Work
- ❑ Data Generation
- ❑ Attributes
- ❑ The Network
- ❑ Results
- ❑ Analysis
- ❑ Conclusion
- ❑ Appendix

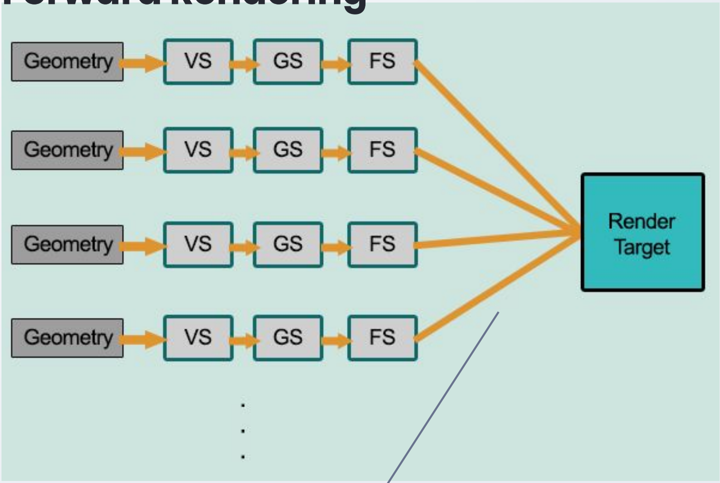
Background For Graphics Pipeline



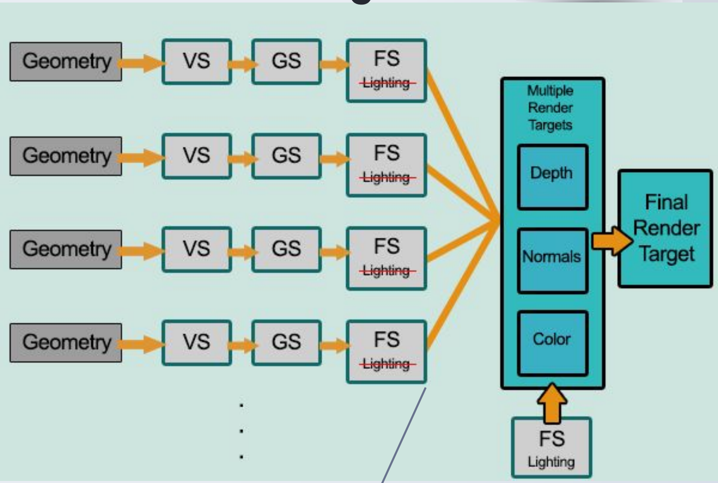
http://web.cse.ohio-state.edu/~shen.94/5542/Site/Slides_files/hardware5542.pdf

Background For Graphics Pipeline

Forward Rendering



Deferred Rendering



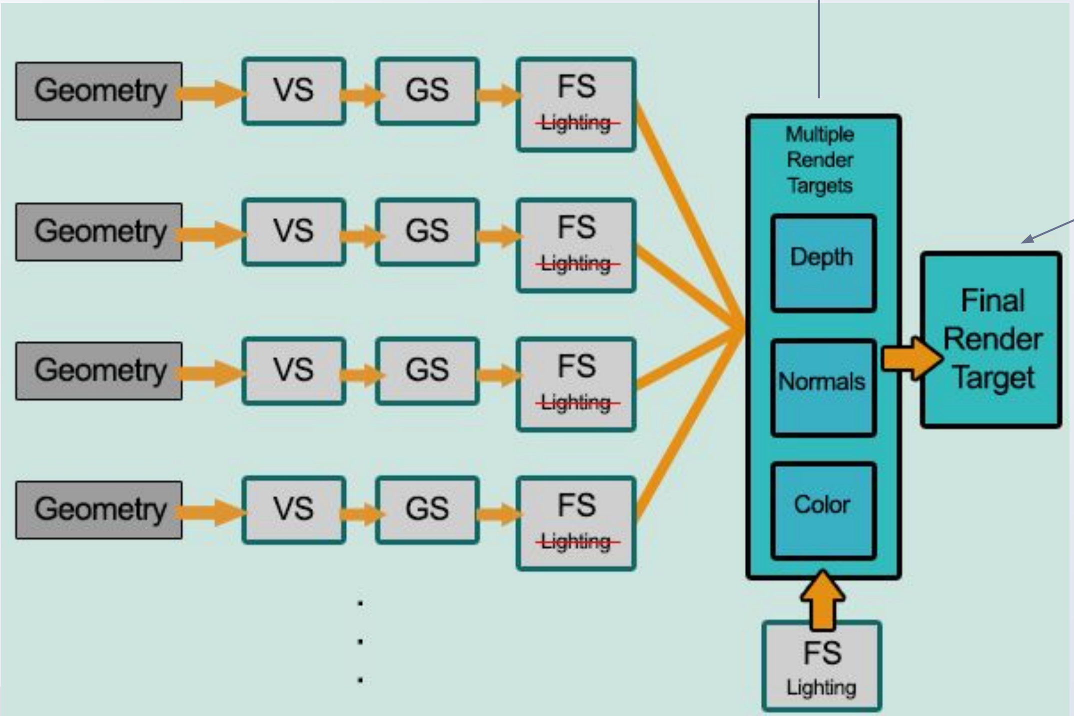
Calculate the final pixel values directly in Fs

https://gamedevelopment.tutsplus.com/articles/forward-rendering-vs-deferred-rendering--gamedev-12342

Create Per-fragment attribute in screen space then apply some screen space technique

Objectives

Given This



Predict This Depending on the Screen Space techniques

Previous Work

- Screen Space Shading Techniques
  - Ambient Occlusion (AO)
  - Directional Occlusion (DO)
  - Indirect Light (GI)
  - Sub-Surface Scattering (SSS)
  - Depth-of-field (DOF)
  - Motion Blur (MB)
  - Image-Based Lighting (IBL)
  - Anti-Aliasing (AA)
- Static 3D scenes rendering

Data Generation

- 61,000 pairs of deferred shading buffers (contains per-fragment attributes) (Computed Using OpenGL)
- Corresponding shaded reference images
- 54,000 used for training
- 6,000 used for validation
- 1,000 used for testing

Attributes

Attribute Name	Space	Notation
Position	Screen	P_s
Normals	Camera/World	N_s/N_w
Depth	Screen	D_s
Distance to the focal plane	Screen	D_{focal}
Radius of the circle of confusion of the lens system	//	B
Normalized direction to the camera	World	C_w
Material parameters	//	R

Attribute Name	Space	Notation
RGB diffuse and specular colors	//	R_{diff}/R_{spec}
Scalar glossiness	//	R_{gloss}
Scatter	//	R_{scatt}
Direct light	//	L
Direct light for diffuse	//	L_{diff}

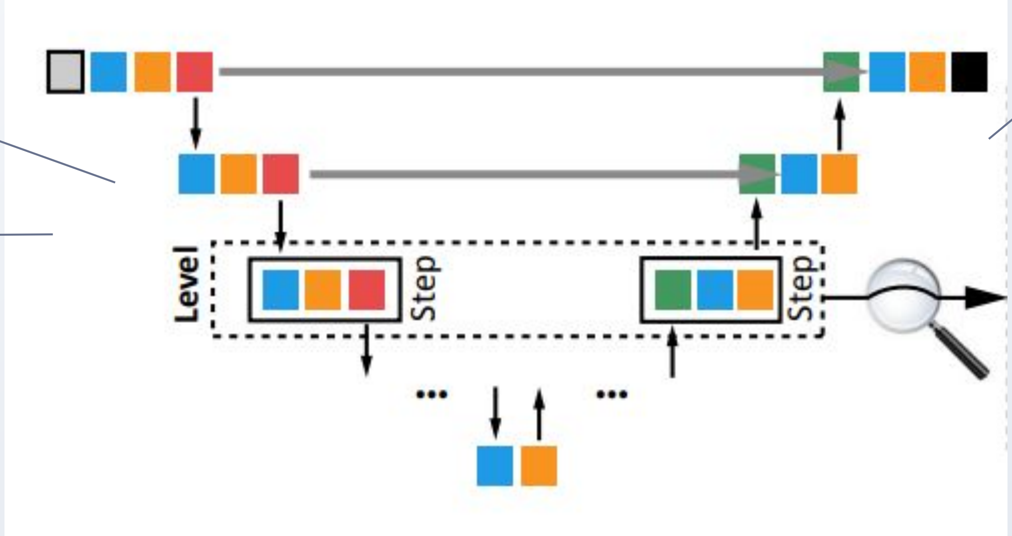
Attributes Used For Each Effect

Effect	Attributes
AO	N_s, P_s
GI	N_s, P_s, L_{diff}
DoF	$D_{focal} \cdot B, D_s, L$
MB	F, L, D_s
SSS	P_s, R_{scatt}, L
AA	D_s, L
Multi	(see AO, DoF)
IBL	N_w, C_w, R
DO	N_w, N_s, P_s
RS	N_s

The Network

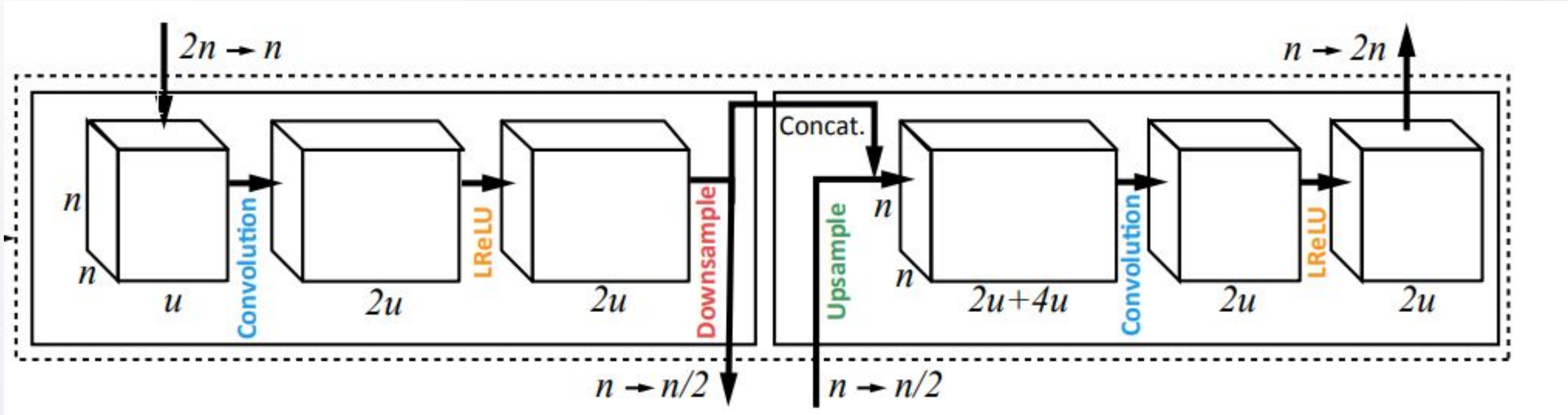
Down branch:
Reduce spatial
resolution

U-shaped



Up branch:
Increase Spatial
resolution

The Network



More information on each level

Loss Function

- Based on structural similarity (SSIM) (More on appendix)
- The loss function is Structural dissimilarity (DSSIM)
- DSSIM = (1 - SSIM)/2

Note:

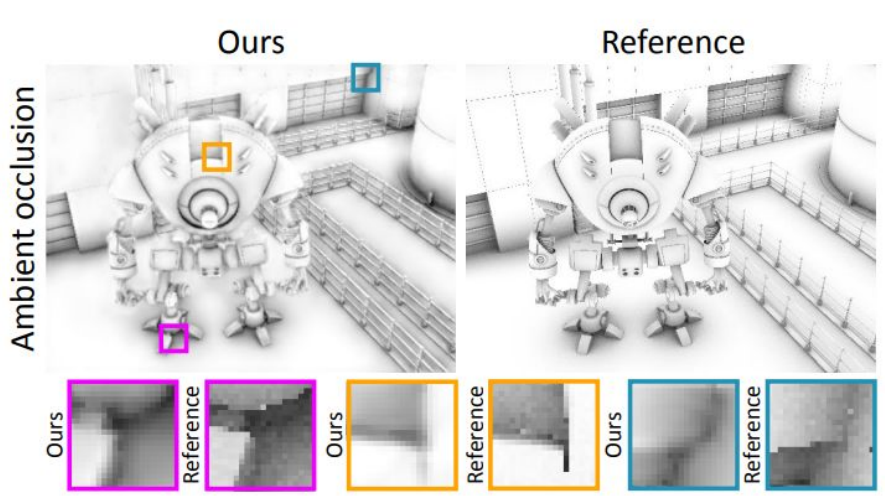
1. The network is the same for all the effects. But the number of kernels on each level and the number of levels vary.
2. The convolutions have a fixed extent in the spatial domain, but may vary depending on the effects.

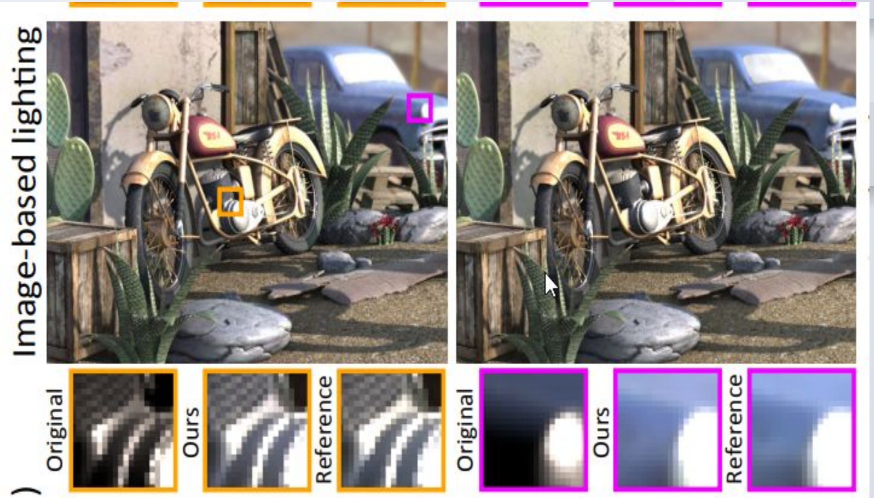
Result Table

Effect	Attributes	Albedo	Mono	u_0.	Lev.	Ker.	Size	SSIM	Time
AO	N_s, P_s	✗	✓	8	6	3	71 K	.805	9 ms
GI	N_s, P_s, L_{diff}	✓	✓	16	5	3	134 K	.798	28 ms
DoF	$D_{focal} \cdot B, D_s, L$	✓	✓	16	5	3	133 K	.959	27 ms
MB	F, L, D_s	✓	✓	16	5	3	133 K	.937	26 ms
SSS	P_s, R_{scatt}, L	✓	✓	16	5	3	133 K	.905	27 ms
AA	D_s, L	✓	✓	8	1	5	1.2 K	.982	1.8 ms
Multi	(see AO, DoF)	✓	✗	16	5	3	135 K	.933	26 ms
IBL	N_w, C_w, R	✓	✗	300	1	1	3.9 K	.973	21 ms
DO	N_w, N_s, P_s	✗	✗	16	5	3	135 K	.589	26 ms
RS	N_s	✓	✗	16	5	5	370 K	.622	80 ms

Albedo: A special shading attribute
Mono: Each RGB channel done separately
U_0: number of kernel on level 0
Time: Training Time

Some Results





Analysis

Limitations

- Inherited artifacts from screen space attributes shading
    - Partial information
- Range of Values are theoretically unbounded
- Effect Radius depends on training resolution

Choices for design

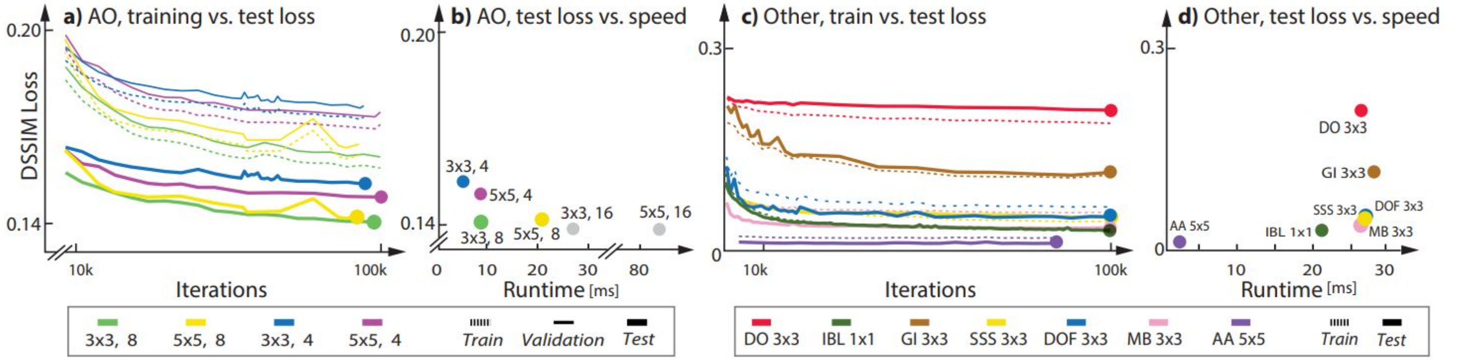


Figure 13: *Analysis of different network structures. We here compare different design choices for different effects in terms of compute time and DSSIM loss. The vertical axes on all plots corresponds to DSSIM loss (less is better). The horizontal axes of the line plots range over the number of training iterations. The scatter plots have computation time of the Deep Shader as the horizontal axis.* a) *Train, test and validation loss as a function of iterations for different designs of AO (curves).* b) *Relation of final loss and compute time for different designs for AO.* c) *Loss as a function of iterations for the chosen designs for other effects (curves).* d) *Comparison of compute time and final loss for the other effects, as a means of placing their relative complexity.*

Conclusion

- Turn Attributes to scenery.
- Inherently limited due to screen space rendering.
- Limited to the attribute range.
- Some results are did not turn out ideal.

Appendix

SSIM: https://www.mathworks.com/help/images/ref/ssim.html

ReLU:
https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/

Appendix

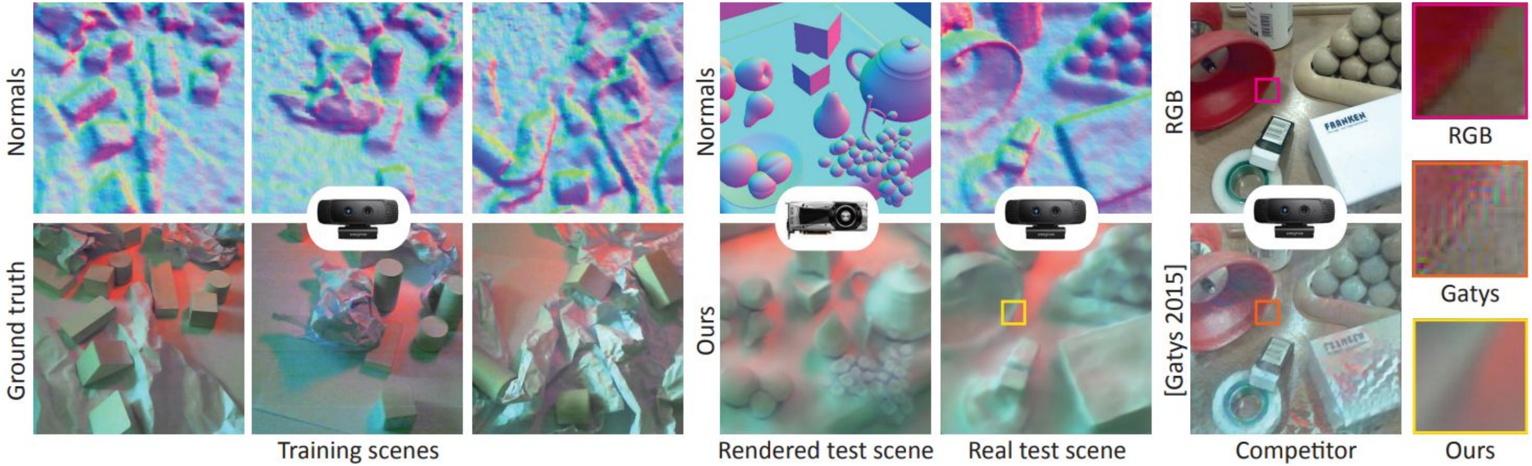


Figure 9: *Deep Shading learnt from RGBD video that captures screen space normals* (top) *and appearance* (bottom). *Deep Shading can learn the correlation including directional light, occlusion and bounces and transfer it to novel synthetic or captured normal images* (4th and 5th column). *This performs better than established deep-learning based style transfer [GEB15] from RGB to RGB images* (last column).

Appendix

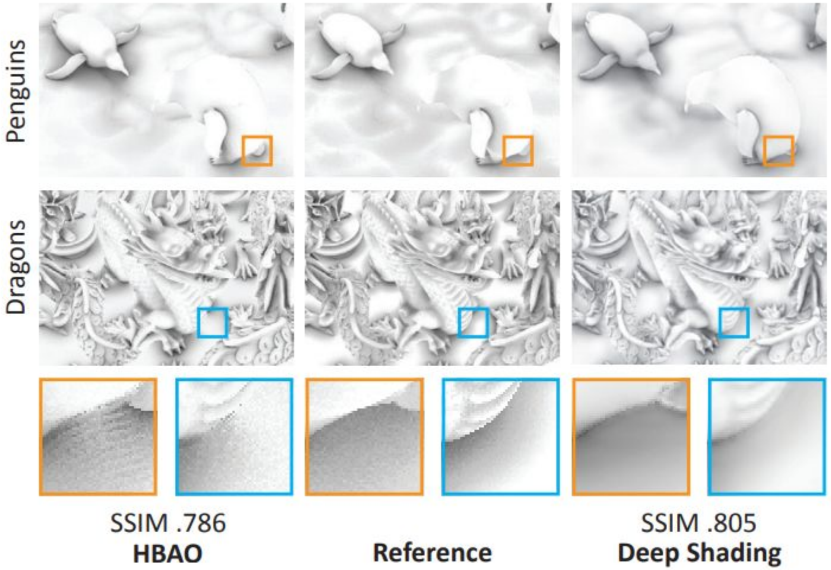


Figure 7: *In a same-time comparison, Deep Shading for AO is on-par with state-of-the-art methods like HBAO, both numerically and visually. The given SSIM values are w.r.t. the full AO test set.*

Appendix

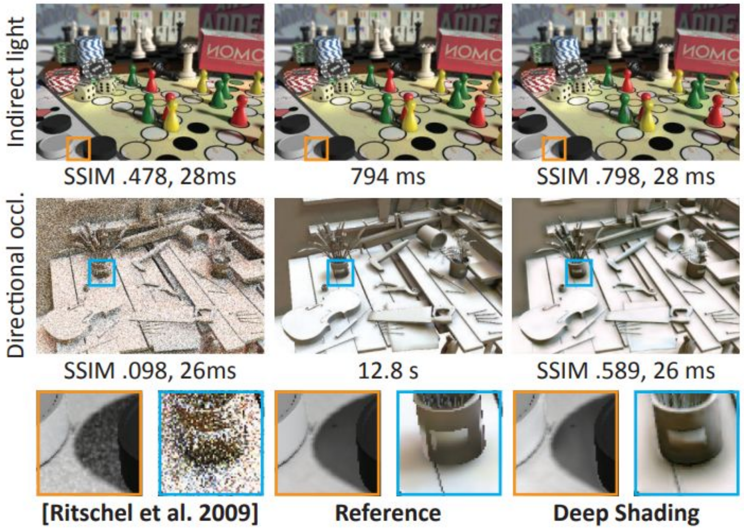


Figure 8: *Same-time comparisons for GI and DO, comparing with the method of Ritschel et al. [RGS09]. The SSIM values are w.r.t. the full test sets.*

Appendix

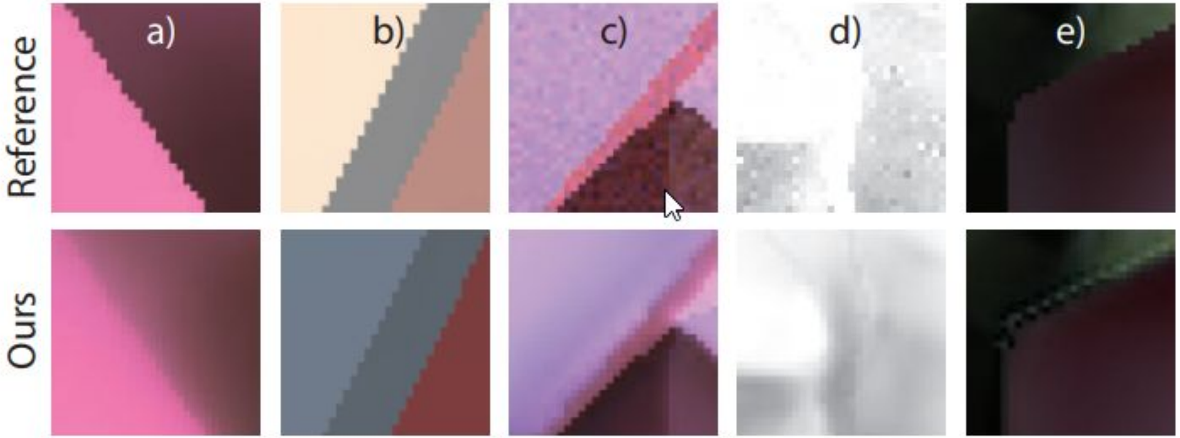


Figure 10: *Typical artifacts of our approach:* a): *Blur.* b): *Color shift.* c): *Ringing.* d): *Background darkening.* e): *Attribute discontinuities.*

Appendix

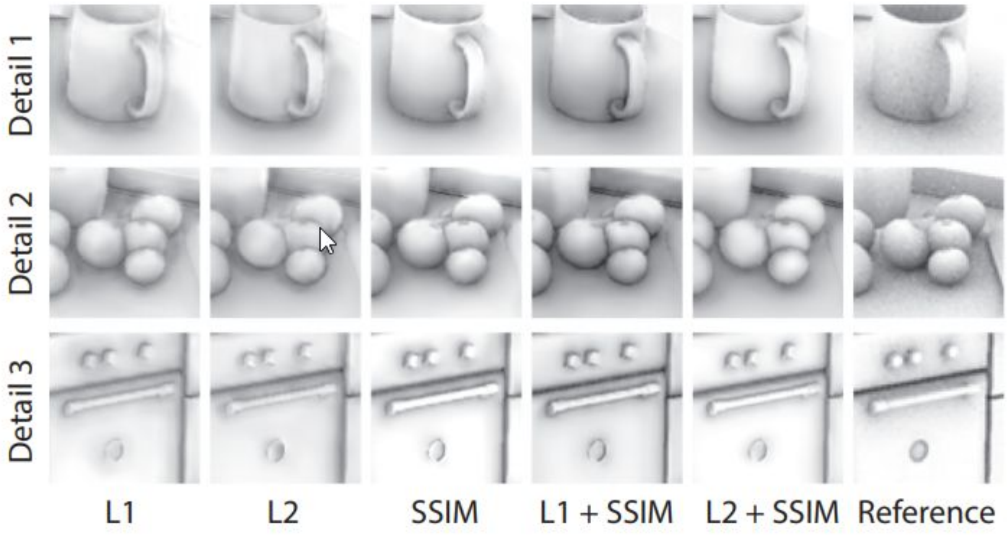


Figure 14: *Outputs produced by the same network trained with different loss functions for the case of AO.*

Appendix

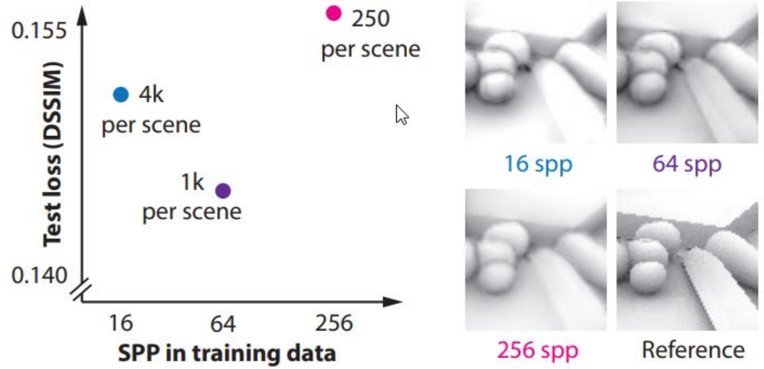


Figure 15: Left: *Data points correspond to the same time budget to produce training data but using different numbers of samples per pixel. The resulting number of individual views per scene (before data augmentation) is given for each point.* Right: *AO produced by the corresponding trained networks.*

Appendix

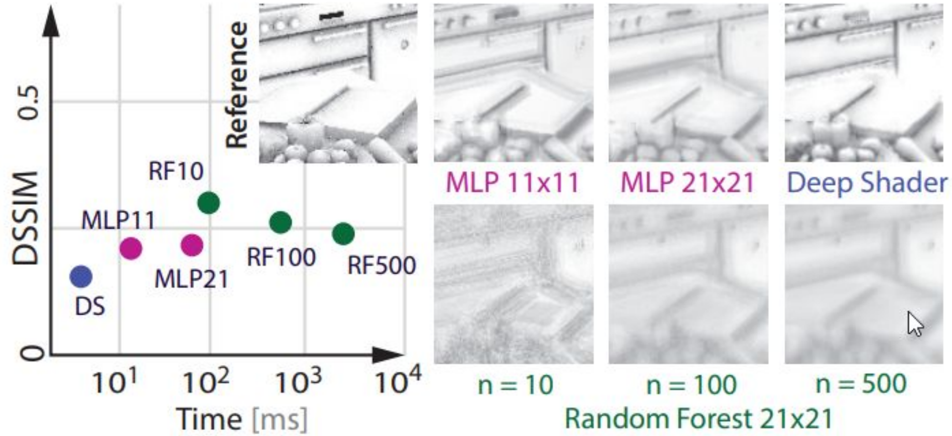


Figure 17: *AO computed using random forests, shallow MLPs and Deep Shading. The vertical axis is image error (DSSIM) on a linear scale. The horizontal axis is compute time for* 256×256 *px images on a logarithmic scale.* n *indicates the number of trees.*