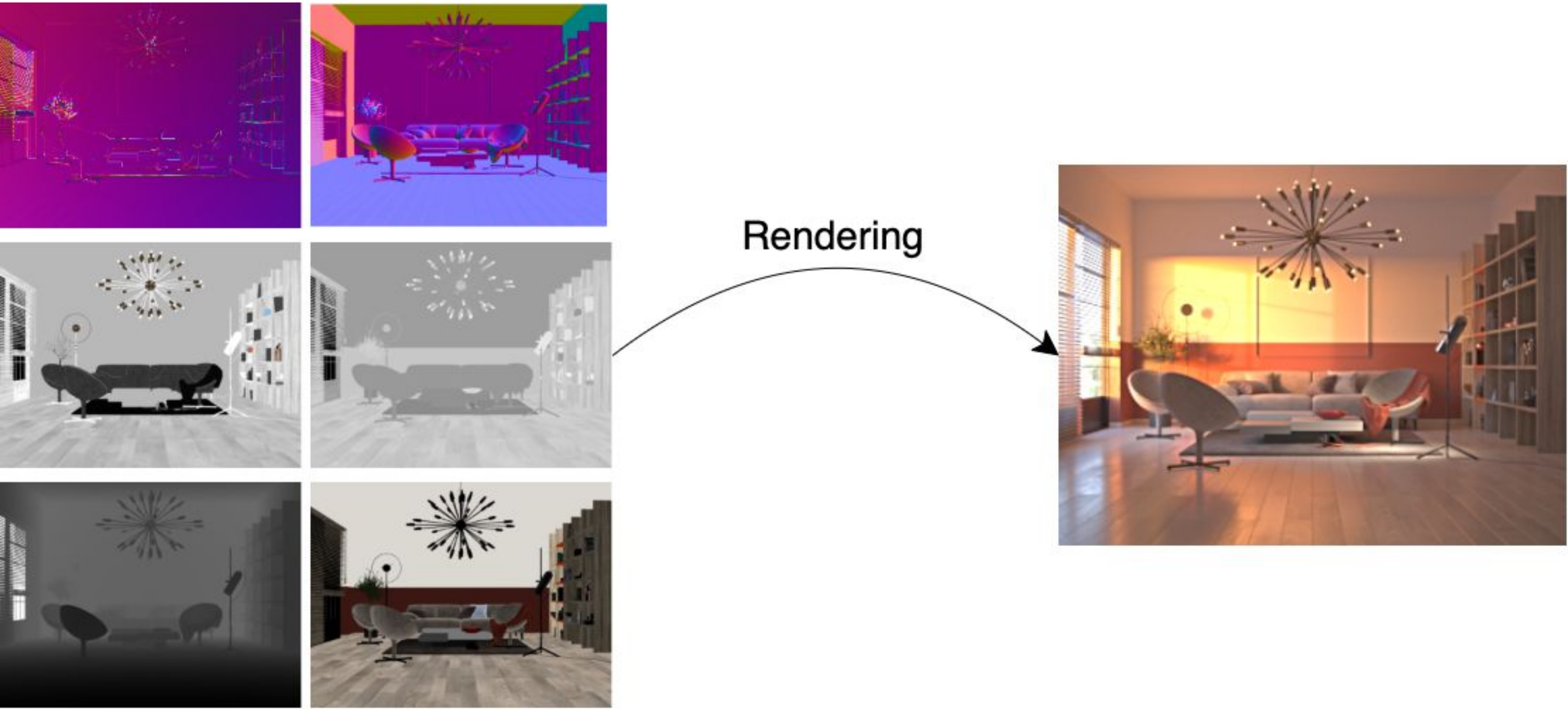
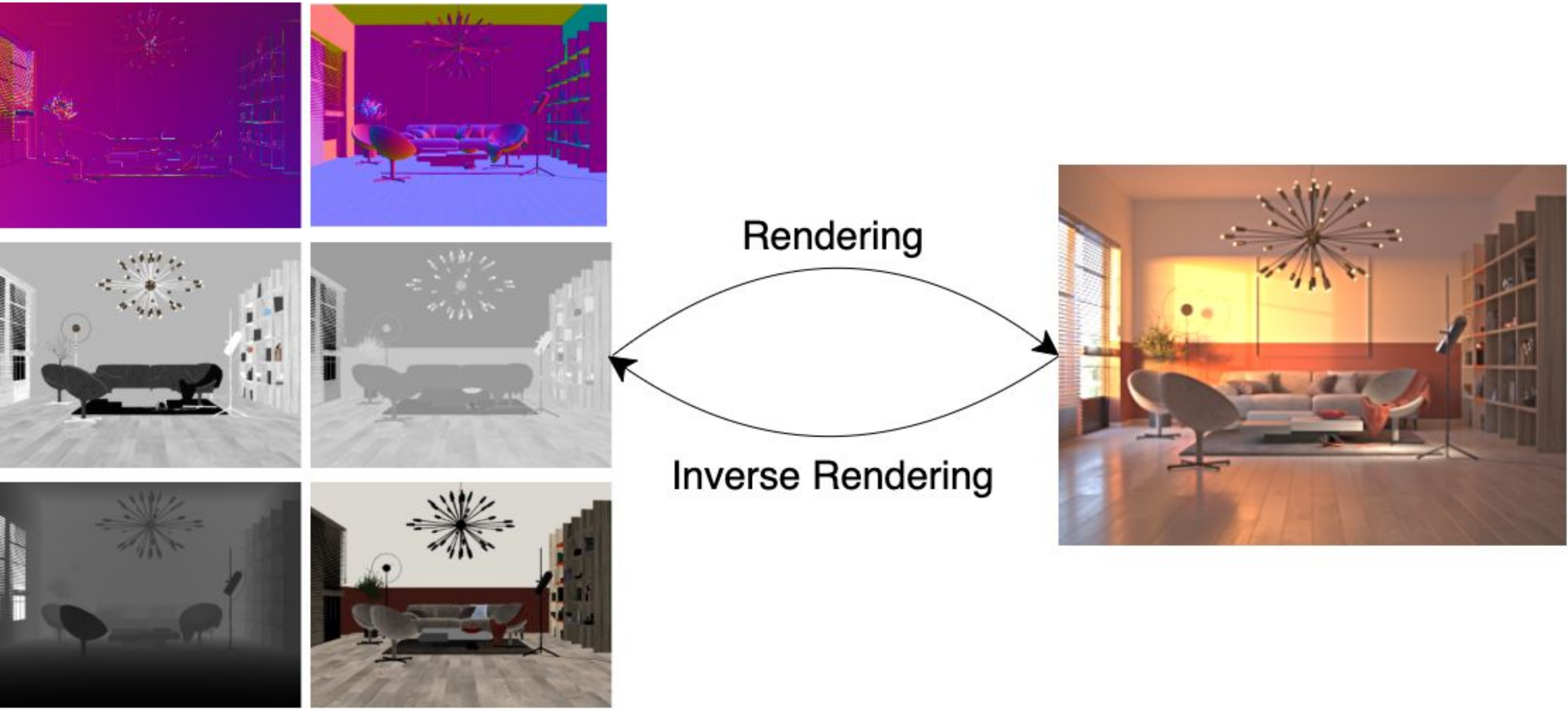
Material Picker

Filip Jurčák
Supervisor: prof. RNDr. Roman Ďurikovič, PhD.
Consultant: Mgr. Petr Vévoda (Corona Renderer)


Rendering

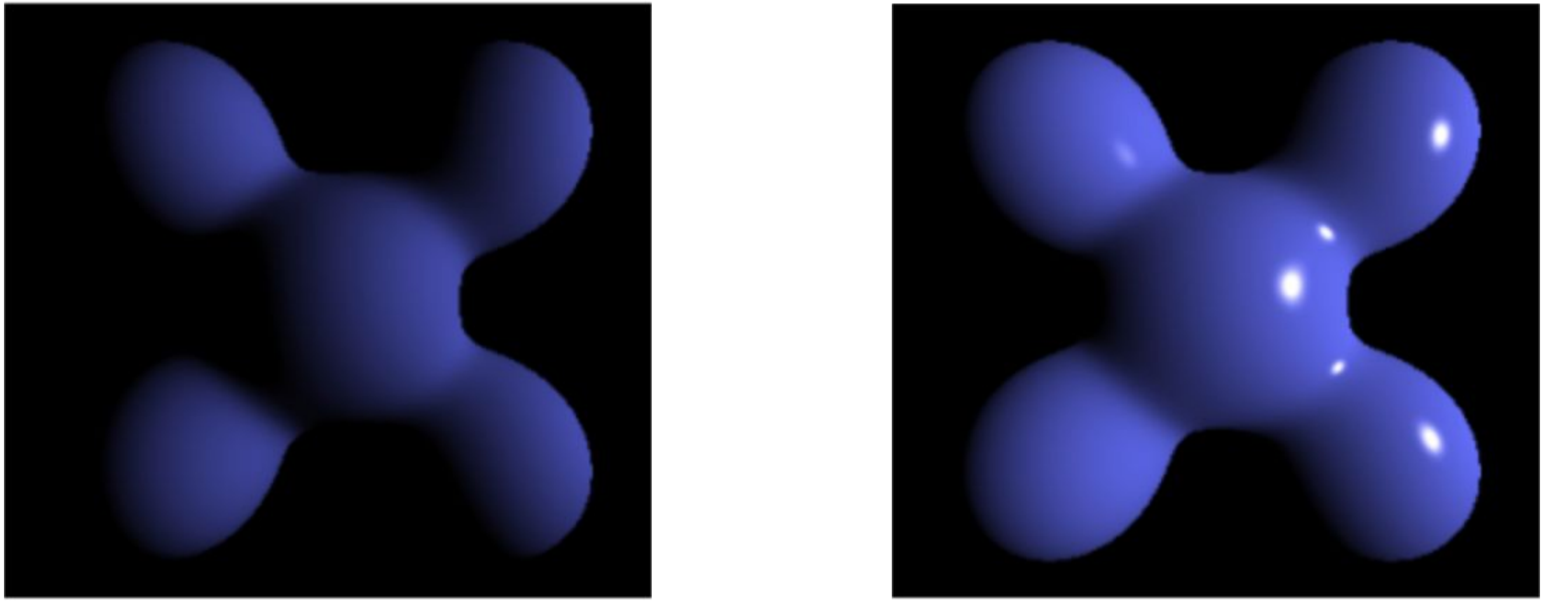

Rendering
Inverse Rendering

BRDF (Bidirectional reflectance distribution function)

$$f_r(\omega_i, \omega_o) \;\; (f_r(\omega_i \;\rightarrow\; \omega_o))$$

- Material model, gives probability that photon arriving from direction ω_i gets reflected into direction ω_o

$$f_r = \frac{\rho_d}{\pi} \qquad\qquad f_r^{Phong} = \frac{\rho_d}{\pi} + \frac{\rho_s(n+2)\cos^n\theta_r}{2\pi}$$



Background

Reflection equation

$$L_r(\omega_i \to \omega_o) = L_i(\omega_i) \cdot f_r(\omega_i \to \omega_o) \cdot \cos\theta_i$$

Reflection equation

$$L_r(\omega_i \to \omega_o) = L_i(\omega_i) \cdot f_r(\omega_i \to \omega_o) \cdot \cos\theta_i$$

$$\downarrow$$

$$L_r(\omega_o) = \int_{H(x)} L_i(\omega_i) \cdot f_r(\omega_i \to \omega_o) \cdot \cos\theta_i \, d\omega_i$$

Neural network - intuition

Neural network - intuition

- Purpose of neural network is to mimic human brain - create a graph where nodes represent neurons in brain, edges between neurons possibility to transfer information

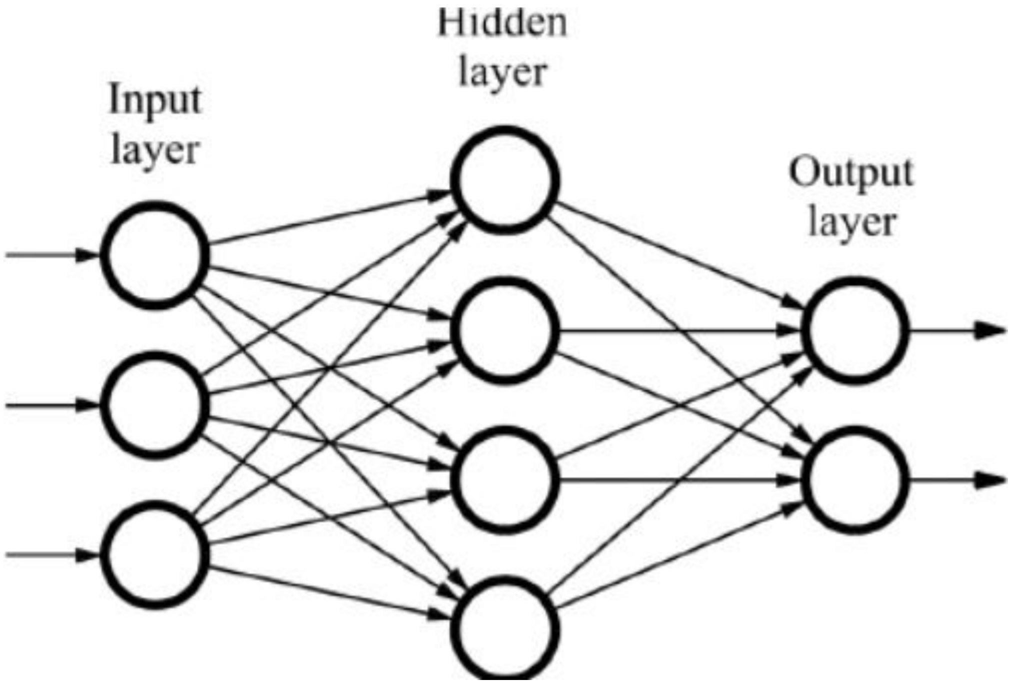
Neural network - intuition

- Purpose of neural network is to mimic human brain - create a graph where nodes represent neurons in brain, edges between neurons possibility to transfer information
- Neuron will “activate” when specific input is provided (just like neurons in brain activate to specific stimulus)

Neural network - intuition

- Purpose of neural network is to mimic human brain - create a graph where nodes represent neurons in brain, edges between neurons possibility to transfer information
- Neuron will “activate” when specific input is provided (just like neurons in brain activate to specific stimulus)
- Neural networks need to have employ some kind of mechanism to adjust themselves when they predict wrong answer

Neural networks - architecture



Neural networks - training

- Dataset

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

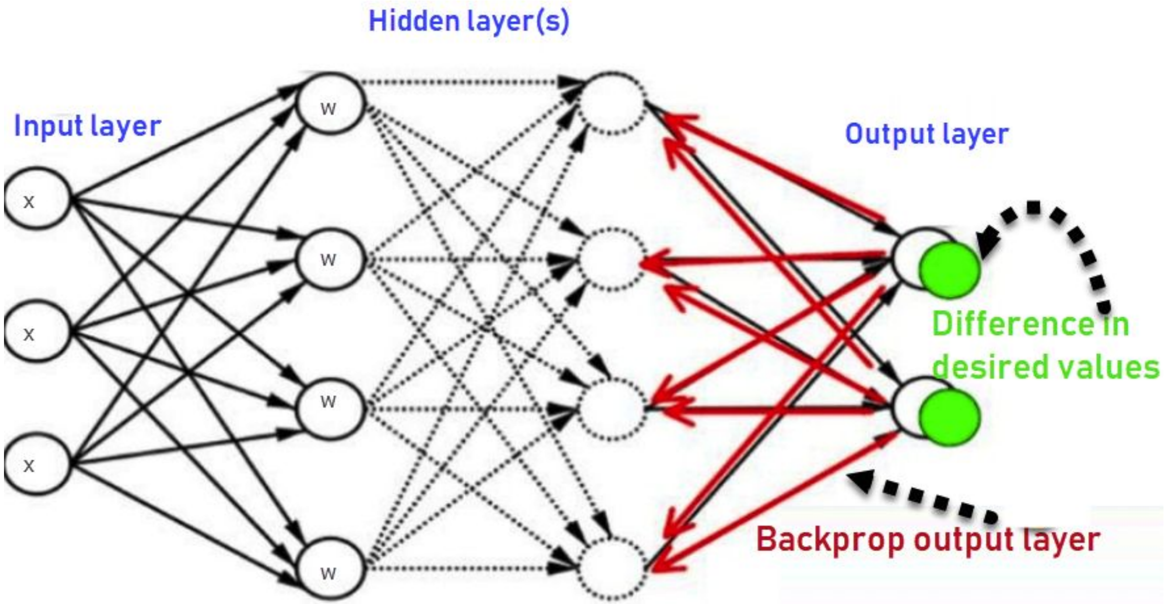
Neural networks - training

- Dataset
- Forward propagation

$$
\begin{aligned}
z^{[l]} &= W^{[l]}a^{[l-1]} + b^{[1]} \\
a^{[l]} &= g(z^{[l]}) \\
&\dots \\
z^{[L]} &= W^{[L]}a^{[L-1]} + b^{[L]} \\
a^{[L]} &= \sigma(z^{[L]})
\end{aligned}
$$

Neural networks - training

- Dataset
- Forward propagation
- Backpropagation



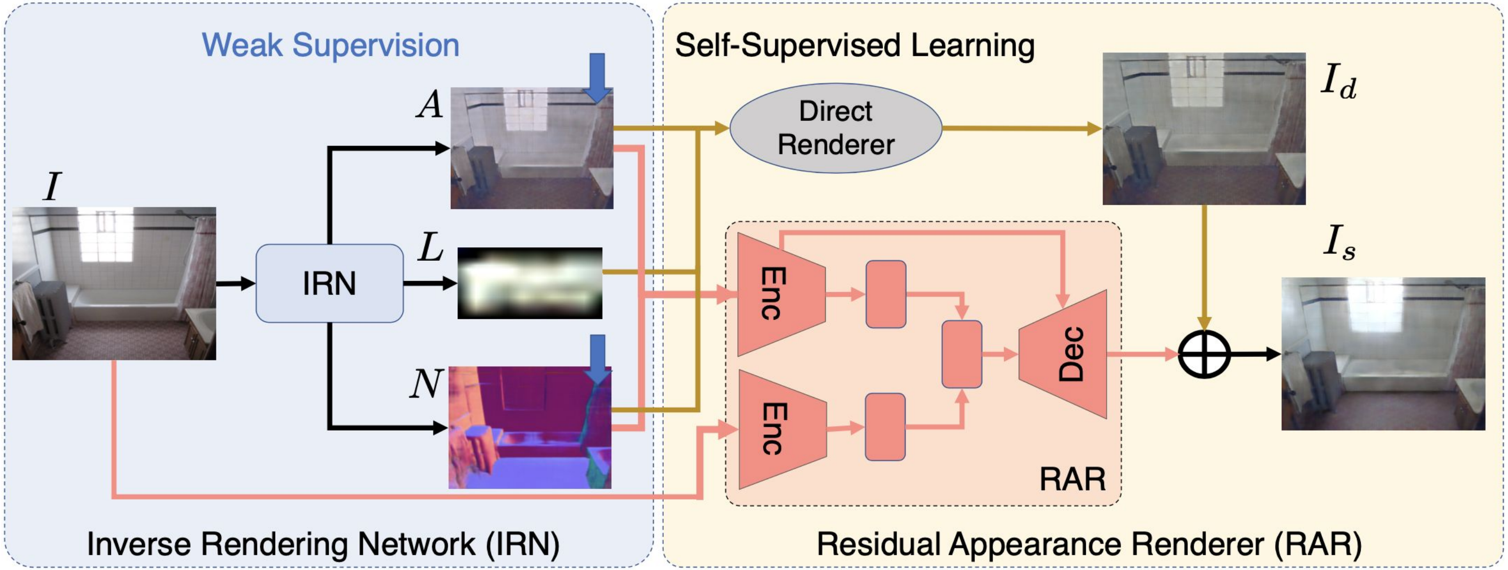
Convolutional neural networks

- In our work, we worked with an improved type of neural networks, called convolutional neural networks
- These networks employ convolution function when making prediction, which works better on images than plain neural networks

Our approach

Starting point

- We decided to replicate approach presented in [1]



Original direct render function

$$f_{direct} = \frac{4\pi * \frac{\pi}{2}}{648} \sum_{i=1}^{648} \frac{\rho_d}{\pi} * L(\omega_i) * \cos\theta_l * (\omega_i \cdot N)$$

Our approach

- On top of estimating normals, diffuse albedo and lighting, also predict **specular albedo, glossiness and view vector**
- Not use ideal diffuse BRDF, but physically correct Phong BRDF, yielding better images from direct renderer

$$f_{direct} = \frac{4\pi * \frac{\pi}{2}}{648} \sum_{i=1}^{648} \frac{\rho_d}{\pi} * L(\omega_i) * \cos\theta_l * (\omega_i \cdot N)$$

$$\downarrow$$

$$f_{Phong} = \frac{4\pi * \frac{\pi}{2}}{648} \sum_{i=1}^{648} \left(\frac{\rho_d}{\pi} + \frac{\rho_s (n+2) \cos^n \theta_r}{2\pi}\right) * L_i(\omega_i) * \cos\theta_l * (\omega_i \cdot N)$$

Comparison of direct render implementations

Original image

Fixed direct renderer

Our direct renderer

Comparison of direct render implementations

Original image

Fixed direct renderer

Our direct renderer

Generating dataset

- 1041 unique geometries

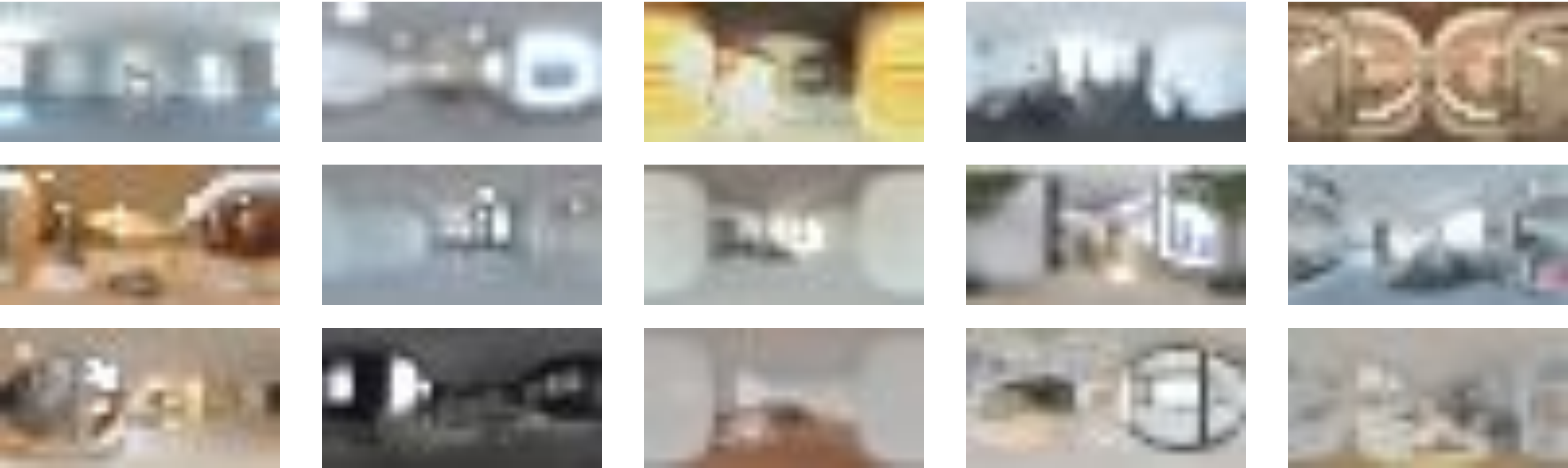
Generating dataset

- 1041 unique geometries
- 5709 images rendered in the dataset (with light modifications)

Generating dataset

- 1041 unique geometries
- 5709 images rendered in the dataset (with light modifications)
- 1005 env maps (compared to 205 used in [1])



Generating dataset

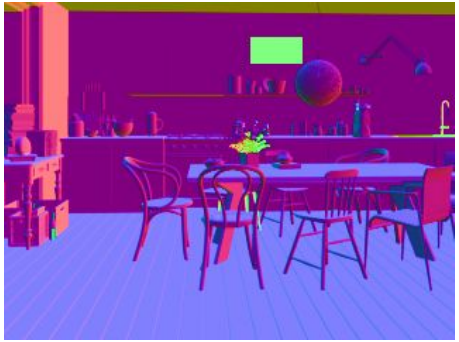
- 1041 unique geometries
- 5709 images rendered in the dataset (with light modifications)
- 1005 env maps (compared to 205 used in [1])
- Took about 1000 hours to render

GT data for one scene

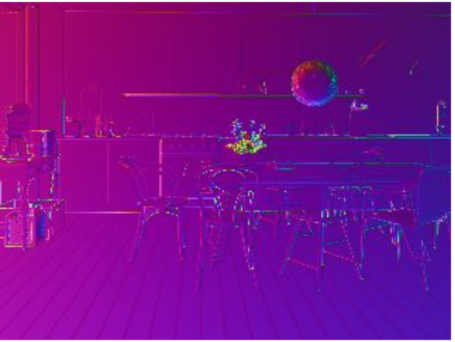

Environment map


Original image


Diffuse albedo


Normals


Specular albedo

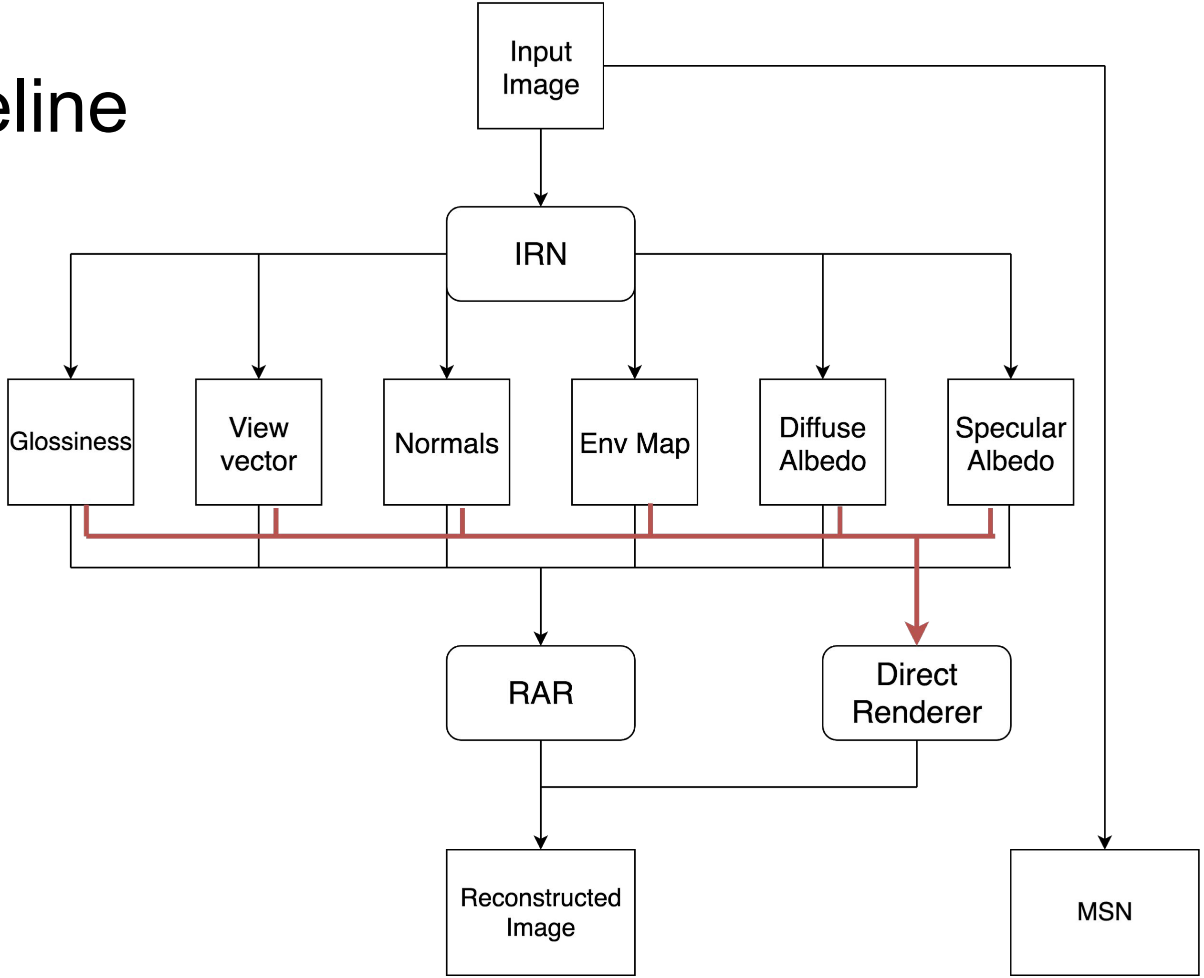

View vector


Depth

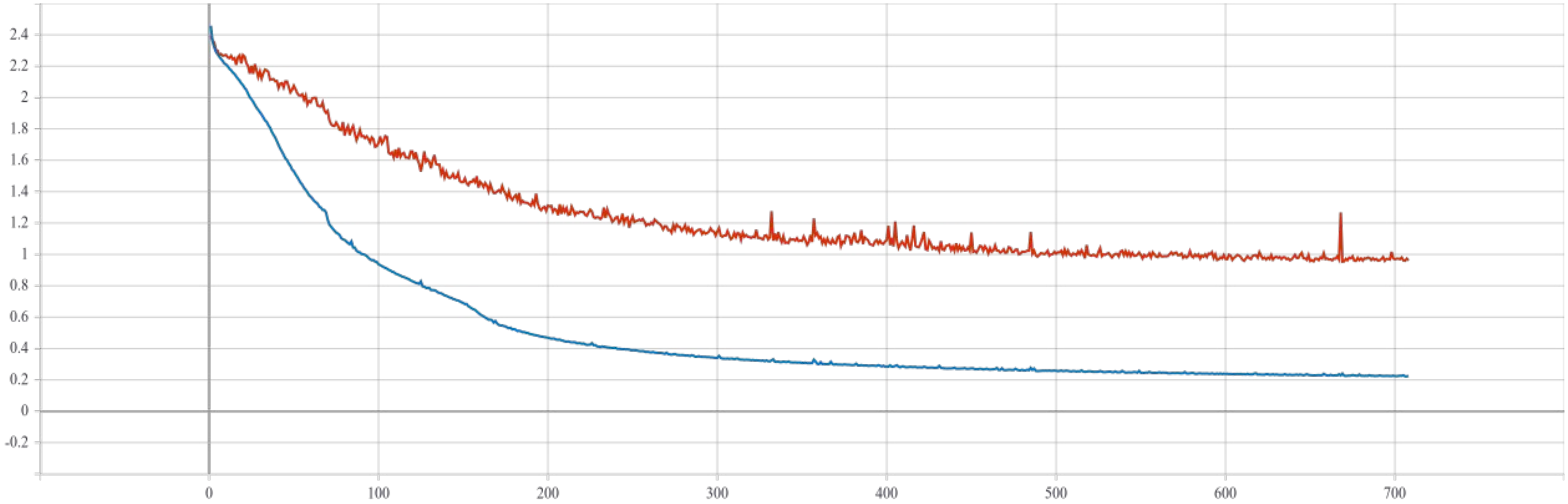

Glossiness


Material segmentation

Our pipeline



Results



Training IRN, validation (red) and training error (blue)

Train set		
Average Error	Lowest error	Highest error
0.221	0.133	1.924

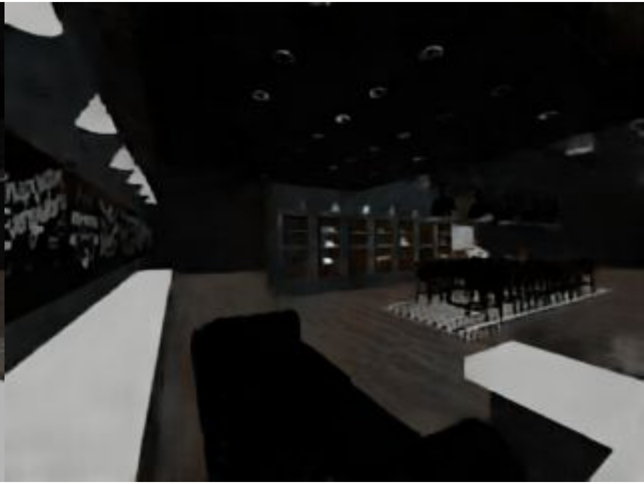
Test set		
Average Error	Lowest error	Highest error
0.973	0.202	3.131

Training IRN

- Results on training data

Original image

GT Diffuse Albedo



Predicted Diffuse Albedo

Average Error	This image error
0.221	0.222

Training IRN

- Results on training data

Original image

GT Specular Albedo

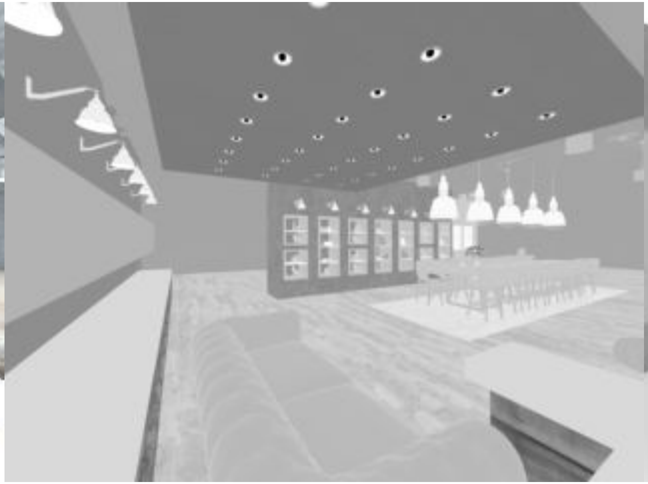
Predicted Specular Albedo

Average Error	This image error
0.221	0.222

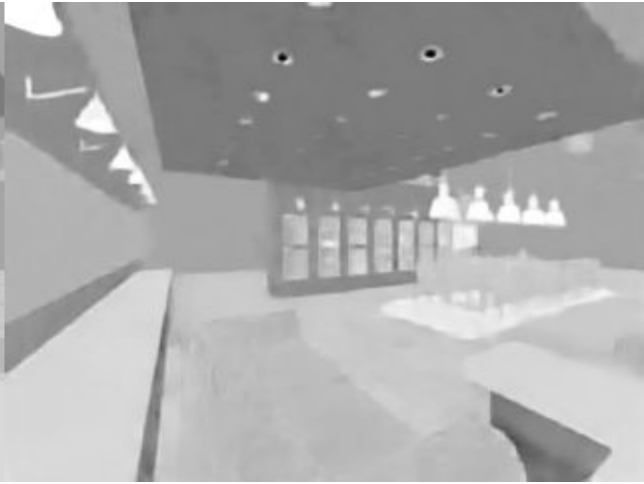
Training IRN

- Results on training data

Original image



GT Glossiness



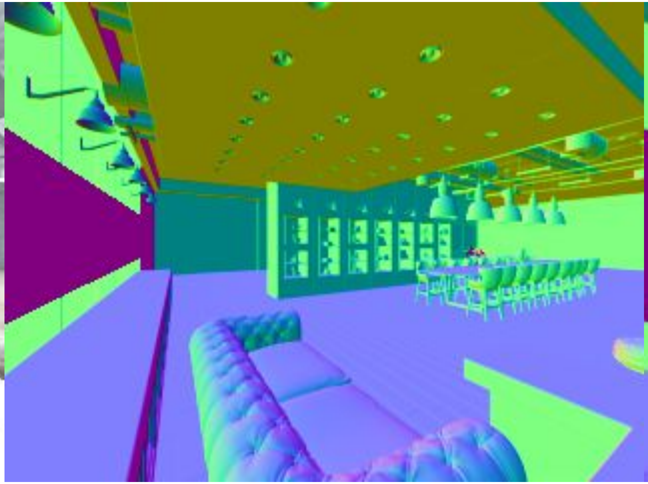
Predicted Glossiness

Average Error	This image error
0.221	0.222

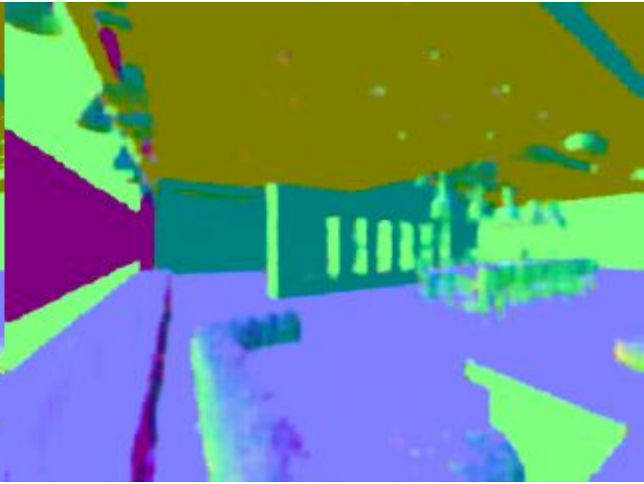
Training IRN

- Results on training data

Original image



GT Normals



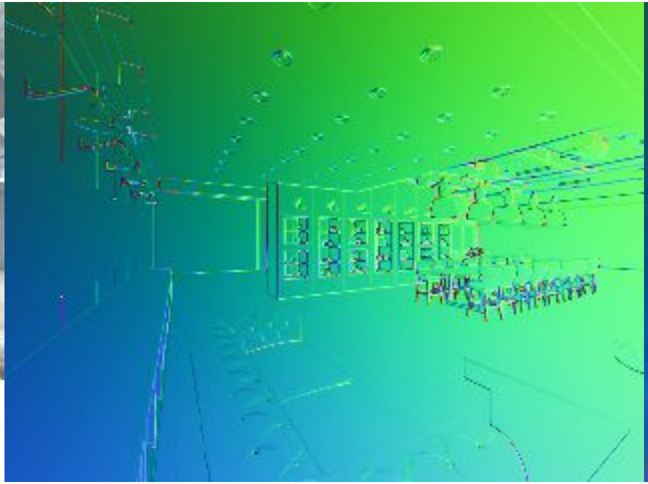
Predicted Normals

Average Error	This image error
0.221	0.222

Training IRN

- Results on training data

Original image



GT View vector

Predicted View vector

Average Error	This image error
0.221	0.222

Training IRN

- Results on testing data

Original image

GT Diffuse Albedo

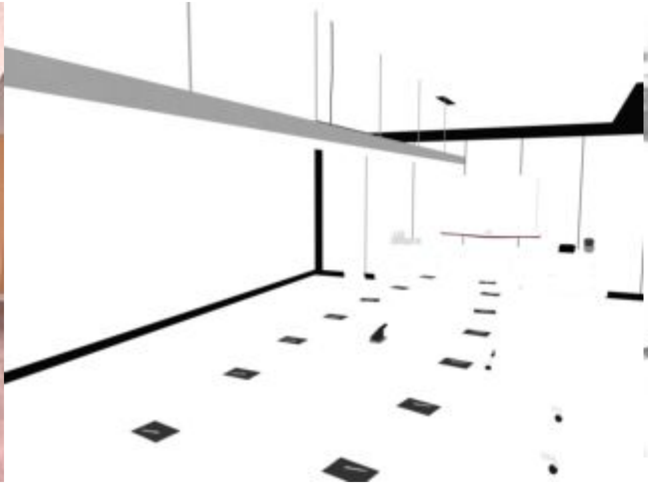
Predicted Diffuse Albedo

Average Error	This image error
0.973	0.940

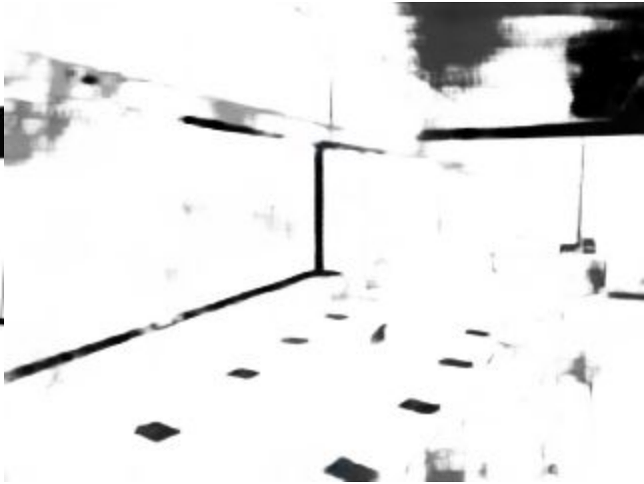
Training IRN

- Results on testing data

Original image



GT Specular Albedo



Predicted Specular Albedo

Average Error	This image error
0.973	0.940

Training IRN

- Results on testing data

Original image

GT Glossiness

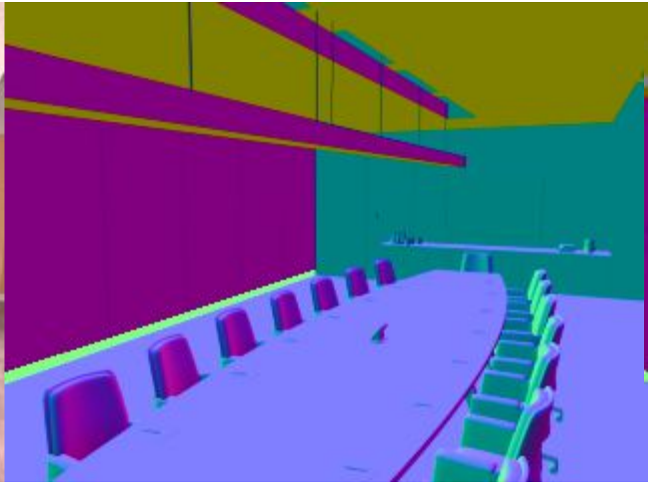
Predicted Glossiness

Average Error	This image error
0.973	0.940

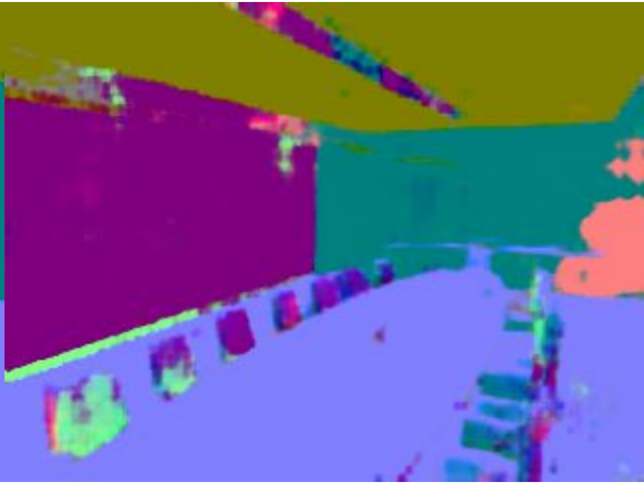
Training IRN

- Results on testing data

Original image



GT Normals



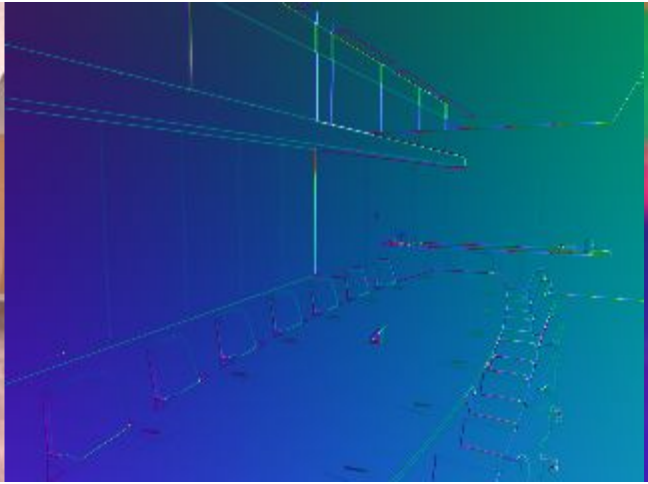
Predicted Normals

Average Error	This image error
0.973	0.940

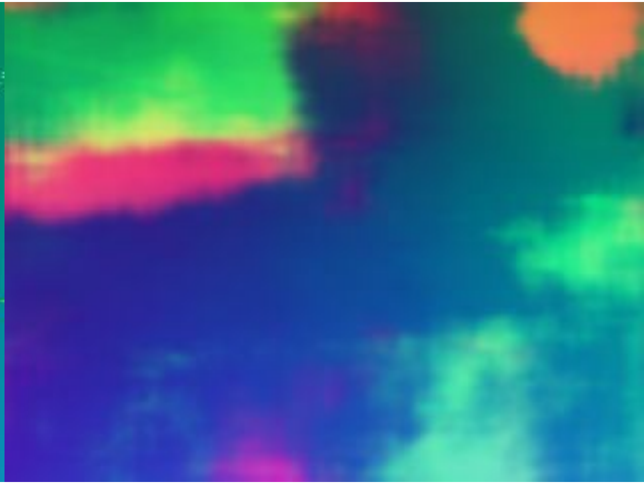
Training IRN

- Results on testing data

Original image

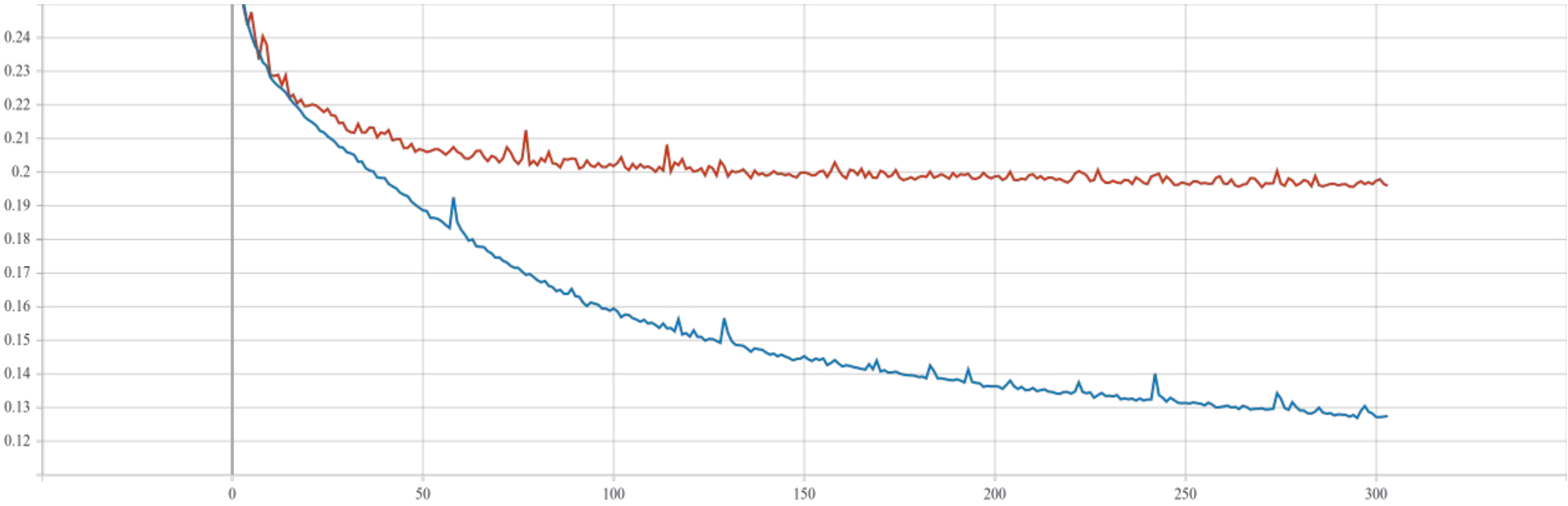


GT View vector



Predicted View vector

Average Error	This image error
0.973	0.940



Training RAR, validation (red) and training error (blue)

Train set		
Average Error	Lowest error	Highest error
0.128	0.043	0.436

Test set		
Average Error	Lowest error	Highest error
0.194	0.082	0.471

Training RAR - reconstruction

- Results on training data

Original image

Direct renderer

Reconstructed image

Average Error	This image error
0.128	0.128

Training RAR

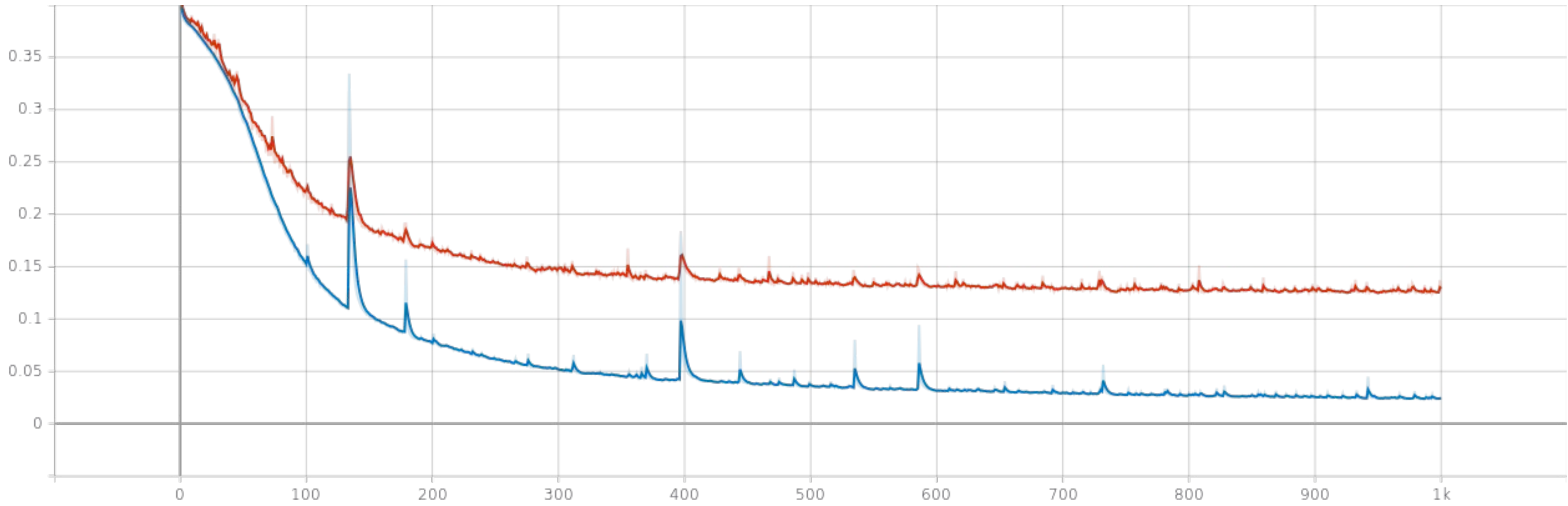
- Results on testing data

Original image

Direct renderer

Reconstructed image

Average Error	This image error
0.194	0.190



Training MSN, validation (red) and training error (blue)

Train set		
Average Error	Lowest error	Highest error
0.032	0.004	0.080

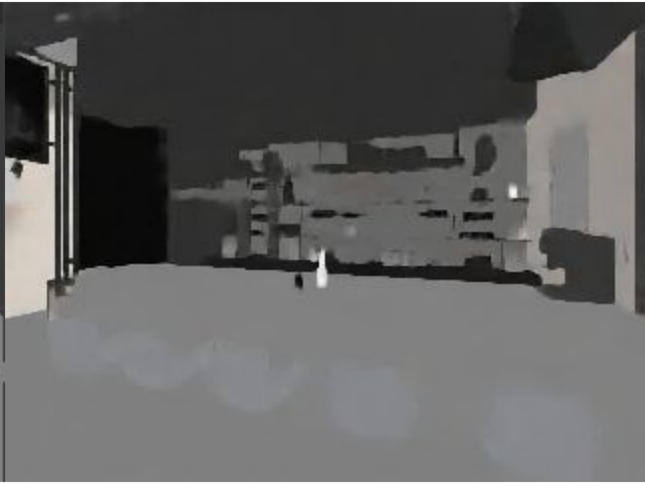
Test set		
Average Error	Lowest error	Highest error
0.137	0.005	0.615

Training MSN

- Results on training data

Original image

GT Segmentation



Predicted Segmentation

Average Error	This image error
0.032	0.032

Training MSN

- Results on testing data

Original image

GT Segmentation

Predicted Segmentation

Average Error	This image error
0.137	0.138

Conclusion

- We proved that neural networks are powerful learning representations that can learn useful priors even when it comes to such unconstrained problems like inverse rendering or material segmentation

Conclusion

- We proved that neural networks are powerful learning representations that can learn useful priors even when it comes to such unconstrained problems like inverse rendering or material segmentation
- We acknowledge that our results are limited, as we don’t have dataset of enough size to generalize well

Future work

- Enlarge dataset
- Predict better lighting (spatially varying lighting)
- More parameters (leading to better BRDF)
- Texture transfer

Sources

[1] Neural Inverse Rendering of an Indoor Scene From a Single Image

Thank you for your attention

Supervisor’s questions

- **Q**: Pri terajšom rozsahu navrhnutej topológie ste zvládli trénovanie na 6000 obrázkoch. Viete odhadnúť kde je limit dát aby sme sieť nepreučili alebo ináč aby sme nepokazili dobre nastavené parametre príliš veľkým množstvom dát?

Supervisor’s questions

- **Q**: Pri terajšom rozsahu navrhnutej topológie ste zvládli trénovanie na 6000 obrázkoch. Viete odhadnúť kde je limit dát aby sme sieť nepreučili alebo ináč aby sme nepokazili dobre nastavené parametre príliš veľkým množstvom dát?
- **A**: Nemyslíme si že by sa sieť pokazila tým že pridáme viac **dobrých dát** do nášho dataset-u, to znamená takých, že sa zachová alebo zmenší rozdiel v distribúcii dát pre trénovaciu a testovaciu množinu. Takisto chceme zachovať čo najväčšiu rozmanitosť dát. Dôležitá teda nie je len veľkosť, ale aj aké dáta pridávame do dataset-u. Ak sa teda pridávajú len dobré dáta na trénovanie, nemalo by to negatívne ovplyvniť výsledky siete.

Reviewer’s questions

- **Q**: Je k testovaným modelom okrem číselného ohodnotenia toho, akého množstva chýb sa dopúšťajú, možné slovne charakterizovať nejaké konkrétne bežné typy chýb, ktoré je možné vo výstupe pozorovať?

Reviewer’s questions

- **Q**: Je k testovaným modelom okrem číselného ohodnotenia toho, akého množstva chýb sa dopúšťajú, možné slovne charakterizovať nejaké konkrétne bežné typy chýb, ktoré je možné vo výstupe pozorovať?
- **A**: Najväčší problém bol s trénovaním normál. To je vidieť hlavne pri hranách objektov v scénach s veľmi malými objektmi, kde mala sieť problém rozoznať hranice a tým pádom často bol výsledný obrázok na týchto častiach dosť nepresný. Takisto nebolo úplne presný odhad osvetlenia. Keďže sme na odhad svetla používali veľmi malú mriežku, pre testovacie dáta občas sieť nebola schopná zachytiť správne osvetlenie.