CMPT732 Course Project Final Report

Topic: Single image based SVBRDF Recovery

Team Member:	**Yingqian Gu**
	Zixuan Li
	Bichun Zhang
Project Contact Person:	**Changqing Zou**
Project Supervisor:	**Ali Mahdavi-Amir**
Submission Date:	**12/12/2021**

Background

The images of real-life objects are the outcome of interactions between lighting, geometrics and materials. Humans can perceive the material appearance in a single image based on texture, highlights, shading and some other visual cues. However, estimating the properties of a material from a single photograph using a computer requires unraveling these complex interactions and is challenging, let alone even more complex global illumination effects. In order to address the problem, materials can be characterized by spatially-varying bi-directional reflectance distribution functions (SVBRDFs), which are high-dimensional functions that depend on viewing and lighting conditions. In recent years, the advent of convolutional neural networks (CNNs) made it possible to recover SVBRDFs using just a single image.

Introduction

Our work is to compare 3 existing models:

1. *Single-Image SVBRDF Capture with a Rendering-Aware Deep Network*, Valentin Deschaintre, Miika Aittala, Fredo Durand, George Drettakis, Adrien Bousseau. ACM Transactions on graphics (SIGGRAPH Conference Proceedings), 2018
2. *Deep Inverse Rendering for High-resolution SVBRDF Estimation from an Arbitrary Number of Images*, Duan Gao, Xiao Li, Yue Dong, Pieter Peers, Kun Xu, Xin Tong, ACM Transactions on Graphics (SIGGRAPH Conference Proceedings), 2019
3. *Adversarial Single-Image SVBRDF Estimation with Hybrid Training*, Xilong Zhou, Nima Khademi Kalantari, Eurographics, 2021

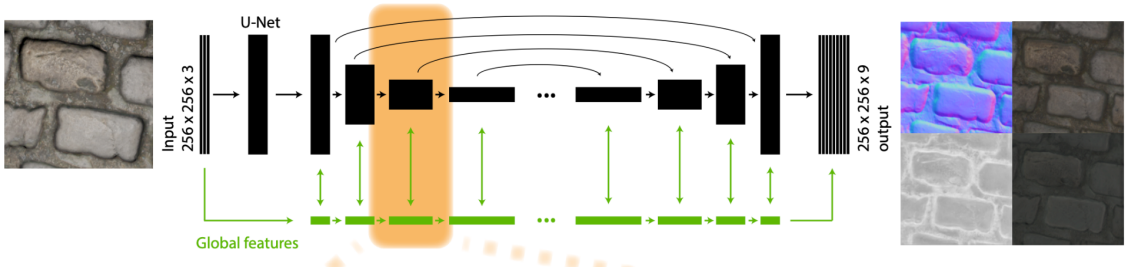
Inference the 3 models on the same set of testing data, including both real world materials and synthesis materials. Compare the re-rendered images and output feature maps to the original texture and reference feature maps using a numerical way. Then include the run time factor to reach a conclusion of which model has the best performance, or which model is the best choice in specific circumstances.

Comparison of Different Models

For the convenience of referring to the 3 models, we use a short abbreviation based on the order of their publication years.

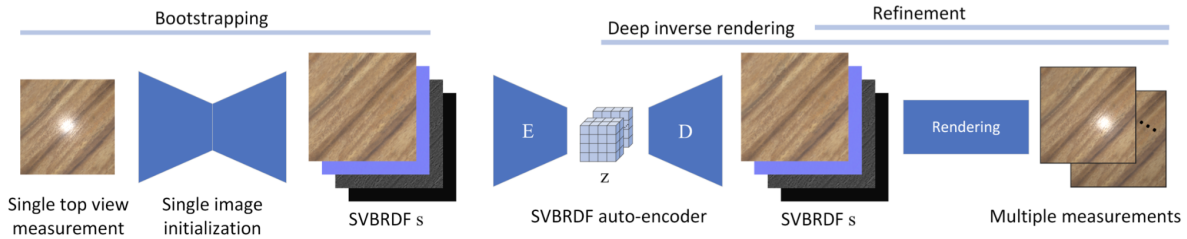
• Algorithm Comparison

Model 1: Single-Image SVBRDF Capture with a Rendering-Aware Deep Network



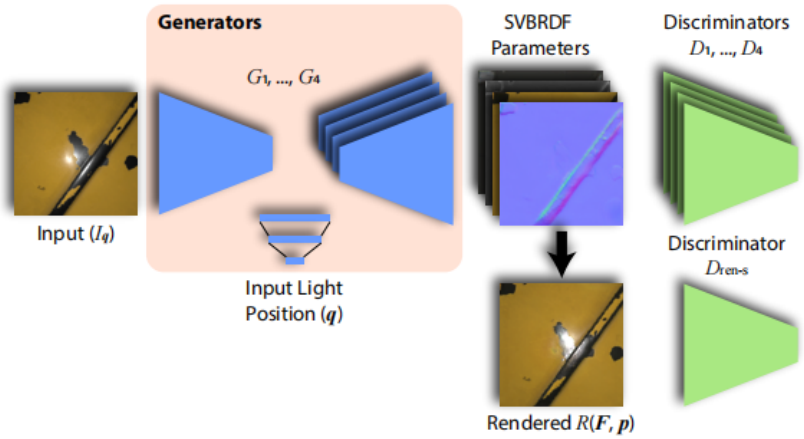
- Model 1 uses U-Net architecture. There are 8 CNNs for downsampling and 8 CNNs for upsampling.
- The black part is an Encoder-Decoder convolutional track for local feature extraction. The green track uses fully-connected layers to extract and propagate global features, and it processes vectors instead of feature maps.
- Skip connections are used between same-sized layers of the Encoder and Decoder, which helps the Decoder to synthesize details of the input.
- Training Loss:
  Model 1 compares the L1 distance between the input and the rendered result pixel-wise.

Model 2: Deep Inverse Rendering for High-resolution SVBRDF Estimation from an Arbitrary Number of Images

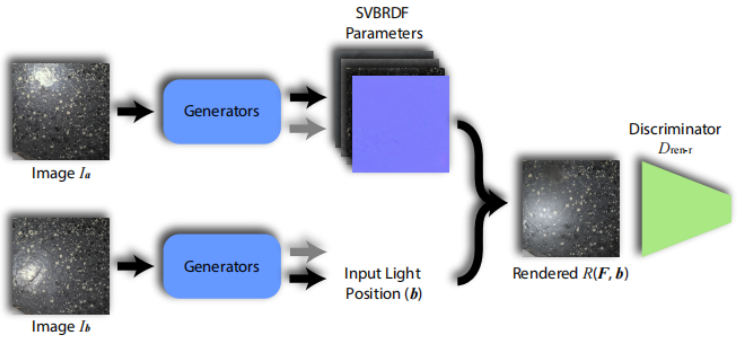


- The main goal of Model 2 is to generate a latent space(z) for deep inverse rendering and for optimization.
- Bootstrapping: use model 1 to generate feature maps of inputs for training.
- Auto-encoder network structure: use 6 CNNs for downsampling and 6 CNNs for upsampling.
- Refinement: add details to the output by directly comparing the L1 distance between the rendered prediction and the input.
- Training Loss:
  A combination of L1 loss between the predicted maps and input maps plus mean of render loss on 9 visualizations of the materials.
  Model 2 also introduces an additional Smoothness Loss during the Encoder training part, which makes sure that a small change in the latent code should result in a small change in the SVBRDF maps(vice versa).

Model 3: Adversarial Single-Image SVBRDF Estimation with Hybrid Training
Model3 is trained on two Generative Adversarial Networks and the final training loss is a combination of both GANs



- The aim of the above GAN is to output the final 4 feature maps.
- Input: synthetic images with a flashlight at position q.
- Generators:an encoder-decoder architecture with a shared encoder and four decoders to estimate the four parameters given an input image.
- Light position: estimated by the encoded features through a set of fully connected layers.
- Discrimators:a set of 4 discriminators with an additional discriminator to evaluate the quality of both the estimated parameters and the rendered images.
- Training Loss:
  The training loss includes three parts: the parameter loss, the rendering loss and the loss for light position.
  The parameter loss is a summation of adversarial loss based on least square formulation and the L1 distance between the features at different layers of the discriminator.
  The rendering loss is computed the the way as the parameter loss except that it adds a VGG based perceptual loss to ensure the consistent results of all generators.
  The loss for lighting is also the L1 distance between the estimated and the ground truth.



- The aim of the above GAN is to output 4 feature maps as the ground truth for the other GAN.
- Input: a pair of real images with different lighting positions.

- Generators:estimate the parameters and light position, then the parameters and light position are used to render an image.
- Discriminators:evaluate the quality of the rendered image with image Ib used as ground truth.
- Training Loss:
  The training loss is computed in the similar way of the other GAN.

• Testing Result Comparison

- ❖ We evaluate 3 models on a set of 84 synthetic and 13 real images in terms of *LPIPS* and *RMSE*.
- ❖ N, D, R and S refer to normal, diffuse albedo, roughness, and specular albedo respectively. Ren refers to corresponding renderings.

1. Learned Perceptual Image Patch Similarity (LPIPS) metric

LPIPS evaluates the distance between image patches.
Higher means further/more different. Lower means more similar

2. Root Mean Square Error (RMSE)

RMSE measures the standard deviation of the residuals (prediction errors).
Higher means further/more different. Lower means more similar.

Real-world images:

The following evaluates the re-rendered predictions of 5 real images both in terms of *LPIPS* and *RMSE*.
Note:

1. *There are no ground truth SVBRDFs for real-world images, so we only calculate loss of the feature maps on synthetic images.*
2. *The dynamic range of flash photographs can be large. Model 1 normalized the input by transforming the input image into logarithmic space and compacting it to the range[0, 1].*

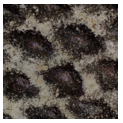
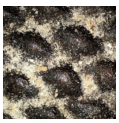
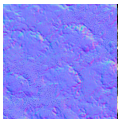
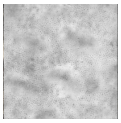
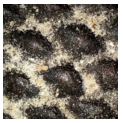
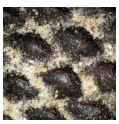
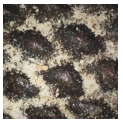
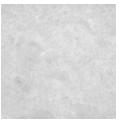
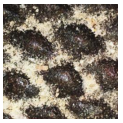
image1	input	N	D	R	S	Ren	LPIPS	RMSE
model 1							*0.3101*	*2.2433*
model 2							***0.0044***	***1.0638***
model 3							*0.0977*	*2.8641*

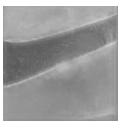
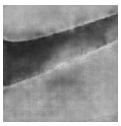
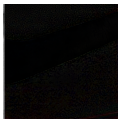
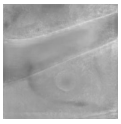
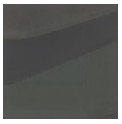
image2	input	N	D	R	S	Ren	LPIPS	RMSE
model 1							*0.2552*	*2.2817*
model 2							***0.0007***	***0.4788***
model 3							*0.1189*	*3.0234*

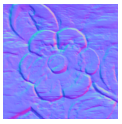
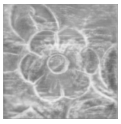
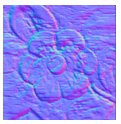
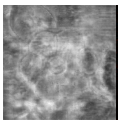
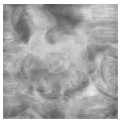
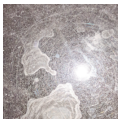
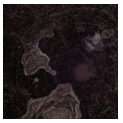
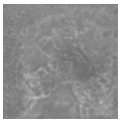
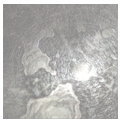
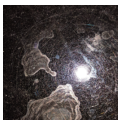
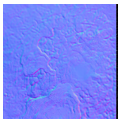
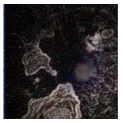
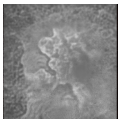
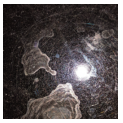
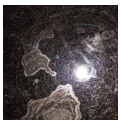
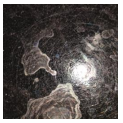
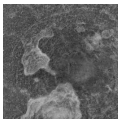
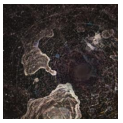
image3	input	N	D	R	S	Ren	LPIPS	RMSE
model 1							*0.4598*	*2.1722*
model 2							***0.0009***	***0.8703***
model 3							*0.1765*	*3.1787*

image4	input	N	D	R	S	Ren	LPIPS	RMSE
model 1							*0.2608*	*2.2190*
model 2							***0.0004***	***0.5546***
model 3							*0.0592*	*2.0958*

image5	input	N	D	R	S	Ren	LPIPS	RMSE
model 1							*0.3483*	*2.0523*
model 2							***0.0002***	***0.3838***
model 3							*0.1089*	*2.8267*

Synthetic images:

The following evaluates the predicted feature maps and the corresponding re-rendered images on 5 synthetic images in terms of both LPIPS and RMSE.

image6	input	N	D	R	S	Ren
model 1						
RMSE			*9.5851*			*2.2478*
LPIPS			*0.3718*			*0.3524*
model 2						
RMSE			9.4004			***0.7806***
LPIPS			*0.3488*			***0.0008***
model 3						
RMSE			7.6797			*2.5324*
LPIPS			*0.1964*			*0.0500*
reference						

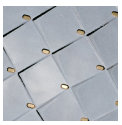
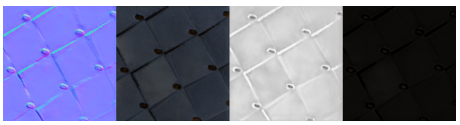
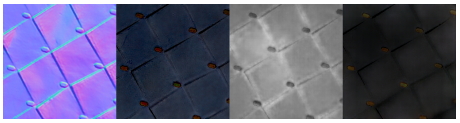
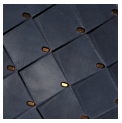
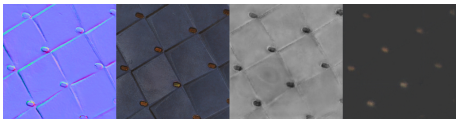
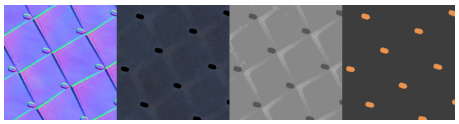
image7	input	N	D	R	S	Ren
model 1						
RMSE			*8.3118*			*2.2228*
LPIPS			*0.3892*			*0.3240*
model 2						
RMSE			*9.2202*			***0.6892***
LPIPS			*0.3071*			***0.0023***
model 3						
RMSE			*7.9082*			*2.2562*
LPIPS			*0.2810*			*0.0668*
reference						

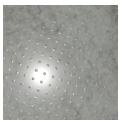
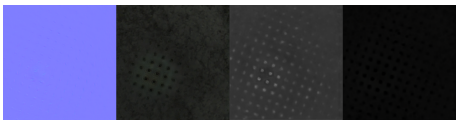
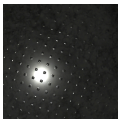
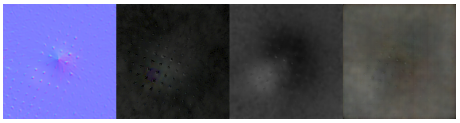
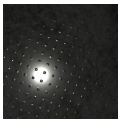
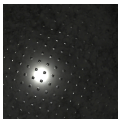
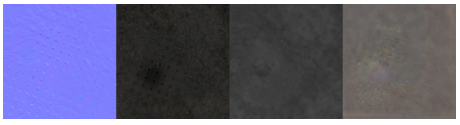
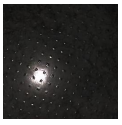
image8	input	N	D	R	S	Ren
model 1						
RMSE			*6.4678*			*2.7805*
LPIPS			*0.4414*			*0.3835*
model 2						
RMSE			*7.0996*			***0.4563***
LPIPS			*0.3700*			***0.0001***
model 3						
RMSE			*6.1045*			*2.6144*
LPIPS			*0.3235*			*0.0849*
reference						

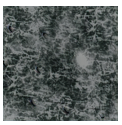
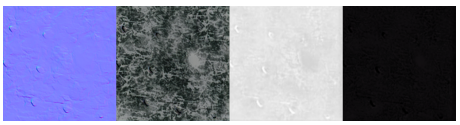
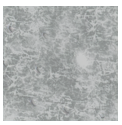
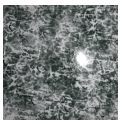
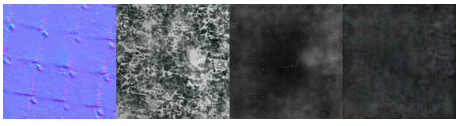
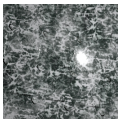
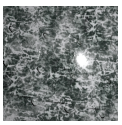
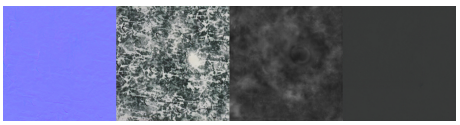
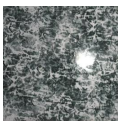
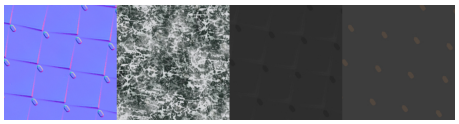
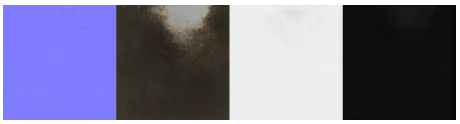
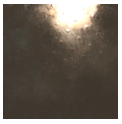
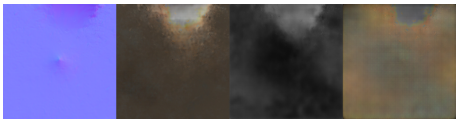
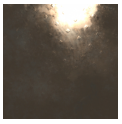
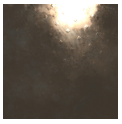
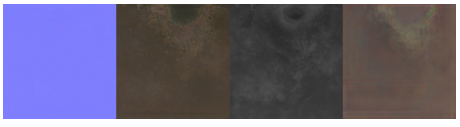
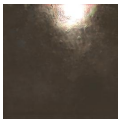
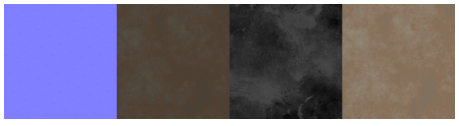
image9	input	N	D	R	S	Ren
model 1						
RMSE			*9.3413*			*2.2831*
LPIPS			*0.4798*			*0.2658*
model 2						
RMSE			*8.7560*			***0.8582***
LPIPS			*0.5146*			***0.0060***
model 3						
RMSE			*6.3731*			*2.4571*
LPIPS			*0.3684*			*0.0700*
reference						

image10	input	N	D	R	S	Ren
model 1						
RMSE			*8.2160*			*2.2944*
LPIPS			*0.5042*			*0.1577*
model 2						
RMSE			*8.0893*			***0.3118***
LPIPS			*0.3337*			***0.0001***
model 3						
RMSE			6.5872			*2.3110*
LPIPS			*0.2741*			*0.1191*
reference						

Average result comparison:
The following tables show the average loss on 97 test images in terms of both LPIPS and RMSE.

LPIPS			
	Synthetic images		Real-world images
	AVG of (N,D,R,S)	Ren	Ren
model 1	0.4317	0.3157	0.3200
model 2	0.3578	0.0010	0.0013
model 3	0.3002	0.0939	0.1087

Table1: Average LPIPS comparison both on the set of real images and synthetic images. The first column is the average LPIPS of the feature maps. The second and third column are the average LPIPS of the re-rendered results.

RMSE						
	Synthetic images					Real-world images
	N	D	R	S	Ren	Ren
model 1	8.6760	6.1905	10.1411	10.4656	3.1701	3.2107
model 2	9.5474	6.0568	9.2545	9.7333	0.4825	0.5338
model 3	7.4298	6.1401	9.0145	7.4149	2.4308	2.6775

Table2: Average RMSE comparison both on the set of real images and synthetic images. The first 4 columns are the average RMSE of the feature maps. The last 2 columns are the average RMSE of the re-rendered results.

- **Time Performance Comparison**
  Using NVIDIA GeForce GTX 1070 Ti 6GB. Testing on 84 synthesis images
  - **Model 1:** 40.55s
  - **Model 2:** 1836.09 (using the default settings of 4000 iterations for improving resolution and 200 iterations for refinement)
  - **Model 3:** 33.12s

Limitations and advantages

Model 1 has the simplest network structure, yet it can only process images with relatively low resolution and tends to produce correlated structures in different maps. Also, materials outside the scope of the training data cannot be reproduced properly.

Model 2 allows multiple inputs under different lighting conditions and supports high-resolution inputs and outputs. It improves the quality of SVBRDFs estimated from single-image inputs compared to **Model 1**, especially when the target material is outside the training dataset. But this method is highly underconstrained, therefore it often introduces unnecessary details to the normal maps.
We need to choose the starting point carefully, since nonlinear mapping from latent code to rendered images is like any nonlinear optimization: a suboptimal starting point may lead to a local minimum.
Long running time is another limitation, because Model 2 needs to initialize an arbitrary number of input photos.

Model 3 does not rely on the feature maps of synthetic images,like model 1 or model2, but instead generates feature maps from real image pairs. Thus it performs better when it comes to real examples. However, in some cases with strong specular highlights, model3 fails to properly remove the highlights from the estimated parameters. Furthermore, model 3 is not able to properly estimate the map in the regions away from the highlights,since the single input image does not provide useful information in these areas.

Conclusion

Based on the re-rendered result, model 2 not only has the best performance but also has the highest resolution, owing to its iterative steps for minimizing render loss and refinement. The only problem is that the run time is much longer than the other two models. So if time is an important factor to be considered while resolution isn’t essential, model 3 could also produce acceptable results with a decent run time.

Comparing the inferenced feature maps, model 3 has the best result. Although a low feature maps loss doesn’t lead to a low render loss directly, it still means the network of model 3 has the best performance on analysing the spacial variant of a texture (i.e. bump, valley, roughness). This could be crucial if analysing the property or rendering the texture in a 3D environment.