Final Project for Fundamentals of Digital Media Technology

Name: Bo Li ID: 1500012204
School of Life Sciences
Computer Science and Technology, Double Major

2018/01

Abstract

The bidirectional reflectance distribution function (BRDF) is a kind of description for the surface reflectance properties. Given the information about incident light radiance and BRDF of a point on the surface, the reflected outgoing radiance can be computed according to rendering equation:

$$L_o(p, w_o) = \int_{S^2} f(p, w_o, w_i) L_i(p, w_i) |cos\theta_i| dw_i \tag{1}$$

f is the BRDF function with p (position on the surface), w_o (out direction) and w_i (incident direction) as parameters.

BRDF plays an important role in the representation of material's optical appearance and has wide-spread applications especially in computer graphics. According to the defination, BRDF can be sampled using physical devices to measure the light's radiance or irradiance on incident and outgoing directions. In practical situations, however, a physically accurate estimate of BRDF is unnecessary and time-consuming. Many image-based BRDF modeling techniques have been developed. In this project, I followed a research by MSRA described in the paper *Modeling Surface Appearance from a Single Photograph using Self-augmented Convolutional Neural Networks* presented on SIGGRAPH 2017 and reproduced their methods in Tensorflow.

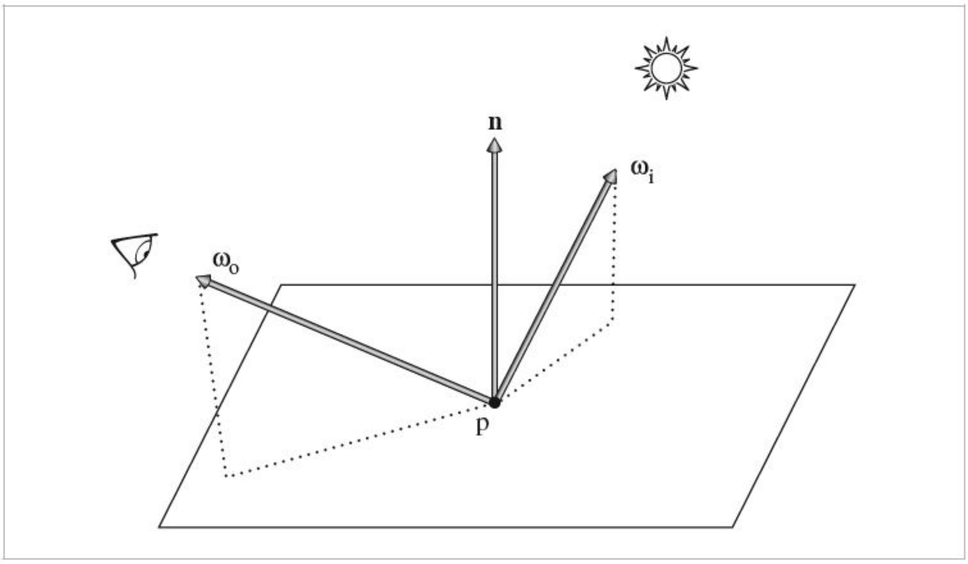

Figure 1: BRDF is a function over pairs of incident and outgoing directions that describes how much light along w_i is scattered from the surface in the direction w_o. The figure comes from reference 2.

Contents

1 Introduction

Generally speaking, modeling BRDF from a single photo is a ill-posed problem. In addition, the unknown illumination condition may lead to ambiguity between lighting and BRDF (scaling lighting can be compensated by scaling the BRDF with a inverse factor, but reflectance remains unchanged). Fortunately, data-driven machine learning methods have shown successful solutions to such underconstrained problems. In the paper, the researchers proposed a convolutional neural network (CNN) to extract spatial-varying BRDF (SVBRDF) from a single photo under unknwon lighting conditions.

However, deep learning methods rely heavily on large training datasets. In this problem, the labels of training data refer to their corresponding BRDFs. However, obtaining BRDF labels by artists manually is a very time-consuming job. Thus, unlabeled data with synthesized labels must be fed together with truly labeled data. They called this strategy "self-augmentation". Its details and effects will be presented later.

2 BRDF Model and Network Stucture

The previous description about BRDF is just a schematic one. In practice, many BRDF models has been proposed. They have their own properties and complexities, but all obey two laws: *reciprocity* ($f(p, w_o, w_i) = f(p, w_i, w_o)$) and *energy conservation* ($\int_{S^2} f(p, w_o, w)|cos\theta|dw \leq 1$).

In this work, we don't consider the geometry of surface (i.e., the input image appears to be a flat plane without discontinuity, occlusion or other unexpected situations). A BRDF model with similar form as Ward model is used:

$$f(w_i, w_o, x) = \frac{\rho_d(\mathbf{x})}{\pi} + \rho_s \frac{e^{-tan^2\delta/\alpha^2}}{4\pi\alpha^2\sqrt{(w_i \cdot \mathbf{n}(\mathbf{x})(w_o \cdot \mathbf{n}(\mathbf{x})}} \tag{2}$$

The reflection is separated to two parts: diffuse (Lambertian) and specular reflection, notated by d and s. In this work, our target is to predict ρ_d, ρ_s, α and $\mathbf{n}$. The former two factors are the diffuse and albedo specular. The latter two, roughness and surface normal are both related to specular reflections. δ is used as the angle between normal and halfway vector $w_h = (w_i + w_o)/||w_i + w_o||$. Note that $\mathbf{n}$ and ρ_d changes per pixel $\mathbf{x}$. ρ_s is homogeneous constant in order to reduce the complexity of the question. Here we show the structure of CNN used:

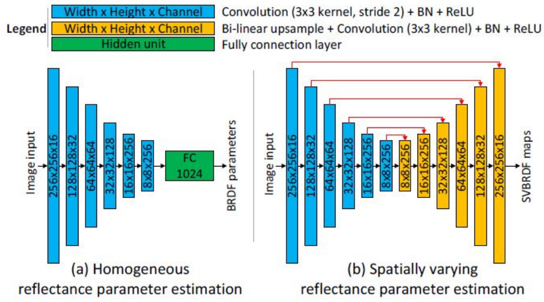

Figure 2: A schematic representation of the SVBRDF-Net. Copied from the original paper. The left and right subfigures are for homogeneous and spatially varying BRDF parameters respectively.

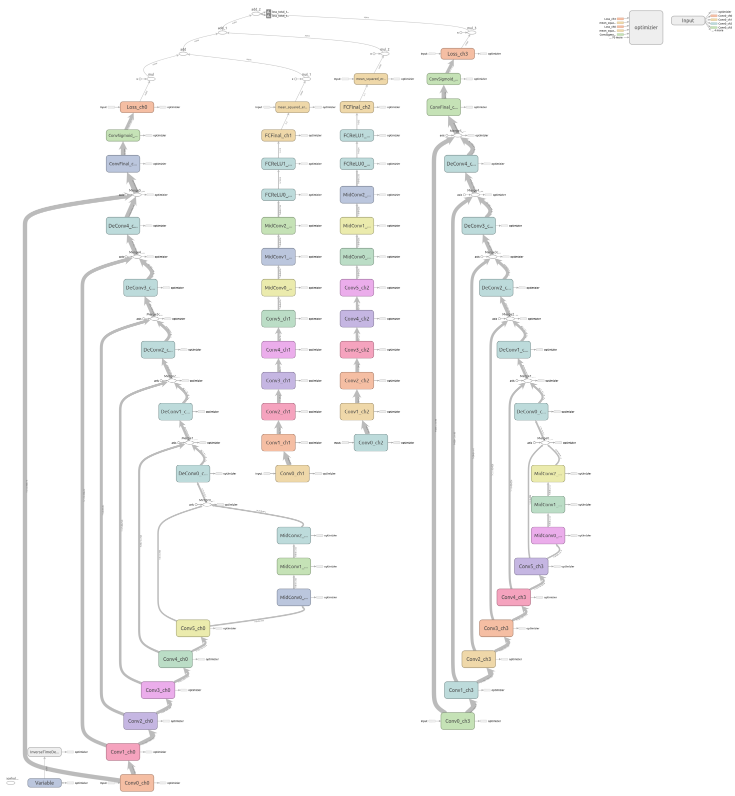

Figure 3: A detailed network structure generated by Tensorboard. The network contains four channels, giving prediction for diffuse albedo, specular albedo, roughness and normals with four losses. The final loss is their weighted summation.

Similar to other CNNs, the SVBRDF-Net implement a series of convolution layers to reduce image resolution by half and analyze features. Each convolution layer is follwed by a batch-normalization layer (accelerating training) and a ReLU activation layer. But note the different synthesis subnetwork for homogeneous and spatially-varying parameters. Homogeneous ones (roughness is a scalar, and specular albedo is a 3D vector) are synthesized using three fully connection layers. As for spatially-varying ones, they are re-synthesized using deconvolution (actually a convolution with a bilinear upsampling layer) opposed to convolution layers. Note that we also concatenate the deconvoluted feature maps from the corresponding analysis layers to re-introduce high frequency information lost in further convolution. The concatenated data are fed together as the input of subsequent layers.

3 Training and Self Augmentation

The input image is 256×256, with 4 labels as shown. As we have mentioned before, gaining labeled data for this task is a difficult job. Only 384×384 labels are provided in the dataset, we can directly render a image with given BRDF and randomly selected lighting (cube map). Rendering is implemented by the researchers via Monte Carlo integration on the hemisphere of the incident directions using CUDA on GPU. I referred to their implementation on Caffe and directly reused the rendering engine, as well as some utilities such as functions for reading or writing images in their code. Combined with different rotations, crops and lightings, nearly 130k labeled data is derived. The data is stored in .pfm HDR format, so when reading or writing via .jpg format instead, we must do the corresponding gamma correction.

Because the materials contained in the labeled dataset are too few to make plausible predictions for other materials, unlabeled images are introduced to refine the net. Using labeled data, we can generate an initial network like in Figure 5(b). Now given an unlabeled image, a rough prediction can be generated by the network. It's the key that we know how to render a new image with BRDF labels, so it means that we can get an exact labeled data using an unlabeled one. The produced data is then feed into the network for training along with labeled ones. In this way, the searching space is expanded and overfitting is somewhat prevented.

To solve the problems of ambiguity between albedo and lighting, the average value of diffuse albedo is assumed to be 0.5 and training data is normalized before being fed into the net. Also, the log of specular and roughness is used instead of their real value for numerical stability.

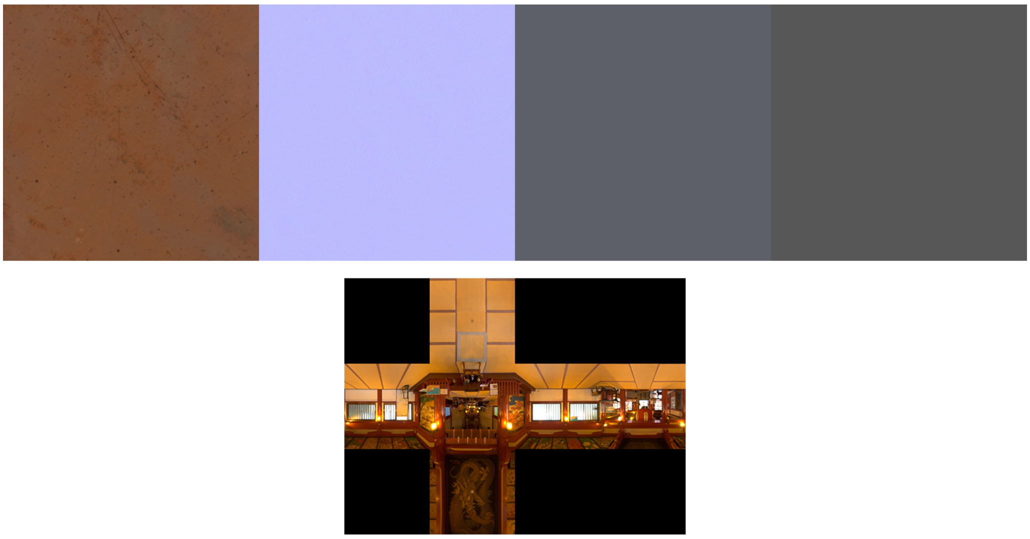

Figure 4: Training data example. Above: BRDF labels are diffuse, normal, specular and roughness maps. Below: cube map provides illumination.

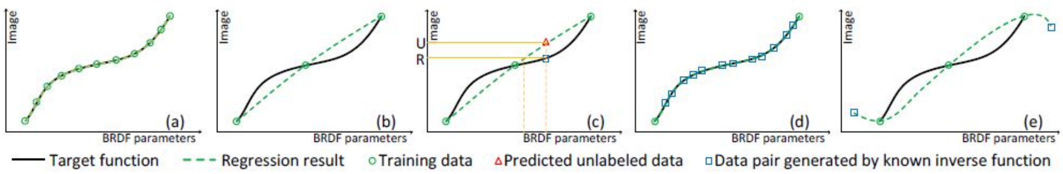

Figure 5: A schematic illustration to explain the self-augmentation.

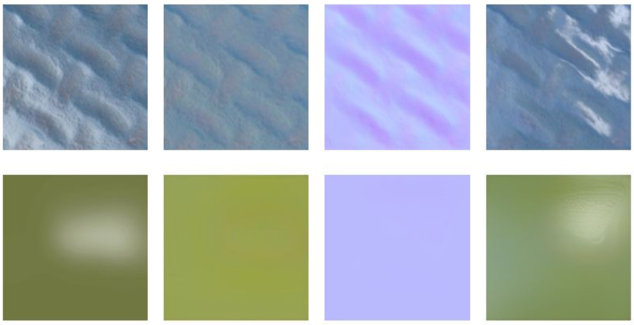

Figure 6: The input image, predicted diffuse albedo, normal and rerendered image under novel lighting conditions. Note that the rerendered surface is a bit blue because of the environment map.

4 Result and Discussion

The training is finished after 300k iterations, each of which contains a batch of 16 training data pairs. I used a Tesla K80 in our laboratory to do the training. The computation on the GPU itself is fast with the aid of powerful GPU, but because the GPU cluster is busy and I/O becomes the bottleneck. It still took a few days to finish the training.

Now we show some experiment results. The dataset is divided to three subsets: plastic, metal and wood. I trained the net with the plastic data. The figure shows the predicted diffuse albedo and normal map. Specular albedo and roughness are monochromatic and not shown here. We also use the predicted BRDF to rerender a image illuminated by hybrid light source consisting of environmental light map and directional light. Because of the difference between original and novel lighting, the input image and rendered one cannot be same. Note that the environmental lighting map we used here is a little blue and so are the rerendered images. If the light intensity of the input image is too bright or dark, the result will seem unnatural, as shown in figure 7.

The researchers of the paper also show the strong effect of self-augmentation. I didn't do this experiment because training a model costs too much time. I refer to their results in the paper here. The SA-SVBRDF model estimates more plausible and shows less artifacts than the regular SVBRDF trained with limited labeled data.

Problems may occur when doing cross estimation, i.e., using a network trained with plastic data to estimate images of metal. Some strange artifacts appeared in the predicted albedo map, and the rendered image failed to show the glossy property of metal. It suggests the different features between

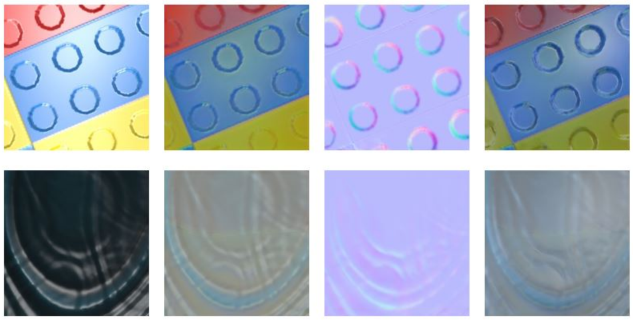

Figure 7: When the input image is either too dark or too bright, the synthesized result will look quite different.

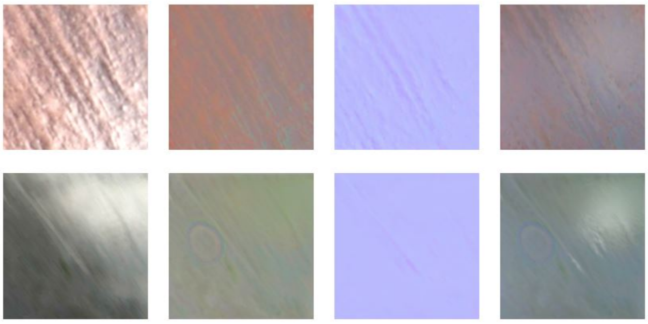

Figure 8: Artifacts of cross estimation. When using our plastic model to estimate a metal surface, some unexpected artifacts may occur. And the metal loses its glossy reflection and have a visual texture of plastic instead.

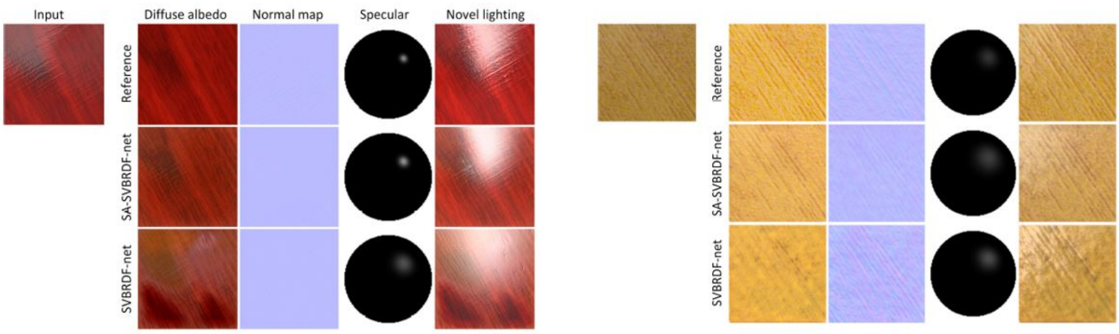

Figure 9: This figure is from the paper. Self-augmentation improves the quality of predicted BRDF and reduces artifacts.

materials. When estimating BRDFs of a different kind of material, a new model must be trained. Similarly, if the input image does not contain rich information to predict BRDF, for example, overexposure or underexposure just as in figure 7, the prediction will deviate far from ground truth. These are all the limitations of this work. Finally, the material is constrained to be planar, which is not real in many situations, and they concluded generalizing the proposed SVBRDF-Net to non-planar samples as their future work.

References

[1] Xiao Li, Yue Dong, Pieter Peers, Xin Tong. Modeling Surface Appearance from a Single Photograph using Self-augmented Convolutional Neural Networks. ACM Trans. Grp. 36, 4 (July 2017), 45:1-45:11

[2] Pharr M, Jakob W, Humphreys G. Physically based rendering: From theory to implementation[M]. Morgan Kaufmann, 2016.

[3] Tensorflow docs

[4] A blog about batch-normalizaion