BRDF reconstruction using MERL database

YUAN MENG, Carnegie Mellon University, USA

In the final project, I explored the method of reconstructing material using linear combination of MERL database.

CCS Concepts: • **Computing methodologies → Reflectance modeling**.

1 ABOUT FINAL PROJECT

This is the final project report of CMU 15-663 Computational Photography course. I've (re)implemented below features:

- Reconstruction of isotropic material using linear combination of MERL database
- Optimization of sample directions

2 INTRODUCTION

There are a lot of ways to acquire the BRDF real-life materials. This final project focuses on using limited samples from the material and reconstruct it assuming it could be expressed by a linear combination of all the materials in MERL database. There are two main tasks. The first is coming up with a way to compute the linear combination. The second one is finding the good sample directions as we don't have the resources to evaluate every ω_i, ω_o direction pair. Experiments are done comparing the results of these methods.

3 PRELIMINARIES

3.1 BRDF

The Bidirectional Reflectance Distribution Function (BRDF) is the abstraction to describe light reflection at a surface. Ignoring the spacial difference of a material, one can express the BRDF as $f_r(\omega_i, \omega_o)$. Given any incident and outgoing direction on a surface, we will use the BRDF value to describe the material.

3.2 MERL Database and Material Isotropy

There are lots of analytic BRDFs, but analytic BRDF can not express various materials in real-life. People have come up with many ways to acquire the BRDF of real-life material and to use it in the renderer. One can indeed get a very accurate real-life material BRDF if he takes enough sample, covering every degree of the ω_i, ω_o pair, which has four parameter $\theta_i, \phi_i, \theta_o, \phi_o$. Even taking the samples at the resolution of 1° will require a total of $90^2 * 360^2 = 1049760000$ samples and this task is impossible to do in real life. However, thanks to the isotropy of many materials, we have a way[5] to reduce the parameter dimension to 3, and reduce the domain of ϕ to 180°. The Rusinkiewicz coordinate system uses $\theta_{half}, \theta_{diff}, \phi_{diff}$ to express isotropic BRDF where θ_{half} is the θ value of half vector $\omega_{half} = (\omega_i + \omega_o)/2$ and $\theta_{diff}, \phi_{diff}$ is the difference vector expressing the location of ω_i as if ω_h is the Z-axis in the coordinate system. Under Rusinkiewicz coordinate system, the MERL database evaluate 100 materials with a resolution of 1°(This is not accurate as θ_h of MERL is not uniformly distributed). It requires $90 * 90 * 180 = 1458000$ samples.

Author's address: Yuan Meng, yuanm2@andrew.cmu.edu, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh, Pennsylvania, USA, 15213.

3.3 Principal Components of MERL database

Once we have all the data in MERL database, we can extract the principal components of the material and focusing on combine the materials in MERL to create new material or fit a new material to the combination of MERL materials. Note there is an assumption for this method: the material should fall in the subspace of all MERL material or it will never recover the new material. Also, direct PCA on the raw BRDF data has very poor performance due to the high dynamic range of BRDF data.

4 IMPLEMENTATION

4.1 Data and Mapping

A popular way of dealing with high dynamic range is taking the log of the data. So Nieson[4] suggest using a log mapping

$$ln(\frac{\rho cos_{weight} + \epsilon}{\rho_{ref} cos_{weight} + \epsilon})$$

. This mapping reduce the dynamic range and make the data more normally distributed with $\rho_{ref} = median(materials)$ Using this mapping we can get an observation matrix $X \in R^{300\times1458000}$. Using the mean substracted matrix of X we can use singular alue decomposition to get the principal components.

$$(X - \tilde{\mu}) = U\Sigma V^T$$

Then we can get a matrix $Q \in R^{1458000*300}$ of scaled principal components.

$$Q = V\sigma$$

4.1.1 *Invalid BRDF Space.*

In the full $\theta_h, \theta_d, \phi_d$ space. There are some directions(about 300000) that are below the positive hemisphere. During any computation, we should exclude those values and directions.

4.2 Reconstruction

Given limited samples on the new material, we can only acquire few lines/rows of the observation matrix $\tilde{X} - \tilde{\mu}$. We want to use these observation to get a linear combination c, where

$$(\tilde{X} - \tilde{\mu} = \tilde{Q}c)$$

$$c = argmin||(\tilde{X} - \tilde{\mu}) - \tilde{Q}c||^2$$

This is a least square solution, but it suffers from over fitting problem. So a regularization term is added to the optimization problem

$$c = argmin||(\tilde{X} - \tilde{\mu}) - \tilde{Q}c||^2 + \eta||c||^2$$

with a closed form solution

$$c = (\tilde{Q}^T\tilde{Q} + \eta I)^{-1}\tilde{Q}^T(\tilde{X} - \tilde{\mu})$$

Note that Q and μ only depends on the MERL database data, so we can compute them one time and load them from disk after that. $\tilde{Q}$ is simply selecting some rows corresponding to the observation direction in Q

4.3 Finding Optimal Directions

In total, we have $90 * 90 * 180 = 1458000$ directions in MERL database. It's possible that choosing some of them is enough to generate good results. The idea is evaluating the error of selecting a few rows of the $\tilde{Q}$ matrix and try to optimize that error.

4.3.1 Using condition Number. The condition number of matrix $\tilde{Q}$ can be a metric to evaluate the noise and numerical error, minimizing this using gradient descent could give us the optimal sampling directions. However, the invalid space in the full BRDF space and the fact that we can only move the parameter in the resolution of 1 prevents us from using general optimization solver. I need to follow the algorithm described in the paper to optimize the condition number

4.3.2 Using New error metrics. In the following paper[6], there is a new metric that did better than condition number. This error metric is more comprehensive.

4.4 Using more samples

Choosing the sampling direction require us to use equipment like gonioreflectometer. If someone just take an image of the material, he can still reconstruct the material. At the same time, all pixels in the image could be used as samples. Thus we will have more samples than just generating points, this could possibly improve the accuracy of reconstruction. The idea of [6] is optimizing two camera direction/light direction pair and the use all the samples in those two images to do the reconstruction. To implement this, I need to compute the fov, the image size and use the near-field assumption to compute the corresponding direction for each pixel. Then add all of these valid sample to $\tilde{Q}$

5 RESULTS AND EVALUATION

This section includes the results of my experiment and the analysis. All rendered images use Mitsuba renderer[3] with bidirectional path tracer and 2048spp. The MERL plugin for Mitsuba is adapted from[1]

5.1 Finding Optimized direction

I did the experiment on finding ten optimized directions using condition number. The direction I got are:

θ_h	θ_d	ϕ_d
4.01	14	16
69.3	17	151
8.1	82	48
2.84	89	15
17.77	41	92
0.9	3	33
1.87	69	145
26.67	71	56
0.01	89	59
1.87	50	70

These directions have a lower condition number than the optimal direction given by the paper, and one can find the similarities between these two directions set, especially the θ_h and θ_d column

θ_h	θ_d	ϕ_d
3	12	28
63	19	89
5	77	77
2	60	180
15	4	130
1	6	37
2	79	110
39	76	89
0	71	104
5	75	180

For getting the camera direction, the error metrics computation, even after vectorization, is extremely slow. Each iteration will take 20 seconds and there are countless iteration during gradient descent and finding a good initial guess. Also it takes more than 32GB of memory and swap file which will cause the program getting killed by system after running for 2 to 3 hours. So it is impossible to run this process locally using my implementation. For the following experiment, I just use the proposed optimal direction for 25° fov in the paper[6].

θ_h	θ_d	ϕ_d
6	23	41
16	79	87

5.2 Reconstruction on three merl material

For the reconstruction, I first test my implemntation on three MERL materials

- green-plastic
- alum-bronze
- light-red-paint

5.2.1 green-plastic. This is a plastic like material which will have *random* specular highlights. The results are

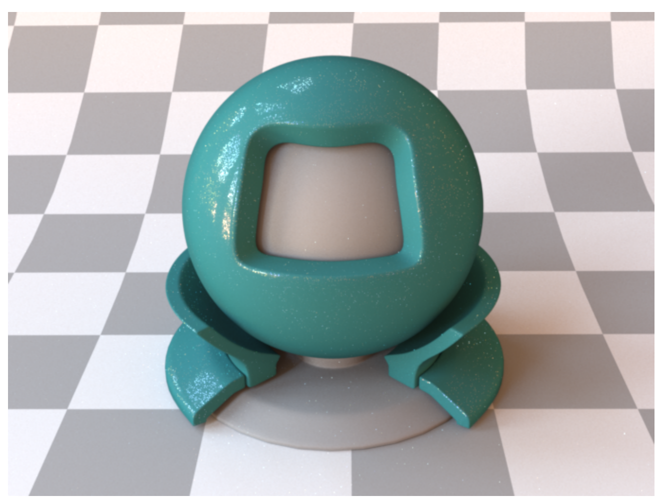

Fig. 1. Green plastic ground truth

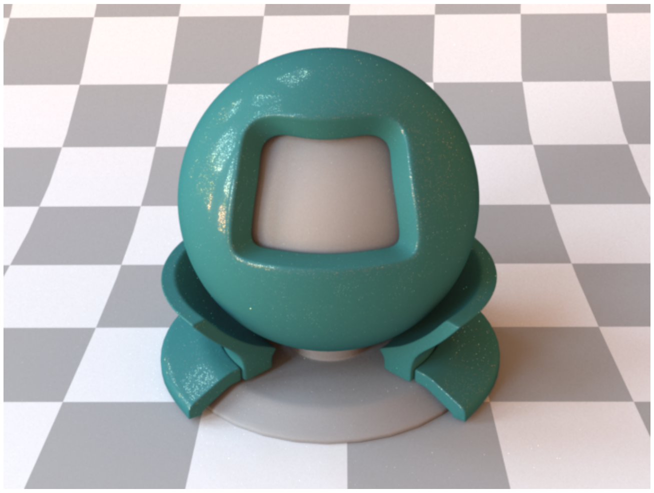

Fig. 2. Green plastic self-generated 10 points

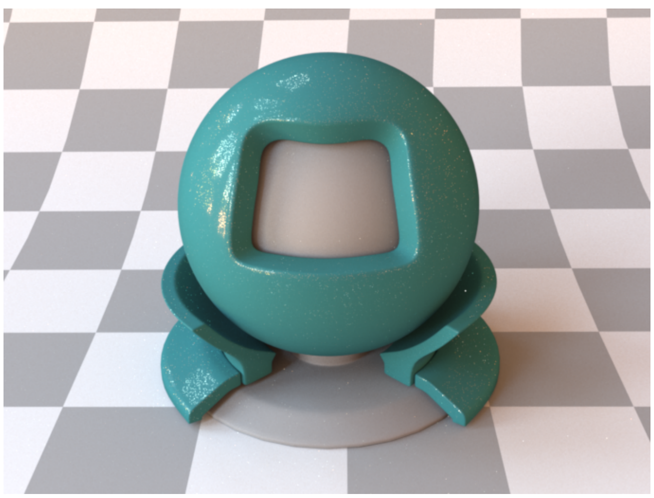

Fig. 3. Green plastic reference two shots

From the rendered images, the self-generated 10 points look darker than ground truth. Also the locations of the white plastic specular highlight is kind of random

5.2.2 alum-bronze. This is a metal like material with light bronze color. The result of self-generated 10 points and reference two shot near field are not the same as the ground truth. The outer rim of two shots are lighter. And the self-generated 10 points is more like a darker brown color than the bronze color. I tried using the reference ten points directions to reconstruct the material and it looked better, but it still has a darker brown color at the outer rim. So it looks like none of these methods faithfully reconstruct the alum-bronze material in MERL database, with the reference 10 points perform relativelty better.

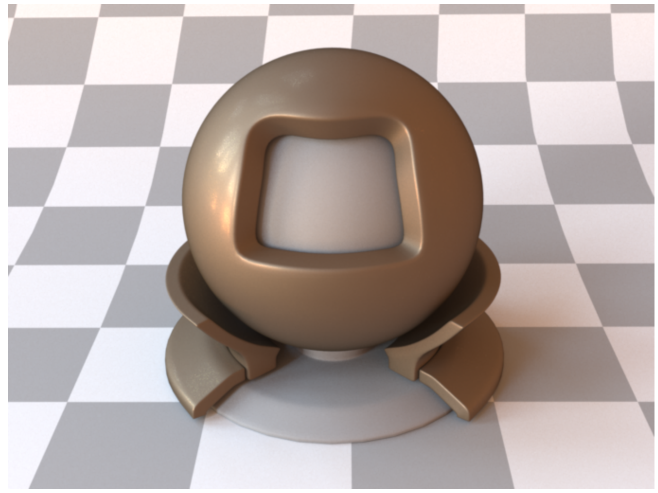

Fig. 4. alum bronze ground truth

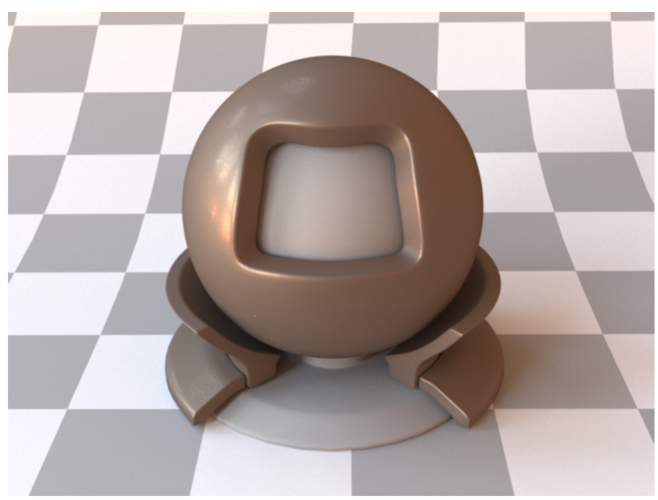

Fig. 5. alum bronze self-generated 10 points

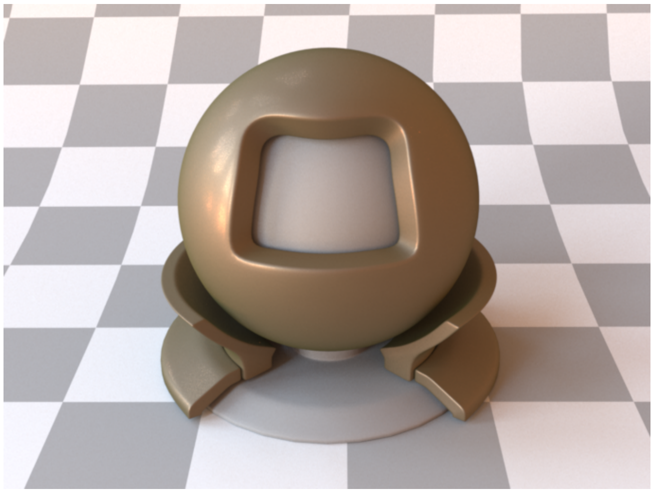

Fig. 6. alum bronze reference two shot

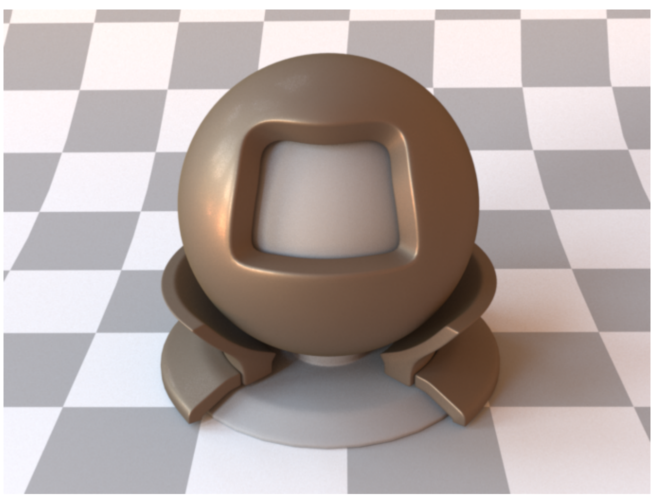

Fig. 7. alum bronze reference 10 points

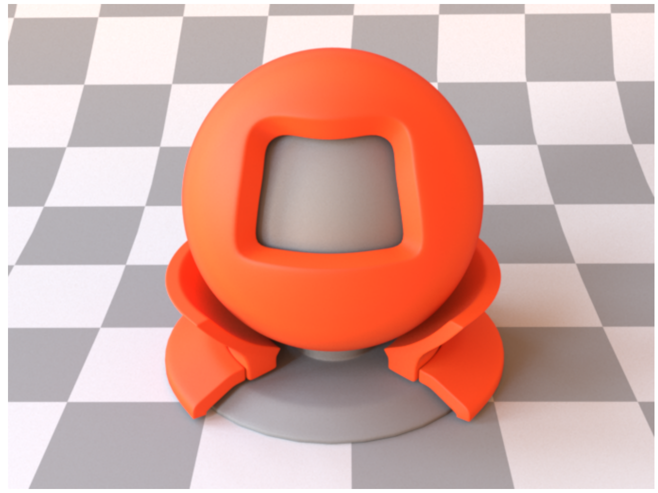

Fig. 8. light red ground truth

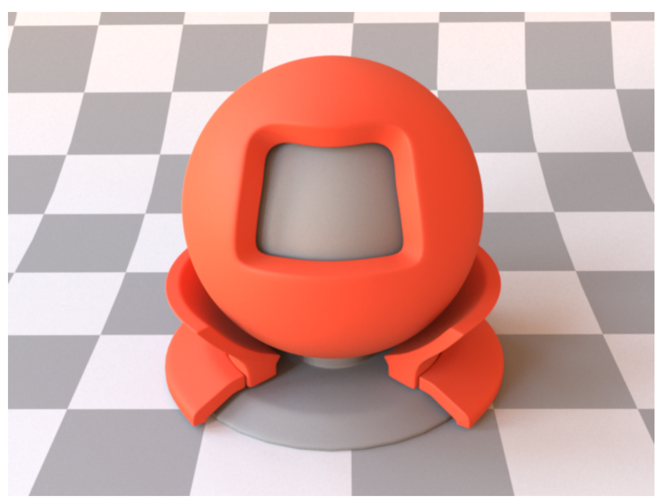

Fig. 9. light red self-generated 10 points

5.2.3 light red paint. This is a very light paint material, I would say it's of orange color rather than red. The reconstruction results of two shots and reference 10 points directions look close to ground truth. But the self-generated 10 points(which has a lower condition number and should perform better theoretically) failed to generate the orange color. It looks much more reddish(or I could say it's darker). The reference 10 points also has the problem of being darker. The two shot image reconstruction wins in this comparison

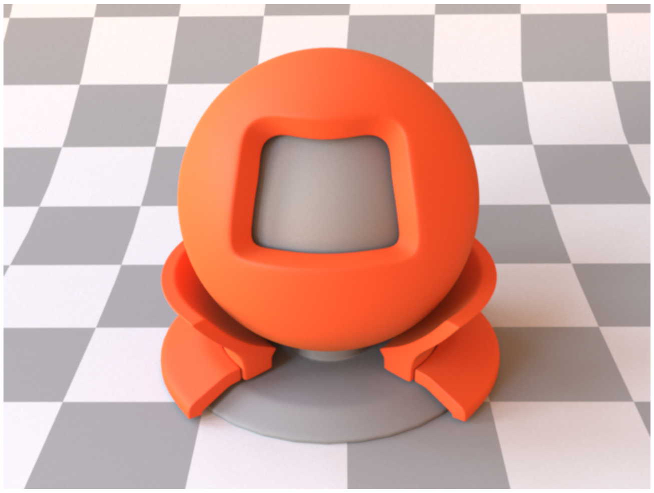

Fig. 10. light red reference 10 points

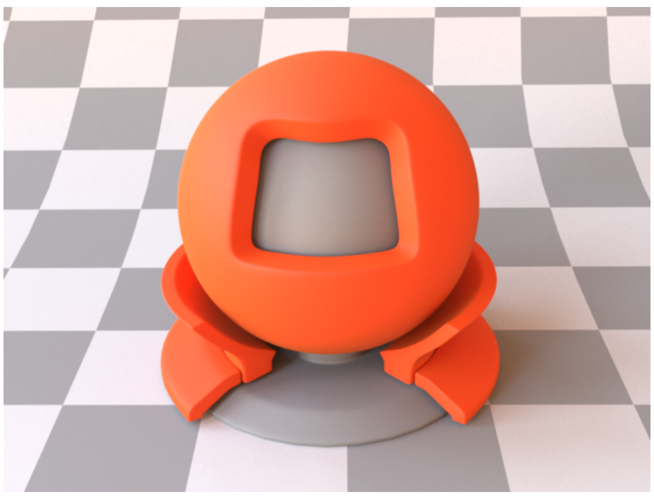

Fig. 11. light red reference two shots

5.2.4 Analysis of RMSE. However, if I compute the root mean square error:

material	self-generated 10p	ref 10p	ref two shot
light red paint	0.398	0.710	0.732
alum bronze	32.26	68.77	52.89

The self-generated 10 points result always have the lowest RMSE, but it never looks close to the ground truth(except for the green-plastic material) in actual rendered image. This is really strange and I still havn't figured out why this happened.

5.3 General problem with the reconstruction

It you take a closer look at the rendered images, you will find that all of these reconstructed material failed to render the black ring like effect at grazing angles. This is quite obvious for the green plastic material. Also, the self-generated 10 points one always have a darker color than the ground truth.

5.4 More samples

I also wonder what happened if I put more sample into this reconstruction pipeline. There is a obvious paradox in the condition number theory. If the number of samples(directions) increases, the condition number will almost surely get larger. And a larger condition number means more noise and error. To test this, I just use my McMC rountine to generate around 8000 points and feed them into the pipeline. The number of sample is lower than the two shots cases(typically it will have more than 25000 valid samples in the two shot setup). And they have one thing common: the sample they generated are correlated. The two shots sample are spatially correlated as moving one pixel is just moving a little bit of ω_i. The McMC sample is still spatially correlated(but with some probability of taking random global step, which is 30 percent in my setup). The result of alum-bronze is below. This is the closest result to the ground truth. So it looks like if we have more samples, we should just use all of them.

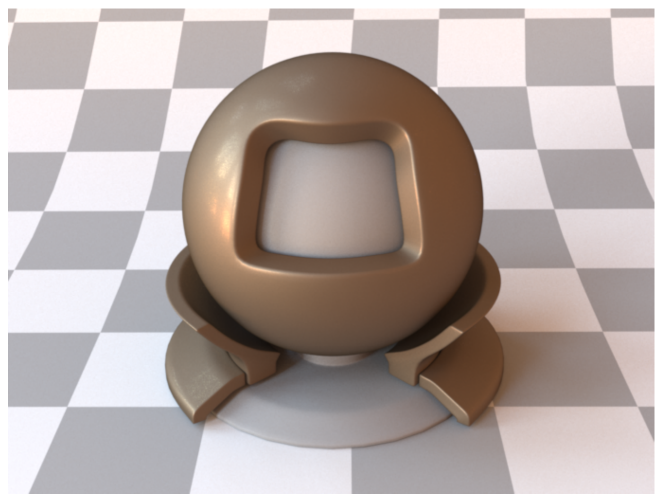

Fig. 12. alum bronze mc 8k points

5.5 Reconstruction on other material

For this part I only tested one purple material from the adaptive material database[2] If there is no coordinate system mismatch due to different convention of phi during conversion from io space to hd space, this method can not reproduce the purple material, the images looks much brighter than ground truth.

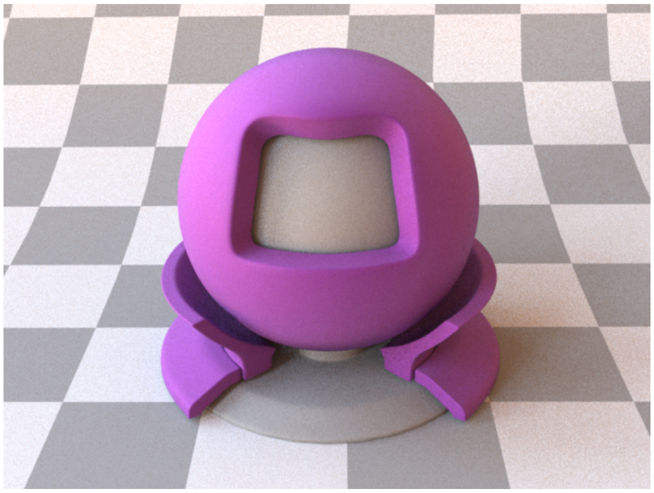

Fig. 13. Purple ground truth

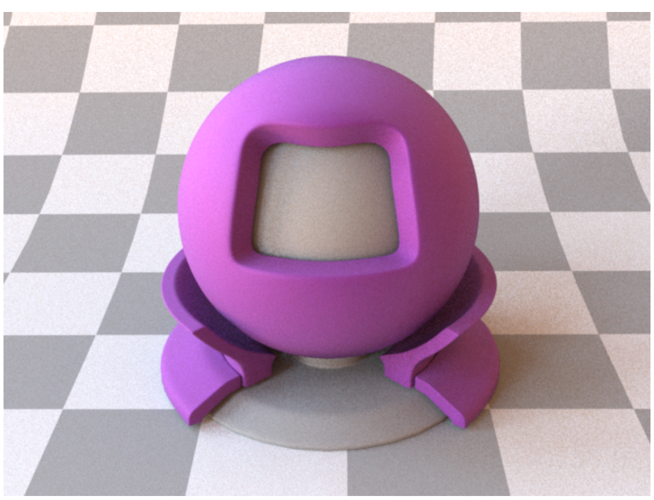

Fig. 14. purple mcmc

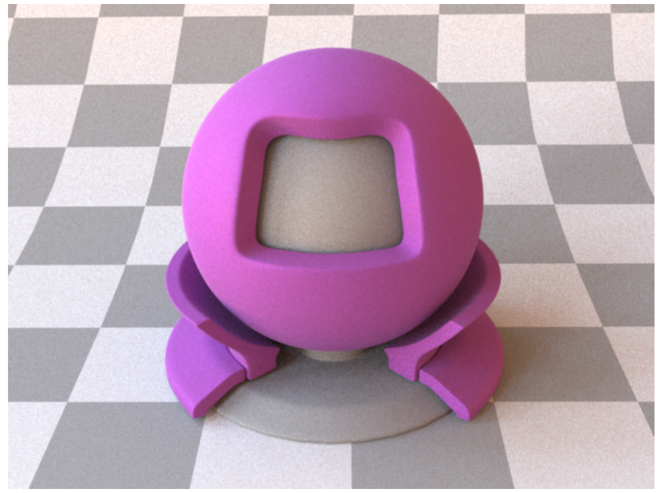

Fig. 15. purple ref 10 points

5.6 Error Metric

Although I can not use the new error metric to compute the new direction. I did evaluate the self-generated 10 points with the reference 10 points. The self-generated 10 points have a higher error metric(280) than the ref 10 points(240). This could be another evidence that condition number is not a good way to predict the reconstruction quality for some sample sets.

6 PROBLEMS AND POSSIBLE IMPROVEMENTS

I've addressed most problems in the previous section, this section is to summarize

6.1 Grazing angles

This method sometimes ignore the darker color of ground truth MERL material at grazing angles.

6.2 A faster but better error metric

In optimizing the directions, I can not run the error metric locally with a laptop of 32GB memory. It would take more than 10 hours to optimize 10 point direction.

6.3 Putting more materials into the BRDF space

If we have more materials other than MERL, we can put all of them into the material space. Thus the reconstruction could be more robust as there is more liklihood for a new material to lie in the subspace of all known material.

6.4 Low RMSE Wrong Rendered Image

Reconstruction having the lowest RMSE is the most different from the ground truth during the rendering comparison.

REFERENCES

[1] Jonathan Dupuy. 2016. DJ BRDF. https://github.com/jdupuy/dj_brdf.
[2] Jonathan Dupuy and Wenzel Jakob. 2018. An Adaptive Parameterization for Efficient Material Acquisition and Rendering. *ACM Trans. Graph.* 37, 6, Article 274 (dec 2018), 14 pages. https://doi.org/10.1145/3272127.3275059

[3] Wenzel Jakob. 2010. Mitsuba renderer. http://www.mitsuba-renderer.org.
[4] Jannik Boll Nielsen, Henrik Wann Jensen, and Ravi Ramamoorthi. 2015. On Optimal, Minimal BRDF Sampling for Reflectance Acquisition. *ACM Trans. Graph.* 34, 6, Article 186 (nov 2015), 11 pages. https://doi.org/10.1145/2816795.2818085
[5] Szymon M. Rusinkiewicz. 1998. A New Change of Variables for Efficient BRDF Representation. In *Rendering Techniques '98*, George Drettakis and Nelson Max (Eds.). Springer Vienna, Vienna, 11–22.
[6] Zexiang Xu, Jannik Boll Nielsen, Jiyang Yu, Henrik Wann Jensen, and Ravi Ramamoorthi. 2016. Minimal BRDF Sampling for Two-Shot near-Field Reflectance Acquisition. *ACM Trans. Graph.* 35, 6, Article 188 (dec 2016), 12 pages. https://doi.org/10.1145/2980179.2982396