

# Homework 1: Photometric Stereo

Suiliang Mai, 22521175

## 1 Problem Setting

For each object, given 16-bit integer PNG images with resolution of 612x512 from 96 different lighting directions, light directions and light intensities, our goal is to recover the normal map and albedo map of the specific fixed viewpoint.

## 2 Method

The reflectance models is assumed to be the Lambert's model, with the following simplified render equation,

$$I_C = L \cdot \rho_C \cdot (\mathbf{n} \cdot \mathbf{l}), \quad C = \{R, G, B\} \quad (1)$$

where  $L$  is the known light intensity,  $\mathbf{l}$  is the known light direction,  $\rho_C$  is the unknown surface albedo and  $\mathbf{n}$  is the unknown normal.

Since we have  $N = 96 > 3$  images from different light directions and light intensities, the problem becomes an optimization problem. For each pixel  $\mathbf{p}$  in each channel of the  $i$ -th captured image, we suppose  $\mathbf{l}_i = (l_{i,0}, l_{i,1}, l_{i,2})$  the light direction,  $I_i$  the reflectance,  $L_i$  the light intensity. Typically, photometric stereo utilizes least squares method to minimize

$$\|A\mathbf{x} - \mathbf{b}\| \quad (2)$$

where

$$A_{N \times 3} = \begin{pmatrix} l_{0,0} & l_{0,1} & l_{0,2} \\ \vdots & \vdots & \vdots \\ l_{N-1,0} & l_{N-1,1} & l_{N-1,2} \end{pmatrix} \quad (3)$$

and

$$\mathbf{b}_{N \times 1} = \left( \frac{I_0}{L_0}, \dots, \frac{I_{N-1}}{L_{N-1}} \right)^\top \quad (4)$$

The result is a  $3 \times 1$  vector  $\mathbf{x}$ . Since  $x = \rho \cdot \mathbf{n}$ , the albedo  $\rho$  and normal  $\mathbf{n}$  can be calculated by

$$\mathbf{n} = \frac{\mathbf{x}}{\|\mathbf{x}\|}, \quad \rho = \|\mathbf{x}\| \quad (5)$$

Actually, there are three channel RGB for each image. Practically, we solve each channel respectively with the results  $\mathbf{x}_R$ ,  $\mathbf{x}_G$  and  $\mathbf{x}_B$ , and then utilizes the following method to aggregate the final result,

$$\mathbf{n} = \frac{\mathbf{x}_R + \mathbf{x}_G + \mathbf{x}_B}{\|\mathbf{x}_R + \mathbf{x}_G + \mathbf{x}_B\|}, \quad \rho = (\|\mathbf{x}_R\|, \|\mathbf{x}_G\|, \|\mathbf{x}_B\|)^\top \quad (6)$$

## 2.1 outlier

Shadows due to geometry visibility and highlights due to complex material model can be observed from the captured images, which are not considered in our simple Lambert's model. In other words, they are the outliers. A simple solution is to discard certain percentage of the darkest and brightest pixels to get rid of shadow and highlight respective. We set both the darkest and brightest ratio 0.2 in our experiments.

## 3 Experiments

Our method is tested in four different objects in the real capture scene. The qualitative results are shown in fig. 1 and the quantitative results are shown in table 1. The error maps calculate the differences between the re-render images and the ground-truth images. Large errors mainly appear in the specular part of the objects, which demonstrates close approximations of our methods.

We further show the difference of albedo recovery for the original version and the robust version that discards the outliers in fig. 2. The results show that our robust version successfully reduce the specular part of the object as expected.

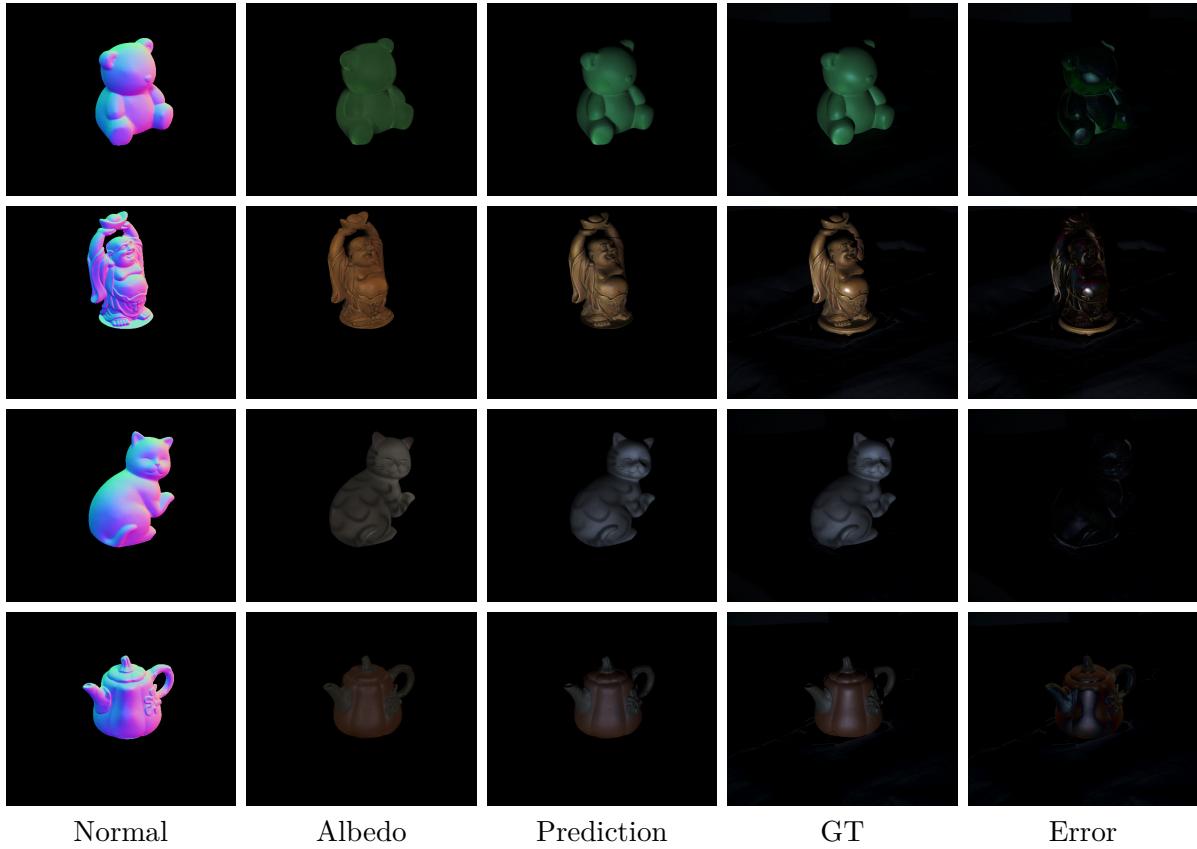


Figure 1: Qualitative results of photometric stereo. We increase the exposure by 2.0 in **bear**, **buddha** and **cat** object, increase the exposure by 3.0 in **pot** object for better visualization.

## 4 Discussion

### 4.1 Acceleration of the Robust Version

To discard the darkest and brightest part for better robustness, a general idea is to discard for each pixel with a loop. Each time, the loop fetches the images, indexes the pixel, sorts the

	PSNR	SSIM
bear	33.77	0.7353
buddha	32.20	0.5422
cat	34.73	0.7836
pot	43.20	0.7770

Table 1: Quantitative results of photometric stereo. We use PSNR and SSIM as metrics to evaluate the re-rendered results.

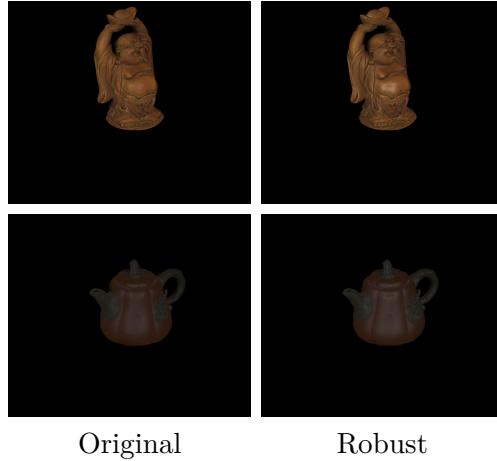


Figure 2: Comparision of the original version and the robust version. The robust version discards the outliers that are too dark or too bright. We increase the exposure by 2.0 in `buddha` object and increase the exposure by 3.0 in `pot` object for better visualization.

data, discards the outliers and finally solves the optimization problem by least squares, which drastically slows down the running time from 3 second to 10 second using `Matlab` in my PC.

To accelerate this process, we can introduce a mask for each pixel to mask out the outliers. The mask is pre-computed before the pixel loop by a general sorting and a filtering by the darkest and brightest ratio. Code details can be found in `myPMS_robust.m` and `myPMS_robust_accelerate.m`.

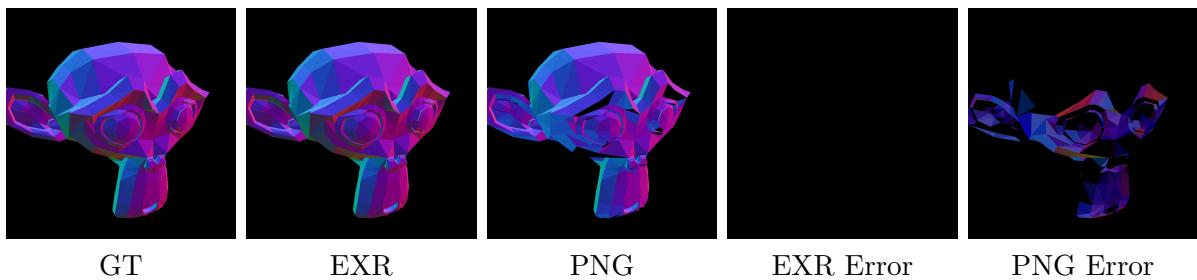


Figure 3: Comparisons of the influences of the light directions. `EXR` means we use the lossless OpenEXR images which store all information including the clipped negative values, while `PNG` means lossy PNG images are used which are clipped with negative values.

## 4.2 Influence of the Choice of the Light Directions

On one hand, if we visualize the light directions of the provided data, we can get the following result, which is a uniform distribution in a hemisphere of the positive z-axis.

On the other hand, the normal recovery of the four objects we used shows that the normal is also in the positive z-axis direction. That means we give the normal direction  $\mathbf{n}$  is on the same side as light direction  $\mathbf{l}$ , or mathematically, we have

$$0 < \mathbf{n} \cdot \mathbf{l} < 1 \quad (7)$$

However, when the light directions are not satisfied with this condition, due to lighting is clipped in the "negative" hemisphere. Finally It causes the shadow effect as well and becomes an outlier.

We validate this consideration in a synthetic scene rendered by **Blender** and the results are shown in fig. 3. Some of the ground-truth normal of the image is in the  $-\mathbf{z}$  direction. For the EXR images, it stores all information and therefore the recovered normal is almost the same. While the lossy PNG shows poor results where the normal is in the  $-\mathbf{z}$  direction.