Multi-View 3D Object Reconstruction Using Photometric Stereo and Multi-View Light Sources

Sepehr Mousaviyan Arash Dehghani

Spring 2025

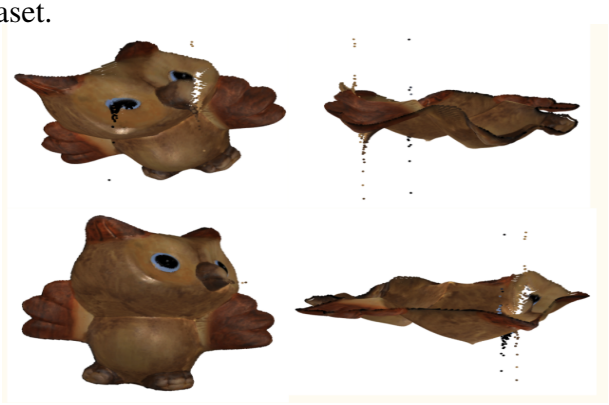
Course: “CS217A: Light and Geometry in Computer Vision”
Instructor: Dr. Charles Fowlkes

Abstract

This project develops a method for 3D object reconstruction by integrating photometric stereo and multi-view geometry, utilizing the ‘DiLiGenT-MV’ dataset. Surface normals estimated from multiple light sources are used to reconstruct 3D surfaces for each view, which are combined to create accurate 3D models. Extending prior photometric stereo techniques from the class works.

1 Introduction

Extending our prior work in photometric stereo, where we estimated surface normals from a single view using multiple light sources, as explored in the third homework assignment with the “Owls” dataset.



Initially, we attempted to create our own dataset by purchasing a chrome sphere from Amazon to calibrate lighting directions, aiming to replicate the “Owls” data collection process.

However, we realized that accurate 3D reconstruction requires the camera's intrinsic and extrinsic parameters for the matching view porpuse, which were challenging to obtain with our setup. Subsequently, We decided to use the 'DiLiGenT-MV' dataset [DiLiGenT-MV Team, 2018], a comprehensive multi-view photometric stereo dataset. It includes images of five objects with complex bidirectional reflectance distribution functions (BRDFs), captured from 20 views, each illuminated by 96 calibrated point light sources, along with calibration data. This project leverages the 'DiLiGenT-MV' dataset to integrate single-view 3D estimates, derived from surface normals using two distinct methods, across multiple perspectives to construct accurate 3D object models.

2 Theory for extracting 3D Surface from Normals

Reconstructing a 3D surface from surface normals, obtained via photometric stereo, involves estimating the depth or height map of the object's surface. Surface normals, represented as vectors $\mathbf{n} = (n_x, n_y, n_z)$, describe the orientation of the surface at each point. The goal is to recover the depth function $z(x, y)$ such that the surface's gradient matches the normal field. Mathematically, the surface normal is related to the depth by:

$$\mathbf{n} = \frac{\left(-\frac{\partial z}{\partial x}, -\frac{\partial z}{\partial y}, 1\right)}{\sqrt{\left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2 + 1}}. \tag{1}$$

Equivalently, the components of the normal field satisfy:

$$\frac{\partial z}{\partial x} = -\frac{n_x}{n_z}, \quad \frac{\partial z}{\partial y} = -\frac{n_y}{n_z}. \tag{2}$$

Initially, our 3D reconstruction worked well with the "Owls" dataset, producing accurate surfaces. We then applied the same methods to our own dataset, created using the chrome sphere, but encountered significant issues, including distorted reconstructions. After investigation, we identified that the normal field must be integrable, requiring the mixed partial derivatives of the gradient field to be equal (i.e., the curl of the gradient field should be zero):

$$\frac{\partial}{\partial y}\left(-\frac{n_x}{n_z}\right) = \frac{\partial}{\partial x}\left(-\frac{n_y}{n_z}\right). \tag{3}$$

This condition was not satisfied in our dataset due to noise and calibration errors, causing inconsistencies. To address this, we employed two methods to reconstruct the 3D surface:

2.1 Deriving Poisson

Given a normal map $\mathbf{n}(x, y) = (n_x, n_y, n_z)$, we define:

$$p = -\frac{n_x}{n_z}, \quad q = -\frac{n_y}{n_z}$$

These represent the gradients of the surface height function $z(x, y)$:

$$\frac{\partial z}{\partial x} = p, \quad \frac{\partial z}{\partial y} = q$$

We pose this as a Poisson equation:

$$\Delta z = \frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} = \frac{\partial p}{\partial x} + \frac{\partial q}{\partial y}$$

2.2 Fourier solution

Taking the 2D Fourier Transform:

$$\mathcal{F}[\Delta z] = -4\pi^2(u^2 + v^2)\hat{z}(u, v) = \hat{f}(u, v)$$

Solving in the frequency domain:

$$\hat{z}(u, v) = \frac{-\hat{f}(u, v)}{4\pi^2(u^2 + v^2)}, \quad \text{with } \hat{z}(0, 0) = 0$$

Finally, the depth map is obtained via inverse FFT:

$$z(x, y) = \mathcal{F}^{-1}[\hat{z}(u, v)]$$

These methods are applied to each single view's normal field, and the resulting 3D estimates are combined across views using multi-view geometry to form a unified 3D model.

Note that we use DFT that is discrete. Because the world is flat we do not witness much higher frequency components in normal images. So, we can have a approximation of a discrete Fourier formula:

$$\nabla^2 f \Rightarrow (4 - 2\cos(\frac{2\pi u}{M}) - 2\cos(\frac{2\pi v}{N}))F(u,v) \tag{4}$$

where with approximation, we have:

$$\approx (4 - 2(1 - 4\pi^2 u^2/M^2) - 2(1 - 4\pi^2 v^2/N^2))F = (4\pi^2 v^2/N^2 + 4\pi^2 u^2/M^2)F \tag{5}$$

Therefore, that justifies why the method works even for discrete.

3 Methodology

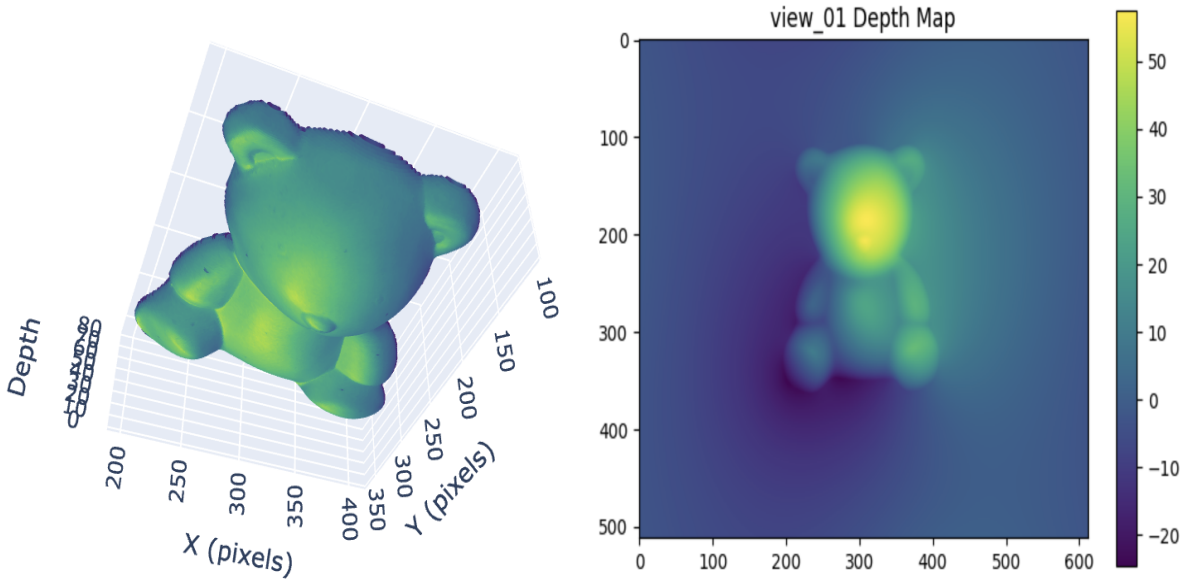
To achieve the project's goals, the following methodology will be employed:

1. **Data Preparation**: Initially, we attempted to create our own dataset, capturing images of objects and using a chrome sphere to determine light directions. This required precisely positioning objects in the same location for each capture and retaking photos with the chrome sphere to calibrate lighting, which proved challenging due to alignment and calibration issues. Consequently, we adopted the 'DiLiGenT-MV' dataset

   [DiLiGenT-MV Team, 2018], which provides images of five objects captured from 20 viewpoints, each illuminated by 96 calibrated point light sources, along with camera intrinsic and extrinsic parameters. We tested results on "Bear" photos.

2. **Photometric Stereo Processing**: Compute surface normals for each view using photometric stereo algorithms, leveraging the dataset's calibrated light sources.

3. **3D Extraction**: Apply normal integration and shape-from-normals optimization to derive 3D models from surface normals for each single view, using the Fourier method described earlier.



4. **Multi-View Integration**: Combine single-view 3D estimates using multi-view geometry techniques, leveraging the dataset's extrinsic parameters to align views into a unified 3D object model.

   We now aim to integrate depth estimation of multiple view. We use this equation, with assumption of calibrated cameras:

$$\mathbf{x}_{\text{world}} = R_i^{\top}(\mathbf{x}_i - T_i) \tag{6}$$

   This transforms the camera coordinate to the world.

   One important point is that when solving the differential equation, we did not use boundary conditions. Hence, for a point in multi-view, we have different estimations. Assume that x_1 and x_2 are coordinates of the same point from different views in camera 1 and 2 respectively. Let $x_2 = x_1 + d$

$$\mathbf{x}^1_{\text{world}} = R_1^T(x_1 - T_1) \tag{7}$$

   For camera 2 we have:

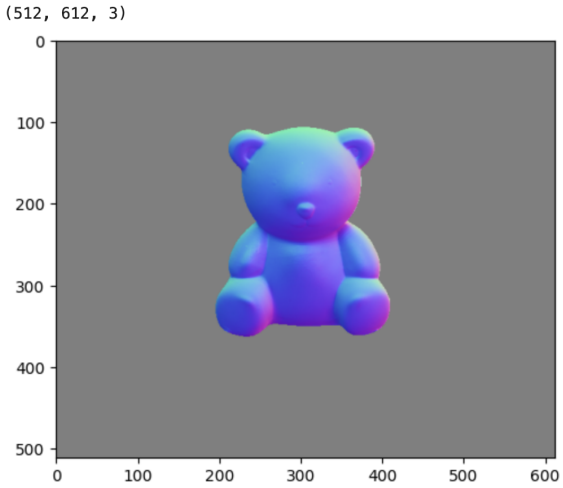
$$\mathbf{x}^2_{\text{world}} = R_2^T(x_1 + d - T_2) = R_2^T x_1 + \tilde{d} \tag{8}$$

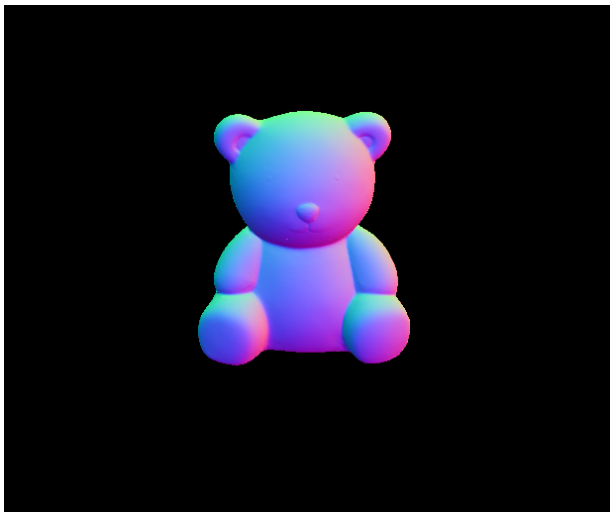
   which means we can match the point by computing just an offset. After rotating back the views, only the translation terms(all offset in calculation) are left to integrate multiple views. So, all points differ by :

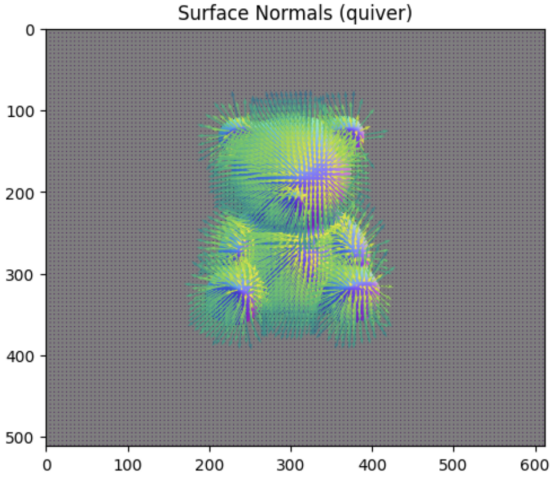
$$\tilde{d} = R_1^T x_1 - R_2^T x_2 \tag{9}$$

   Since equation $\mathbf{x}^1_{\text{world}} = \mathbf{x}^2_{\text{world}}$ must hold.

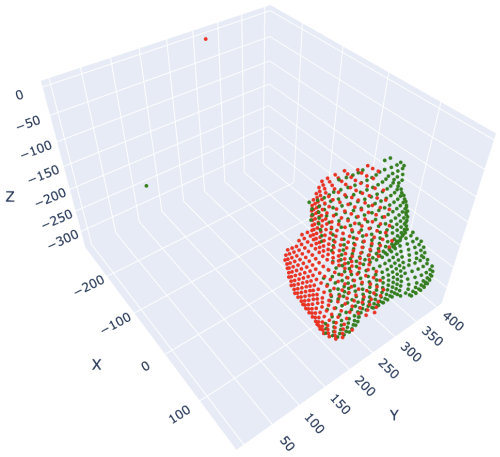
Example of surface normal finding(3D surface reconstruction provided on section 2) which the left is the estimated normal with showing the x, y ,z as R, G, B channels and right is the ground truth which fits perfectly.

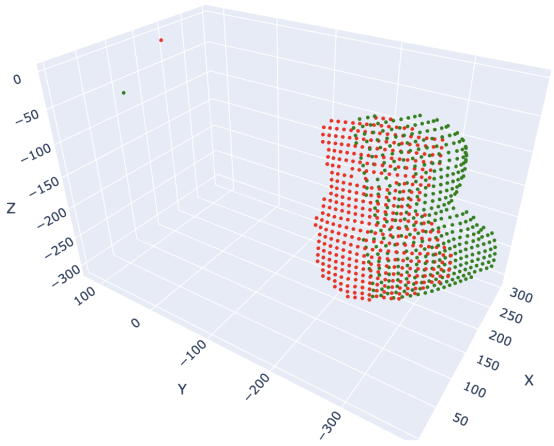


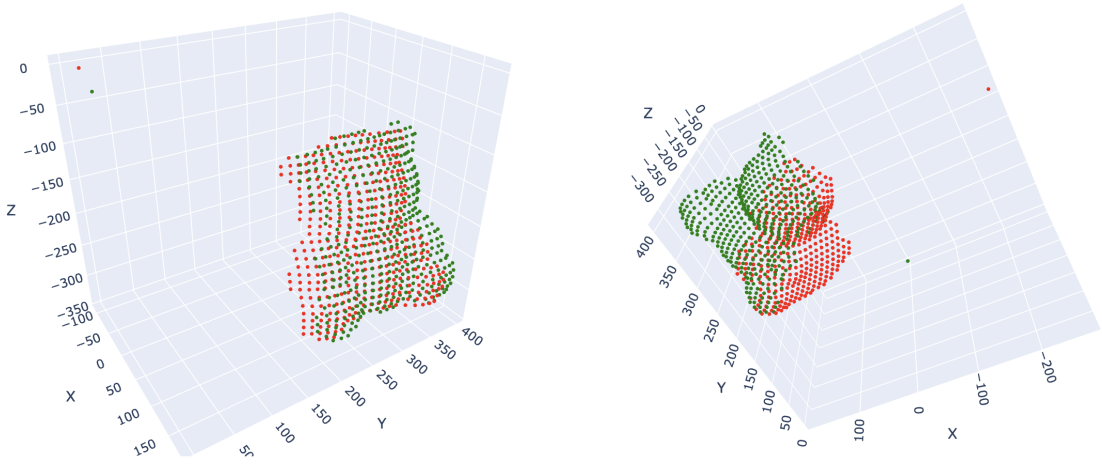




Here are some examples matching the points using two different view(Also in the code with details of which view's has been used.







Code (.Ipynb file) is also provided in the zip file containing this document.

4 Results and summary

We developed a framework for combining multiple views of an object to estimate its 3D structure. Each stage of the pipeline is explained in detail, with corresponding results presented both in this document and within the code.

5 Discussion

In the feature matching stage, traditional methods like SIFT often perform poorly on 3D data, especially near surface edges and in regions with nearly symmetrical structures. To mitigate these issues, incorporating a robust estimation method like RANSAC significantly improves the reliability of feature correspondences. RANSAC helps by filtering out outliers and enforcing geometric constraints, such as the epipolar relationship.

Additionally, incorporating more views of the object may introduce time efficiency challenges, even though the current two-view setup completes in under a minute. This suggests a potential need for parallelization or the use of more computationally efficient methods to maintain scalability as the number of views increases.

6 Contributions

As a group, we collaboratively tried to overcome the bugs happening during the process. Some tasks, such as dataset creation, were done together. For the 3D view reconstruction, Arash worked on the theoretical aspects, while Sepehr implemented the reconstruction from surface normals and depth estimation. In the multi-view stage, Arash handled most of the work, including finding matching points and merging the 3D data across views.

References

DiLiGenT-MV Team. DiLiGenT-MV: Multi-View Photometric Stereo Dataset. https://sites.google.com/site/photometricstereodata/mv, 2018.