

Development and Evaluation of 3D Reconstruction Framework for General Objects

by

Kai Wu

Bachelor of Engineering, Beijing University of Posts and Telecommunications

2014

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

Master of Applied Science

in

THE FACULTY OF APPLIED SCIENCE
(Department of Electrical and Computer Engineering)

The University of British Columbia
(Vancouver)

April 2017

© Kai Wu, 2017

Abstract

Advancements in state-of-the-art 3D reconstruction algorithms have sped ahead of the development of interfaces that improve the algorithms' accessibility for application developers. In light of this discrepancy, we propose a novel abstraction framework specifically for shape capture techniques, designed to allow developers to create their own camera setups and apply the most effective method to retrieve an object's shape and appearance for their particular application.

We see that details of 3D reconstruction algorithms can be hidden by an abstraction, that uses a description based on camera setup and the characteristics of the object. We demonstrate that the description can be mapped to one or more methods to provide the user's requested result. Given that no single algorithm can work in all situations, we select a suite of three algorithms, each working in substantially different conditions, to provide an example of how the space of 3D reconstruction can be filled.

We evaluate our abstraction through a proof-of-concept implementation of the algorithm framework and a synthetic dataset where each object has been imaged with the appropriate setup for each algorithm. We demonstrate that the mapping from object characteristics to 3D shape is effective, and provides an illustration of an accessible form of 3D shape reconstruction for non-experts in computer vision, such as application developers.

Preface

At **UBC!** (**UBC!**), a preface may be required. Be sure to check the **GPS!** (**GPS!**) guidelines as they may have specific content to be included.

Table of Contents

Abstract	ii
Preface	iii
Table of Contents	iv
List of Tables	viii
List of Figures	x
List of Acronyms	xiv
Acknowledgments	xv
Dedication	xvi
1 Introduction	1
1.1 Outline	2
1.1.1 Related Work	3
1.1.2 A Taxonomy of 3D Reconstruction	4
1.1.3 A Description of 3D Reconstruction	4
1.1.4 A Mapping of 3D Reconstruction	4
1.1.5 An Interpretation of 3D Reconstruction	4
1.2 Contributions	5
1.3 Organization	6

2	Related Work	8
2.1	ToolBoxes	8
2.2	3D Reconstruction Techniques	8
2.2.1	Stereo Correspondence	9
2.2.2	Shading	13
2.2.3	Silhouette	17
2.2.4	Texture	19
2.2.5	Defocus	20
3	A Taxonomy of 3D Reconstruction	23
3.1	A New Perspective of Taxonomy	24
3.1.1	Object class	24
3.1.2	Class labels	24
3.1.3	Classes of algorithms	26
3.2	Working conditions of algorithms	26
3.2.1	Multi-view Stereo	27
3.2.2	Shape from Shading	27
3.2.3	Photometric Stereo	28
3.2.4	Structured Light	31
3.2.5	Visual Hull	31
3.3	Summary	32
4	A Description of 3D Reconstruction	34
4.1	Definition	35
4.1.1	Basic notations	35
4.1.2	Segment and Scell	35
4.1.3	Photo-consistency	37
4.1.4	Formal Definition	37
4.1.5	Applied Definition	38
4.2	Model	38
4.3	Representation	39
4.3.1	Texture	39
4.3.2	Lightness	40

4.3.3	Reflectance	43
4.3.4	Roughness	43
4.3.5	Concavity	45
4.4	Expression	45
5	A Mapping of 3D Reconstruction	47
5.1	Synthetic setup	48
5.2	Structure of Datasets	48
5.3	Evaluation metrics	49
5.4	Selected methods	50
5.5	Baseline	50
5.6	Effective Problem Domain	52
5.6.1	EPD of PMVS	53
5.6.2	EPD of EPS	57
5.6.3	EPD of GSL	62
5.7	Mapping Construction	66
5.7.1	Mapping of PMVS	66
5.7.2	Mapping of EPS	69
5.7.3	Mapping of GSL	70
5.8	Summary	73
6	An Interpretation of 3D Reconstruction	74
6.1	Evaluation Methodology	75
6.1.1	Objective	75
6.1.2	Key Evaluation Questions and Steps	75
6.2	Parameter Setting	77
6.3	Validation of Mapping	77
6.3.1	Synthetic Datasets	78
6.4	Interpreter	82
6.5	Validation of Interpreter	84
6.5.1	Real-world Datasets	84
6.6	Summary	87

7 Conclusions	88
7.1 Summary	88
7.2 Future directions	88
7.2.1 Geometric Model	88
7.2.2 Property Parameters	88
7.2.3 Metric	88
7.2.4 Mapping Construction	89
7.2.5 Interpreter	89
7.3 Concluding remarks	89
Bibliography	90
A Supporting Materials	97
A.1 Material of real-world objects	97
A.2 Parameters of real-world objects	97
A.3 Results of real-world objects	97

List of Tables

Table 2.1	Classes of algorithms that utilize each visual/geometric cue. Note that the abbreviations will be used extensively in the theis, and the actuall meaning of SfS can be deduced from the context.	9
Table 2.2	Assumptions made by different classes of photometric stereo. . .	14
Table 3.1	Label of the six classes of objects.	26
Table 3.2	Selected algorithms from each class of algorithms	26
Table 3.3	Working conditions of typical Multi-view Stereo algorithms. .	27
Table 3.4	Working conditions of typical Shape from Shading algorithms.	28
Table 3.5	Working conditions of typical Photometric Stereo algorithms. .	30
Table 3.6	Working conditions of typical Structured Light algorithms. .	31
Table 3.7	Working conditions of typical Photometric Stereo algorithms. .	32
Table 3.8	Algorithm classification based on the new taxonomy	33
Table 4.1	Model of the 3D reconstruction problem: properties	39
Table 4.2	Model of the 3D reconstruction problem: requirements	39
Table 4.3	Expression of the reconstruction problem for the object class 1, 2, 3, 5.	46
Table 5.1	Summary of the selected algorithms for the framework, and the corresponding working conditions in theory.	50
Table 5.2	Summary of the baseline algorithms for the framework, and the corresponding working conditions in theory.	51
Table 5.3	Problem conditions for establishing the <i>effective problem do-</i> <i>main</i> of PMVS.	55

Table 5.4	The <i>effective problem domain</i> of PMVS in terms of accuracy and completeness.	57
Table 5.5	Problem conditions for establishing the <i>effective problem domain</i> of EPS.	59
Table 5.6	The <i>effective problem domain</i> of EPS in terms of the <i>angular difference</i>	60
Table 5.7	Problem conditions for establishing the <i>effective problem domain</i> of GSL.	64
Table 5.8	The <i>effective problem domain</i> of GSL in terms of accuracy and completeness.	66
Table 5.9	The condition matrix of PMVS in terms of the two metrics <i>accuracy</i> and <i>completeness</i>	68
Table 5.10	The condition matrix of example-based PS in terms of the metric <i>angular error</i>	70
Table 5.11	The condition matrix of Gray-code SL in terms of the two metrics <i>accuracy</i> and <i>completeness</i>	72
Table 6.1	Property settings of the three testing objects: ‘bottle’, ‘knight’, ‘king’, which have increasing degree of concavity.	78
Table 6.2	Property list for the real-world objects	86
Table A.1	Material of Real-world objects.	98
Table A.2	Property list for the real-world objects	100

List of Figures

Figure 1.1	The three layers of the 3D reconstruction framework.	3
Figure 2.1	Illustrations of MI-based VH. (a) shows one object (top left) and its silhouette with 2D lines traced over it to find intersections along rays in the X, Y and Z ray-set of the MI, respectively. (b) shows the MI data structure and conversion algorithm in a 2D example. Image courtesy of M. Tarini.	19
Figure 2.2	Three distortion effect: distance distortion, position distortion, and foreshortening distortion.	20
Figure 2.3	A thin lens of focal length f focuses the light from a plane a distance z_0 in front of the lens at a distance z_i behind the lens, where $\frac{1}{z_0} + \frac{1}{z_i} = \frac{1}{f}$. If the sensor plane moved forward Δz_i , the image are no longer in focus and the <i>circle of confusion</i> c depends on the distance of the sensor plane motion Δz_i relative to the lens aperture diameter d	20
Figure 2.4	shape from focus	22
Figure 3.1	A list of properties for object classes.	25
Figure 3.2	The effect of GBR ambiguity	29
Figure 3.3	Six classes of objects of interest, and the algorithms that could work reliably for these classes.	32
Figure 4.1	Relation between a scell and a segment	36
Figure 4.2	Light-matter interaction	41
Figure 4.3	The light-matter interaction.	42

Figure 4.4	The light-lens interaction.	42
Figure 4.5	The light-sensor interaction.	42
Figure 4.6	A red specular sphere. The surface reflects light in a mirror-like way, and no diffuse reflection exist, thus the colour of the surface is no longer visible.	44
Figure 4.7	Surface Slope Distribution Model	45
Figure 5.1	Structure of the synthetic dataset.	49
Figure 5.2	A right-skewed distribution, which is a typical graph of the angular error.	52
Figure 5.3	Shape estimation results with varied mean and median values. The algorithm is less sensitive to large mean or median value while more sensitive to the difference between the mean and median value.	53
Figure 5.4	Performance of PMVS under six pairwise conditions. For instance, (a) shows the performance under changing <i>texture</i> and <i>albedo</i> values while the others are fixed. The property values are set based on settings in Table 5.3.	54
Figure 5.5	(a) shows the reflection of light off a specular surface. V_1 received the diffuse component while V_2 receives the specular component. (b), (c) shows the images observed from these two views. The specular area (red circle) observed in V_2 is visible in V_1	56
Figure 5.6	(a)-(c). The albedo is set as 0.2, (d)-(f). the specular is set as 0.2. According to energy conservation, as the specular component increases, the diffuse component decreases.	56
Figure 5.7	Performance of Example-based PS under six pairwise conditions. For instance, (a) shows the performance under changing <i>texture</i> and <i>albedo</i> values. The property values are assigned based on the settings in Table 5.5 (a).	58
Figure 5.8	(a)-(c). The texture is set as 0.5. The estimated normal map and recovered surface becomes consistently worse as the specular level rises.	60

Figure 5.9	According to energy conservation, as the specular component increases, the diffuse component decreases. (a)-(c): the estimated normal map and recovered height map becomes consistently worse as the albedo decreases; (c)-(e): the estimated normal map and recovered height map becomes consistently worse as the specular increaess.	61
Figure 5.10	The ‘peculiar’ effect of roughness on PS. Albedo is set as 0.8, and specular is set as 0.8. (b) demonstrate that a medium level roughness would lead to worse normal estimation since it blurs the specular lobe.	62
Figure 5.11	Performance of Gray-encoded SL under six pairwise conditions. For instance, (a) shows the performance under changing <i>texture</i> and <i>albedo</i> values. The property values are assigned based on settings in Table 5.7 (a).	63
Figure 5.12	(a)-(c): the specular is set as 0.2, albedo has a positive effect on completeness; (d)-(e): the albedo is set as 0.2, specular has a negative effect on completeness.	65
Figure 5.13	(a)-(c): the roughness is set as 0.2, specular has a negative effect on completeness; (d)-(e): the specular is set as 0.8, roughness has a positive effect on competeness.	65
Figure 5.14	Performance of PMVS under varied conditions of changing property values. The baseline method serves as the guidelines to determine the performance of PMVS.	67
Figure 5.15	Performance of EPS under varied conditions of changing property values. Varied statistical measures of angular error are compared to the baseline method to determine the performance of EPS.	69
Figure 5.16	Performance of GSL under varied conditions of changing property values. The baseline method serves as the guidelines to determine the performance of GSL.	71

Figure 6.1	The UI for determining the property settings, including albedo, specular, and roughness of the surface. The albedo is set as the value channel of HSV colour. In this case, the albedo is set as 0.8, and the specular and roughness is set as 0.5, 0.2, respectively. (a) demonstrates the effect of the property settings on a sphere while (b) on a teapot.	78
Figure 6.2	The synthetic dataset and groundtruth for the evaluation of the extensiveness of the mapping to objects with different shapes. Three objects with varied degrees of concavity are selected, each is configured with four properties settings.	79
Figure 6.3	The first column shows the best algorithm chosen by the mapping. The quantitative and qualitative performance of each technique on the synthetic dataset. The red dots are from the ground truth while the black ones the reconstruction.	80
Figure 6.4	The first column shows the best algorithm chosen by the mapping. The quantitative and qualitative performance of each technique on the synthetic dataset. The red dots are from the ground truth while the black ones the reconstruction.	81
Figure 6.5	The first column shows the best algorithm chosen by the mapping. The quantitative and qualitative performance of each technique on the synthetic dataset. The green dots are from the ground truth while the black ones the reconstruction.	83
Figure 6.6	Two components of the Interpreter layer.	84
Figure 6.7	The rerepresentatives of the six classes of objects used for evaluation.	85
Figure 6.8	The evaluation of the effectiveness of the mapping using real-world object. The well reconstructed object is label by green rectangle.	87
Figure A.1	Reconstruction results of MVS, PS, SL	99
Figure A.2	Reconstruction results of MVS, PS, SL (cont'd)	100

List of Acronyms

- **3D:** 3-dimensional
- **BRDF:** Bi-directional Reflectance Distribution Function
- **CAD:** Computer Aided Design
- **EPD:** Effective Problem Domain
- **EPS:** Example-based Photometric Stereo
- **GSL:** Gray code Structured Light
- **MVS:** Multi-View Stereo
- **PMVS:** Patch-based Multi-View Stereo
- **PS:** Photometric Stereo
- **SfS:** Shape from Shading
- **SL:** Structured Light
- **VH:** Visual Hull

Acknowledgments

Thank those people who helped you.

Don't forget your parents or loved ones.

You may wish to acknowledge your funding sources.

Dedication

献给我的爷爷吴国利先生

Chapter 1

Introduction

Modeling of the 3D world has been an active research topic in computer vision for decades and has a wide range of applications including 3D mapping and navigation, online shopping, 3D printing, computational photography, video games, visual effects, and cultural heritage archival. The goal in 3D modeling is to reconstruct a 3D geometric model represented by point cloud, voxel grid, depth maps, or surface mesh, from RGB or range sensors, optionally incorporating the material of the surface.

Achieving this goal is an extremely challenging task, as it involves the reverse process of image formation, which is highly likely to result in a variety of possible results and solutions. To overcome this challenge, some assumptions must be made in terms of materials, viewpoints, and lighting involved. In turn, a solid understanding of the interaction of lighting with surface geometry and material is a prerequisite to fully take advantage of the existing techniques. In past decades, we have witnessed a variety of tools and approaches to 3D modeling applied successfully to an assortment of sub-domains, such as Computer Aided Design (CAD) tools [1], arm-mounted probes, active methods [2, 4, 13, 35] and passive image-based methods [19, 20, 23, 34]. Among the existing approaches, active techniques such as laser scanners [35], Structured Light (SL) systems [13], and Photometric Stereo (PS) [63], as well as passive methods such as Multi-View Stereo (MVS) [52], have been the most successful. Laser scanners and structured light techniques are seen to generate the most accurate results, but are generally complicated to set up and

calibrate, time consuming to scan, and demanding to store and process in terms of memory. Photometric Stereo is able to achieve highly detailed reconstruction comparable to that of laser scanners, but the true depth information is lost due to the use of a single viewpoint. Further, MVS requires minimal setup and can work in both controlled, small scale lab settings as well as outdoor, medium to large scale environments. However, the quality of reconstruction is generally noisier, and is susceptible to the texture and material property of the surface. All of the aforementioned techniques require an understanding of calibration, stereo correspondence, physics-based vision, and so on, which are not easy tasks to master.

Regardless of past successes and strong demands across various areas, we have not yet witnessed any substantial progress in terms of making the mentioned techniques accessible to application developers who generally have little or no computer vision expertise. We've made two key observations about computer vision algorithms: 1) none of these methods works well under all circumstances, nor do they share the same setup or inputs/outputs, making it difficult for developers to choose an optimal method for their particular application; 2) expertise knowledge is a prerequisite to fully exploit the potentials of existing vision techniques. These observations lead us to the following question which we address in this thesis: is it achievable to create a framework that can return a reliable reconstruction by one of the best possible algorithms based on the descriptions of the object or scene to be reconstructed?

The mental model to our approach is similar to that of the game ‘name that object’: one participant makes guesses of what the object is based solely on the descriptions of the appearance provided by the other participant. In our case, the key idea is to construct an algorithm-free framework above the algorithms themselves, so that one or multiple appropriate algorithms can be selected based on the ‘appearance’ of the object described by the developers.

1.1 Outline

The problem addressed by this thesis can be described as follows: construct a framework for 3D reconstruction that can return a reliable reconstruction result by one of the best-suited algorithms, which is determined by the description of the

problem condition, see Figure 1.1. More specifically, a taxonomy is proposed that transforms the 3D reconstruction problem from one requiring knowledge of algorithmic details to one that is based on the correlation between the problem space and algorithms. Next, a well defined model and representations are developed to describe the problem space definitively. Lastly, mapping between the problem space and the algorithms is discovered, from which a proof-of-concept interpreter is proposed. A rigorous evaluation is then carried out to verify the effectiveness and robustness of the interpreter.

The framework consists of the following three layers: the *description layer* sits on top and acts as the interface between the user and the lower layers. It is through this that the user provides a description of the 3D reconstruction problem. The description is passed to the *interpreter* layer, which chooses appropriate algorithms given the description, and then configures each algorithm's parameters. The interpreter can also define any necessary pre or post-processing operations (such as noise removal or image scaling). The lowest layer of the three is where the *algorithms* sit.

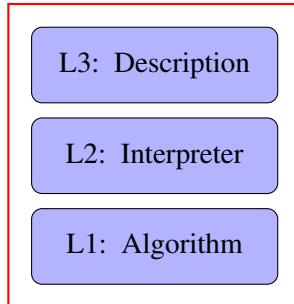


Figure 1.1: The three layers of the 3D reconstruction framework.

1.1.1 Related Work

We discuss the existing software and toolboxes for 3D reconstruction, and present the required vision background needed to fully take advantage of these toolboxes. A review of the 3D acquisition techniques is also provided, organized by the visual and geometric cues used for reconstruction.

1.1.2 A Taxonomy of 3D Reconstruction

The proposed taxonomy categorizes algorithms based not on how well they work, but on the problem domain that they can reliably work under. First, the problem space is developed, with each axis representing a key property of the object's material or geometry. Next, the selected classes of algorithms are mapped to the problem conditions based on reports in the relevant literature.

1.1.3 A Description of 3D Reconstruction

In previous cases, the mapping from a problem space to an algorithms has been ambiguous due to the problem spaces that are less well-defined. Here, we set out to provide a rigorous definition of the problem space itself. First, a formal and practical definition of the 3D reconstruction problem based on Set Theory is proposed. Second, a model consisting of key object properties is developed. Third, the representations of the problem are proposed. We select key elements that can affect the properties of the model and use them as components of the representation. Lastly, common 3D reconstruction tasks are expressed using the proposed model and representations.

1.1.4 A Mapping of 3D Reconstruction

To derive more precise mapping, we need to evaluate the performance of the selected algorithms under varied properties and their combinations. We use synthetic datasets to achieve this goal. Part of the challenge in establishing a comprehensive set of experiments for such an evaluation is the large variations of shapes and material properties. To overcome this issue, we first establish the *effective problem domain* (EPS) by finding the effective and dependent properties. Then we evaluate the performance of each algorithm under the conditions of all dependent properties and their combinations, which serves as the basis of the mapping.

1.1.5 An Interpretation of 3D Reconstruction

We conduct the evaluation of the framework around two key evaluation questions: 1) can the derived mapping be extended to an object with a different shape; 2) can the framework return a reliable result given the correct description to the problem

condition. To answer these questions, we carry out two separate experiments: 1) we use synthetic objects with the same configurations as the ones used to derive the mapping, and check if the algorithms can return reliable results that are consistent with the mapping; 2) we use real-world objects to test the use of the framework.

1.2 Contributions

The main contribution of this thesis is the development and evaluation of a framework for the 3D reconstruction problem in computer vision, to hide the details of specific methods. An abstraction may be employed by users to describe the conditions of the vision problems they are trying to solve, and our novel interpreter uses such descriptions to select an appropriate algorithm and return a reliable result. This endeavor is non-trivial for two reasons: 1). currently, not one approach can achieve satisfactory results for an object with general material and geometric properties. Our framework can, to an extent, solve this problem by incorporating multiple algorithms that are designed for a range of problem conditions; 2). a solid understanding of reconstruction algorithm details is a prerequisite to fully take advantage of the existing techniques, which is difficult for general developers to obtain. The descriptive language proposed in this thesis can allow application developers to bypass the aforementioned hurdles, and thus is a developer-friendly approach. The significant aspects of our approach are presented in further detail below:

1. A taxonomy of the 3D reconstruction problem that focuses on problem conditions instead of algorithmic details.

Typical taxonomies generally focus on one class of algorithms and are algorithm centric. Normally they describe and classify intra-class algorithms based on *how* an algorithm solves a problem. For instance, MVS algorithms can be categorized based on visibility models or scene representations, and PS methods can be classified by reflectance models. While this type of taxonomy provides a decent basis for comparison of intra-class algorithms, it provides little insight into the conditions where techniques can perform well, which is crucial when it comes to designing an application that requires reliable reconstruction techniques. Thus this thesis introduces a new perspective of taxonomy for 3D reconstruction that is

based on the conditions surrounding the problem. Further, we address the lack of progress in certain areas of this research field, which will be helpful for redirecting research efforts to less explored territories.

2. A description of the 3D reconstruction problem that allows increased precision in mapping from a well-defined problem space to algorithms.

Much of the recent research in this area has been focused on technical novelties. However, we need to pay equal attention to the conditions under which algorithms are designed, in order to return reliable results. One of the reasons for this oversight is the ambiguity of the problem domain. Information regarding the set of conditions under which a given algorithm performs well is difficult to convey without an agreed upon model representing the problem space itself. Conversely, knowledge of which algorithm best suits a particular set of conditions in the problem space is also difficult to determine without a model to represent such conditions. Therefore, it's crucial to have a better understanding of the problem space so that we can exploit the working space of algorithms.

3. A mapping from the description of the object to appropriate algorithms.

Given an algorithm, the conditions under which this specific algorithm works well is largely unclear. Additionally, given a specific problem condition, the knowledge of which algorithm performs well under this problem condition is empirical. Therefore, we need to find a precise mapping from the well defined problem space to determine a reliable solution. Here the effective conditions of a specific algorithm is discovered by evaluating performance under the well defined problem conditions.

1.3 Organization

We organize this thesis as follows. First we will discuss related works in Chapter 2. In Chapter 3, we provide a taxonomy of 3D reconstruction based on the conditions surrounding a problem. In Chapter 4, we provide a formal description of the 3D reconstruction problem, which can be applied to current existing techniques. In Chapter 5, we discuss the process of generating a synthetic dataset to evaluate the performance of a selected set of techniques under varied problem conditions, from which a mapping can be derived. In Chapter 6, we use both synthetic

and real-world datasets to demonstrate the interpretation of the 3D reconstruction description and the validity of the proposed mapping.

Chapter 2

Related Work

Section 2.1 discusses the existing toolboxes for 3D reconstruction. Section 2.2 presents a comprehensive review of the field of image-based 3D reconstruction based on varied visual/geometric cues, which include *stereo correspondence, shading, silhouette, texture distortion, and (de)focus*.

2.1 ToolBoxes

There have been many attempts in developing computer vision or image processing frameworks that support rapid development of vision applications. There are multiple general vision libraries in the field including OpenCV [16], VLFeat [59], VXL [5] and multiple Matlab libraries [33, 40]. These libraries often provide tools to multiple image processing and computer vision problems, including low-vision tasks such as feature detection and matching, middle-level vision tasks such as segmentation, tracking, and high-level vision problems such as classification and recognition. All of these software frameworks and libraries provide vision components and algorithms without any context of how and when they should be applied, and so often require expert vision knowledge for effective use.

2.2 3D Reconstruction Techniques

Image-based 3D reconstruction attempts to recover the geometry and optionally the material of the object from images under different viewpoints or illuminations.

The goal can be described as “given a set of images of an object or a scene, estimate the most likely 3D shape that explains those images, under the assumption of known materials, viewpoints, and lighting conditions”. This definition reveals that if those assumptions are violated, this becomes an ill-posed problem since multiple combinations of geometry, viewpoint and illumination can produce exactly the same images [45], thus making it an extremely challenging task.

The 3D reconstruction techniques exploits a variety of visual and geometric cues to extract geomtry from images: stereo correspondence, shading, contour, texture, (de)focus, etc, please refer to Table 2.1 for an overview. The algorithms are organized based on the cue used for reconstruction.

Cue	Algorithm
Stereo correspondence	Stereoscopy Trinocular Stereo Multi-view Stereo (MVS) Laser scanning Structured light (SL)
Shading	Shape from Shading (SfS) Photometric Stereo (PS)
Contour	Shape from Silhouette (SfS)
Texture	Shape from Texture
(De)focus	Shape from (De)focus

Table 2.1: Classes of algorithms that utilize each visual/geometric cue. Note that the abbreviations will be used extensively in the theis, and the actuall meaning of SfS can be deduced from the context.

2.2.1 Stereo Correspondence

Stereo correspondence is one of the most widely used visual cues in 3D vision. Passive methods, including stereoscopy, trinocular stereo, and MVS, identify correspondences across different views, and estimate the 3D point by triangulation. However these passive approaches suffer from uniform or periodic surfaces. The active techniques attempt to overcome the correspondence problem by replacing one of the cameras with a controllable illumination source, e.g., single-point laser,

slit laser scanner, and temporal or spatially modulated Structured Light (SL), we refer the readers to the survey article by Blais for recent development of active methods. Two most popular methods, MVS and SL, are reviewed in depth, and organized based on the reconstruction algorithms and projection patterns used, respectively.

Volumetric methods

The first class computes the cost function in a 3D volume, then extracts a surface from this volume. One successful algorithm is voxel colouring, which traverses a discretized 3D space in depth-order to identify voxels that have a unique colouring, constant across all possible interpretations of the scene [51]. Another thread of work formulates the problem in the Markov Random Field (MRF) framework and extracts the optimal surface by Graph-Cut algorithms [47, 60, 61].

Surface Evolution

The second class works by iteratively evolving a volume or surface to minimize a cost function. The class includes methods based on voxels, level set, and surface meshes. Space Carving technique achieves least-commitment shape [41] by iteratively removing inconsistent voxels from the scene [34]. Level-set techniques cast the problem as a variational one, and use a set of PDE's as cost functions, which are deformed from an initial set of surfaces towards the objects to be detected [19]. Other approaches use a deformable model and represent the scene as surface meshes that moves as a function of internal and external forces [18]. Hiep et al. presented a visibility-based method that transforms a dense point cloud into a surface mesh, which is feed into a mesh-based variational refinement that captures small details, smartly handling photo-consistency, regularization and adaptive resolution.

Region Growing

The third class starts with a sparse set of scene points, and propagates these points to spatial neighbours and refine the cost function with respect to position and orientation of the points. Otto and Chau proposed one of the first work on region grow-

ing stereo search. The essence of the algorithm is: start with an approximate match between a point in one image and a point in another, use an adaptive least-squares correlation algorithm to produce a more accurate match, and use this to predict approximate matches for points in the neighbourhood of the first match. A two-view quasi-dense approach first sorts the list of point correspondences into a list of seed points by correlation score. At each step of the propagation, A ‘best’ seed point is chosen. Then in the immediate spatial neighborhood of this seed point, new potential matches are checked and the bests are added to the current list of seed points [36, 37]. This best-first strategy guarantees convergence by choosing only new matches that have not yet been selected. A patch based approach undergoes multiple iterations of matching, propagation, and filtering [20]. A stereoscopic approach called PatchMatch Stereo, which is inspired by an approximate nearest neighbour matching algorithm called PatchMatch [9]. The method starts by randomly assigning an oriented plane to each pixel in two views. Then each pixel goes through three iterations of propagations and refinement. The plane is propagated to spatial neighbours, corresponding pixel from another view, and across time. It can achieve sub-pixel accuracy, but is computational heavy and difficult to parallelism. There has been some efforts to extend PatchMatch Stereo to multi-view scenario [21, 58, 64] or proposing new propagation scheme to increase the computational efficiency [21].

Depthmap Merging

The fourth class is image-space based methods that computes a per-view depthmap. By treating a depthmap as a 2D array of 3D points, multiple depthmaps can be considered as a merged 3D point cloud. The winner-takes-all approach takes a set of discretised depth values and pick the one with the highest photo-consistency score for each pixel independently. Uniform depth sampling may suffice for simple and compact objects. However, for complex and large scenes, a proper sampling scheme is crucial to achieve high speed and quality. More sophisticated cost function are derived to account for occlusion or non-Lambertian effects which might add noise to the photo-consistency score [23, 61]. In the case of severe occlusion, spatial consistency can be enforced under the assumption that neighbouring pixels

have similar depth values. This can be formulated under the Markov Random Field (MRF) framework, where the problem becomes minimizing the sum of a unary $\Phi(\cdot)$ and pairwise term $\Psi(\cdot, \cdot)$. The unary term reflects the photo-consistency score of assigning a depth value d_p from a depth set to the pixel p , whereas the pairwise term enforces the spatial regularization, and assigns the cost of setting depth label k_p, k_q to a pair of neighbouring pixels p and q , respectively.

$$E(\{k_p\}) = \sum_p \Phi(k_p) + \sum_{(p,q) \in \mathcal{N}} \Psi(k_p, k_q)$$

Structured Light

Structured light is considered one of the most accurate reconstruction technique. It is based on projecting a temporally or spatially modulated pattern onto the surface and viewing the illuminated surface from one or more points of view. The correspondence is easily detected from the projected and imaged pattern, which is triangulated to obtain the 3D point. Each pixel in the pattern is assigned a unique codeword, and the codeword is encoded by using grey level, colour or geometric representations. Structured light is classified based on the coding strategy: temporal, spatial and direct codification [48]. Temporal techniques generate the codeword by projecting a sequence of patterns. Spatial codification represents each codeword in a unique pattern. Direct codification techniques define a codeword for every pixel, which is equal to its grey level or colour.

Temporal encoding A sequence of patterns are successively projected onto the surface, the codeword for a given pixel is formed by the sequence of illumination values for that pixel across the projected patterns. This kind of pattern can achieve high accuracy due to two factors: 1). the codeword basis is small, e.g., two for binary pattern, therefore, each bit is easily distinguishable; 2). a coarse-to-fine strategy is used, and the position of the pixel becomes more precise as the patterns are successively projected. We further classify these techniques as follows: 1). binary codeword; 2). n -ary codeword; 3). gray code combined with phase shifting; 4). hybrid techniques.

Spatial encoding This kind of technique concentrate all the coding in a unique

pattern. The codeword that labels a certain pixel is obtained from a neighbourhood of the pixels around it. Normally, the visual features gathered in a neighbourhood are the intensity or colour of the pixels or groups of pixels around it.

Direct encoding There are ways that can directly represent the codeword in each pixel. To achieve this, there is a need to use either a large range of colour values or introduce periodicity. However, this kind of pattern is highly sensitive to noise because the “distance” between codewords is nearly zero. Moreover, the perceived colour depends not only on the projected colour, but also the intrinsic colour of the surface, therefore, reference images must be taken. This kind of coding can be classified as: 1). codification based on grey levels; 2). codification based on colour.

2.2.2 Shading

The shading variations can reveal the surface normal orientation, which can be further integrated into a 2.5D height map. Shading variation depends on the shape (surface normal orientation), reflectance (material), and lighting (illumination), therefore is generally a ill-posed problem because difference shapes illuminated under different light conditions might produce the same image. This leads to a novel technique called Photometric Stereo in which surface orientation is determined from two or more images. The idea of Photometric Stereo is to vary the direction of the incident illumination between successive views while holding the viewing direction constant. This provides enough information to determine surface orientation at each pixel [62]. This technique can produce a surface normal map with the same resolution of the input image, i.e., to produce the pixel-wise surface normal map. Since the coefficients of the normal are continuous, the integrated height map can reach an accuracy that cannot be achieved by any triangulation methods. Therefore, photometric stereo is more desirable if the intrinsic geometric details are of great importance.

Shape from Shading

The problem of recovering the shape of a surface from the intensity variation is first proposed by Horn [28]. It assumes that the surface under consideration is of

a uniform albedo and reflectance, and that the direction of the single distant light source is either known or can be calibrated by the use of a reference object. Thus the intensity $I(x, y)$ becomes purely a function of the local surface orientation. The information of reflectance, illumination, and viewing geometry can be combined into a single function called reflectance map $R(p, q)$, that relates surface orientation directly to image intensities

$$I(x, y) = R(p(x, y), q(x, y))$$

$$I(x, y) = \rho(\vec{n}, \vec{l}) \vec{n}^\top \vec{l} \quad (\text{Lambertian model})$$

where $(p, q) = (z_x, z_y)$ are surface gradients. Unfortunately, measurements of the brightness at a single pixel only provide one constraint whereas surface orientation requires two. Thus additional constraints such as smoothness or integrability is required to estimate (p, q) .

Photometric Stereo

Category	Camera	Light source	Reflectance
Original PS	Orthographic	Directional, known intensity and direction	Lambertian
Generalized lighting PS	Orthographic	unknown intensity and direction, ambient	Lambertian
Generalized reflectance PS	Orthographic	Distant, known intensity and direction	Non-Lambertian

Table 2.2: Assumptions made by different classes of photometric stereo.

Original Photometric Stereo This method, first proposed by Woodham [63], utilised multiple light sources from different directions to overcome the ambiguity of Shape from Shading. Assume there are P pixels per image, and Q illumination

directions, the intensity of the i th pixel under j th illumination would be

$$\begin{aligned} I_{i,j} &= \rho_i \vec{n}_i^\top \vec{l}_j \\ \Rightarrow \mathbf{I} &= \mathbf{N}^\top \mathbf{L} \end{aligned}$$

where

- $\mathbf{I} \in \mathbb{R}^{P \times Q}$ stores the pixel intensity from all images. Each column contains pixels from each image while each rows contains intensity of each pixel under all illumination conditions
- $\mathbf{N} \in \mathbb{R}^{P \times 3}$ encodes the albedo-scaled surface normal for each pixel, i.e., $N_{i,:} = \rho_i \vec{n}_i^\top$
- $\mathbf{L} \in \mathbb{R}^{3 \times Q}$ encodes the light source directions, i.e., $L_{:,j} = \vec{l}_j$

This surface reflectance, i.e., spatially varying albedo, and the normal can be estimated by

$$\begin{aligned} N &= \mathbf{I} \mathbf{L}^+ \\ \rho_i &= \|N_{i,:}\| \\ n_i &= \frac{N_{i,:}^\top}{\|N_{i,:}\|} \end{aligned}$$

The key problem is how to generalize the assumptions of photometric stereo. For the camera assumption, orthographic projection can be achieved by using a lens with long focus and placing the objects far from the camera. The nonlinear response can be solved by performing radiometric calibration. The shadow and other global light transportation are one of the sources of errors, some approaches consider them as outliers and remove them before normal estimation. The reflectance and lighting assumptions, however, are the most complicated ones since the reflectance properties depends on material property and the microscopic structure, and the lighting can have arbitrary or fixed position, orientation, and intensity. Therefore the research on Photometric Stereo are generally on two directions: 1). generalization of reflectance; 2). generalization of lighting conditions.

Generalization of Lighting It is possible to estimate the surface orientation without knowing the light directions, a case also known as *uncalibrated Photometric Stereo*, see Table 2.2. Most such techniques assume Lambertian techniques and are based on factorization technique proposed in [25]. Recall the Irradiance equation:

$$I = N^\top L$$

However, an infinite number of candidates \hat{N} and \hat{L} make the above equality met. In fact, any invertible 3×3 matrix G defines a candidate pair $\hat{N} = N \cdot G, \hat{L} = G^{-1}L$. Thus the normal N and light source direction L can only be recovered up to a linear transformation.

Other generalized lighting conditions are anything other than the ideal case of using a single distant point light source in a dark room. Therefore, any general cases like natural ambient light, multiple point light sources with/without ambient lighting, etc. To make the problem more tractable, the reflectance model should no longer be a general one, otherwise, the problem would have too many degrees of freedom, which means many different shapes with an incorrectly estimated general reflectance, and an incorrectly estimated general lighting would generate the same image appearance with much higher probability.

Generalization of Reflectance This class of techniques relax the assumption of Lambertian reflectance.

Outlier rejection The fact that the reflectance of non-Lambertian surfaces can be approximated by the sum of a diffuse and a specular lobe has been exploited extensively. The specular pixels are considered as outliers in [17] and [10]. The assumption that the color of the specular lobe differs from that of the diffuse lobe allows the separation of the specular and diffuse components [39, 49, 50].

Reference object A different approach uses a reference object that has the same material as the target object. This is proposed in [54] and later revisited in [26]. It can deal with arbitrary BRDFs as long as the reference and target object has the same material. Multiple reference objects are needed for spatially-varying BRDFs as the BRDF at each point on the target object is a linear combination of the basis BRDFs defined by the set of reference objects.

Parametric reflectance model More sophisticated BRDF models can replace

the reference objects. An isotropic Ward model is used as basis BRDF, and the surface orientation and parameters of the reflectance models are estimated iteratively [24].

Invariants of BRDF While parametric reflectance models are very good at reducing the complexity of BRDFs, they are usually only valid for a limited class of materials. An alternative is to exploit the invariants of BRDFs. Typical ones include energy conservation, non-negativity, Helmholtz reciprocity, isotropy, etc [7, 65].

2.2.3 Silhouette

In some cases, it's an easy task to perform a foreground segmentation of the object of interest, which leads to a class of techniques that reconstructs a 3D volumetric model from the intersection of the binary silhouettes projected into 3D. The resulting model is called a *visual hull*.

The basic idea of shape from silhouette algorithms is that the object lies inside the intersection of all visual cones back-projected from silhouettes. Suppose there are multiple views V of the target object. From each viewpoint $v \in V$, the silhouette s_v can be extracted, which is the region including the object's interior pixels and delimited by the line(s) separating the object from the background. The silhouette s_v are generally non-convex and can represent holes due to the geometry of the object. A cone-like volume $cone_v$ called (truncated) extended silhouette is generated by all the rays starting at the center of projection and passing through all the points of the silhouette. The target object is definitely internal to $cone_v$ and this is true for every view $v' \in V$; it follows that the object is contained inside the volume $c_V = \cap_{v \in V} c_v$. As the size of the V goes to infinity, and all possible views are included, c_V converges to a shape known as the *visual hull* vh of the target object.

[computational complexity] intersection of many volumes can be slow. Simple polyhedron-polyhedron intersection algorithms are inefficient. To improve performance, most methods 1) quantize volumes, 2) perform intersection computation in 2D instead of 3D.

Voxel based methods

First the object space is split up into a 3D grid of voxels; each voxel is intersected with each silhouette volume; only voxels that lie inside all silhouette volumes remain part of the final shape.

Marching intersections based methods

The marching intersection (MI) structure consists of 3 orthogonal sets of rays, parallel to the X , Y , and Z axis, which are arranged in 2D regular arrays, called the $X-rayset$, $Y-rayset$, $Z-rayset$ respectively. Each ray in each rayset is projected to the image plane to find the intersections with the silhouette. These intersections are un-projected to compute the 3D intersection between the ray and the extended silhouette on this ray. This process is repeated for each silhouette, and the un-projected intersections on the same ray are merged by the boolean AND operation.

Once the MI data structure representing the intersection of all extended silhouettes, a triangular mesh is extracted from it. This is done by the MI technique proposed in [46] which traverses the “virtual cells” implicitly defined by the MI, builds a proper marching cube (MC) entry for them that in turn is used to index a MC’s lookup table.

Exact polyhedral methods

The silhouette is converted into a set of convex or non-convex 2D polygons with holes allowed. The resulting visual hull with respect to those polygonal silhouettes is a polyhedron. The faces of this polyhedron lie on the faces of the original cones. The faces of the original cones are defined by the center of projections and the edges in the input silhouettes. The idea of this method is: for each input silhouette s_i we compute the face of the cone. Then we intersect this face with cones of all other input silhouettes, i.e., a polygon-polyhedron intersection. The result of these intersections is a set of polygons that define the surface of the visual hull.

All of the cues above are most widely used ones, and achieved decent results. These following two cues haven’t resulted in as much success. Therefore, we only discuss the general idea rather than the technical details.

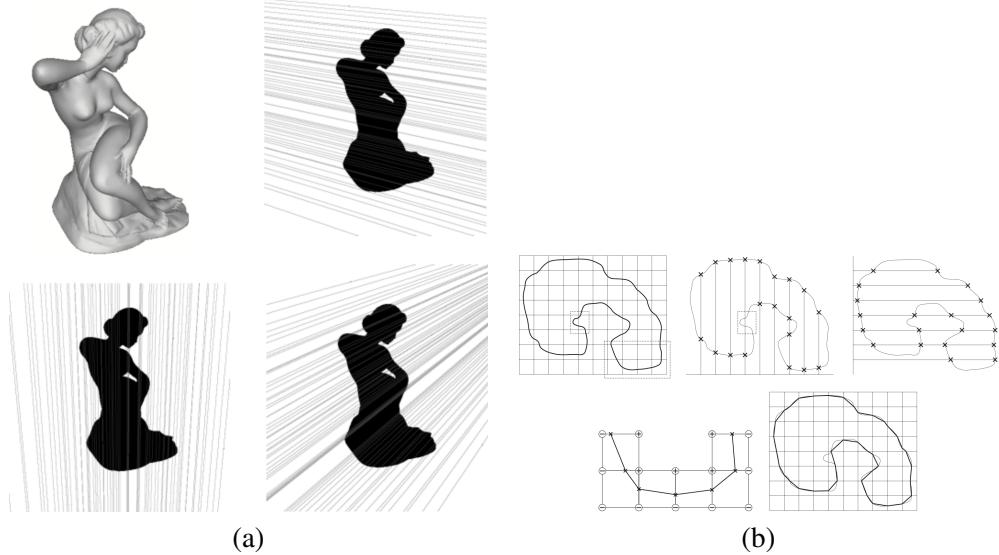


Figure 2.1: Illustrations of MI-based VH. (a) shows one object (top left) and its silhouette with 2D lines traced over it to find intersections along rays in the X, Y and Z ray-set of the MI, respectively. (b) shows the MI data structure and conversion algorithm in a 2D example. Image courtesy of M. Tarini.

2.2.4 Texture

The basic principle behind shape from texture is the *distortion* of the individual texel. In general, the image formation process introduces three distortion effects: the *distance effect*, which makes objects in view appear larger when they are closer to the image plane; the *position effect* which makes objects appear differently when the angle between the line of sight and the image plane different; and the *foreshortening effect*, which distort the objects depending on the angle between the surface normal and the line of sight. Besides, different effects take place under different projection models: the orthographic projection captures only the foreshortening effect whereas the perspective projection captures all three. Therefore, shape from texture methods which use orthographic projection are valid only in a limited domain, where the other two effects can be ignored, and the perspective model captures all three effects, but the resulting algorithms are complicated and involves the solution of nonlinear equations.

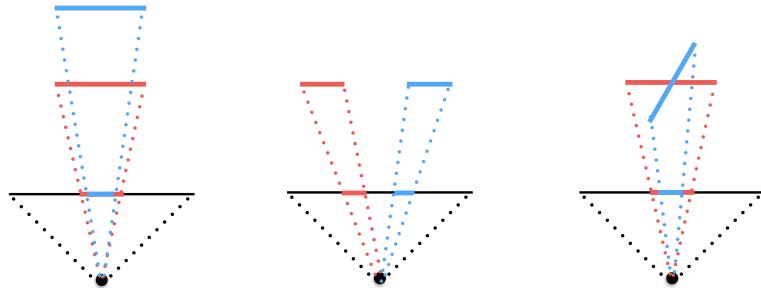


Figure 2.2: Three distortion effect: distance distortion, position distortion, and foreshortening distortion.

To calculate the surface curvature at any point is far from trivial. Therefore, the surface shape is reconstructed by calculating the surface orientation (surface normal). A map of surface normals specifies the surface's orientation only at the points where the normals are computed. But, assuming that the normals are dense enough and the surface is smooth, the map can be used to reconstruct the surface shape.

2.2.5 Defocus

Shape from focus A strong cue for object depth is the amount of blur, which increases as the object moves away from the camera's focusing distance. As shown in Figure 2.3, moving the object surface away from the focus plane increases the circle of confusion.

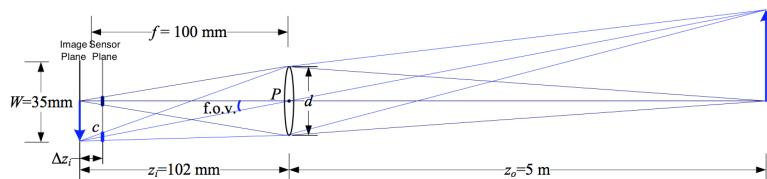


Figure 2.3: A thin lens of focal length f focuses the light from a plane a distance z_0 in front of the lens at a distance z_i behind the lens, where $\frac{1}{z_0} + \frac{1}{z_i} = \frac{1}{f}$. If the sensor plane moved forward Δz_i , the image are no longer in focus and the *circle of confusion* c depends on the distance of the sensor plane motion Δz_i relative to the lens aperture diameter d .

Figure 2.3 shows the basic geometric image formation. The relationship between the object distance z_o , focal distance of the lens f , and the image distance z_i , is given by the Gaussian lens law:

$$\frac{1}{z_o} + \frac{1}{z_i} = \frac{1}{f}$$

All light rays that are radiated from the object and intercepted by the lens to converge at a single point on the image plane, thus a *focused* image $I_f(x,y)$ is formed on the image plane. If, however, the sensor plane does not coincide with the image plane and is displaced from the image plane by a distance Δz_i , the energy received from the object is uniformly distributed over a circular patch on the sensor plane. The relationship between the radius c of the circle of confusion and the sensor displacement Δz_i is as follows:

$$c = \frac{\Delta z_i r}{z_i}$$

The defocused images can be obtained in three ways: by displacing the sensor with respect to the image plane, by moving the lens, or by moving the object with respect to the object plane. The first two ways can cause the following problems:

- The magnification of the system varies, thereby causing the image coordinates of the object points to change.
- The area on the sensor plane over which light energy is distributed varies, thereby causing a variation in image brightness.

To address this issue, the degree of focus is changed by moving the object with respect to a fixed configuration of the optical system and sensor. This approach ensures that the focused areas of the image are always subjected to the same magnification.

The idea is as follows: the stage is moved in increments of Δd , and an image is captured at each stage position ($d = n\Delta d$). By studying the behaviour of the focus measure, an interpolation method is used to compute the accurate depth estimates from a small number of focus measures. An important feature of this method is the local nature, the depth estimate at an image point is computed only from focus measures recorded at that point.

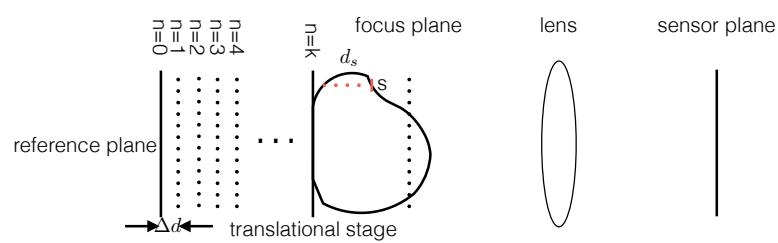


Figure 2.4: shape from focus

Chapter 3

A Taxonomy of 3D Reconstruction

Existing taxonomies of 3D reconstruction techniques generally focus on one category of techniques: the Multi-view Stereo taxonomy in [52] proposed classification of MVS algorithms from various perspectives. Reviews of Structured Light techniques generally classify techniques based on the type of projection pattern used [22, 48]. Photometric Stereo algorithms are classified by the assumptions or generalizations made, for instance, unknown/known reflectance, unknown/known light conditions (uncalibrated/calibrated), etc [53]. These frameworks provide a way to categorize intra-category algorithms, but is unsuitable to evaluate the performance of inter-category algorithms. Furthermore, the algorithms under consideration are targeted at a limited categories of objects. It's well known that these algorithms are highly likely to fail on other categories of objects, and this knowledge of algorithmic applicability is largely empirical, with each algorithm roughly maps to a problem domain that is poorly defined. Thus we need a more object-centered approach to the taxonomy so that a more precise mapping is available.

It's crucial to understand where algorithms perform well and where they fail when designing an application for reconstruction. Under the previous framework of taxonomy, this knowledge is largely empirical, with each algorithm mapped roughly to a sub-volume in the problem space. However, this mapping is ambiguous, i.e., the sub-space is not well defined, and also too general, i.e., algo-

rithm within the same category cannot be effectively distinguished. To overcome this limitation, we need to 1). propose a well-defined problem space; 2). bypass the algorithmic details and focus on properties that are not intuitive to understand and perceive.

The taxonomy proposed in this chapter defines the 3D reconstruction techniques from an object-centered viewpoint, i.e., categorize algorithm based on object class. This taxonomy transforms the 3D reconstruction problem from one requiring knowledge and expertise of specific algorithms in terms of how and when to use them, to one requiring knowledge of the visual and geometric properties of the target object.

3.1 A New Perspective of Taxonomy

Typical taxonomies generally focus on one class of algorithms, and are algorithm centric, describing and classifying intra-class algorithms based on *how* an algorithm solves the problem. However, this approach 1). gives very little insight to what conditions does a specific algorithm work well; 2). requires vision knowledge to understand and use these algorithms. The new taxonomy approaches this problem from a different perspective, more specifically, the algorithms are categorized based on the type of objects/problem conditions that they can reliably work under. We first give an overview of object classes, which serves as the bases to the taxonomy.

3.1.1 Object class

In Figure 3.1 (a), we show a taxonomy of object classes with different material and shape properties. There are in total $3 \times 3 \times 2 \times 4 \times 2 \times 5 = 720$ classes of objects, which still don't fully capture the variations exhibited by real world objects, for instance, effects such as occlusion, discontinuity, emission, etc are not considered.

3.1.2 Class labels

We propose the following labels to the differentiate object classes reviewed above. The order of the labels are: translucency, texture, lightness, reflection model, sur-

Translucency	Texture	Lightness	Reflection	Roughness	Concavity
Opaque	Textureless	Bright	Diffuse		Smooth 
Translucent	Repeated Texture 	Dark	Mixed diffuse and specular		Rough 
Transparent	Textured 		Subsurface scattering		Convex 
			Refraction		Concave 

Figure 3.1: A list of properties for object classes.

face roughness, and concavity.

- **Translucency:** **O:** opaque, **Tl:** translucent, **Tp:** transparent.
- **Texture:** **T:** textured, **Tr:** repeated textured, **Tl:** textureless.
- **Lightness:** **B:** bright, **D:** dark.
- **Reflection:** **D:** diffuse model, **S:** specular model, **M:** mixture of diffuse and specular, **Ss:** subsurface scattering, **Rf:** refraction
- **Roughness:** **S:** smooth, **R:** rough
- **Concavity:** **Cx:** convex, **Cv:** concave

Most techniques that have been developed over the past decades can only tackle a subset of all possible object classes, with a focus on opaque, diffuse objects. For specular, refractive, and translucent or transparent objects, only very specialized algorithms are applicable for reconstruction [29]. Therefore, we consider only the six classes of objects listed in Table 3.1.

Class #	Translucency	Texture	Lightness	Refection	Roughness	Concavity
1	O	Tl	B	D	R	Cx
2	O	Tl	B	M	S	Cx
3	O	T	B	D	R	Cx
4	O	T	B	M	S	Cx
5	O	T	D	D	S	Cx
6	O	T	D	M	S	Cx

Table 3.1: Label of the six classes of objects.

3.1.3 Classes of algorithms

Here is a list of selected algorithms that will be looked into in depth, a summary is listed in Figure 3.2.

Algo. class	Technique
SfS	Horn [28]
MVS	Furukawa [20], Goesele [23], Vogiatzis [61], Hernández [18], Faugeras [19]
Lamberian PS	Woodham [63], Hayakawa [25], Belhumeur [11], Alldrin [8]
Non Lambertian PS	Coleman [17], Barsky [10], Schluns [50], Sato [49], Mallick [39], Alldrain [6], Goldman [24], Silver [54], Hertzmann [26], Zickler [65]
SL	Inokuchi [30]
VH	Szeliski [55], Matusik [42], Tarini [57]

Table 3.2: Selected algorithms from each class of algorithms

3.2 Working conditions of algorithms

This section investigate the cases where each categories of algorithm is capable to work under based on the reported literature.

3.2.1 Multi-view Stereo

The working conditions of Multi-view Stereo algorithms are summarized in Table 3.3.

High texture

Multi-view Stereo algorithms take advantage of the textural information to establish point correspondences across various viewpoints, thus they work best under high textured conditions.

Diffuse reflectance

Most MVS algorithms require that the matching image region with similar or same appearances from different angles, and hence, most of the algorithms assume Lambertian reflectance. While pure Lambertian surfaces are rare in practice, it is known and empirically verified that MVS algorithms work very well on non-Lambertian surface: as long as they contain some diffuse reflectance component, and the photo-consistency function is able to identify and ignore images whose non-diffuse effects(e.g., specular highlights) are strong, then utilize the diffuse component in the remaining images. However, there are some attempts to overcome this limitations, a pure passive methods was proposed that directly model and analyze non-Lambertian effects for MVS algorithms [31, 32].

Technique	Texture	Lightness	Reflectance	Roughness	Concavity
MVS	Textured	-	Diffuse or mixed	-	-

Table 3.3: Working conditions of typical Multi-view Stereo algorithms.

3.2.2 Shape from Shading

Shape from Shading, first proposed by Horn is targeted specifically for known isotropic Lambertian surfaces. By assuming orthographic projection, and known light source intensity and direction, surface orientation can be estimated from the shading variations. The working condition is shown in Table 3.4.

Technique	Texture	Lightness	Reflectance	Roughness	Concavity
SfS	Textureless	Bright	Lambertian	-	Convex

Table 3.4: Working conditions of typical Shape from Shading algorithms.

3.2.3 Photometric Stereo

The working conditions of Photometric Stereo algorithms are presented below. Visual texture can be thought of as a pattern or variance of intensity appearing on an object's surface. In this thesis, the visual texture will be considered as resulting from non-uniform surface albedo. Thus uniform albedo represents uniform texture while non-uniform albedo represents textured surface. The working conditions are shown in Table 3.7.

Albedo

Photometric Stereo can only work well on surfaces with high albedo.

The original PS was developed for surfaces with uniform albedo, PS can be extended easily for spatially varying albedo. First the albedo-scaled normal can be first estimated as usual, then the albedo is retrieved as the magnitude of the scaled normal [63].

Lambertian reflectance

The traditional Photometric Stereo can be considered as an extension of the original Shape from Shading, which incorporates additional light sources to eliminate ambiguity [63], thus are only applicable to Lambertian surface with uniform albedo.

To avoid the tedious process of light calibration, Silver [54] proposed a look-up table scheme that relies on reflectance object with the same reflectance as the target, a uniform Lambertian surface in this case. This approach is later adapted to surface with non-Lambertian reflectance with varying albedo or material in [26].

The original Photometric Stereo require light calibration, *uncalibrated photometric stereo* has been proposed to avoid this tedious process. One approach used six or more pixels with the same albedo, and was able to solve for normals up to a

rotation ambiguity[25]. It can be further proved that a 3-parameter subset of these transformations, known as the Generalized Bas-Relief (GBR) ambiguity, preserve surface integrability [11]. Thus, given three or more images of a Lambertian object acquired under light sources of unknown direction and strength, the surface can be reconstructed up to GBR transformation by enforcing surface integrability, see Figure 3.2 for the effect of GBR-ambiguity.

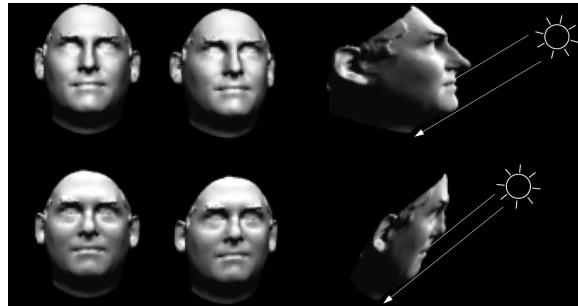


Figure 3.2: The effect of GBR ambiguity

Non-Lambertian reflectance

The *reference object* approach, first proposed by Silver, and later revisited by in [26], can be used for surfaces with spatially varying reflectance. The basic assumption is that the BRDF at each point is a linear combination of the “basis” BRDFs defined by the set of reference objects.

One approach exploits the fact that the reflectance of non-Lambertian surfaces can be approximated by **diffuse component and specular lobe**. Coleman and Jain and Barsky and Petrou who treat specular pixels as outliers, and Schluns, Sato and Ikeuchi, and Mallick et al. who assume the color of the specular lobe differs from the color of the diffuse lobe, allowing separation of the specular and diffuse components.

Due to the high complexity of the BRDF, some methods utilize analytical reflectance models. Goldman et al. uses an isotropic Ward model for each basis BRDF, and the surface orientation and parameters of the reflectance models are estimated iteratively. Alldrin et al. proposed a data-driven approach that got rid of the parametric reflectance model, and employed an novel bi-variate approxima-

tions of isotropic reflectance functions. By combining this approximation with the weighted basis BRDFs, a per-pixel surface normal, global set of non-parametric basis BRDFs, and the corresponding weights are able to be independently estimated. Though the parametric reflectance model can significantly reduce the complexity of BRDFs, they are typically restricted to a limited classes of materials.

Another alternative to using BRDF models is to take advantage of the properties of BRDFs, include energy conservation, non-negativity, Helmholtz reciprocity, or isotropy. Helmholtz stereopsis introduced by Zickler et al. exploits the reciprocity to obtain the surface reconstruction. Isotropy is another physical property which holds for material without “grain”. Tan et al. use both symmetry and reciprocity present in isotropic BRDFs to resolve the generalized bas-relief ambiguity. Alldrin and Kriegman show that isotropy, with no further assumptions on surface shape or BRDF, can be utilized to recover the surface normal at each surface point up to a plane.

Convex

Active methods that assumes a local reflectance model such as most PS can work more reliably for surfaces without casting shadow. Thus the concave surfaces pose a great challenge for this technique since the indensity is no longer determined solely by the light source, surface normal and viewing direction.

Technique	Texture	Lightness	Reflectance	Roughness	Concavity
Lambertian uniform albedo	PS, uniform albedo	Textureless	Bright	Lambertian	-
Lambertian non-uniform albedo	PS,	Textured	Bright	Lambertian	-
Non-Lambertian PS, uniform albedo		Textureless	Bright	Mixed	-
Non-Lambertian PS, non-uniform albedo		Textured	Bright	Mixed	-

Table 3.5: Working conditions of typical Photometric Stereo algorithms.

3.2.4 Structured Light

For stereo correspondence based methods, actively projected patterns have to be used for the lack of surface texture. Since the surface is diffuse, there is no specular reflection to cause severe noise. Refer to Table 3.6 for the working condition of SL.

High albedo

The surface should be sufficiently light, otherwise, the lit areas would be erroneously decoded as unlit, causing correspondence errors.

Diffuse reflectance

Traditional Structured Light techniques don't cope well with highly specular surface since the pattern is undistinguishable at areas with specular highlights.

Concex

Active methods that assumes a local reflectance model such as most PS can work more reliably for surfaces without casting shadow. Thus the concave surfaces pose a great challenge for this technique since the indensity is no longer determined solely by the light source, surface normal and viewing direction.

Technique	Texture	Lightness	Reflectance	Roughness	Concavity
Structured Light	-	Bright	Lambertian	-	Convex

Table 3.6: Working conditions of typical Structured Light algorithms.

3.2.5 Visual Hull

Visual Hull algorithms don't rely on material properties as long as the foreground of the image can be reliably segmented, thus is applicable for all visual properties. However, it fails to carve out the concavities in the object, thus is unsuitable to high concave objects.

Technique	Texture	Lightness	Reflectance	Roughness	Concavity
VH	-	-	-	-	Convex

Table 3.7: Working conditions of typical Photometric Stereo algorithms.

3.3 Summary

Our taxonomy focuses on the visual cues detected in images, which is utilized by various techniques. Conceptualize these visual cues as dimension of the 3D reconstruction problem, we have an abstraction which allow us to think of algorithms as volumes within a n -dimensional problem space. Existing algorithms can be introduced into this framework based on the main visual cue used for reconstruction. Instances where these algorithms have been reported as supporting other forms of variation have been outlined, providing an initial mapping of the space that is summarized below in Table 3.8.

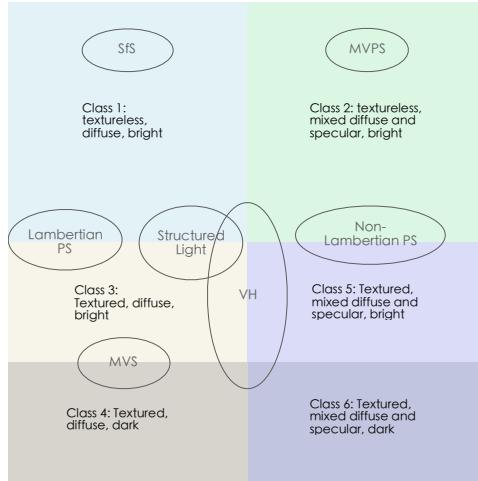


Figure 3.3: Six classes of objects of interest, and the algorithms that could work reliably for these classes.

Class #	Algorithms
1	Horn [28], Woodham [63], Hayakawa [25], Belhumeur [11], Alldrin [8]
2	Coleman [17], Barsky [10], Schluns [50], Sato [49], Mallick [39], Alldrain [6], Goldman [24], Silver [54], Hertzmann [26], Zickler [65]
3	Woodham [63], Hayakawa [25], Belhumeur [11], Alldrin [8], Furukawa [20], Goesele [23], Vogiatzis [61], Hernández [18], Faugeras [19], Inokuchi [30]
4	Furukawa [20], Goesele [23], Vogiatzis [61], Hernández [18], Faugeras [19]
5	Coleman [17], Barsky [10], Schluns [50], Sato [49], Mallick [39], Alldrain [6], Goldman [24], Silver [54], Hertzmann [26], Zickler [65]
6	Szeliski [55], Matusik [42], Tarini [57]

Table 3.8: Algorithm classification based on the new taxonomy

Chapter 4

A Description of 3D Reconstruction

In Chapter 3, we introduce a taxonomy of 3D reconstruction which maps algorithms according to the visual/geometric characteristics of the object. However, without a formal ‘language’, i.e., a model and representations, the mapping from an algorithm to a column of the problem space would be largely empirical. Without a formal definition of the problem space, expressing the condition that an algorithm works well is not a well defined problem.

Computer vision problems require, among other factors, a model of the problem domain [38]. The relevant properties of the elements in the domain must be characterized and their relations must be analyzed. Representations describe the object properties selected by the model to facilitate solution of the problem. For instance, surface orientation is a geometric model, and surface normal or curvature are representations of this model. More specifically in the case of reconstruction problem, we first need to establish a model that defines the problem domain, then properties of the objects and surfaces are investigated to see their influence on the problem domain, which serves as the representation of the reconstruction problem.

In this chapter, we attempt to extend this taxonomy by providing a description of 3D reconstruction problem which allows for a well defined specification of the visual cues surrounding the problem and of the range of the desired solution, abstracting away from the functional specification of *how* to estimate a recon-

struction. We first propose a formal definition of the 3D reconstruction problem in Section 4.1. Section 4.2 discusses various key *aspects* of the problem space that are crucial to describe the appearance of the object. Section 4.3 the actual concrete representations of the proposed model. Section 4.4 provides examples of expressing common 3D reconstruction problems using the proposed model and representations. These four layers: Definition, Representation, Model, and Expression represent our framework of accessible 3D reconstruction.

4.1 Definition

We first give the definitions of some basic concepts, which include general computer vision concepts such as scene, camera, and image. We then define some other notions that are close related to the reconstruction problem before a formal definition is introduced. We then provide some reasonable approximations for a more practical definition.

4.1.1 Basic notations

We use the following notations: $\{C_n\}_{n=0}^{N-1}$ represents the camera set, which include both the intrinsic and extrinsic parameters; $\{I_n\}_{n=0}^{N-1}$ represents the set of all images; $\{L_n\}_{n=0}^{N-1}$ represents the set of light sources.

Definition 1 (Scene) The scene S is the four-dimensional joint spatio-temporal target of interest.

Definition 2 (Image) The 2D observation of the 3D scene S on the image plane of camera C_i at time t_0 , which is modelled as: $I_i = T(S, C_i, L_0, t_0)$, or on the image plane of C_0 under the light source L_i at time t_i , $I_i = T(S, C_0, L_i, t_i)$, where T is the geometric/radiometric transformation.

The transformation T can be a geometric one which determines the 2D coordinates of a 3D point, or a radiometric one which determines the intensity/irradiance information from the information of illumination, viewing direction and surface orientation, or both.

4.1.2 Segment and Scell

Definition 3 (Segment) A segment is a distinct region in the image.

Segment is the most basic element in the image, can be considered as a generalized pixel. For instance, a segment can be a pixel, a window area, an edge, a contour, or a region of arbitrary size and shape.

Definition 4 (Cue) cues are the visual or geometric characteristics of the segments seg that can be used for reconstruction, denoted as $cue(seg)$.

For instance, the cue can be texture within a window area, intensity/colour value of a pixel, or object contour, etc.

Definition (scell) A Scell (scene element) is a volume in the scene which corresponds to at least one segment.

A scell can be considered as a generalization of a voxel. However, a scell is not necessarily distinct since

Definition (Property) Properties are the visual and geometric characteristics of the Scell sc , which would influence the cues of a segment, denoted as $prop(sc)$.

The property of the scell can be the 3D position or orientation information, visual texture, reflectance, surface orientation, roughness, convexity, etc.

The relation between the notions define above is shown in Figure ??.

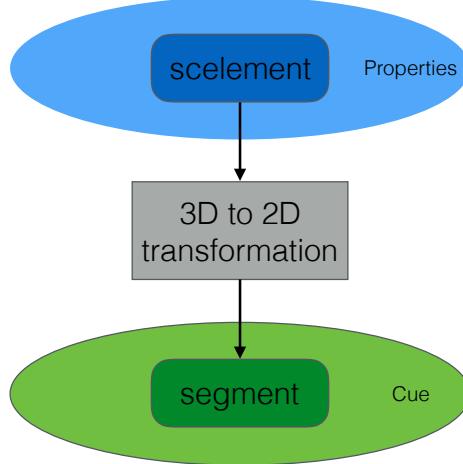


Figure 4.1: Relation between a scell and a segment

4.1.3 Photo-consistency

Every photograph of a 3D scene taken from a camera C_i partitions the set of all possible shape-radiance scene descriptions into two families, those that reproduce the photograph and those that do not. We characterize this constraint for a given shape and a given radiance assignment by the notion of *photo-consistency*.

Definition (Photo-consistency criterion) The photo-consistency criterion checks whether the properties of a scell sc can produce the cues observed in the corresponding segment seg .

$$consist(prop(sc), cue(seg)) = 1 \Rightarrow \text{photo consistent}$$

$$consist(prop(sc), cue(seg)) = 0 \Rightarrow \text{not photo consistent}$$

Definition (Segment photo-consistency) Let S be the scene. A scell $s \in S$ that is visible from C_i is photo-consistent with the image I_i if and only if the photo-consistency check is true.

Definition (Image photo-consistency) A scene S is image photo-consistent with image I_i if any scell $\forall s \in S$ visible from the camera C_i is segment photo-consistent with this image.

Definition (Scene photo-consistency) A scene S is scene photo-consistent with a set of images $\{I_n\}_{n=0}^{N-1}$ if it's image photo-consistency with each image $I_i \in \{I_n\}_{n=0}^{N-1}$ in the set.

4.1.4 Formal Definition

Definition (3D reconstruction) Given a set of images $\{I_n\}_{n=0}^{N-1}$ captured by cameras $\{C_n\}_{n=0}^{N-1}$, or under a set of light sources $\{L_n\}_{n=0}^{N-1}$, find a set of scells $\{sc_n\}_{n=0}^{M-1}$ such that any scell is photo-consistent with the image set $\{I_n\}_{n=0}^{N-1}$, i.e., $\forall sc_i \in \{sc_n\}_{n=0}^{M-1}$, we have $consist(prop(sc_i), cue(seg_{(i,n)})) = 1$.

where $seg_{(i,n)}$ is the corresponding segment of sc_i in camera C_n . Alternatively, 3D reconstruction tries to find a set of scelments $\{sc_n\}_{n=0}^{M-1}$ that are scene photo-consistent with the image set $\{I_n\}_{n=0}^{N-1}$

4.1.5 Applied Definition

While the definition presented above gives a formal definition of the problem of 3D reconstruction, it is not necessarily applicable in a practical setting. We extend in this section this formal definition to an approximate, but more applied version.

Definition (Photo-consistency score) The photo-consistency score measures the similarity between a scell sc and the corresponding segment seg .

$$\begin{aligned} consist(prop(sc), cue(seg)) &= x, x \in [0, 1] \\ consist(prop(sc), cue(seg)) &= 1 \Rightarrow \text{photo consistent} \\ consist(prop(sc), cue(seg)) &= 0 \Rightarrow \text{not photo consistent} \end{aligned}$$

Definition (Applied photo-consistency check) A scell sc and a segment seg are considered photo-consistent if the photo-consistency score is above a pre-defined threshold ε .

$$consist(prop(sc), cue(seg)) > \varepsilon$$

Some more definitions $\sum_{n \in I'} consist(prop(sc_i), cue(seg_{(i,n)}))$

Definition (Applied 3D Reconstruction) Given a set of images $\{I_n\}_{n=0}^{N-1}$ captured by cameras $\{C_n\}_{n=0}^{N-1}$, or under a set of light sources $\{L_n\}_{n=0}^{N-1}$, find a set of scells $\{sc_n\}_{n=0}^{M-1}$ such that the photo-consistency score between the set of scells and their corresponding segments $\{seg_{(i,n)}\}_{i=0, j=0}^{M-1, N-1}$ are maximized.

$$\text{maximize} \quad \sum_{n=0}^{N-1} \sum_{i=0}^{M-1} consist(prop(sc_i), cue(seg_{(i,n)}))$$

4.2 Model

Models and representations are fundamental for vision problem solving. Models select characteristic properties of an object, and representation describe object properties selected by the model to facilitate solution of a class of problem. A model facilitates the representation of aspects of reality useful in a particular problem domain [15]. For instance, surface orientation is one component of surface geometry model, and the corresponding representation can be surface normal or

curvature; another example is: colour is a component of material model, and RGB space is the corresponding representation of the colour.

We select the subset of the properties used for object taxonomy in Chapter 3 as the main components of our model. The model consisting of the key properties are shown in Table 4.2.

Property	Texture	Lightness	Reflectance	Roughness	Concavity
----------	---------	-----------	-------------	-----------	-----------

Table 4.1: Model of the 3D reconstruction problem: properties

In addition to the properties, there are requirements that can be imposed to the final reconstruction result. These requirements include but not exclude:

Requirement	Accuracy-first	Completeness-first	Orientation-first	Roughness	Concavity
-------------	----------------	--------------------	-------------------	-----------	-----------

Table 4.2: Model of the 3D reconstruction problem: requirements

4.3 Representation

Based on the proposed definitions and model of 3D reconstruction problem, we need to further define the representations so that 3D reconstruction problem can be expressed using the proposed model. Now we need to turn to how to represent the properties used in the proposed model, and these factors impact the corresponding properties.

4.3.1 Texture

Texture is one of the most important cues for many computer vision algorithms. It is generally divided into two categories, namely *tactile* and *visual* textures. Tactile textures refer to the immediate tangible feel of a surface whereas visual textures refer to the visual impression that textures produce to human observer, which are related to local spatial variations of simple stimuli like colour, orientation and intensity in an image. We focus only on visual textures as it's the most widely used ones in the stereo vision research, thus the term ‘texture’ thereafter is exclusively referred to ‘visual texture’ unless mentioned otherwise.

Although texture is an important component in computer vision, there is no precise definition of the notion texture. The main reason is that natural textures often exhibit different yet contradicting properties, such as regularity versus randomness, uniformity versus distortion, which can hardly be described in a unified manner.

There are various properties that make the texture distinguishable: scale/size-/granularity, orientation, homogeneity, randomness, and etc. However, due to the diverse and complexity of natural textures, it's a challenging task to map from these semantic meanings to the precise properties of a synthetic texture. The stereo vision community often take a simplified approach, classifying them into two categories: regular and stochastic ones by their degree of randomness. A regular texture is formed by regular tiling of easily identifiable elements (texels) organized into strong periodic patterns. A stochastic texture exhibits less noticeable elements and display rather random patterns. Most of the real world texture are mixtures of these two categories. We adopt another simplification and consider *texture coverage*, which is the ratio of the surface that is textured. Stereo vision, in theory, attempts to find the correspondences based on the ‘distinctiveness’ of the texture. Therefore, as long as the surface is covered by distinctive texture, it make little difference what the basic building texture element is.

4.3.2 Lightness

When light strikes a surface, it may be reflected, transmitted, absorbed, or scattered; usually, a combination of these effects occur. The intensity/colour information received by the sensor is thus determined, among other factors, the amount of light after these interaction. We consider intensity caused solely by reflection as it is the most common phenomenon and the easiest to analyse. Generally, we assume that all effects are local, thus global effects such as inter-reflection, transmission, and etc are omitted, which is called a **local interaction model**. Lightness ranges from ‘black’ to ‘white’ in the grey scale axis. Colour is a superset intensity, which takes account into the spectral composition of light. Both terms depend on illumination, surface normal, surface reflectance, and viewing direction.

In order to understand the contributing factor of pixel intensity/colour, we need

a in-depth understanding of reflection, i.e., how light is reflected off of a surface patch, and the relation between material and intensity value. The radiometric formation of an image consists of three separate process, *light-matter interaction*, *light-lens interaction*, and *light-sensor interaction*.

Definition of Radiometric Terms

Here is a list of radiometry terms, see Figure 4.2 for an illustration:

- Solid angle ($d\omega$): 3D counterpart of angle, $d\omega = \frac{dA \cos \theta_i}{R^2}$ (steradian).
- Projected solid angle ($d\Omega$): $d\Omega = \cos \theta d\omega$.
- Incident radiance ($\mathbf{L}_i(\theta_i, \phi_i)$): light flux received from the direction (θ_i, ϕ_i) on a unit surface area, unit ($\text{watt} \cdot \text{m}^{-2} \cdot \text{steradian}^{-1}$).
- Irradiance ($\mathbf{E}_i(\theta_i, \phi_i)$): light Flux (power) incident per unit surface area from all direction, $\mathbf{E}_i(\theta_i, \phi_i) = \int_{\Omega_i} L_i(\theta_i, \phi_i) d\Omega_i$ (watt/m^2).
- Surface radiance ($\mathbf{L}_r(\theta_r, \phi_r)$): light flux emmited from a unit surface area in the direction (θ_r, ϕ_r) , unit ($\text{watt} \cdot \text{m}^{-2} \cdot \text{steradian}^{-1}$).

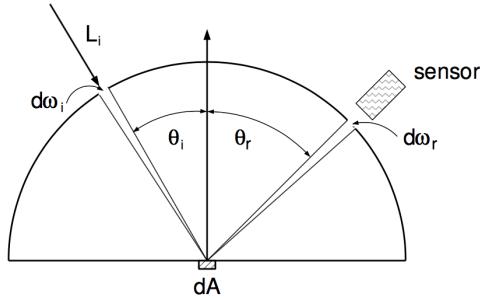


Figure 4.2: Light-matter interaction

Light-matter interaction

The relation between the incoming illumination and reflected light is model using the *bidirectional reflectance distribution function*, usually abbreviated BRDF. The BRDF is define as

Definition (BRDF) the ratio of the surface radiance $L_r(\theta_r, \phi_r)$ to the irradiance $E_i(\theta_i, \phi_i)$, i.e., $f(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{E_{surface}(\theta_i, \phi_i)}{L_{surface}(\theta_r, \phi_r)}$.

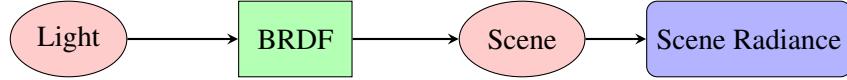


Figure 4.3: The light-matter interaction.

Diffuse Albedo or surface lightness is the proportion of incident light that is reflected by the surface. It should be noted that albedo is not an intrinsic property of a surface. Instead, for any surface, the albedo depends on the spectral and angular distributions of the incident light.

Light lens interaction

The assumption made in vision is that radiance is constant as it propagates along ray. Therefore the scene radiance is the same as the radiance hitting on the camera sensor. It can be further shown that the image irradiance received by the sensor is proportional to the scene radiance, thus the relation between *scene radiance* and *image irradiance* is linear.

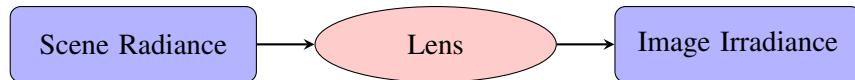


Figure 4.4: The light-lens interaction.

Light sensor interaction

The camera response function relating image irradiance at the image plane to the measured pixel intensity values is a non-linear mapping. A linear relation can be retrieved by radiometric calibration.

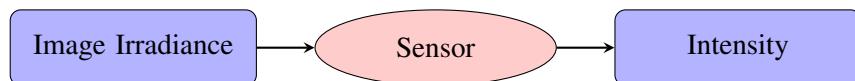


Figure 4.5: The light-sensor interaction.

In conclusion, if *light sensor* is assumed as a linear mapping as most of vision

algorithms do, or calibrated as a pre-processing step. The factor that influence the intensity is the BRDF value. There are 4 DoF for spatially-invariant BRDF, and for a special, simple case - Lambertian reflectance, the BRDF is degenerated to the diffuse *albedo*, which is the representation we adopt for intensity.

4.3.3 Reflectance

Specular surfaces reflect light in almost a single direction when the microscopic surface irregularities is small compared to light wavelength, and no subsurface scattering present [43]. Unlike diffuse reflections, which we experience the lightness and colour of an object, specular reflections carry information about the structure, intensity, and spectral content of the illumination field. In other words, specular reflection is simply image of the environment, or the illumination field, distorted by the geometry of the reflecting surface. See Figure 4.6, the image no long reflect the original colour of the surface (red), instead it shows a distorted image of the environment. A purely specular surface is a mirror. Purely specular surfaces are rare in nature. Most natural materials exhibit a mix of specular and diffuse reflection. Variations in microscopic surface geometry can cause specular reflections to be scattered, blurring the image of the environment in an amount proportional to surface roughness. We use a numeric *specularps* value to denote the proportion of specularity of the material, with 0 being completely diffuse, and 1 being completely specular or mirror light.

4.3.4 Roughness

Roughness, which is characterized as the microscopic shape characteristics of the surface, contributes to the way in which light is reflected off of a surface. A smooth surface may reflect incident light in a single direction, while a rough surface may scatter the light in various directions. We need prior knowledge of the microscopic surface irregularities, or a model of the surface to determine the reflection of incident light.

The possible surface models are divided into 2 categories: surface with exactly known profiles and surfaces with random irregularities. An exact profile may be determined by measuring the height at each point on the surface by means of a



Figure 4.6: A red specular sphere. The surface reflects light in a mirror-like way, and no diffuse reflection exist, thus the colour of the surface is no longer visible.

sensor such as the stylus profilometer. This method is cumbersome and impractical. Hence, it's more reasonable to model the surface as a random process, where it is described by a statistical distribution of either its height above a certain mean level, or its slope w.r.t its mean (macroscopic) slope. The section only discusses these second statistical approach.

Slope Distribution Model

We can think of a surface as a collection of planar micro-facets.

A large set of micro-facets constitutes an infinitesimal surface patch that has a mean surface orientation \vec{n} . Each micro-facet has its own orientation, which may deviate from the mean surface orientation by an angle α .

We will use the parameter α to represent the slope of individual facets. Surfaces can be modeled by a statistical distribution of the micro-facet slopes. If the surface is isotropic, the probability distribution of the micro-facet slopes can be assumed to be rotationally symmetric w.r.t the mean surface normal \vec{n} . Therefore, facet slopes can be described by a one-dimensional probability distribution function. For instance, the surface may be modeled by assuming a normal distribution

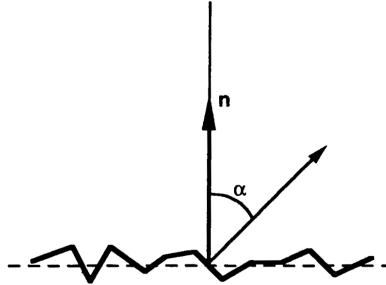


Figure 4.7: Surface Slope Distribution Model

for the facet slope α , with mean value $\bar{\alpha} = 0$ and standard deviation σ_α , and larger σ_α can be used to model rougher surfaces:

$$p_\alpha(\alpha) = \frac{1}{\sqrt{2\pi}\sigma_\alpha} e^{-\frac{\alpha^2}{2\sigma_\alpha^2}}$$

4.3.5 Concavity

Concavity can cause self-shadow or inter-reflection effect, which can severely impede the accuracy of intensity based algorithms. Since concavity is not shown in the silhouette image, methods that utilize silhouette information may also fail to reconstruct concavities. Concavity is measured by *surface curvature*.

4.4 Expression

Now with the proposed definition and representation of 3D reconstruction problem, we can express some existing 3D reconstruction algorithms under this framework. The expression of the reconstruction problem is shown in table 4.3.

object	Texture coverage	Albedo	Specular	Roughness	Concavity
Class 1	0.2	0.8	0.2	0.8	0.2
Class 2	0.2	0.8	0.5	0.2	0.2
Class 3	0.8	0.8	0.2	0.8	0.2
Class 5	0.8	0.8	0.5	0.2	0.2

Table 4.3: Expression of the reconstruction problem for the object class 1, 2, 3, 5.

Chapter 5

A Mapping of 3D Reconstruction

Most of the vision work focuses on developing algorithmic novelties, and very few investigates the rigorous conditions under which these algorithms work. Thus this knowledge is only known empirically, without a rigorous definition of the application domain or problem conditions. This section builds upon the 3D description proposed in Chapter 4, and attempts to find out the optimal algorithms under a well defined condition.

To achieve this goal, we need a dataset to evaluate the performance of each algorithm under varied conditions, which is not the goal of most online datasets. To the best of our knowledge, current existing 3D benchmarks focus on one specific class of algorithms, for example, the Middlebury dataset is targeted at MVS algorithms, and the ‘DiLiGenT’ dataset is for Photometric Stereo algorithms. This makes them only suitable to the evaluation of the within-category algorithms. There is no dataset that evaluates 3D reconstruction across differ categories, not to mention one that covers a range of properties of material and geometry and all their combinations. The reasons for the lack of such a dataset are: 1). it’s already tedious to create a real-world dataset for one specific category of algorithms, it would be even more challenging to create a dataset for a larger range of algorithms with the ground truth; 2). it’s practically impossible to change one property, e.g., , noise level, lighting configuration, material, etc.while fixing the others in order to conduct a thorough evaluation.

We propose a synthetic dataset created by phisically-based rendering software

- Blender, to evaluate the 3D reconstruction algorithms. The dataset includes a collection of images of a scene under different materials or lighting conditions. The camera/projector intrinsic and extrinsic parameters are computed directly from the configurations of the synthetic setup, and the ground truth, including the 3D model point cloud and normal map, are generated directly from Blender.

5.1 Synthetic setup

We use the physically-based rendering engine named Cycles in Blender to generate the synthetic dataset. For each technique, the configuration of the camera remains fixed. The image resolution is 1280×720 , with a focal length of 35mm or 1400pix .

For the Multi-View Stereo setup, there are five rings of cameras, of which the elevation angle is $15^\circ, 30^\circ, 45^\circ, 60^\circ, 90^\circ$. The between-angle of two neighbouring cameras is $30^\circ, 30^\circ, 45^\circ, 45^\circ$, and 360° . Thus there are in total $12 + 12 + 8 + 8 + 1 = 41$ cameras.

For the photometric stereo setup, since increasing the number of images is only important up to a point, the experimental results showed that most algorithms reaches to optimum when 15 images are used [12]. To make a balance between algorithm performance and rendering time, we use 25 light sources, which are distributed on four different rings with elevation angle of $90^\circ, 85^\circ, 60^\circ$, and 45° . The azimuth angle between two neighbouring light sources is 45° .

For the structured light setup, the baseline angle between the camera and the projector is 10° , and only one camera is used, thus only a portion of the object is visible. The resolution of the projector is 1024×768 , thus 10 Gray code patterns are needed. To counter the effect of inter-reflection, each pattern and its inverse are projected, which makes it less sensitive to scattered light.

5.2 Structure of Datasets

Due to the number of properties and number of levels for each property, it would be unrealistic to render all their combinations. For instance, if there are N properties and each is discretized into L levels, the number of different combinations is L^N , and for each combination, there are in total $41 + 25 + 42 = 108$ images to render. Therefore, we take another approach: 1). we investigate the *effete problem do-*

main which consists of only the *effective* and *dependent* properties; 2). generate synthetic images for the *effective* and *dependent* properties and all their combinations. The structure of the dataset is as follows

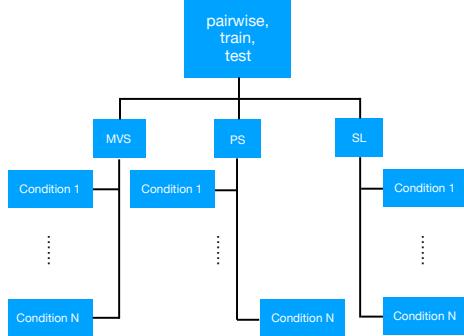


Figure 5.1: Structure of the synthetic dataset.

5.3 Evaluation metrics

We use the metric proposed in [52] to evaluate MVS and SL algorithms. More specifically, we compute the accuracy and completeness of the reconstruction. For accuracy, the distance between the points in the reconstruction R and the nearest points on ground truth G is computed, and the distance d such that $X\%$ of the points on R are within distance d of G is considered as accuracy. Thus the lower the accuracy value, the better the reconstruction result. For completeness, we compute the distance from G to R . Intuitively, points on G is not “covered” if no suitable nearest points on R found. A more practical approach computes the fraction of points of G that are within an allowable distance d of R . Note that as the reconstruction gets better, the “accuracy value” goes down, but the “accuracy” is often claimed as improved, which is contradictory at first glance. To make it more consistent to the natural context, we say the accuracy goes up when the reconstruction gets better.

For photometric stereo, the depth information is lost since only one viewpoint is used. Thus the previous metrics are not applicable. We employ another evaluation criteria that is widely adopted by the community, which is based on the statistics of angular error. For each pixel, the angular error is calculated as the

angle between the estimated and ground truth normal, i.e., $\arccos(n_g^T n)$, where n_g and n is the ground truth and estimated normals respectively. In addition to the mean angular error, we also calculate the standard deviation, minimum, maximum, median, the first quartile, and the third quartile of angular errors for each estimated normal map.

5.4 Selected methods

We have selected one representative algorithm from three major classes of algorithms presented in Chapter 3: the PMVS proposed in [20], the example-based photometric stereo proposed in [26], and the Gray code structured light technique, see Table 5.1 for a summary of the selected algorithms. The current implementation of SL projects both column and row patterns, and depth values are computed using these two kinds of patterns individually. A depth consistency checking step is performed to reject erroneous triangulations.

Technique	Texture	Albedo	Specular	Roughness
PMVS: patch-based, seed points propagation MVS.				
PMVS	High	-	Low	-
EPS: example-based Photometric Stereo				
EPS	-	High	Low	High
GSL: Gray code Structured Light technique				
GSL	-	High	Low	High

Table 5.1: Summary of the selected algorithms for the framework, and the corresponding working conditions in theory.

5.5 Baseline

A baseline algorithm that works sufficiently well under most conditions should be chosen so that it's possible to determine the performance of selected algorithm within the framework. We choose the Visual Hull technique as one of our baseline algorithms since 1) it works relatively well as long as the silhouette of the object can be reliably extracted thus is insensitive to material properties; 2). the true scene is always enclosed by the reconstruction result thus the outcome is always

Technique	Texture	Albedo	Specular	Roughness
VH: volumetric Visual Hull				
VH	-	-	-	-
LLS-PS: linear least squares Photometric Stereo.				
LLS-PS	-	High	Low	High

Table 5.2: Summary of the baseline algorithms for the framework, and the corresponding working conditions in theory.

predictable.

A simple linear least squares based Photometric Stereo (LLS-PS) is selected to evaluate Photometric Stereo algorithms. However, there is currently no such algorithm that works reasonably well under various conditions. Thus we run this baseline algorithm under the optimal condition to ensure that a best possible result is achieved. To assess the performance of the selected PS algorithm against the baseline, we compare the following characteristics of the angular error:

Measures of Central Tendency

Mean and median are both valid measures of central tendency. But as the skewness increases, mean would be dragged in the direction of the skew, thus the median is generally considered to be the best representative of the central location of the data. The more skewed the distribution, the greater the difference between the median and mean, and the greater emphasis should be placed on using the median as opposed to the mean. See Figure 5.2.

Variation

- Interquartile range
- Standard deviation

Skewness (right/positive-skewness)

The normal estimation is more insensitive to angular error when the mean and median are close. The shape can be relatively well recovered with a mean/median angular error of 10° , see Figure 5.3 (a)-(f). The normal estimation becomes

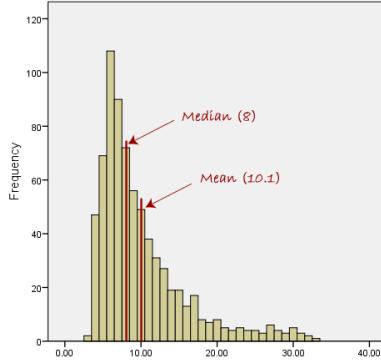


Figure 5.2: A right-skewed distribution, which is a typical graph of the angular error.

more susceptible as the difference between the mean and median increases, see Figure 5.3 (g)-(l). This can be explained as follows: when the normals are reliably estimated, the angular error should follow a Gaussian distribution. Therefore, the mean and median are close to each other. However, if normals are poorly recovering, the mean would become larger since large angular errors exist while the median would change far less since only a small amount of pixels are affected. Thus the difference between mean and median would be larger when surface normals are poorly recovered.

5.6 Effective Problem Domain

The biggest challenge in conducting a comprehensive evaluation is the large variations in shapes and material properties, which results in a problem domain that is too large to cope with. Therefore, the first step is to establish the *effective problem domain* (EPD) by finding the effective properties so that the dimension of the problem conditions would become more manageable. We conduct comprehensive experiments that evaluate the performance of the algorithms by changing two properties at a time while fixing the others. The goals are to 1). identify effective properties; 2). identify dependent properties that would impact the algorithm differently with different combinations of values.

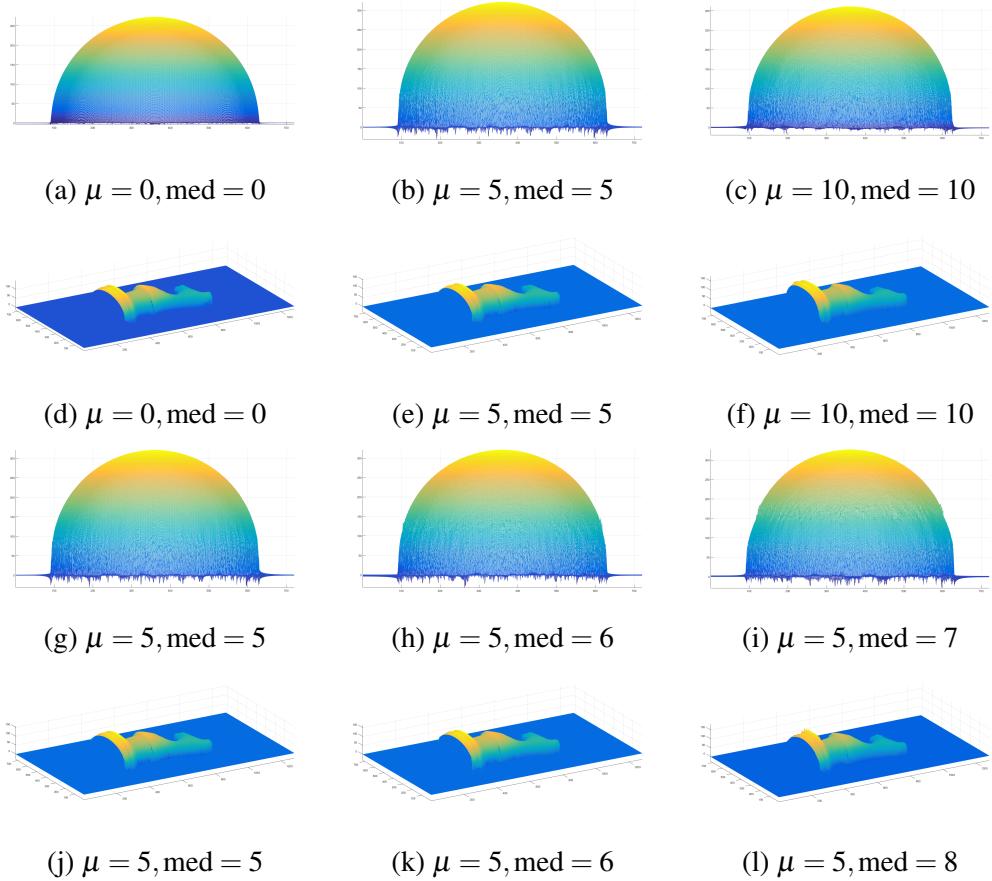


Figure 5.3: Shape estimation results with varied mean and median values.

The algorithm is less sensitive to large mean or median value while more sensitive to the difference between the mean and median value.

5.6.1 EPD of PMVS

We evaluate the performance of PMVS in terms of accuracy and completeness under varied combinations of properties, the settings of the properties are listed in Table 5.3.

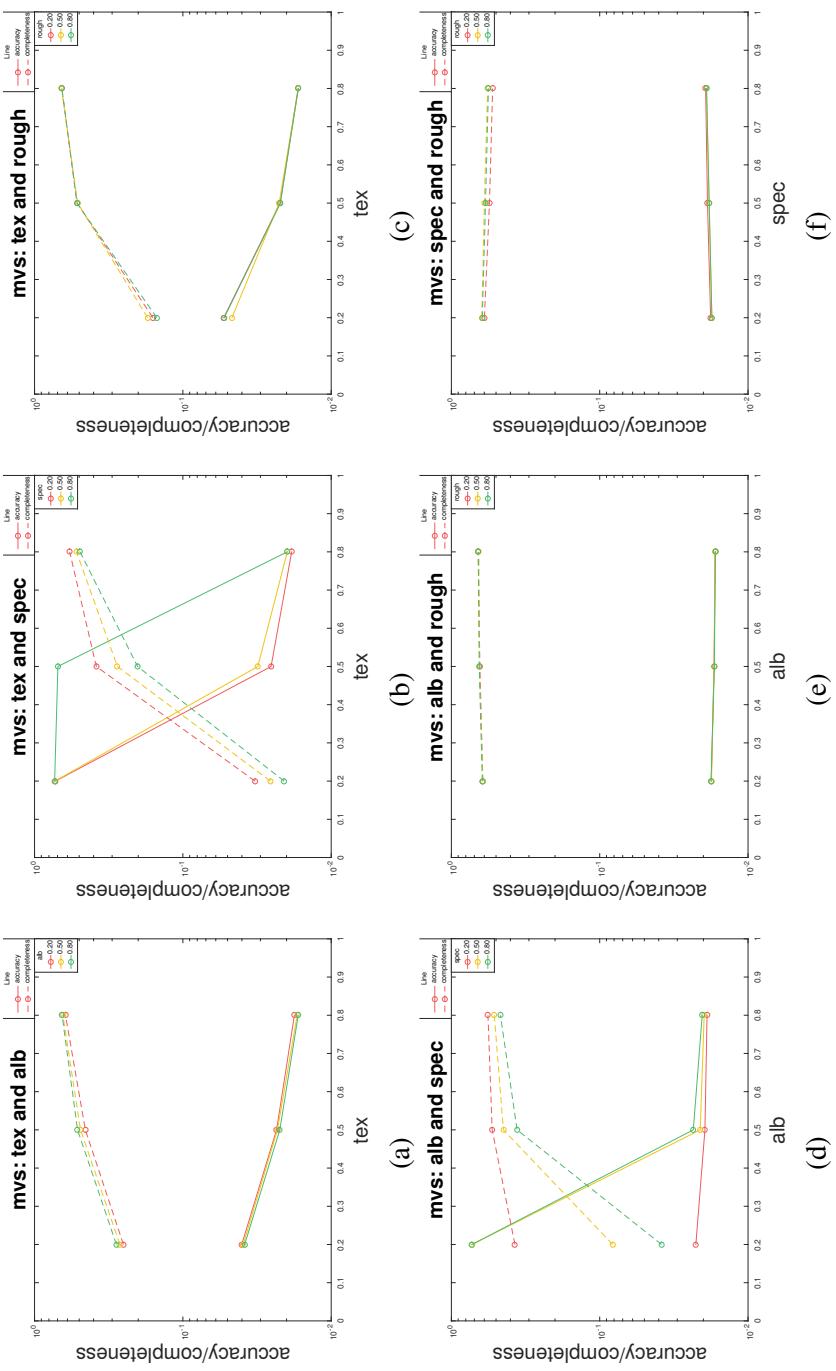


Figure 5.4: Performance of PMVS under six pairwise conditions. For instance, (a) shows the performance under changing *texture* and *albedo* values while the others are fixed. The property values are set based on settings in Table 5.3.

Group	Texture	Albedo	Specular	Roughness
(a)	[0.2, 0.8]	[0.2, 0.8]	0.0	0.0
(b)	[0.2, 0.8]	0.8	[0.2, 0.8]	0.0
(c)	[0.2, 0.8]	0.8	0.0	[0.2, 0.8]
(d)	0.8	[0.2, 0.8]	[0.2, 0.8]	0.0
(e)	0.8	[0.2, 0.8]	0.0	[0.2, 0.8]
(f)	0.8	0.8	[0.2, 0.8]	[0.2, 0.8]

Table 5.3: Problem conditions for establishing the *effective problem domain* of PMVS.

Effective and Dependent Properties

We investigate how each property affects the reconstruction in terms of accuracy and completeness.

(a) Texture and Albedo The texture has a positive effect on the reconstruction in terms of accuracy and completeness while the effect of albedo is negligible.

(b) Texture and Specular Specular has a negative effect on both the accuracy and completeness of the reconstruction. However, the level of impact varies as the texture varies, more specifically, the effect of specular is more substantial on a lower textured surface than that on a higher textured one. This could be explained as follows: the specular lobe can only be observed by cameras positioned and oriented towards the specular lobe, such as the camera V_2 shown in Figure 5.5 (a) and (c). Cameras positioned otherwise would observe the true surface, such as camera V_1 shown in Figure 5.5 (a) and (b). The algorithm would exploit the texture information provided by views like V_1 , and thus is able to reconstruct a specular surface.

(c) Texture and Roughness Roughness doesn't have a significant effect on the results.

(d) Albedo and Specular Albedo has a positive effect whereas specular a negative effect on the reconstruction. Furthermore, the effect of specular is far more substantial on a lower albedo surface than that on a higher albedo one. This can be explained as follows: according to the energy conservation law, as the specular component increases, the diffuse component decreases, thus the diffuse area becomes less discernible, see Figure 5.6 (a)-(c). Increasing the diffuse albedo can

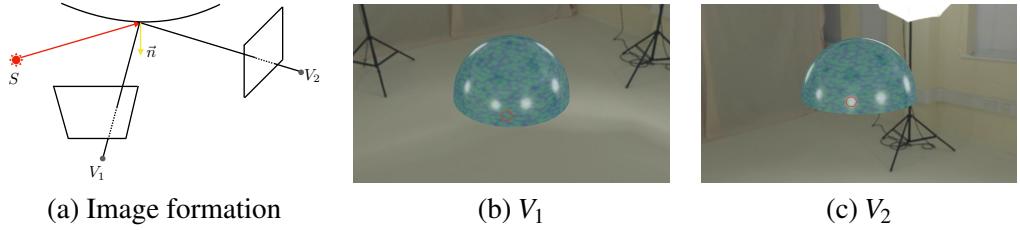


Figure 5.5: (a) shows the reflection of light off a specular surface. V_1 received the diffuse component while V_2 receives the specular component. (b), (c) shows the images observed from these two views. The specular area (red circle) observed in V_2 is visible in V_1 .

counteract the effect of specular and make the texture visible again, see Figure 5.6 (d)-(f).

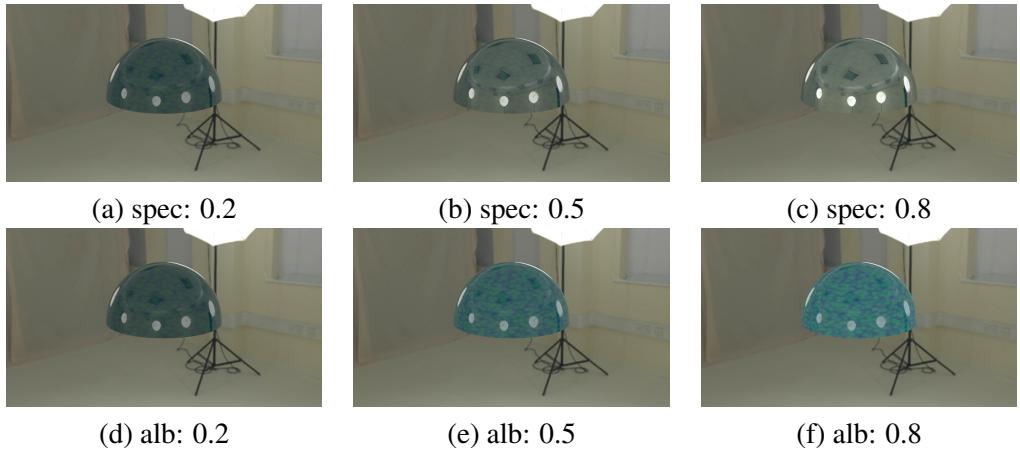


Figure 5.6: (a)-(c). The albedo is set as 0.2, (d)-(f). the specular is set as 0.2. According to energy conservation, as the specular component increases, the diffuse component decreases.

(e) Albedo and Roughness The albedo and roughness have a negligible effect on the results.

(f) Specular and Roughness Surface roughness can effectively diminishes the specular component and makes the surface appear more diffuse. Since specular has a negative impact on the reconstruction, in theory, roughness should have a positive impact on the reconstruction. However, this effect is closely related to

surface texture: surface reconstruction is always poor for low textured surfaces or good for highly textured ones regardless of the amount of specular component, as shown in Figure 5.4 (b). The result shown in Figure 5.4 (f) happens to be tested in highly textured surface, therefore achieves consistently good results. Since high specular, high roughness surfaces visually resemble low specular ones, and achieve similar reconstruction results as well, we would only consider specular, and ignore roughness for simplicity.

Summary

The effective properties of PMVS are: texture, albedo, and specular, of which texture is the most important one. Specular would deteriorate the reconstruction for lower textured and lower albedo surfaces. The effective problem domain is shown in Table 5.4.

Metric	Texture	Albedo	Specular	Roughness
Accuracy	✓	✓	✓	✗
Completeness	✓	✓	✓	✗

Table 5.4: The *effective problem domain* of PMVS in terms of accuracy and completeness.

5.6.2 EPD of EPS

We evaluate the performance of example-based PS in terms of angular error under varied combinations of properties. The statistical measures that we used include median, mean, standard deviation, first and third quartile of the angular error. We investigate two properties at a time. The settings of the properties are listed in Table 5.5.

Effective and Dependent Properties

We investigate how each property affects the reconstruction in terms of the statistics of the angular difference.

(a) Texture and Albedo Texture has no effect on angular difference while albedo has a positive effect on normal estimation.

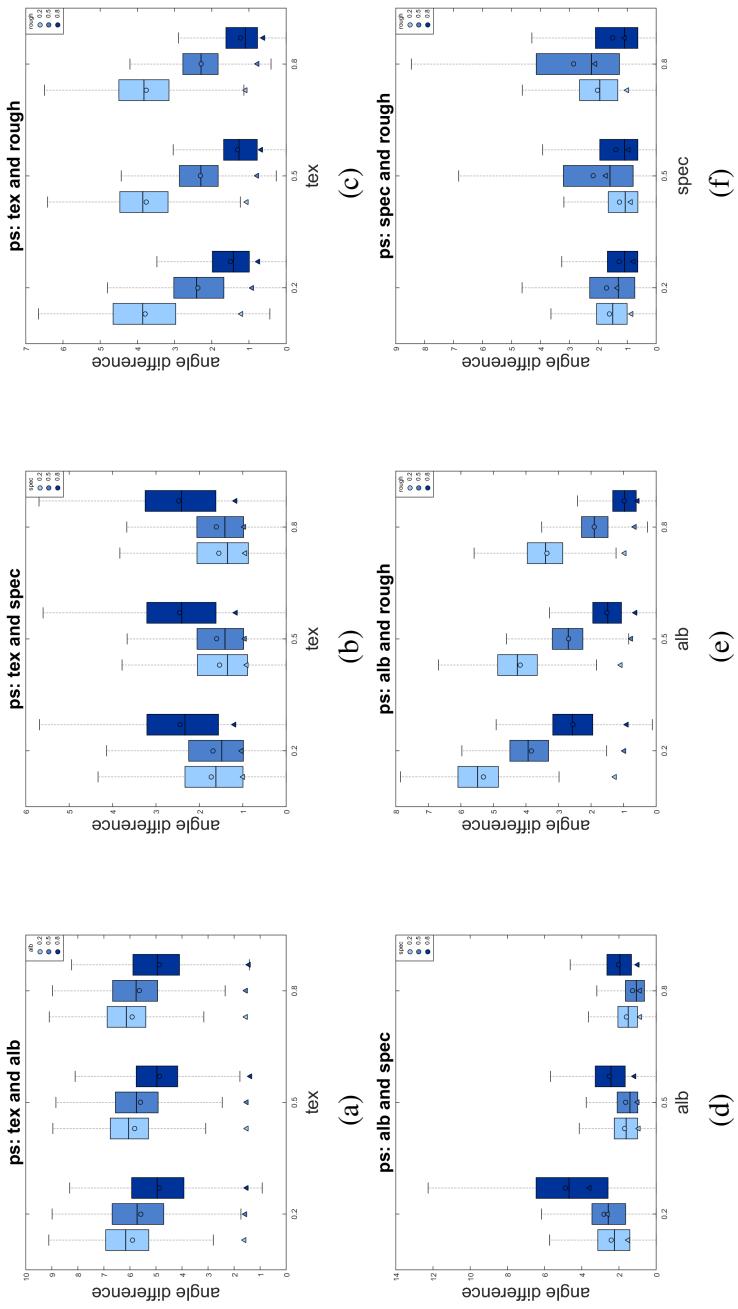


Figure 5.7: Performance of Example-based PS under six pairwise conditions. For instance, (a) shows the performance under changing *texture* and *albedo* values. The property values are assigned based on the settings in Table 5.5 (a).

Group	Texture	Albedo	Specular	Roughness
(a)	[0.2, 0.8]	[0.2, 0.8]	0.0	0.0
(b)	[0.2, 0.8]	0.8	[0.2, 0.8]	0.2
(c)	[0.2, 0.8]	0.8	0.0	[0.2, 0.8]
(d)	0.0	[0.2, 0.8]	[0.2, 0.8]	0.2
(e)	0.0	[0.2, 0.8]	0.0	[0.2, 0.8]
(f)	0.0	0.8	[0.2, 0.8]	[0.2, 0.8]

Table 5.5: Problem conditions for establishing the *effective problem domain* of EPS.

(b) Texture and Specular Texture has no effect on angular difference while specular has a negative effect on normal estimation since the difference of mean and median gets larger as the specular increases. This can be explained as follows: as the specular increases, only the specular regions exhibit erroneous normal estimation while the rest of the surface is reliably estimated, see Figure 5.8. That is why the median value exhibits far less change while the mean value increases significantly as shown in Figure 5.7 (b).

(c) Texture and Roughness Texture has no effect on angular difference while roughness has a positive effect on normal estimation.

(d) Albedo and Specular The albedo has a positive impact on normal estimation, see Figure 5.9 (a)-(c) whereas the specular a negative impact on normal estimation, see Figure 5.9 (d)-(f).

(e) Albedo and Roughness Both albedo and roughness have a positive effect on normal estimation.

(f) Specular and Roughness The specular has a negative impact on normal estimation. However, the roughness has a more complicated effect. We observed that the reconstruction becomes worse when roughness is 0.5, which is counter-intuitive at first sight. However, we argue that it's because the roughness is not strong enough to counteract the specular component, thus resulting in a smoothed and blurred specular lobe with larger area, thus leading to a worse reconstruction result. This effect is also demonstrated in the training stage, see Figure 5.10 for some visual examples.

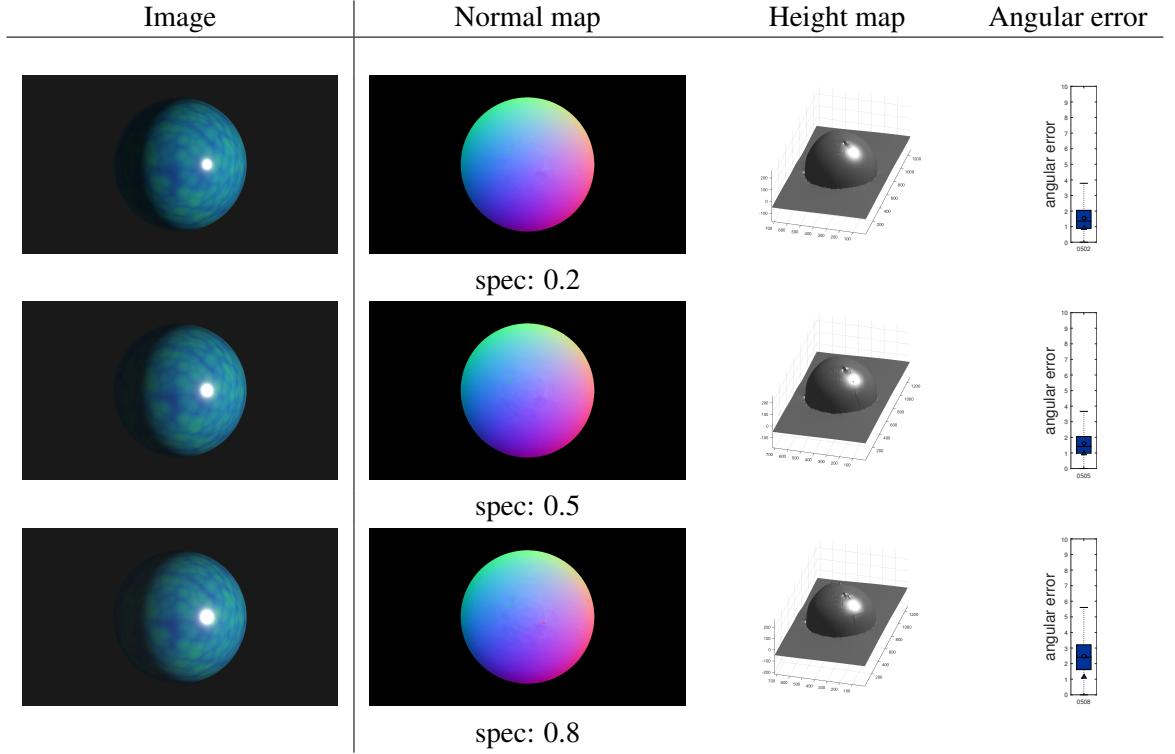


Figure 5.8: (a)-(c). The texture is set as 0.5. The estimated normal map and recovered surface becomes consistently worse as the specular level rises.

Summary

The properties that have an effect on the EPS are: albedo, specular, and roughness, as shown in Table 5.6. Therefore, we will only consider these three properties for all the forthcoming discussion of EPS.

Metric	Texture	Albedo	Specular	Roughness
Angle difference	✗	✓	✓	✓

Table 5.6: The *effective problem domain* of EPS in terms of the *angular difference*.

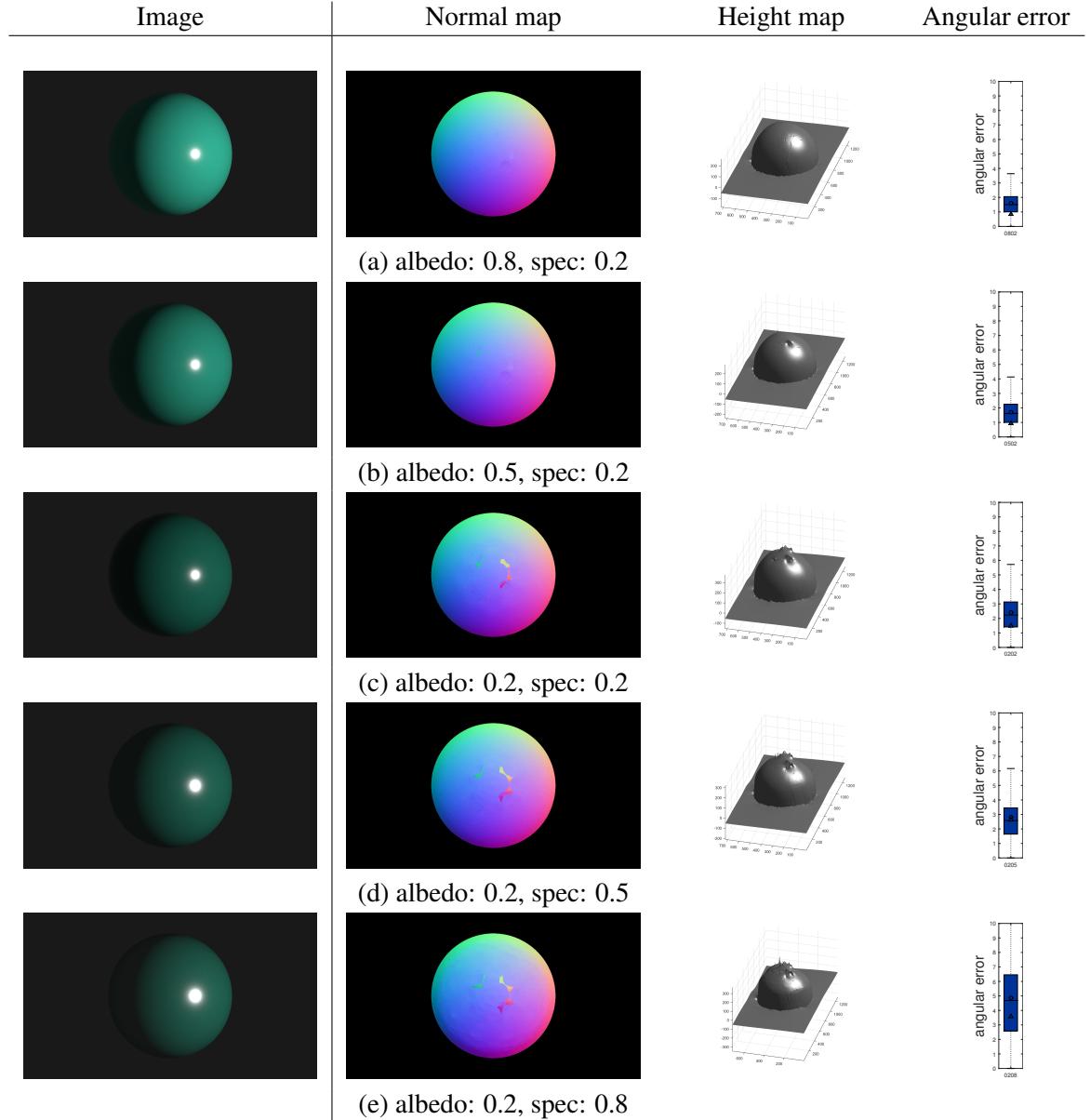


Figure 5.9: According to energy conservation, as the specular component increases, the diffuse component decreases. (a)-(c): the estimated normal map and recovered height map becomes consistently worse as the albedo decreases; (c)-(e): the estimated normal map and recovered height map becomes consistently worse as the specular increases.

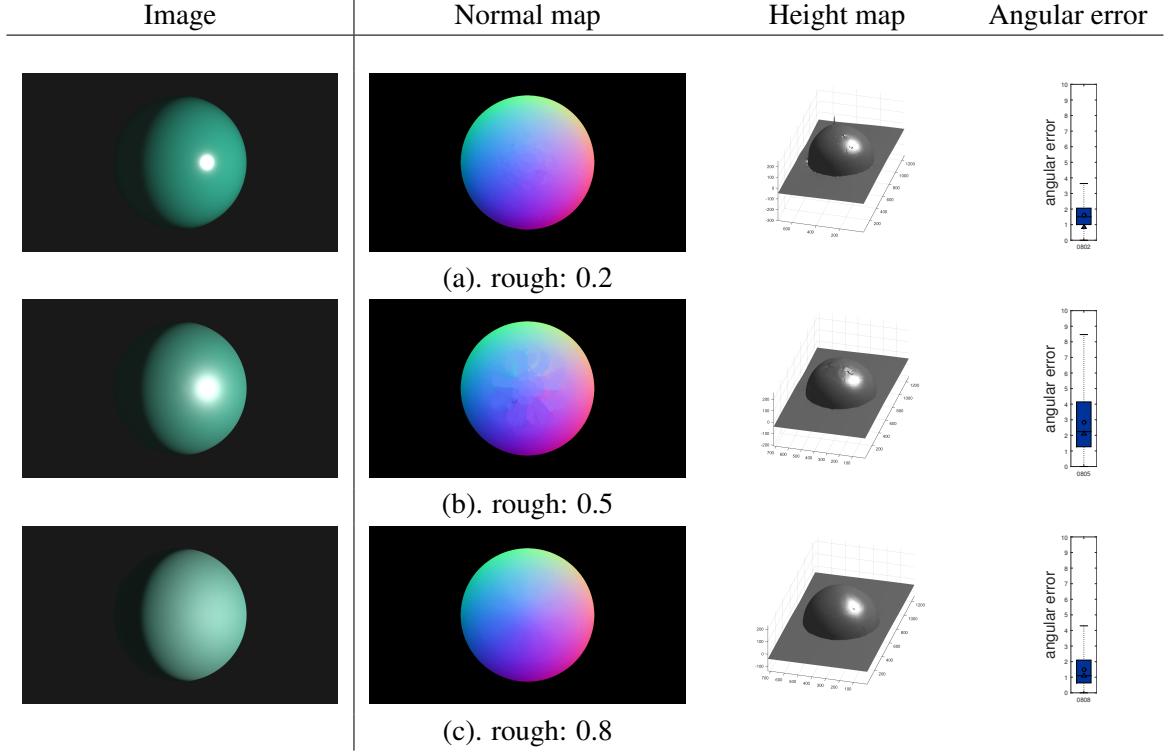


Figure 5.10: The ‘peculiar’ effect of roughness on PS. Albedo is set as 0.8, and specular is set as 0.8. (b) demonstrate that a medium level roughness would lead to worse normal estimation since it blurs the specular lobe.

5.6.3 EPD of GSL

We evaluate the performance of Gray-code SL in terms of accuracy and completeness under varied combination of properties, the settings of the properties are listed in Table 5.7.

Effective and Dependent Properties

We investigate how each property affects the reconstruction in terms of accuracy and completeness. A depth check step is performed to remove erroneous depth, thus the accuracy remain almost constant across all cases.

(a) Texture and Albedo Texture has no significant effect whereas albedo has

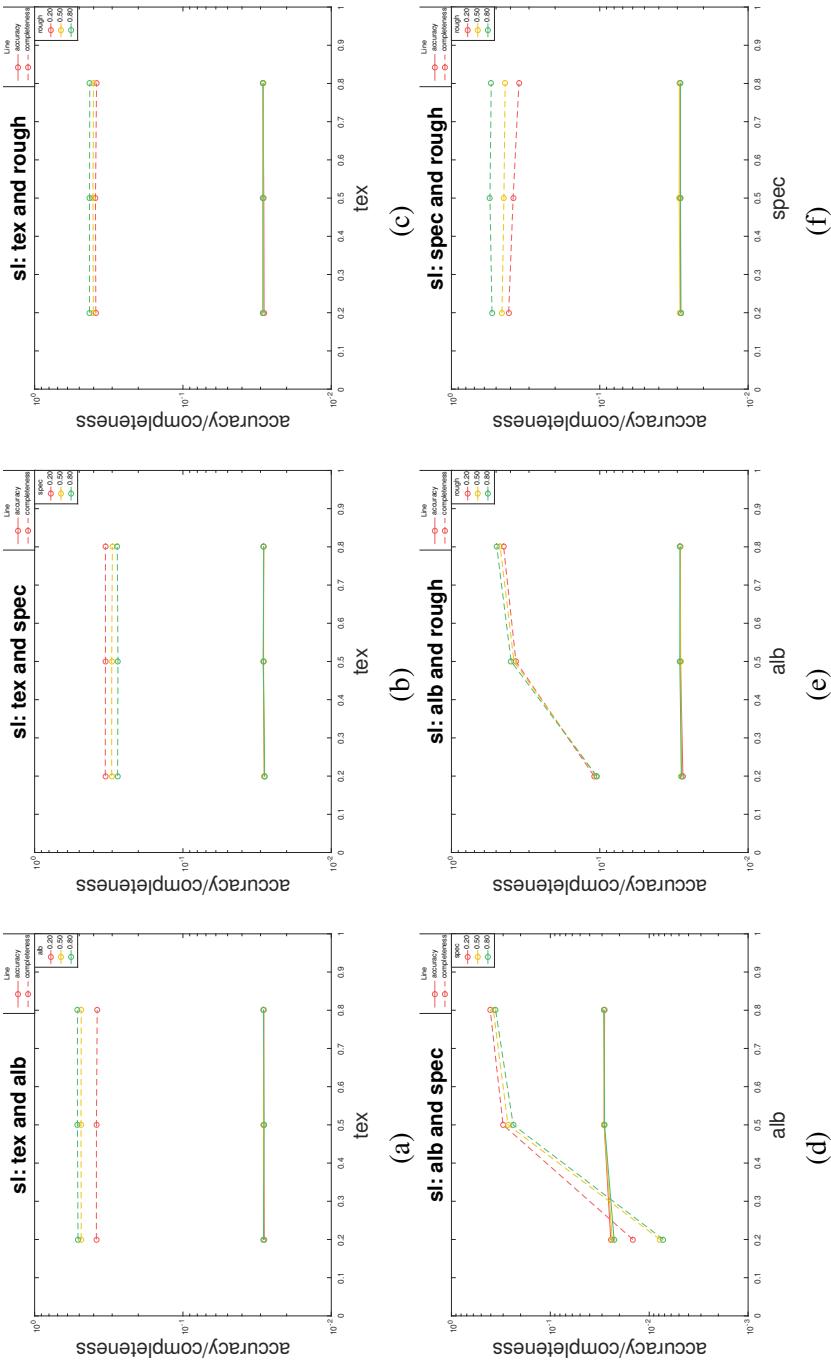


Figure 5.11: Performance of Gray-encoded SL under six pairwise conditions. For instance, (a) shows the performance under changing *texture* and *albedo* values. The property values are assigned based on settings in Table 5.7 (a).

Property	Texture	Albedo	Specular	Roughness
(a)	[0.2, 0.8]	[0.2, 0.8]	0.0	0.0
(b)	[0.2, 0.8]	0.8	[0.2, 0.8]	0.0
(c)	[0.2, 0.8]	0.8	0.0	[0.2, 0.8]
(d)	0.0	[0.2, 0.8]	[0.2, 0.8]	0.0
(e)	0.0	[0.2, 0.8]	0.0	[0.2, 0.8]
(f)	0.0	0.8	[0.2, 0.8]	[0.2, 0.8]

Table 5.7: Problem conditions for establishing the *effective problem domain* of GSL.

a positive effect on completeness. Both properties have no significant effect on accuracy.

(b) Texture and Specular Texture has no significant effect whereas specular has a negative effect on completeness. Both properties have no significant effect on accuracy.

(c) Texture and Roughness Texture has no significant effect whereas roughness has a slightly positive effect on completeness. Both properties have no significant effect on accuracy.

(d) Albedo and Specular Albedo has a positive effect, see Figure 5.12 (a)-(c) whereas specular a negative effect on completeness, see Figure 5.12 (d)-(f). The effect of specular becomes less substantial as the albedo increases, see Figure 5.11 (d). Thus we conclude that the effect of specular is most significant when the albedo is low. Neither property has a significant effect on accuracy.

(e) Albedo and Roughness Albedo has a positive effect whereas roughness has a slightly positive effect on completeness. Both properties have no significant effect on accuracy.

(f) Specular and Roughness Specular has a negative effect, see Figure 5.13 (a)-(c) whereas roughness has a positive effect on completeness, see Figure 5.13 (d)-(f). Neither property has a significant effect on accuracy.

Summary

The properties that have an effect on the GSL are: texture, albedo, specular, as shown in Table 5.8. Therefore, we will only consider these three properties for all

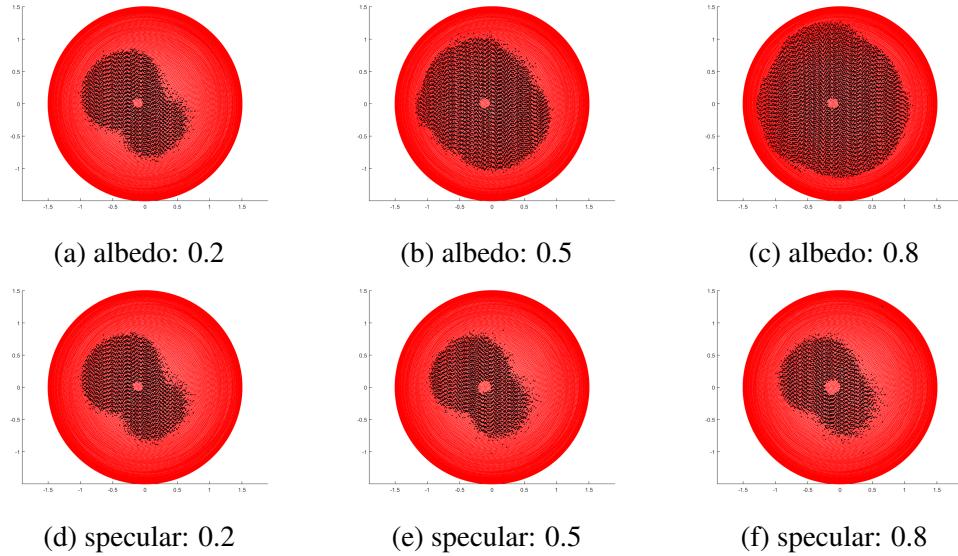


Figure 5.12: (a)-(c): the specular is set as 0.2, albedo has a positive effect on completeness; (d)-(e): the albedo is set as 0.2, specular has a negative effect on completeness.

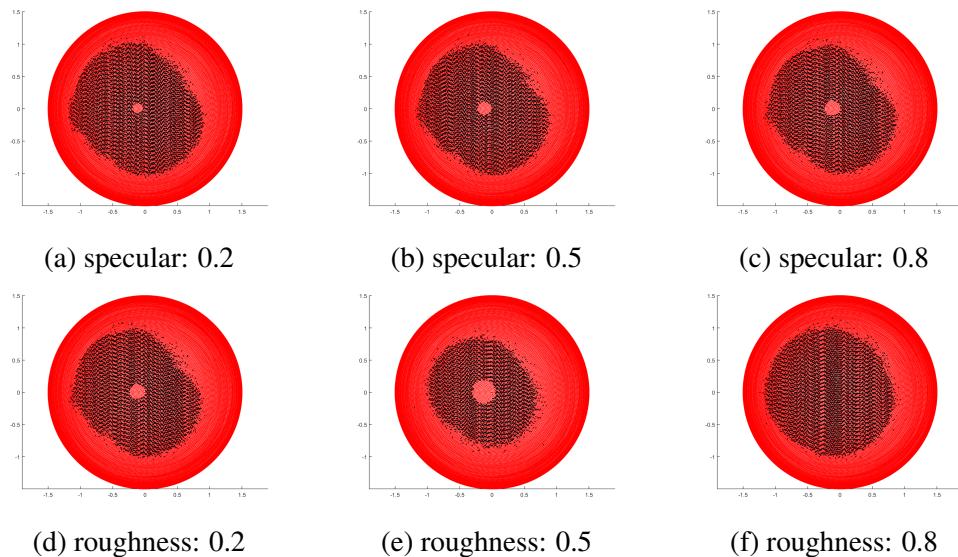


Figure 5.13: (a)-(c): the roughness is set as 0.2, specular has a negative effect on completeness; (d)-(e): the specular is set as 0.8, roughness has a positive effect on completeness.

forthcoming discussion of GSL.

Metric	Texture	Albedo	Specular	Roughness
Accuracy	✗	✗	✗	✗
Completeness	✗	✓	✓	✓

Table 5.8: The *effective problem domain* of GSL in terms of accuracy and completeness.

5.7 Mapping Construction

We generate another synthetic dataset using only the effective and dependent properties and all their combinations. Since there are three effective properties for each selected method, there are in total L^3 different combinations of property values for each technique, where L is the number of discrete values for each property, which is set as 3 representing the discrete values of 0.2, 0.5, 0.8. We present the results as follows: the 27(3^3) results are divided into three plots. Each plot illustrates the results of one fixed property P_1 , and two changing properties P_2 and P_3 . To have a better understanding of the pairwise relation between any two properties, each effective property would be chosen as P_1 once. Therefore, we end up with three groups of graphs with each consisting of three plots.

5.7.1 Mapping of PMVS

The performances of PMVS under different combinations of property values are shown in Figure 5.14, along with the performance of the baseline method. The conditions under which PMVS works well are listed in Table 5.9. We make the following observations from the training results:

(a)-(c) Texture: as the texture level increases, the completeness increases consistently, and accuracy also improves for medium/high albedo and low-medium specular surfaces.

(d)-(f) Albedo: for medium/high textured surfaces, albedo can effectively counteract the effect of specular, i.e., improve both the accuracy and completeness of the reconstruction, see the red and green lines in Figure 5.14 (d)-(f). As for low textured surface, the effect of albedo is less significant, see the blue line in Figure 5.14

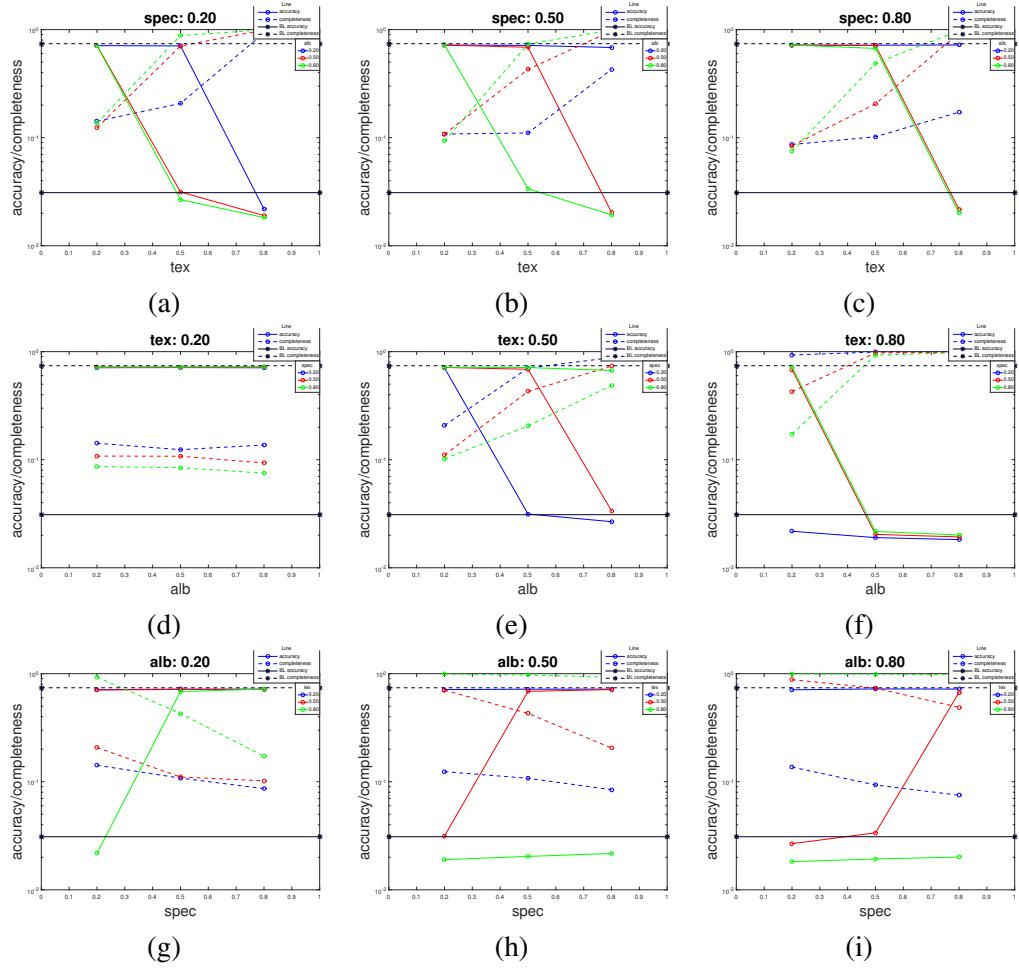


Figure 5.14: Performance of PMVS under varied conditions of changing property values. The baseline method serves as the guidelines to determine the performance of PMVS.

(d)-(f).

(g)-(i) Specular: the effect of specular is closely related to albedo: surface with higher albedo is able to endure higher specular component, see the performance difference between the red ($alb = 0.5$) and green ($alb = 0.8$) line in Figure ?? (g) - (i). For low/medium textured surface, Specular has a consistently negative impact for low/medium textured surface whereas the effect is almost neutralized for highly textured surfaces. This aligns with previous observation illustrated in Figure 5.5.

We could derive the problem conditions that PMVS could reliably work on from the training results. Those conditions are listed in Table 5.9.

Metric	Texture	Albedo	Specular	Roughness
Accuracy	0.5	0.5	0.2	-
	0.5	0.8	0.2	-
	0.8	0.2	0.2	-
	0.8	0.5	0.2	-
	0.8	0.8	0.2	-
	0.8	0.5	0.5	-
	0.8	0.8	0.5	-
	0.8	0.5	0.8	-
	0.8	0.8	0.8	-
Completeness	0.5	0.5	0.2	-
	0.5	0.8	0.2	-
	0.5	0.8	0.5	-
	0.8	0.2	0.2	-
	0.8	0.5	0.2	-
	0.8	0.8	0.2	-
	0.8	0.5	0.5	-
	0.8	0.8	0.5	-
	0.8	0.5	0.8	-
	0.8	0.8	0.8	-

Table 5.9: The condition matrix of PMVS in terms of the two metrics *accuracy* and *completeness*.

5.7.2 Mapping of EPS

The performances of example-based PS under difference combinations of properties are shown in Figure 5.15, along with the result of the baseline method. The conditions under which example-based PS works well are listed in Table 5.10. We make the following observations from the training results:

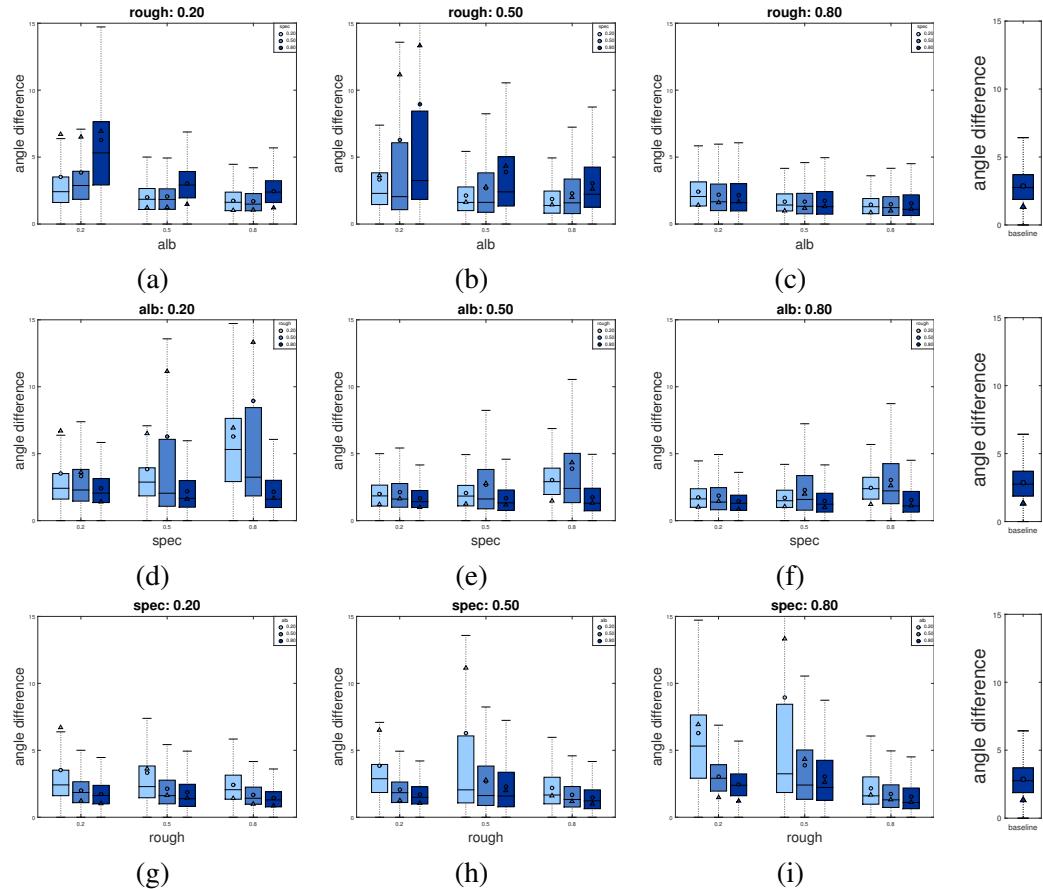


Figure 5.15: Performance of EPS under varied conditions of changing property values. Varied statistical measures of angular error are compared to the baseline method to determine the performance of EPS.

(a)-(c) Albedo: albedo has a consistently positive effect on the reconstruction.

(d)-(f) Specular: specular has a consistently negative impact on the normal estimation, which is manifested by the increasing variation represented by interquartile

range and standard deviation.

(g)-(i) Roughness: roughness has a more complicated effect on reconstruction as illustrated in Figure 5.10, i.e., medium roughness would blur the specular area, and lead to worse normal estimation and shape recovery.

We could derive the problem conditions that EPS could reliably work on from the training results. Those conditions are in Table 5.10.

Metric	Texture	Albedo	Specular	Roughness
Angle difference	-	0.2	0.2	0.8
	-	0.2	0.5	0.8
	-	0.2	0.8	0.8
	-	0.5	0.2	0.8
	-	0.5	0.5	0.8
	-	0.5	0.8	0.8
	-	0.8	0.2	0.2
	-	0.8	0.2	0.8
	-	0.8	0.5	0.2
	-	0.8	0.5	0.8
	-	0.8	0.8	0.2
	-	0.8	0.8	0.8

Table 5.10: The condition matrix of example-based PS in terms of the metric *angular error*.

5.7.3 Mapping of GSL

The performances of Gray-code SL under different combinations of property values are shown in Figure 5.16, along with the result of the baseline method. Only a portion of the scene is visible since there is only one camera, and the percent of visible surface varies from object to object. This value can be approximated by the completeness value obtained under the optimal reconstruction condition. In this case, we claim that the a completeness of 80% of that of the baseline is acceptable. The conditions under which GSL works well are listed in Table 5.11. We make the following observations:

(a)-(i): the accuracy remains almost fixed, thus irrelevant to all properties;

(a)-(c) Albedo: albedo has a consistently positive effect on the completeness of the

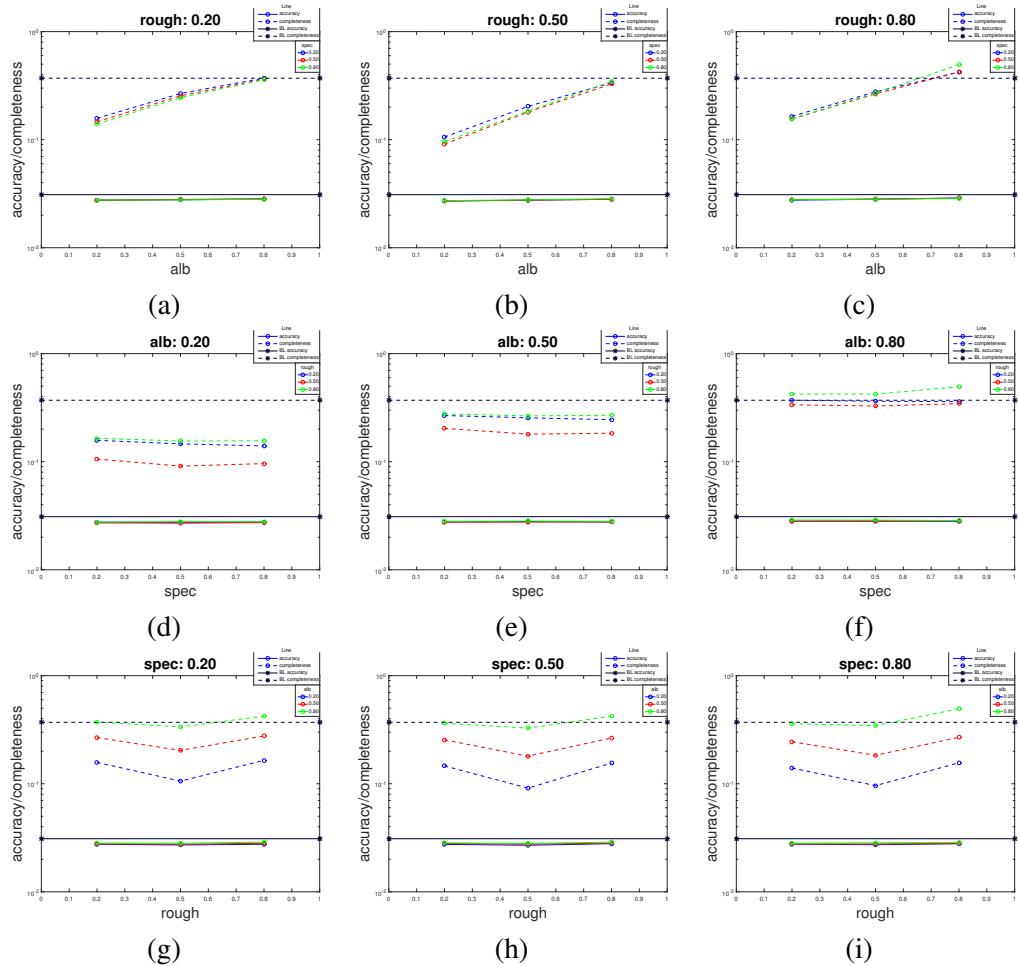


Figure 5.16: Performance of GSL under varied conditions of changing property values. The baseline method serves as the guidelines to determine the performance of GSL.

reconstruction.

(d)-(f) Specular: specular has a negative effect on completeness as shown in Figure 5.12 (d)-(f). However, we did notice in some case, the completeness of the reconstruction improves as the specular level increases. There are two contributing factors to completeness: 1). specular would decrease the completeness since the pattern can no longer be decoded in the glossy area, thus causing incomplete reconstruction; 2). large roughness would spread the specular lobe into a larger area, leading to a brighter surface, thus increase the completeness of the reconstruction. These two contradicting factors would together determine the completeness of the reconstruction. Thus if the first factor is more substantial, the completeness would decrease whereas if the second one is more significant, the completeness would increase.

(g)-(i) Roughness: roughness has similar effect as that we found on EPS, i.e., medium roughness would blur the specular lobe to a larger area thus causing larger holes in the reconstruction, see Figure 5.10 (d)-(f). Large roughness would effectively counteract the effect of specular, thus improve the completeness of the reconstruction.

We could derive the problem conditions under which GSL could reliably work by considering both the quantitative and qualitative results. Those conditions are listed in Table 5.11.

Metric	Texture	Albedo	Specular	Roughness
Accuracy	-	-	-	-
Completeness	-	0.8	0.2	0.2
	-	0.8	0.5	0.2
	-	0.8	0.8	0.2
	-	0.8	0.2	0.8
	-	0.8	0.5	0.8
	-	0.8	0.8	0.8

Table 5.11: The condition matrix of Gray-code SL in terms of the two metrics *accuracy* and *completeness*.

5.8 Summary

The development of the mapping is an on-going process. For instance, we can include more quantitative metrics such as colour accuracy, ‘ghost reconstruction’, etc. In order to make the mapping applicable to objects with more complex shapes, we need to consider more sophisticated geometric properties besides roughness, such as concavity, depth-discontinuity, occlusion, etc. Furthermore, the incorporation of more algorithms to another way to make sure that the problem space is well covered.

Chapter 6

An Interpretation of 3D Reconstruction

So far, we proposed a well-defined problem space for 3D reconstruction problem and a precise mapping from the problem space to the algorithm space. We need to validate that the derived mapping can be reliably applied to object with a different shape, and demonstrate the usage of the framework, i.e., the interpretability from the problem centric description to a reliable reconstruction result must be shown.

However, such an evaluation faces several challenges: 1). the mapping doesn't pose very stringent constraints on the types of material and geometry, however an exhaustive evaluation would require a vast amount of objects to reach to a solid conclusion, which is not a practical approach; 2).

Although only three algorithms and two baseline methods are implemented within the current framework, all of which are the top performers in the corresponding field, thus are sufficient to validate the framework's ability to translate the descriptive model into a reconstruction. The integration of new algorithms requires that they be evaluated with the same problem conditions presented in Chapter 5, thus allowing researchers to contribute novel algorithms to the framework once they become available.

Section 6.1 gives a the roadmap of our evaluation which is centered around two key evaluation questions: extensiveness of the mapping, and interpretability of the algorithm-free framework. Section 6.3 investigates cases under which the mapping

can be reliably applied to other objects. Section 6.5 presents real-world use cases of the framework, where a satisfactory reconstruction result is return given the correct description of the object.

6.1 Evaluation Methodology

This section formulates the methodology of evaluation. We start with the objective, which gives a brief introduction of what needs to be evaluated. Then two key evaluation questions are proposed, with evaluation steps, criteria and expected outcomes to determine if the evaluation is successful.

6.1.1 Objective

This evaluation intends to validate that 1). the mapping from Chapter 5 can be extended to objects with different shapes, and demonstrate cases where it succeeds and fails; 2). demonstrate the real-world use cases of the proposed framework. For the first goal, objects with varied degrees of shape changes are used, and the corresponding results are compared to the mapping. We attempt to demonstrate if the mapping, to some extent, is invariant to the changes of shape, and when would it fail to hold. For the second goal, we use real-world objects to demonstrate that the framework can return a satisfactory result when provided with a correct description.

6.1.2 Key Evaluation Questions and Steps

The evaluation attempts to 1) *prove that the mapping can be extended to other objects with different geometries*; 2). *demonstrate that the framework can return a satisfactory reconstruction result given a correct description*.

1. Does the mapping work for objects with a different shape?

We first need to prove that the mapping derived in Chapter 5 is applicable to objects with different shapes. However, the variations of geometry is too vast and complicated to model, it wouldn't be possible to consider all these conditions. Thus we focus on one geometric property that in theory could have an impact on the mapping, which is the concavity of the surface. We use three synthetic objects with

varied degrees of concavity, and verify if the mapping is applicable under those circumstances, and when it would succeed or fail. We use synthetic data to verify the mapping since it would not be practical to change material properties using real world objects. The evaluation steps include:

Data generation: the synthetic data is generated in the Blender using the same setups presented in Chapter 5. We consider the four property settings representing four major classes of real-world objects from Chapter 3.

Validation of mapping: it consists of two steps: 1). the successful algorithm(s) in each case need be identified; 2). the reliable algorithms of each case should be consistent with the results of the mapping. Both quantitative and qualitative results are used to find the successful algorithms under each problem condition. To determine that a method returns a reliable result, the accuracy value should be lower than that of the baseline method while the completeness should be higher, and all the statistical measures of the angular error must be lower than those of the baseline method, including the mean, median, standard deviation, and interquatile range. The qualitative results are used to further confirm the validity of the quantitative results and give a visual sense of the . Once we have identified the successful algorithm(s) in each problem condition, we need to verify that they are consistent with the results of the mapping. If that is not the case, we need to find out how robust each algorithm is with respect to concavity changes, and how concavity change would affect the reconstruction results.

2. Can the framework return a satisfactory reconstruction given the correct description.

Given a correct description of the object, the algorithm chosen by the mapping should give a satisfactory reconstruction result of a real-world object. However, the quantitative results are not available since we don't have the groundtruth data. Therefore, visual inspection is utilized to determine the quality of reconstruction results. The framework would use the algorithm determined by the mapping for reconstruction, the result of which is then compared to the baseline algorithm to determine if the quality is acceptable. As previously mentioned, the baseline method is chosen so that it can always provide a decent reconstruction under most circumstances. The evaluation steps are presented as follows:

Data generation: the real-world data are captured using similar setups as the synthetic counterparts: for MVS, a Nikon D700 camera with [focal] lens are used; for photometric images, a Nikon D700 camera with [focal] lens, a handheld lamp, and two reference objects are used, a diffuse one and a glossy one; for structured light techniques, a Nikon 700 camera and a [??] projector are used, which is positioned with a bewteen angle of around 10° . We used nine everyday objects with varying texture, reflectance properties, and shapes with low concavity.

Output: the output of Photometric Stereo algorithms, typically a normal map, is integrated to a surface mesh for visual analysis.

Use case of framework: we need to demonstrate that: 1). the interpreter would return a reliable reconstructed model given the correct description, and a less successful one given an invalid description; 2). the algorithm chosen by the interpreter should be one of the top performers among all the algorithms implemented within the framework. The quality of the reconstruction is determined by comparing the result to the baseline method.

6.2 Parameter Setting

The first step is to estimate the settings of properties of the object. We use the BRDF explorer developed by Disney Animation [3] to visualize the rendered object with chaning properties. A try-and-see approach was taken to obtain the parameters, more specifically, the user would change the value of each property and see if the rendered result looks like the real object. A similar approach can be found in the [12] where the author also used a synthetic dataset to find the contributing factors of various Photometric Stereo algorithms.

6.3 Validation of Mapping

We first validate that the derived mapping can be applied to object with a different shape. The idea is that given the description of an arbitrary object, we use all three techniques for reconstruction, and see if the algorithm that has the best quantitative or qualitative result is consistent to the algorithm chosen by the mapping.

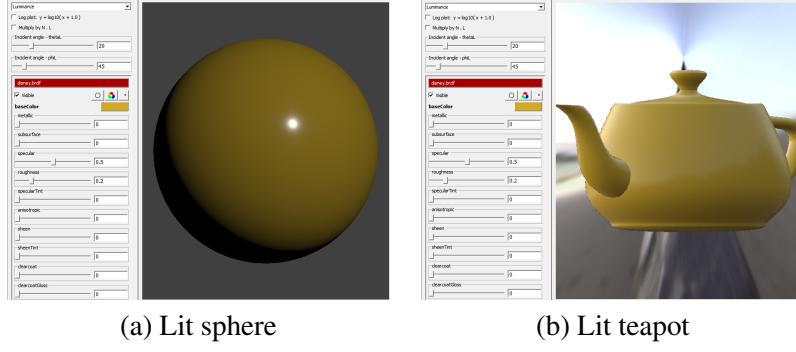


Figure 6.1: The UI for determining the property settings, including albedo, specular, and roughness of the surface. The albedo is set as the value channel of HSV colour. In this case, the albedo is set as 0.8, and the specular and roughness is set as 0.5, 0.2, respectively. (a) demonstrates the effect of the property settings on a sphere while (b) on a teapot.

6.3.1 Synthetic Datasets

We use three objects with increasing degree of concavity, which are ‘bottle’, ‘knight’, and ‘king’, as shown in Figure 6.2. We select four property settings representing the four classes of most commonly seen objects discussed in Chapter 4, which is shown in Table 6.1 to assess the validity of the mapping. We present the results of the mapping in Table 6.1 as a reference to check if the quantitative and qualitative results from the testing data is consistent with the results of the mapping. The results are shown in Figure 6.3, 6.4, and 6.5.

Class	Texture	Albedo	Specular	Roughness	Metrics		
					Accuracy	Completeness	Ang diff
(a)	0.2	0.8	0.2	0.8	GSL	GSL	EPS
(b)	0.2	0.8	0.5	0.2	GSL	-	-
(c)	0.8	0.8	0.2	0.8	PMVS, GSL	PMVS, GSL	EPS
(d)	0.8	0.8	0.5	0.2	PMVS, GSL	PMVS	-

Table 6.1: Property settings of the three testing objects: ‘bottle’, ‘knight’, ‘king’, which have increasing degree of concavity.

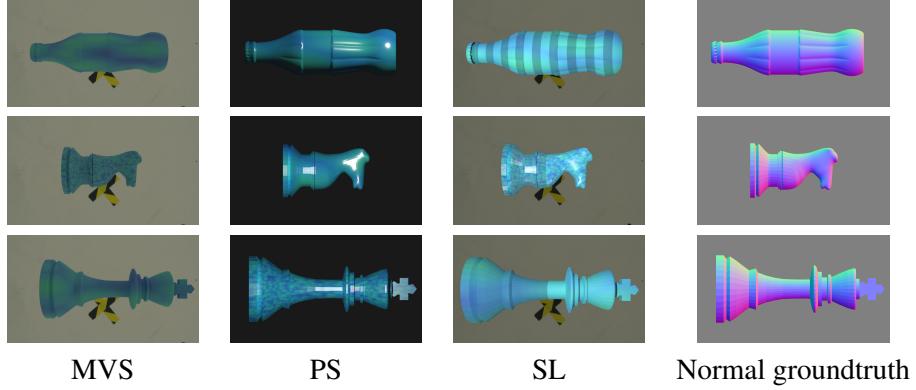


Figure 6.2: The synthetic dataset and groundtruth for the evaluation of the extensiveness of the mapping to objects with different shapes. Three objects with varied degrees of concavity are selected, each is configured with four properties settings.

Data 1: bottle

The first test object is a ‘bottle’, which has shallow indentations on the surface, thus low level concavity. The synthetic object is configured with the four property settings listed in Table 6.1. The first column presents the results of the mapping, and the algorithms that produce acceptable results are labeled in green box. All quantitative and qualitative results align with those of the mapping, thus we claim that the mapping is successfully applied to surface with low concavity.

Data 2: knight

The second object is a chess piece knight, which has medium concavity. In this case of medium concavity, we can see that the results of PMVS and GSL are still consistent with the mapping. However, the EPS fails to return reliable reconstruction for cases of high specular, such as (b) and (d) due to cast shadow effect, which is manifested by an increase of the variation of the angular error, represented by standard variation and interquatile range. Thus we claim that the mapping is still valid for PMVS, GSL and for EPS in low specular cases.

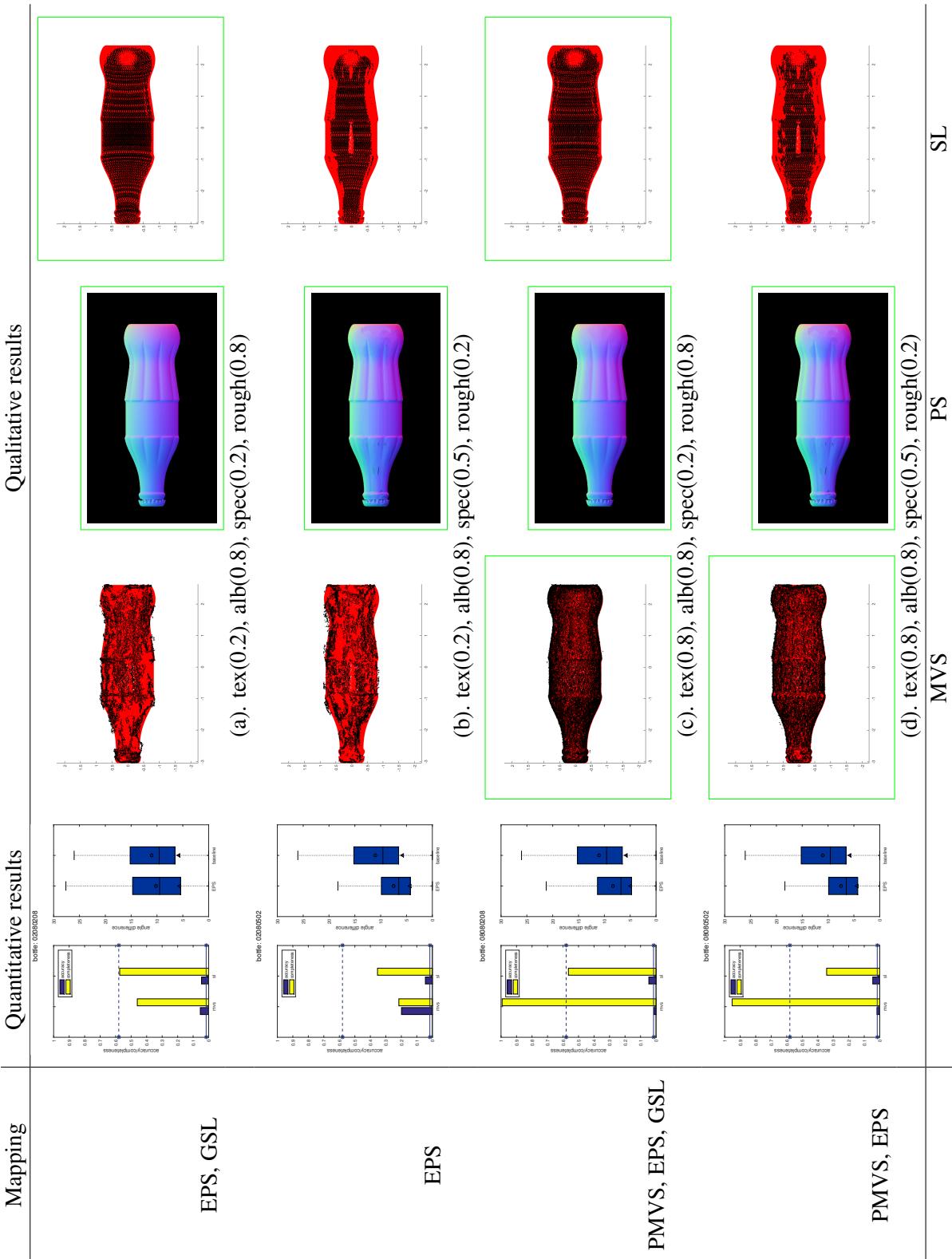


Figure 6.3: The first column shows the best algorithm chosen by the mapping. The quantitative and qualitative performance of each technique on the synthetic dataset. The red dots are from the ground truth while the black ones are the reconstruction.

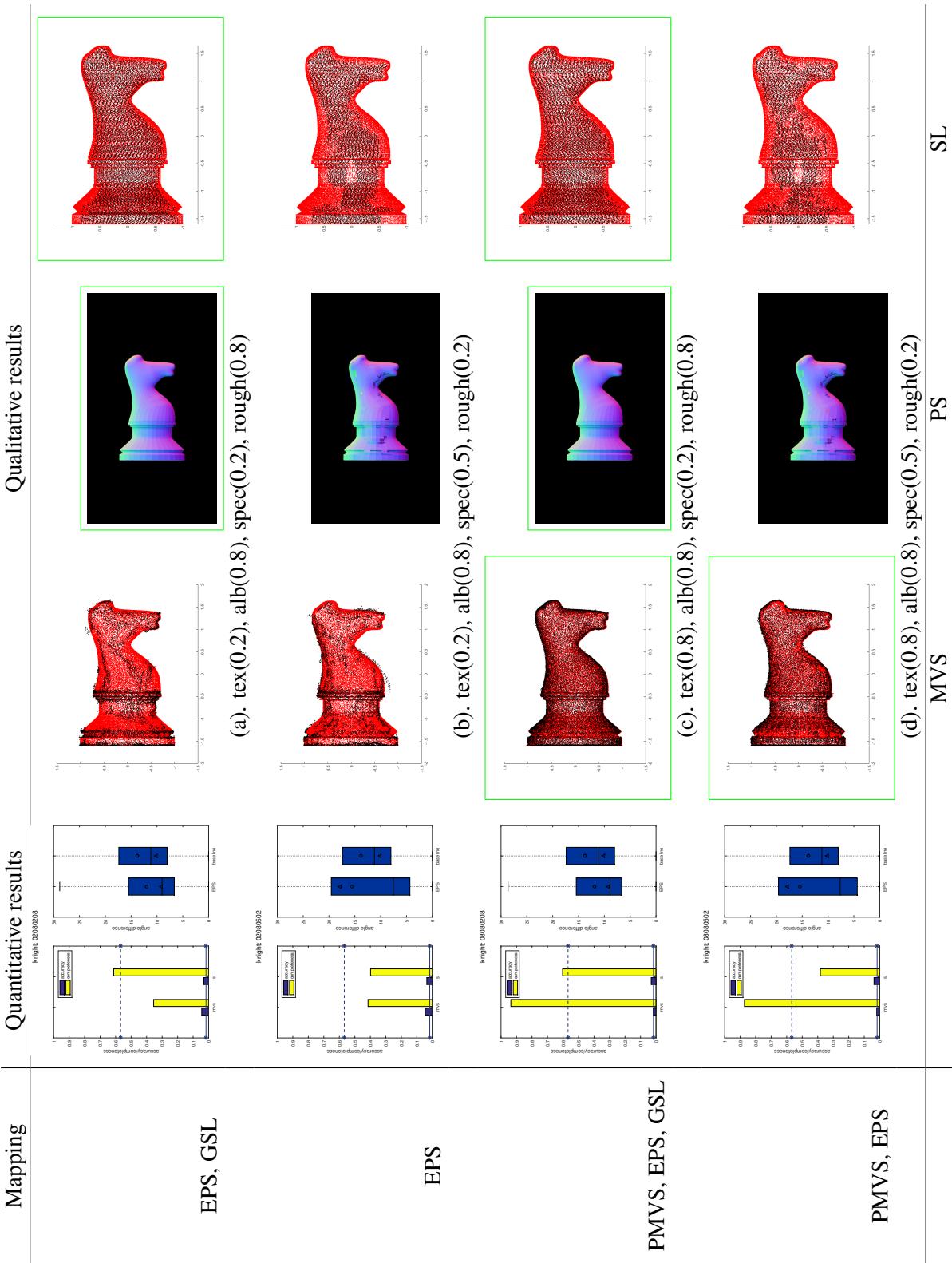


Figure 6.4: The first column shows the best algorithm chosen by the mapping. The quantitative and qualitative performance of each technique on the synthetic dataset. The red dots are from the ground truth while the black ones are the reconstruction.

Data 3: king

The last synthetic object is the chess piece king, which has the largest concavity. In the case of high concavity, the quantitative results of PMVS and GSL are still consistent with that of the mapping. However, the results of the EPS become inconsistent, which is the result of the cast shadow due to the large concavity. Though the result of EPS under condition (a), (c) are still better than that of the baseline, the medium angular error is above the acceptable threshold, which is 10° in most cases. We can see with more clarity from the normal maps that the cast shadow on the ‘cross’ leads to completely inaccurate normal estimation, which is labeled by red rectangle.

Summary

We can conclude that the mapping of PMVS and GSL are robust to concavity whereas EPS is relatively more sensitive to concavity due to cast shadows. Therefore, we should put more effort on developing Photometric Stereo algorithms that can reliably deal with cast shadow so that they can be reliably applied to shape with more complex shapes.

6.4 Interpreter

The framework consists of three separate layers: The first layer is the actual implementation of the algorithms. The second layer is the description of the problem domain. The third layer is the interpreter which receives the description from the user and return an acceptable result. It is responsible for choosing an appropriate 3D reconstruction algorithm based on the description of the problem domain and additional requirements. There are many ways to use the mapping to interpret the problem description. As a proof of concept, we proposed a proof of concept interpreter that consists of two components: mapping and constraints.

It requires an understanding of algorithmic performance across difference range of problem space to create an interpreter of the 3D reconstruction problem. The mapping constructed using the synthetic dataset in Chapter 5 provides us with a detailed understanding of how the different combinations of properties affect the performance of algorithms. This allows us to select an appropriate algorithm based

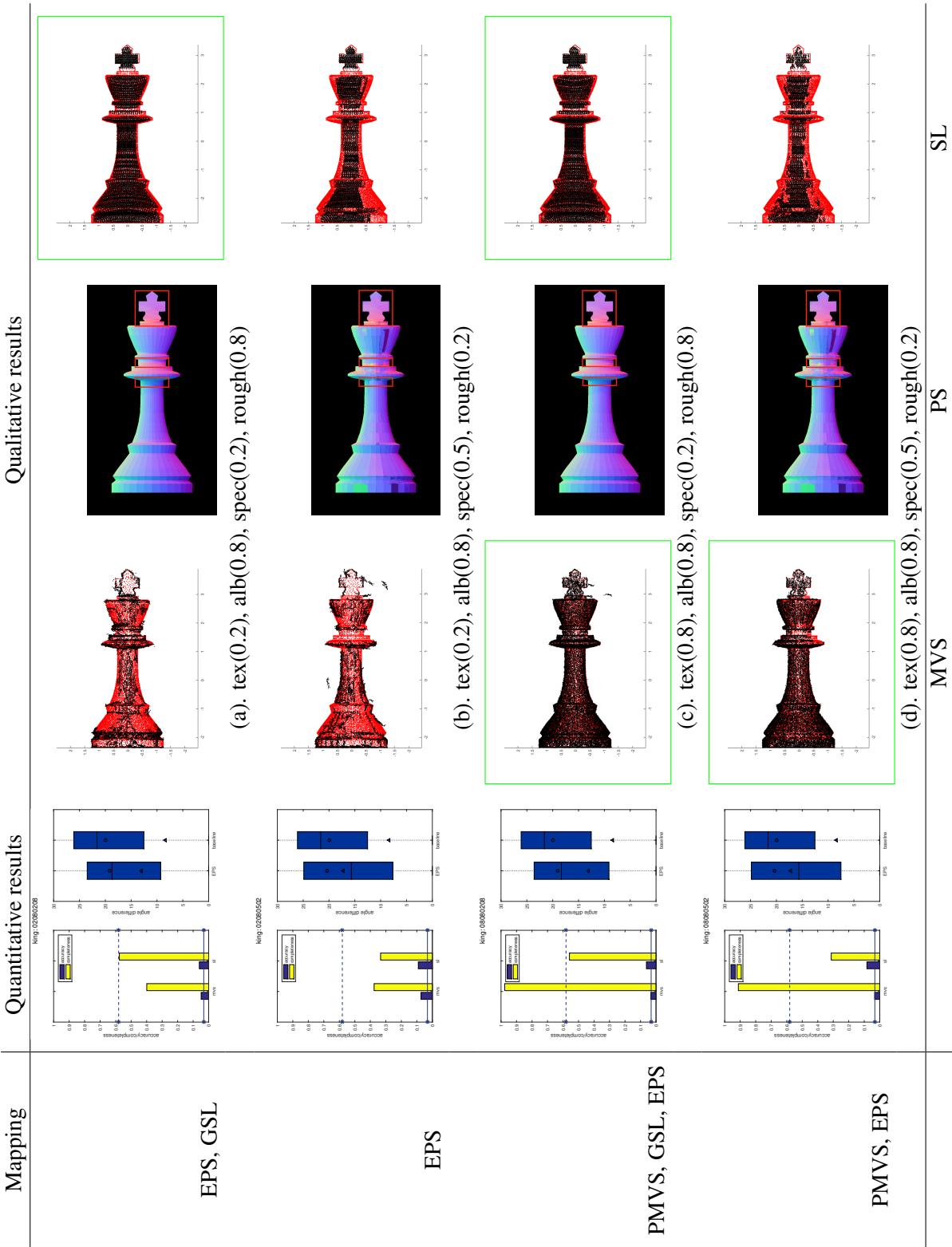


Figure 6.5: The first column shows the best algorithm chosen by the mapping. The quantitative and qualitative performance of each technique on the synthetic dataset. The green dots are from the ground truth while the black ones the reconstruction.

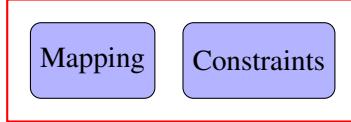


Figure 6.6: Two components of the Interpreter layer.

on a description of properties.

Next we turn our sights to define the constraints of the framework. Constraints are provided to allow the user to describe the expected result so that a model that most resembles the user's request can be returned by the framework when multiple algorithms can achieve satisfactory results. The following constraints are provided:

- depth-first/shape-first: methods that reconstructs surface orientations from a single viewpoint can't retrieve the depth information, thus is referred to as 2.5D reconstruction, such as Shape from Shading, Photometric Stereo, and Shape from (de)focus, etc.. However, these methods generally can reconstruct small scale details, thus achieve much higher detailed results. Intuitively, depth-first would return model with true depth information whereas shape-first would prioritize details and depth information.
- accuracy-first/completeness-first: methods that achieve high accuracy don't necessarily achieve high completeness. This allows the user to choose the algorithm based on the priority level of accuracy and completeness.

6.5 Validation of Interpreter

We demonstrate real-world use cases of the proof of concept interpreter, which should return a satisfactory result given a valid description, or a less successful one given an incorrect description. We choose four different problem conditions where each describes one of the four major classes of objects and provides demonstrative results. Please refer to the appendix for more results.

6.5.1 Real-world Datasets

We use a similar setup to the synthetic settings and captured a real world dataset of nine objects. The property of these objects are listed in Table ???. Since we don't

have the ground truth, we resort to visual analysis to see if the appropriate algorithm gives an acceptable reconstruction compared to that of the baseline method.

class #	Description	Object	Material
1	textureless		
	diffuse		
	bright		
2	textureless		
	mixed diffuse/specular		
	bright		
3&4	textured		
	diffuse		
	dark/bright		
5&6	textured		
	mixed diffuse/specular		
	dark/bright		

Figure 6.7: The representatives of the six classes of objects used for evaluation.

We use the aforementioned methods to retrieve the parameters of each property, the decomposition of material for each object is presented in Figure A.1.

The property settings of each object is listed in Table A.2.

Object	Texture	Albedo	Specular	Roughness	Mapping
status	0.2	0.8	0.2	0.5	EPS, GSL
cup	0.2	0.8	0.2	0.2	EPS, GSL
pot	0.8	0.2, 0.5	0.2	0.2	PMVS
vase	0.8	0.8, 0.2	0.5	0.2	PMVS

Table 6.2: Property list for the real-world objects

Data 1: statue

The first object is a statue with low texture, specular component, medium roughness, and high albedo. PMVS produces a very noise reconstruction due to the lack of surface texture whereas the other two techniques return satisfactory results. The interpreter would return the appropriate result based on the user specified constraints.

Data 2: cup

The second object is a cup with low texture, roughness, and high albedo and specular. PMVS fails to reconstruct the surface while the other two give good results. The quality of detail of EPS are clearly higher than that of GSL, and the interpreter would return the one meets the constraints specified by the user.

Data 3: pot

The third object is a pot with high texture, low specular, medium roughness, and both high and low albedo values. PMVS gives good reconstruction results while the other two suffers from low albedo.

Data 4: vase

The fourth object is a vase with high texture, high specular, low roughness, and both low and high albedo. PMVS gives good results, while the other two mehtods fails to reconstruct the surface due to low albedo an high specular.

Mapping	Qualitative results			Baseline VH
	PMVS	EPS	GSL	
EPS, GSL				
PMVS				
PMVS				

Figure 6.8: The evaluation of the effectiveness of the mapping using real-world object. The well reconstructed object is label by green rectangle.

6.6 Summary

Building upon our description and mapping, we are able to develop a proof of concept interpreter which interprets the model of the problem, selects the most appropriate algorithm base on the mapping and return a reliable reconstruction result.

The development of more complex description of object geometry and material, incorporation of new algorithms and the improvement of the mapping is an ongoing process to improve the framework.

Chapter 7

Conclusions

7.1 Summary

7.2 Future directions

7.2.1 Geometric Model

Current geometric model fails to capture the complexity of the real world object and focuses mainly on the visual properties. Thus one of the future directions is to develop intuitive geometric models to describe complex object.

7.2.2 Property Parameters

We use a try-and-see approach to obtain the property settings of the object, which is base on judgements of the users, thus not very rigorous and seems cumbersome. We can use machine learning techniques to obtain the visual and geometric information

7.2.3 Metric

We utilized three metrics: accuracy, completeness and angular error. However, there are some other measures that worth investigating: colour accuracy, ‘ghost’ reconstruction error, etc. Additional metrics can extend the application of the

framework and provide more option for developers to choose from.

7.2.4 Mapping Construction

The construction of mapping requires evaluation of the algorithm for pairwise properties, which doesn't scale so well with respect to the number of properties. Therefore, we need a more

7.2.5 Interpreter

Current implementation of interpreter is a simplistic one and doesn't fully take advantage of all the information we have obtained from the construction of mapping. Therefore, we should develop a more sophisticated interpreter

7.3 Concluding remarks

Bibliography

- [1] Autodesk. URL <http://en.wikipedia.org/wiki/Autodesk>. → pages 1
- [2] Lidar. URL <http://en.wikipedia.org/wiki/Lidar>. → pages 1
- [3] Brdf explorer. <https://www.disneyanimation.com/technology/brdf.html>. → pages 77
- [4] Kinect. URL <http://en.wikipedia.org/wiki/Kinect>. → pages 1
- [5] Vxl c++ libraries for computer vision research and implementation. <http://vxl.sourceforge.net>. → pages 8
- [6] N. Alldrin, T. Zickler, and D. Kriegman. Photometric stereo with non-parametric and spatially-varying reflectance. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. → pages 26, 29, 33
- [7] N. G. Alldrin and D. J. Kriegman. Toward reconstructing surfaces with arbitrary isotropic reflectance: A stratified photometric stereo approach. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. → pages 17, 30
- [8] N. G. Alldrin, S. P. Mallick, and D. J. Kriegman. Resolving the generalized bas-relief ambiguity by entropy minimization. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–7. IEEE, 2007. → pages 26, 33
- [9] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), Aug. 2009. → pages 11

- [10] S. Barsky and M. Petrou. The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1239–1252, 2003. → pages 16, 26, 29, 33
- [11] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille. The bas-relief ambiguity. *International journal of computer vision*, 35(1):33–44, 1999. → pages 26, 29, 33
- [12] S. Berkiten and S. Rusinkiewicz. An RGBN benchmark. Technical report, Technical Report TR-977-16, Princeton University, Feb. 2016. → pages 48, 77
- [13] F. Bernardini, H. Rushmeier, I. M. Martin, J. Mittleman, and G. Taubin. Building a digital model of michelangelo’s florentine pieta. *IEEE Computer Graphics and Applications*, 22(1):59–67, 2002. → pages 1
- [14] F. Blais. Review of 20 years of range sensor development. *Journal of Electronic Imaging*, 13(1), 2004. → pages 10
- [15] R. C. Bolles and P. Horaud. 3dpo: A three-dimensional part orientation system. *The International Journal of Robotics Research*, 5(3):3–26, 1986. → pages 38
- [16] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library.* ” O’Reilly Media, Inc.”, 2008. → pages 8
- [17] E. N. Coleman and R. Jain. Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry. *Computer graphics and image processing*, 18(4):309–328, 1982. → pages 16, 26, 29, 33
- [18] C. H. Esteban and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding*, 96(3):367–392, 2004. → pages 10, 26, 33
- [19] O. Faugeras and R. Keriven. *Variational principles, surface evolution, pde’s, level set methods and the stereo problem*. IEEE, 2002. → pages 1, 10, 26, 33
- [20] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2010. → pages 1, 11, 26, 33, 50
- [21] S. Galliani, K. Lasinger, and K. Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *Proceedings of the IEEE*

- International Conference on Computer Vision*, pages 873–881, 2015. →
pages 11
- [22] J. Geng. Structured-light 3d surface imaging: a tutorial. *Advances in Optics and Photonics*, 3(2):128–160, 2011. → pages 23
- [23] M. Goesele, B. Curless, and S. M. Seitz. Multi-view stereo revisited. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 2402–2409. IEEE, 2006. → pages 1, 11, 26, 33
- [24] D. B. Goldman, B. Curless, A. Hertzmann, and S. M. Seitz. Shape and spatially-varying brdfs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1060–1071, 2010. →
pages 17, 26, 29, 33
- [25] H. Hayakawa. Photometric stereo under a light source with arbitrary motion. *JOSA A*, 11(11):3079–3089, 1994. → pages 16, 26, 29, 33
- [26] A. Hertzmann and S. M. Seitz. Example-based photometric stereo: Shape reconstruction with general, varying brdfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1254–1264, 2005. → pages 16, 26, 28, 29, 33, 50
- [27] V. H. Hiep, R. Keriven, P. Labatut, and J.-P. Pons. Towards high-resolution large-scale multi-view stereo. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1430–1437. IEEE, 2009. →
pages 10
- [28] B. K. Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. 1970. → pages 13, 26, 27, 33
- [29] I. Ihrke, K. N. Kutulakos, H. Lensch, M. Magnor, and W. Heidrich. Transparent and specular object reconstruction. In *Computer Graphics Forum*, volume 29, pages 2400–2426. Wiley Online Library, 2010. → pages 25
- [30] S. Inokuchi. Range-imaging system for 3d object recognition. In *Proc. of 7th International Conference on Pattern Recognition, 1984*, 1984. → pages 26, 33
- [31] H. Jin, S. Soatto, and A. J. Yezzi. Multi-view stereo beyond lambert. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE*

- Computer Society Conference on*, volume 1, pages I–I. IEEE, 2003. → pages 27
- [32] H. Jin, S. Soatto, and A. J. Yezzi. Multi-view stereo reconstruction of dense shape and complex appearance. *International Journal of Computer Vision*, 63(3):175–189, 2005. → pages 27
 - [33] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. Available from: <<http://www.peterkovesi.com/matlabfn/>>. → pages 8
 - [34] K. N. Kutulakos and S. M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000. → pages 1, 10
 - [35] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, et al. The digital michelangelo project: 3d scanning of large statues. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 131–144. ACM Press/Addison-Wesley Publishing Co., 2000. → pages 1
 - [36] M. Lhuillier and L. Quan. Match propagation for image-based modeling and rendering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1140–1146, 2002. → pages 11
 - [37] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE transactions on pattern analysis and machine intelligence*, 27(3):418–433, 2005. → pages 11
 - [38] J. J. Little. *Recovering shape and determining attitude from extended gaussian images*. PhD thesis, 1985. → pages 34
 - [39] S. P. Mallick, T. E. Zickler, D. J. Kriegman, and P. N. Belhumeur. Beyond lambert: Reconstructing specular surfaces using color. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 619–626. Ieee, 2005. → pages 16, 26, 29, 33
 - [40] G. Mariottini and D. Prattichizzo. Egt: a toolbox for multiple view geometry and visual servoing. *IEEE Robotics and Automation Magazine*, 3(12), December 2005. → pages 8
 - [41] D. Marr. Vision: A computational investigation into the human representation and processing of visual information. 1982. → pages 10

- [42] W. Matusik, C. Buehler, L. McMillan, and S. J. Gortler. An efficient visual hull computation algorithm. *Tech. Rep., MIT LCS Technical Memo 623, MIT Laboratory for Computer Science*, 2002. → pages 26, 33
- [43] S. K. Nayar, K. Ikeuchi, and T. Kanade. Surface reflection: physical and geometrical perspectives. Technical report, DTIC Document, 1989. → pages 43
- [44] G. P. Otto and T. K. Chau. region-growing algorithm for matching of terrain images. *Image and vision computing*, 7(2):83–94, 1989. → pages 10
- [45] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(6035):314–319, 1985. → pages 9
- [46] C. Rocchini, P. Cignoni, F. Ganovelli, C. Montani, P. Pingi, and R. Scopigno. Marching intersections: an efficient resampling algorithm for surface management. In *Shape Modeling and Applications, SMI 2001 International Conference on.*, pages 296–305. IEEE, 2001. → pages 18
- [47] S. Roy and I. J. Cox. A maximum-flow formulation of the n-camera stereo correspondence problem. In *Computer Vision, 1998. Sixth International Conference on*, pages 492–499. IEEE, 1998. → pages 10
- [48] J. Salvi, J. Pages, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern recognition*, 37(4):827–849, 2004. → pages 12, 23
- [49] Y. Sato and K. Ikeuchi. Temporal-color space analysis of reflection. *JOSA A*, 11(11):2990–3002, 1994. → pages 16, 26, 29, 33
- [50] K. Schluns. Photometric stereo for non-lambertian surfaces using color information. In *International Conference on Computer Analysis of Images and Patterns*, pages 444–451. Springer, 1993. → pages 16, 26, 29, 33
- [51] S. M. Seitz and C. R. Dyer. Photorealistic scene reconstruction by voxel coloring. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 1067–1073. IEEE, 1997. → pages 10
- [52] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, volume 1, pages 519–528. IEEE, 2006. → pages 1, 23, 49

- [53] B. Shi, Z. Wu, Z. Mo, D. Duan, S.-K. Yeung, and P. Tan. A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3707–3716, 2016. → pages 23
- [54] W. M. Silver. *Determining shape and reflectance using multiple images*. PhD thesis, Massachusetts Institute of Technology, 1980. → pages 16, 26, 28, 29, 33
- [55] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image understanding*, 58(1):23–32, 1993. → pages 26, 33
- [56] P. Tan, S. P. Mallick, L. Quan, D. J. Kriegman, and T. Zickler. Isotropy, reciprocity and the generalized bas-relief ambiguity. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. → pages 30
- [57] M. Tarini, M. Callieri, C. Montani, C. Rocchini, K. Olsson, and T. Persson. Marching intersections: An efficient approach to shape-from-silhouette. In *VMV*, pages 283–290, 2002. → pages 26, 33
- [58] Y. Uh, Y. Matsushita, and H. Byun. Efficient multiview stereo by random-search and propagation. In *3D Vision (3DV), 2014 2nd International Conference on*, volume 1, pages 393–400. IEEE, 2014. → pages 11
- [59] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008. → pages 8
- [60] G. Vogiatzis, P. H. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 391–398. IEEE, 2005. → pages 10
- [61] G. Vogiatzis, C. H. Esteban, P. H. Torr, and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(12):2241–2246, 2007. → pages 10, 11, 26, 33
- [62] R. J. Woodham. Photometric stereo: A reflectance map technique for determining surface orientation from image intensity. In *22nd Annual Technical Symposium*, pages 136–143. International Society for Optics and Photonics, 1979. → pages 13

- [63] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):191139–191139, 1980. → pages 1, 14, 26, 28, 33
- [64] E. Zheng, E. Dunn, V. Jovic, and J.-M. Frahm. Patchmatch based joint view selection and depthmap estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1517, 2014. → pages 11
- [65] T. E. Zickler, P. N. Belhumeur, and D. J. Kriegman. Helmholtz stereopsis: Exploiting reciprocity for surface reconstruction. *International Journal of Computer Vision*, 49(2-3):215–227, 2002. → pages 17, 26, 30, 33

Appendix A

Supporting Materials

This would be any supporting material not central to the dissertation. For example:

- radiometry
- technical details of MVS, PS, SL, SfS, etc

A.1 Material of real-world objects

A.2 Parameters of real-world objects

A.3 Results of real-world objects

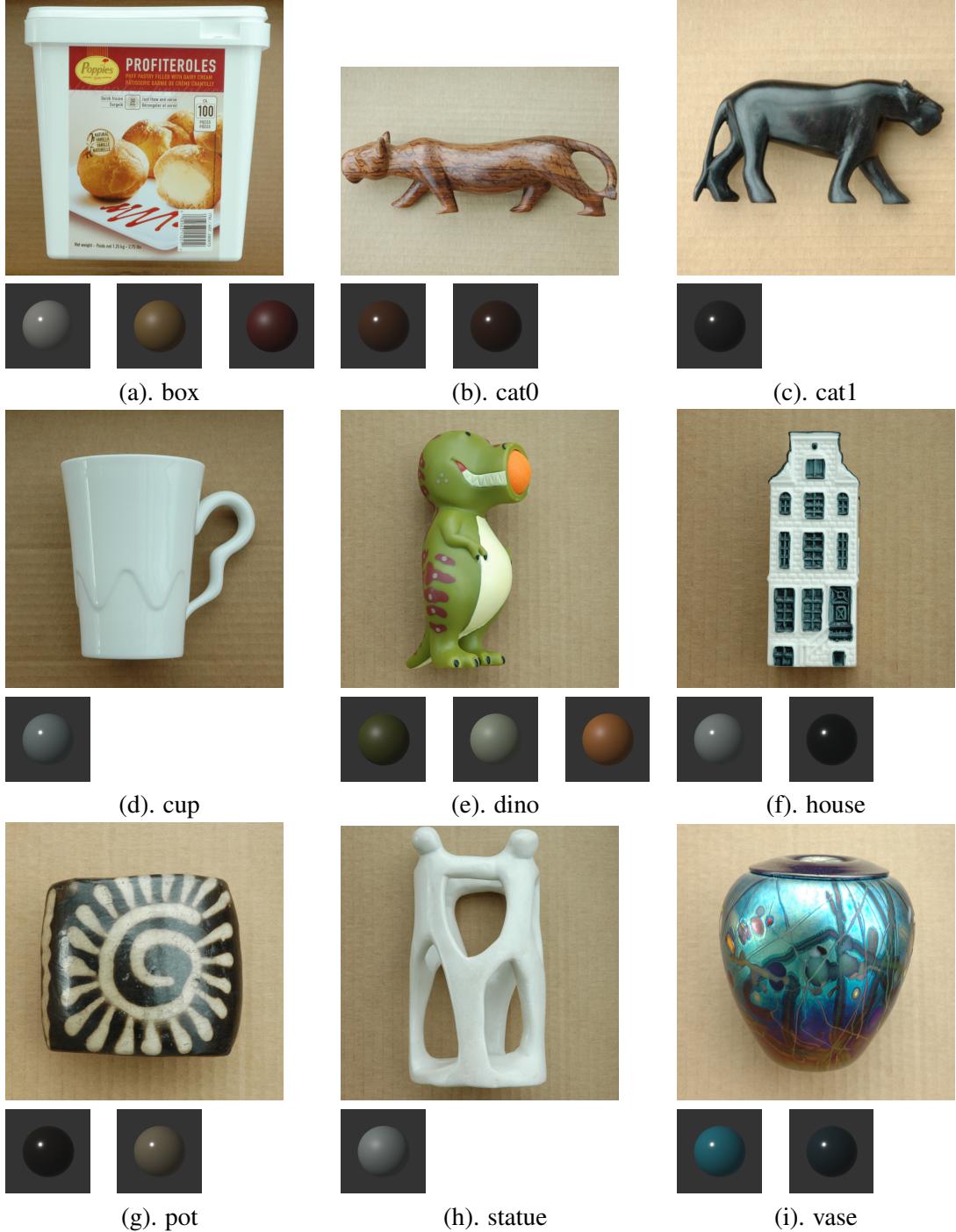


Table A.1: Material of Real-world objects.

Object	PMVS	Example-based PS	Gray SL	Mapping
box				MVS, SL, PS
cat0				None
cat1				None
dino				PS, SL
cup				PS, SL
house				MVS
pot				MVS, SL

Figure A.1: Reconstruction results of MVS, PS, SL

Property	Texture	Albedo	Specular	Roughness	Best-suited Algo.
box	0.2	0.8	0.2	0.2	MVS, SL, PS
	0.5	0.2	0.2	0.5	
	0.8	0.8	0.2	0.5	
cat0	0.5	0.5, 0.2	0.2	0.2	None
cat1	0.2	0.2	0.2	0.2	None
cup	0.2	0.8	0.2	0.2	PS, SL
dino	0.2	0.5, 0.8, 0.8	0.2	0.5	SL
house	0.8	0.8, 0.2	0.2	0.2	MVS
pot	0.5	0.2, 0.5	0.2	0.2	MVS, SL
status	0.2	0.8	0.2	0.5	PS, SL
vase	0.8	0.8, 0.2	0.2	0.2	None

Table A.2: Property list for the real-world objects

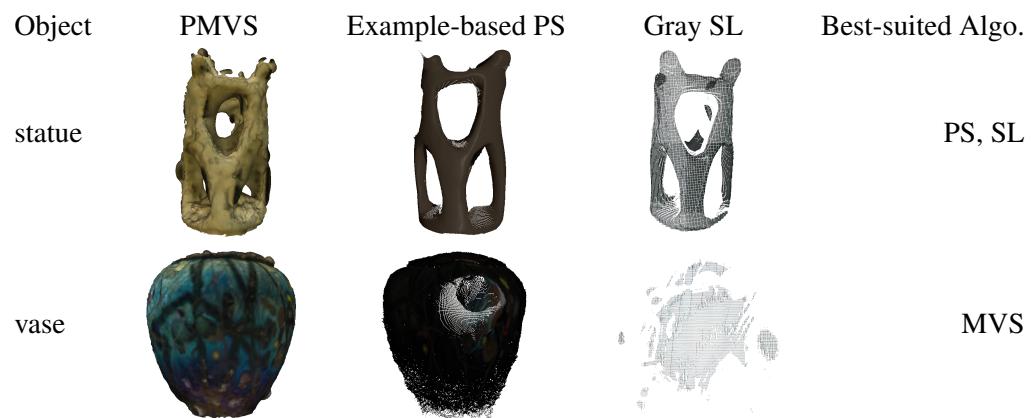


Figure A.2: Reconstruction results of MVS, PS, SL (cont'd)