

# **Thesis**

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  
(Electric and Computer Engineering Department)

The University of British Columbia

(Vancouver)

April 2017

© Kai Wu, 2017

# **Abstract**

This document provides brief instructions for using the `ubcdiss` class to write a UBC-conformant dissertation in L<sup>A</sup>T<sub>E</sub>X. This document is itself written using the `ubcdiss` class and is intended to serve as an example of writing a dissertation in L<sup>A</sup>T<sub>E</sub>X. This document has embedded Unique Resource Locators (URLS) and is intended to be viewed using a computer-based Portable Document Format (PDF) reader.

Note: Abstracts should generally try to avoid using acronyms.

Note: at University of British Columbia (UBC), both the Graduate and Postdoctoral Studies (GPS) Ph.D. defence programme and the Library's online submission system restricts abstracts to 350 words.

# **Preface**

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

# Table of Contents

<b>Abstract</b> . . . . .	<b>ii</b>
<b>Preface</b> . . . . .	<b>iii</b>
<b>Table of Contents</b> . . . . .	<b>iv</b>
<b>List of Tables</b> . . . . .	<b>vii</b>
<b>List of Figures</b> . . . . .	<b>ix</b>
<b>Glossary</b> . . . . .	<b>xi</b>
<b>Acknowledgments</b> . . . . .	<b>xii</b>
<b>1 Introduction</b> . . . . .	<b>1</b>
1.1 Problem definition . . . . .	3
1.1.1 Scope . . . . .	3
1.1.2 Data . . . . .	4
1.2 Thesis outline . . . . .	4
1.3 Contributions . . . . .	5
1.4 Organization . . . . .	6
<b>2 Related Work</b> . . . . .	<b>8</b>
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
<b>3</b>	<b>A new taxonomy of 3D Reconstruction . . . . .</b>	<b>22</b>
3.1	Object class . . . . .	23
3.2	Class 1 . . . . .	23
3.2.1	SfS . . . . .	25
3.2.2	Lambertian PS: uniform reflectance . . . . .	25
3.2.3	SL . . . . .	26
3.3	Class 2 . . . . .	26
3.3.1	Non-Lambertian PS: uniform reflectance . . . . .	26
3.4	Class 3 . . . . .	27
3.4.1	MVS . . . . .	27
3.4.2	Lambertian PS: non-uniform reflectance . . . . .	28
3.5	Class 4 . . . . .	29
3.6	Class 5 . . . . .	29
3.6.1	Non-Lambertian PS: non-uniform reflectance . . . . .	30
3.7	Class 6 . . . . .	30
3.7.1	VH . . . . .	31
3.8	Summary . . . . .	31
<b>4</b>	<b>A Description of 3D Reconstruction . . . . .</b>	<b>33</b>
4.1	Definition . . . . .	33
4.1.1	Basic notations . . . . .	34
4.1.2	Segment and Scell . . . . .	34
4.1.3	Photo-consistency . . . . .	35
4.1.4	Formal Definition . . . . .	36
4.1.5	Applied Definition . . . . .	36
4.2	Model . . . . .	37
4.3	Representation . . . . .	37
4.3.1	Texture . . . . .	38

4.3.2	Lightness . . . . .	38
4.3.3	Reflectance . . . . .	41
4.3.4	Roughness . . . . .	42
4.3.5	Concavity . . . . .	43
4.4	Expression . . . . .	44
<b>5</b>	<b>A Mapping of 3D Reconstruction Techniques . . . . .</b>	<b>45</b>
5.1	Synthetic setup . . . . .	46
5.2	Structure of Datasets . . . . .	46
5.3	Selected methods . . . . .	47
5.4	Evaluation metrics . . . . .	47
5.5	Dependency Check . . . . .	48
5.5.1	PMVS . . . . .	48
5.5.2	Example-based PS . . . . .	50
5.5.3	Single property . . . . .	51
5.5.4	Pairwise properties . . . . .	51
5.5.5	Gray-code SL . . . . .	54
5.5.6	Single property . . . . .	54
5.5.7	Pairwise properties . . . . .	56
5.6	Training and mapping . . . . .	56
5.6.1	PMVS . . . . .	56
5.6.2	Example-based PS . . . . .	57
5.6.3	Gray-code SL . . . . .	57
<b>6</b>	<b>Interpretation of 3D Reconstruction Model . . . . .</b>	<b>62</b>
6.1	Synthetic Datasets . . . . .	62
6.2	Real-world Datasets . . . . .	64
6.3	Summary . . . . .	65
<b>Bibliography</b>	. . . . .	<b>68</b>
<b>A Supporting Materials</b>	. . . . .	<b>74</b>

# 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	Selected algorithms from each class of algorithms . . . . .	23
Table 3.2	Categories of algorithm that are applicable for object class 1. .	25
Table 3.3	Categories of algorithm that are applicable for object class 2. .	26
Table 3.4	Categories of algorithm that are applicable for object class 3. .	27
Table 3.5	Categories of algorithm that are applicable for object class 4. .	29
Table 3.6	Categories of algorithm that are applicable for object class 5. .	29
Table 3.7	Categories of algorithm that are applicable for object class 6. .	31
Table 3.8	Summary of VH representations and reconstruction approach. .	31
Table 3.9	Algorithm classification based on the new taxonomy . . . . .	32
Table 4.1	Model of the 3D reconstruction problem. . . . .	37
Table 4.2	Expression of the reconstruction problem for the object class 1, 2, 3, 5. . . . .	44
Table 5.1	Parameter settings of dependency check of MVS. . . . .	48
Table 5.2	The correlation between each property and the metrics <i>accuracy</i> and <i>completeness</i> . . . . .	50
Table 5.3	Parameter settings of dependency check of example-based PS. .	51
Table 5.4	The correlation between each property and the metric <i>angular difference</i> . . . . .	54

Table 5.5	Parameter settings of dependency check of Gray code SL. . . . .	54
Table 5.6	The correlation between each property and the metrics <i>accuracy</i> and <i>completeness</i> . . . . .	56
Table 5.7	The conditions under which PMVS works well in terms of the two metrics <i>accuracy</i> and <i>completeness</i> . . . . .	58
Table 5.8	The conditions under which example-based PS works well in terms of the two metric <i>angular difference</i> . . . . .	58
Table 5.9	The conditions under which Gray code SL works well in terms of the two metrics <i>accuracy</i> and <i>completeness</i> . . . . .	61
Table 6.1	Property lists of the test objects. Link to the labels in Figure ??, (a): dark blue rectangle, (b): dark green rectangle, (c): light blud rectangle, (d): light green rectangle. . . . .	63
Table 6.2	Property list for the real-world objects . . . . .	65

# List of Figures

Figure 1.1	Thesis overview. Rectangles denote process. Rounded rectangles represents data or component.[To be re-done] . . . . .	7
Figure 2.1	Illustratives 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. . . . .	18
Figure 2.2	Three distortion effect: distance distortion, position distortion, and foreshortening distortion. . . . .	19
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 $\frac{1}{z_o} + \frac{1}{z_i} = \frac{1}{f}$ . If the sensor plane moved forward $\Delta z_i$ , the image are no longer in focus and the <i>circle of confusion</i> $c$ depends on the distance of the sensor plane motion $\Delta z_i$ relative to the lens aperture diameter $d$ . . . . .	20
Figure 2.4	shape from focus . . . . .	21
Figure 3.1	<i>Top:</i> A list of properties for object classes. <i>Bottom:</i> Six object classes of interest. Only texture, lightness, and reflectance are consider. Properties not consider are set as follows: opaque, rough (for Lambertian)/smooth (for Non-Lambertian), low concavity. . . . .	24
Figure 3.2	The effect of GBR ambiguity . . . . .	28

Figure 4.1	Relation between a scell and a segment . . . . .	35
Figure 4.2	Light-matter interaction . . . . .	40
Figure 4.3	The light-matter interaction. . . . .	40
Figure 4.4	The light-lens interaction. . . . .	40
Figure 4.5	The light-sensor interaction. . . . .	41
Figure 4.6	A <b>red</b> 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. . . . .	42
Figure 4.7	Surface Slope Distribution Model . . . . .	43
Figure 5.1	Performance of MVS with varied properties . . . . .	49
Figure 5.2	Performance of PS with varied properties . . . . .	52
Figure 5.3	The ‘peculiar’ effect of roughness on PS. The order of the property is: albedo, specular, and roughness, thus 080205 means albedo: 0.8, specular: 0.2, and roughness: 0.5 . . . . .	53
Figure 5.4	Performance of SL with varied properties . . . . .	55
Figure 5.5	Performance of MVS with varied properties. . . . .	57
Figure 5.6	Performance of PS with varied properties. . . . .	59
Figure 5.7	Performance of SL with varied properties. . . . .	60
Figure 6.1	The synthetic datasets and groundtruth for the evaluation . . .	63
Figure 6.2	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. . . . .	64
Figure 6.3	Reconstruction results of MVS, PS, SL . . . . .	66
Figure 6.4	Reconstruction results of MVS, PS, SL (cont’d) . . . . .	67

# Glossary

This glossary uses the handy `acronym` package to automatically maintain the glossary. It uses the package's `printonlyused` option to include only those acronyms explicitly referenced in the `LATEX` source.

**GPS**      Graduate and Postdoctoral Studies

**PDF**      Portable Document Format

**URL**      Unique Resource Locator, used to describe a means for obtaining some resource on the world wide web

# **Acknowledgments**

Thank those people who helped you.

Don't forget your parents or loved ones.

You may wish to acknowledge your funding sources.

# Chapter 1

## Introduction

Modelling of the 3D world has been an active research topic in computer vision for decades. The goal is to reconstruct a 3D geometric model, represented by point cloud, voxel grid, depth maps, or surface mesh, from RGB or range sensors, optionally with the material of the surface. It 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.

We've witness a variety of tools and approaches such as Computer Aided Design (CAD) tools [1], arm-mounted probes, active methods [2, 3, 12, 33] and passive image-based methods [18, 20, 23, 32] applied successfully to some sub-domains of the problem. Among the existing techniques, active techniques such as laser scanner [33], structured light system (SL) [12], and Photometric Stereo (PS) [60], and passive method such as Multi-view Stereo (MVS) [49] have been the most successful ones. Laser scanners and structured light techniques can generate the most accurate results, but is generally complicated to set up and calibrate, time consuming to scan, and memory demanding to store and process. Photometric Stereo is able to achieve highly detailed reconstruction comparable to that of laser scanner, but the true depth information is lost due to the use of a single viewpoint. MVS requires minimal setup and works in both controlled, small scale lab setting or a outdoor, medium to large scale environments. However, the quality of the reconstruction is generally noisier, and is susceptible to the texture and material property of the surface. All these techniques requires an understanding of calibra-

tion, stereo correspondence, physics-based vision, and etc, which is no easy task to master. Furthermore, this is an extremely challenging task since it's the reverse process of image formation, which is highly likely to have more than one plausible results. To overcome this challenge, some assumptions have to be made in terms of the materials, viewpoints, and lighting, which adds additional layer of complexity to the inherit complexity of the specific reconstruction technique. A solid understanding of the interaction of lighting with surface geometry and material is a prerequisite to fully take advantage of these existing techniques.

Regardless of the success in the past and the substantial need for this technology, we have not yet witnessed any substantial progress in terms of making those techniques accessible to application developers who generally have little or no computer vision expertise. These developers generally focus more on the development of the application, have a good understanding of the properties of the target objects for their application domain, and are good at learning programming API rather than vision algorithms. We've made two key observations about computer vision algorithms: 1) none of these methods works well under all circumstances, nor do they require the same setup or inputs/outputs, making it difficult for developers to choose the 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 question: is it possible to create a computer vision abstraction that makes the selection of a particular algorithm based on the descriptions of the object or scene to be reconstructed. By doing so, we can encapsulate computer vision experts' knowledge of their algorithms strengths within the abstraction so that a developer need only describe the problem they need solved. The mental model to our approach is similar to that of the game 'name that object': one participant takes 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 abstraction around the detailed algorithms and implementations so that one or multiple best suited ones can be selected based on the 'appearance' of the object described by the developers. The developers use the abstraction's description interface that is structured to match how vision problems can be described based on a model of a 3D scene and translated to parameters useful for determining which algorithms would work best.

## 1.1 Problem definition

The problem we address in this thesis can be described as: find a small set of visual and geometric properties, from which an descriptive abstraction is formed to find the best-suited algorithm(s) to reconstruct the target object. The

### 1.1.1 Scope

To limit the scope of this work, we make the following assumptions:

#### Simplified reflectance model

Since the majority of reconstruciton techniques rely on observing light reflected off a surface, surfaces exhibit significant effect of global light tranport present a huge challenge to the reconstruction problem. Surface exhibits global light transport, including *specular, transmission, sub-surface scattering, inter-reflection, self-shadow*, and etc would break the assumptions made by most generic 3D reconstruction algorithms. Thus the global light transport are ignored, and the reflection properties of consideration are *albedo*, i.e., the ratio of reflected light w.r.t the received light, and *specularity*, i.e., the amount of specular reflection. A more comprehensive model should be constructed based on our work to incorporate more complex phenomena to be more comprehensive.

#### Simplified geometric model

It's a challenging task to model geometry using mathematical descriptions. For geometric primitives such as cube, sphere, or cone, etc, it's possible to describe the shape using concise descriptions. However, the task becomes prohibitive when it comes to shapes with varied characteristics. Furthermore it becomes more ambiguous when natural language is employed. Thus we only consider the microscopic roughness of the surface, which has a direct relation with the reflection. Other prominent geometric properties such as *concavity*, which affects self-shadow, inter-reflection, *depth-discontinuity*, which affects the depth estimation, are ignored.

### **Simplified object class**

Only a subset of visual/geometric properties are considered, which include texture, lightness, specular, and roughness. Since we use a simplified reflectance and geometric model, phenomena such as translucency, subsurface scattering, refraction, occlusion, concavity can not be described properly. Thus object exhibiting those properties are not considered for evaluation purpose.

#### **1.1.2 Data**

We use both a synthetic and a real-world dataset. The synthetic dataset is generated by a physically-based renderer Cycles with varied reflectant and geometric properties, including texture, albedo, specular, and roughness. We used the similar setup to capture real-world images of 11 objects to further test the validity of our proposed abstraction.

## **1.2 Thesis outline**

we present a flow chart to summarize the complete working of the system, as shown in Figure 1.1.

### **Related Work**

We discuss the existing softwares and toolboxes for 3D reconstruction, and present the minimum vision background needed to fully take advantage of those toolboxes. Then a review of 3D acquisition techniques is provided, organized by the visual and geometric cues used for reconstruction.

### **Taxonomy of Algorithms**

The majority of taxonomy of 3D reconstruction utilizes the differences of the algorithmic details as taxonomy axes. For instance, MVS algorithms can be categorized based on various visibility models or scene representations, and PS methods can be classified by the reflectance models. However, it doesn't provide the context or the applicability of these techniques. Thus the proposed taxonomy categorize algorithms from an object-centered perspective, i.e., algorithms are classified based

on the class that the object belongs to.

### **Description of 3D Reconstruction**

Once we have a taxonomy of algorithm based on object class, a model of the 3D reconstruction problem needs to be developed. This model should give a clear and distinct description of the task that doesn't require much vision knowledge and general enough that is not object specific. This includes a formal/applied definition of the 3D reconstruction problem, the representation, and lastly, the expression.

### **Mapping of 3D Reconstruction**

The abstraction consists of mappings from a property set and all its combination to algorithms that can achieve satisfactory results. To construct such mappings, we need to evaluate the performance of the selected algorithm 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 variability of shapes and material properties. To overcome this issue, we first investigate the dependent properties, which are properties that have influence on one another, thus must be considered jointly. Then we evaluate the performance the each algorithm under the conditions of dependent properties and all their combinations, which makes up our abstraction.

### **Interpretation of 3D Reconstruction**

We use both synthetic and real-world datasets to evaluate the proposed abstraction. We used three synthetic objects: a cup, a pot, and a vase. For the real-world dataset, we use the similar setups and captured the images for 11 objects with various shape and material properties.

## **1.3 Contributions**

The main contributions are:

- 1. A new taxonomy of 3D reconstruction problem from object-centered perspective.**

The previous taxonomies generally focus on one class of algorithms, and thus can only deal with a specific class of objects. The proposed taxonomy address this issue by considering the criteria from an object-centered perspective, i.e., classify the algorithm based on the object classes that can be reliably reconstructed.

**2. A description of 3D reconstruction that can describe 3D reconstruction problem using visual/geometric properties.**

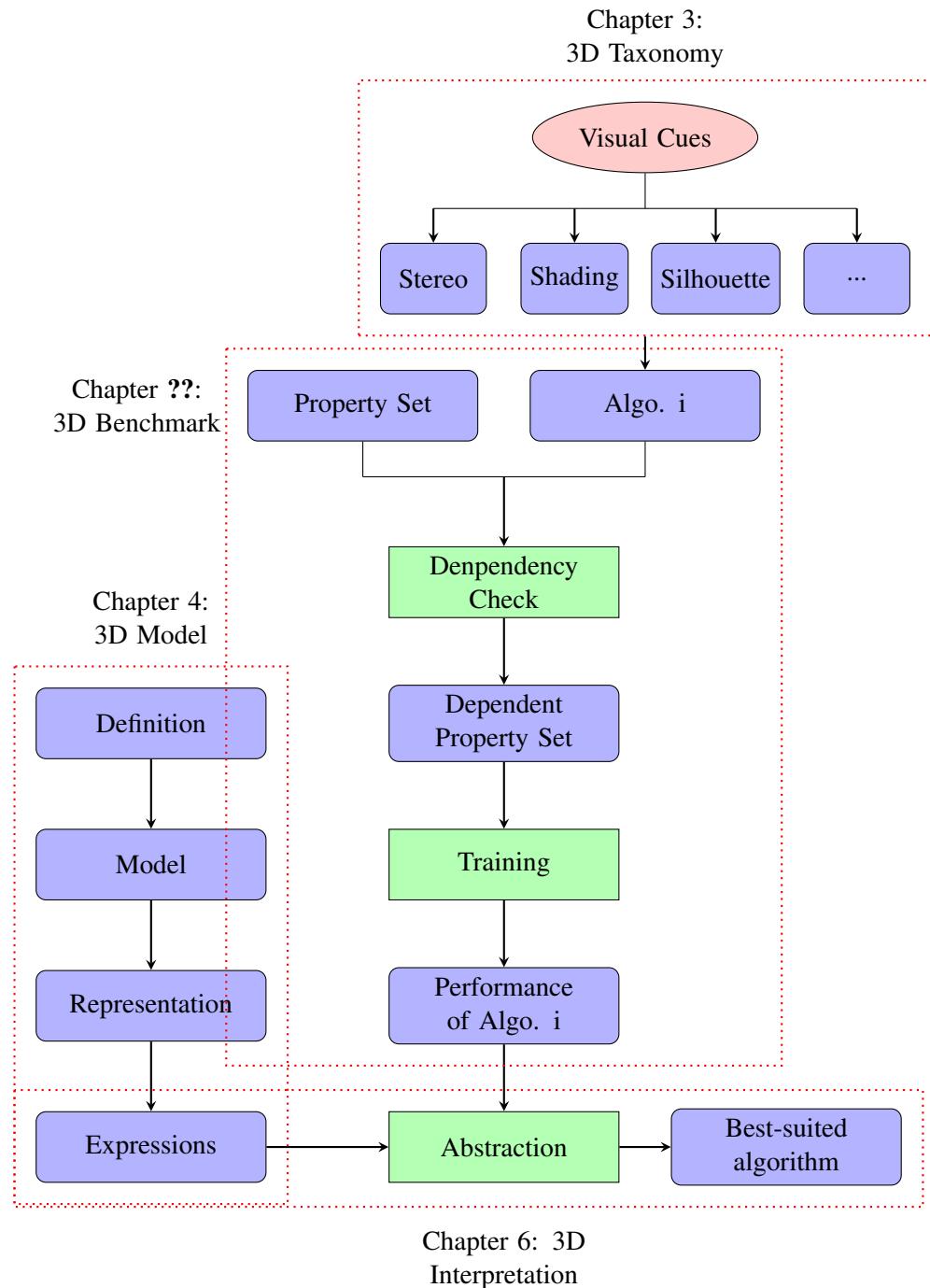
A formal and a practical definition of the 3D reconstruction problem is proposed, followed by a model and representation, which can be used to describe a 3D reconstruction problem, NOT from an algorithmic point of view, but in an object-centered manner.

**3. An mapping of 3D reconstruction that maps problem model to a suite of algorithms.**

A mapping that maps a description of 3D reconstruction problem into a set of appropriate algorithm are derived from a synthetic dataset. The process of finding the mapping is discussed in depth, which includes a step to identify the dependent properties to reduce the dimensions of the problem domain, and a step to evaluate the performance under all dependent properties and their combinations.

## 1.4 Organization

We organize this thesis as follows: we discuss the related work in Chapter 2. In Chapter 3 we provide a new taxonomy of 3D reconstruction based on object class. In Chapter 4, we provide a formal description of 3D reconstruction, which applies to most of the existing techniques, and extendable to future algorithms. In Chapter 5, we discuss the process of generating a synthetic dataset to evaluate the performance of a selected technique under the condition of different properties, which serves as the basis for the mapping of 3D reconstruction. In Chapter 6, we use both synthetic and real-world dataset to demonstrate the interpretation of the 3D reconstruction model and the validity of the proposed mapping.



**Figure 1.1:** Thesis overview. Rectangles denote process. Rounded rectangles represent data or component.[To be re-done]

# 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 [15], VLFeat [56], [VXL] [4] and multiple Matlab libraries [31, 37]. 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 invalid, this becomes an ill-posed problem since multiple combinations of geometry, viewpoint and illumination can produce exactly the same images [42], thus making it an extremely challenging task.

The 3D reconstruction techniques exploit a variety of visual and geometric cues to extract geometry from images: stereo correspondence, shading, contour, texture, (de)focus, etc. Refer to Table 2.1 for the cue used by each class of algorithms. The algorithms are organized based on the cue used for reconstruction in this review.

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 thesis, and the actual 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 [48]. Another thread of work formulates the problem in the Markov Random Field (MRF) framework and extracts the optimal surface by Graph-Cut algorithms [44, 57, 58].

### 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 [38] by iteratively removing inconsistent voxels from the scene [32]. 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 [18]. Other approaches use a deformable model and represent the scene as surface meshes that moves as a function of internal and external forces [17]. 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 [34, 35]. 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 [8]. 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, 55, 61] 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, 58]. 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 [45]. 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 [59]. 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 [60], 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

$$I_{i,j} = \rho_i \vec{n}_i^\top \vec{l}_j$$

$$\Rightarrow \mathbf{I} = \mathbf{N}^\top \mathbf{L}$$

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

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

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  $3 \times 3$  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 [16] and [9]. 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 [36, 46, 47].

*Reference object* A different approach uses a reference object that has the same material as the target object. This is proposed in [51] 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 [6, 62].

### 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 fro 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

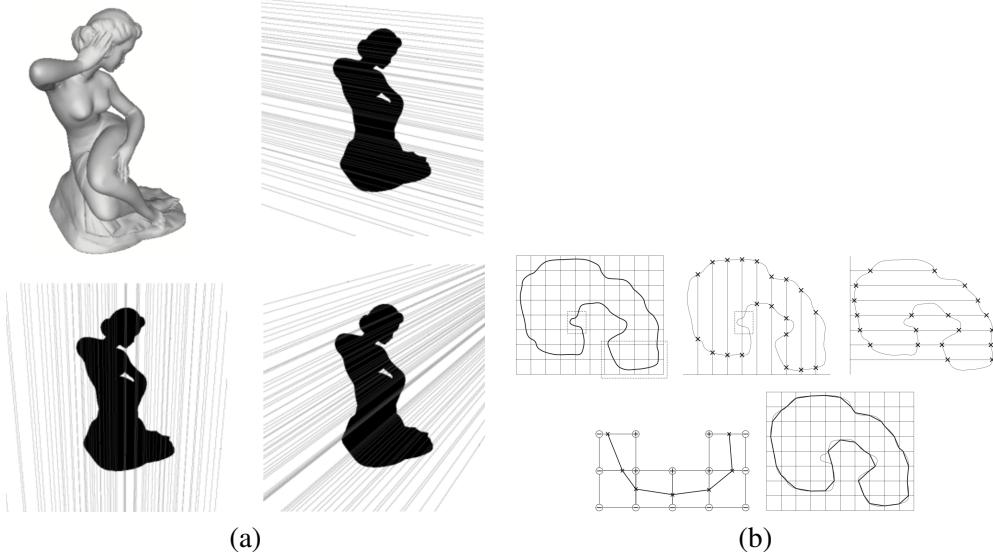
First the object space is split up into a 3D grid of voxels; each voxel is intersected with each silhouette volume; only voxels that lie inside all silhouette volumes remain part of the final shape.

#### 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 [43] 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.



**Figure 2.1:** Illustratives 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.

### Exact polyhedral methods

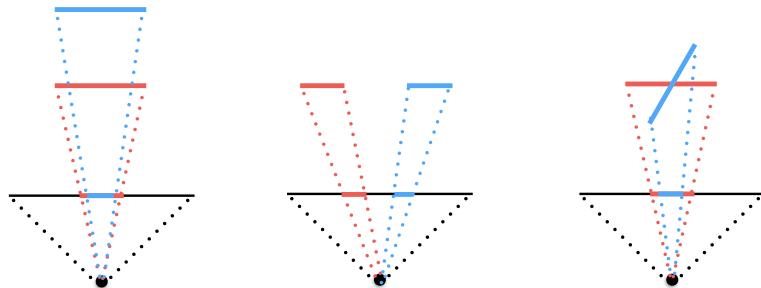
The silhouette is converted into a set of convex or non-convex 2D polygons with holes allowed. The resulting visual hull with respect to those polygonal silhouettes is a polyhedron. The faces of this polyhedron lie on the faces of the original cones. The faces of the original cones are defined by the 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.

#### 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.

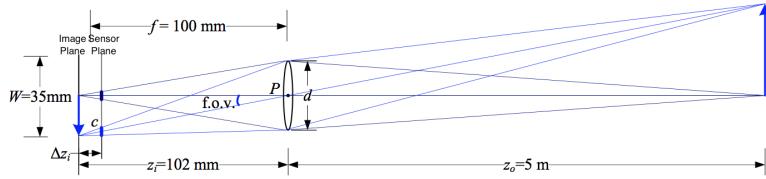


**Figure 2.2:** Three distortion effect: distance distortion, position distortion, and foreshortening distortion.

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  $\Delta z_i$ , the image are no longer in focus and the *circle of confusion*  $c$  depends on the distance of the sensor plane motion  $\Delta z_i$  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  $\Delta z_i$ , 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  $\Delta z_i$  is as follows:

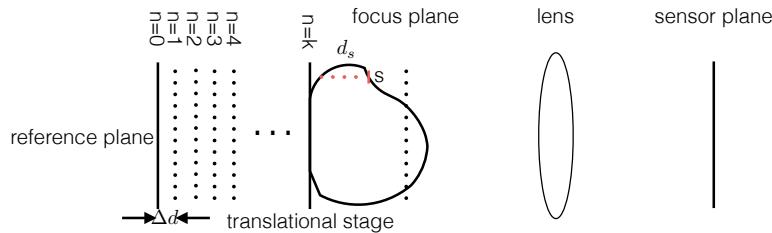
$$c = \frac{\Delta z_i r}{z_i}$$

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  $\Delta d$ , 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 new taxonomy of 3D Reconstruction

Existing taxonomies of 3D reconstruction techniques generally focus on one category of techniques: the Multi-view Stereo taxonomy in [49] 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, 45]. Photometric Stereo algorithms are classified by the assumptions or generalizations made, for instance, unknown/known reflectance, unknown/known light conditions (uncalibrated/calibrated), etc [50]. 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.

The taxonomy proposed in this chapter defines the 3D reconstruction techniques from a 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 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. 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].

To make the problem tackleable, only six classes of objects are being investigated in depth. The reasoning behind the selected object classes is solely based on the [popularity] of the class. See Figure 3.1 (b) for the six classes of objects.

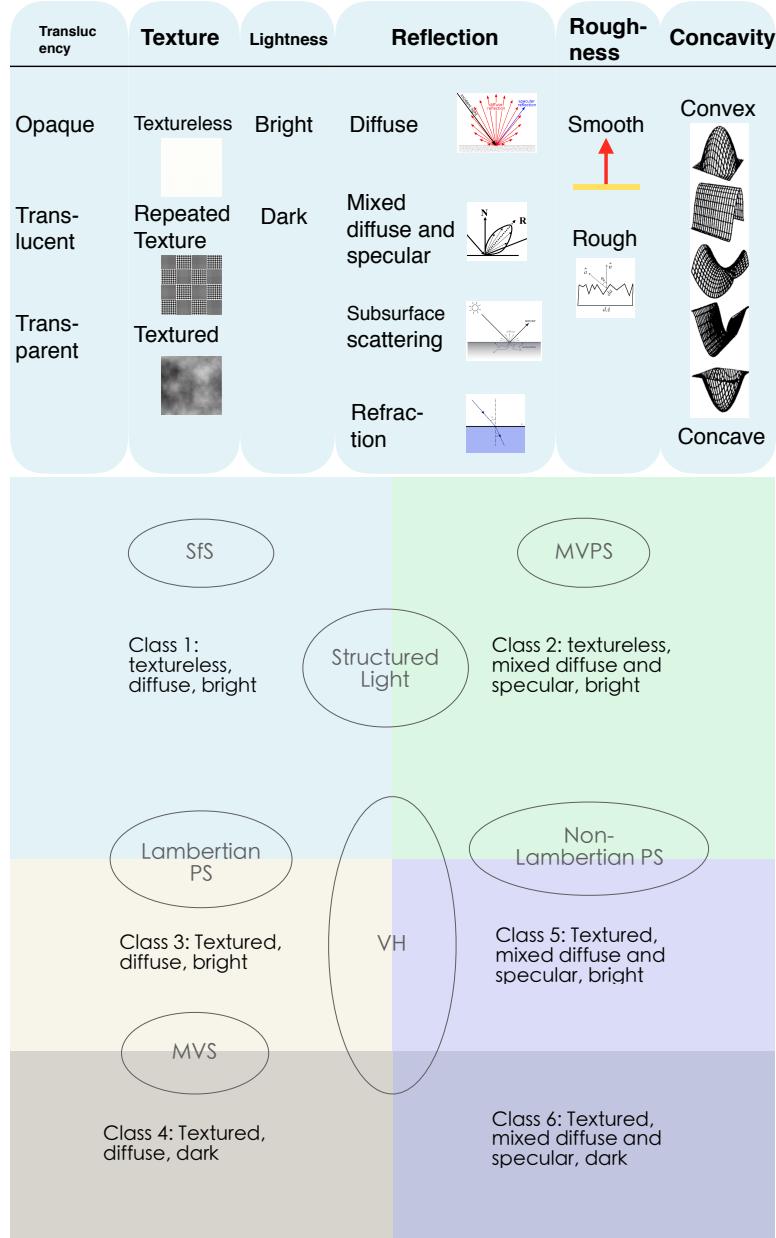
Here is a list of algorithms we will look into in depth, a summary is list in Table 3.1.

Algo. class	Technique
SfS	Horn [28]
MVS	Furukawa [20], Goesele [23], Vogiatzis [58], Hernández [17], Faugeras [18]
Lamberian PS	Woodham [60], Hayakawa [25], Belhumeur [10], Alldrin [7]
Non Lambertian PS	Coleman [16], Barsky [9], Schluns [47], Sato [46], Mallick [36], Alldrain [5], Goldman [24], Silver [51], Hertzmann [26], Zickler [62]
SL	Inokuchi [30]
VH	Szeliski [52], Matusik [39], Tarini [54]

**Table 3.1:** Selected algorithms from each class of algorithms

### 3.2 Class 1

In this section, we discuss techniques for reconstructing bright, textureless, and diffuse surfaces. The reconstruction of this type of surfaces is complicated by the fact



**Figure 3.1:** Top: A list of properties for object classes. Bottom: Six object classes of interest. Only texture, lightness, and reflectance are considered. Properties not considered are set as follows: opaque, rough (for Lambertian)/smooth (for Non-Lambertian), low concavity.

that there is no spatial information available for stereo correspondence searching due to the lack of texture, thus making passive stereo based methods unavailable. Textureless and diffuse surface also indicates uniform reflectance, i.e., uniform diffuse albedo. The conditions under which each categories of algorithms would work or fail are listed in Table 3.2.

Technique	Textureless	Bright	Lambertian
SfS	✓	✓	✓
MVS	✗	✓	✓
Lambertian PS	✓	✓	✓
Non Lambertian PS	✓	✓	✗
MVPS	✓	✓	✓
SL	✓	✓	✓
VH	✓	✓	✓

**Table 3.2:** Categories of algorithm that are applicable for object class 1.

### 3.2.1 SfS

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.

### 3.2.2 Lambertian PS: uniform 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 [60]. The surface orientation can be retrieved from using only two images.

To avoid the tedious process of light calibration, Silver [51] proposed a look-up scheme that relies on reflectance object with the same reflectance as the target, a uniform Lambertian surface in this case. This approach can be applied to surface with non-Lambertian reflectance, and varying colour or material, see Section 3.3 for non-Lambertian surfaces, and Section 3.5, 3.6 for surfaces with varying colour/-material.

### 3.2.3 SL

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.

## 3.3 Class 2

This section discusses the reconstruction of bright, textureless, non-Lambertian surfaces. The specular surfaces reflect light in a single direction which follows the law of reflection, thus the appearance changes as the viewpoint changes, which makes the correspondence search in these regions unreliable. Thus textureless and non-Lambertian properties rule out stereo based techniques. Textureless also indicates that the surface is composed of the same material thus the albedo is uniform across the surface. The conditions under which each categories of algorithms would work or fail are listed in Table 3.3.

Technique	Textureless	Bright	Non-Lambertian
SfS	✓	✓	✗
MVS	✗	✓	✗
Lambertian PS	✓	✓	✗
Non-Lambertian PS	✓	✓	✓
MVPS	✓	✓	✓
SL	✓	✓	✗
VH	✓	✓	✓

**Table 3.3:** Categories of algorithm that are applicable for object class 2.

### 3.3.1 Non-Lambertian PS: uniform reflectance

This section considers techniques that deal with surfaces composed of the same material, i.e., *uniform albedo*. Techniques that can deal with spatially varying reflectance can generally be adapted to deal with surfaces of uniform reflectance. Thus refer to Section 3.6 for more techniques that are designed for surfaces of spatially-varying reflectance.

As discussed briefly in Section 3.2, we can also use a non-Lambertian *reference*

*objects*. The method is first proposed by Silver and later by Hertzmann and Seitz.

### 3.4 Class 3

This section deals with bright, textured, Lamberian surfaces. Multi-view stereo can use the Lambertian, textured surface to reconstruct a (quasi-)dense model. The conditions under which each categories of algorithms would work or fail are listed in Table 3.4.

Technique	Textured	Bright	Lambertian
SfS	✗	✓	✓
MVS	✓	✓	✓
Lamberian PS	✓	✓	✓
Non Lambertian PS	✓	✓	✗
MVPS	✗	✓	✓
SL	✗	✓	✓
VH	✓	✓	✓

**Table 3.4:** Categories of algorithm that are applicable for object class 3.

#### 3.4.1 MVS

The diffuse and textured surface is most suitable for MVS algorithms. However, active technique could work, but the surface texture might interfere with the encoding process thus making the reconstruction inaccurate. Furukawa use wide-baseline stereo matching to recover the 3D coordinates of salient feature points, then shrink a visual hull model so that the recovered points lie on its surface, then refine the result using energy minimization. Goesele et al. compute a depth map from each camera viewpoint (similar to [31]) and merge the results using VRIP [62]. Esteban and Schmitt first compute a depth map from each camera viewpoint and merge the results into a cost volume. They then iteratively deform a mesh, initialized at the visual hull, to find a minimum cost surface in this volume, also incorporating terms to fit silhouettes. Kolmogorov and Zabih [35] compute a set of depth maps using multi-baseline stereo with graph cuts, then merge the results into a voxel volume by computing the intersections of the occluded volumes from each viewpoint. Faugeras and Keriven compute a minimum cost surface by

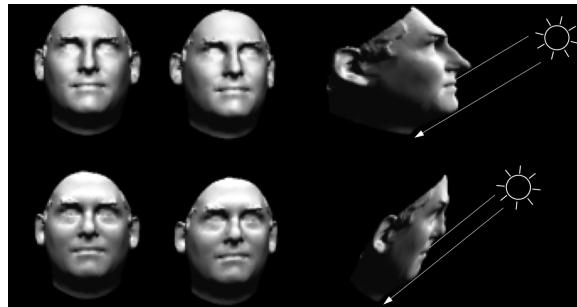
evolving a surface in a level-set frame-work, using a prediction-error measure. Vougiatzis et al. compute a correlation cost volume in the neighborhood of the visual hull. A minimum-cost surface is then computed using volumetric min-cut.

### 3.4.2 Lambertian PS: non-uniform reflectance

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. This section discusses techniques that deal with surfaces of *spatially varying albedo*.

The original photometric stereo can also be used for surfaces with spatially-varying albedo. The albedo-scaled normal can be first estimated as usual, then the albedo is retrieved as the magnitude of the scaled normal [60]. This method requires three instead of two images.

To avoid the tedious process of light calibration, *uncalibrated photometric stereo* has been proposed. 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 [10]. 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

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.

### 3.5 Class 4

The dark surface, i.e., low reflectance makes any active lighting based technique unsuitable. Thus passive stereo based techniques still work in this case, see Section 3.4 for details.

Technique	Textured	Dark	Lambertian
SfS	✗	✗	✓
MVS	✓	✓	✓
Lamberian PS	✓	✗	✓
Non Lambertian PS	✓	✗	✗
MVPS	✗	✗	✓
SL	✗	✗	✓
VH	✓	✓	✓

**Table 3.5:** Categories of algorithm that are applicable for object class 4.

### 3.6 Class 5

Technique	Textured	Bright	Non-Lamberian
SfS	✗	✓	✗
MVS	✓	✓	✗
Lamberian PS	✓	✓	✗
Non Lambertian PS	✓	✓	✓
MVPS	✗	✓	✓
SL	✗	✓	✗
VH	✓	✓	✓

**Table 3.6:** Categories of algorithm that are applicable for object class 5.

### 3.6.1 Non-Lambertian PS: non-uniform reflectance

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 approximations 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.

## 3.7 Class 6

We discuss textured, mixed diffuse and specular, and bright surface, which is challenging for MVS due to the specular nature, and is difficult for any active techniques since the low amount of reflected light.

Technique	Textured	Dark	Non-Lamberian
SfS	✗	✗	✗
MVS	✓	✓	✗
Lamberian PS	✓	✗	✗
Non Lambertian PS	✓	✗	✓
MVPS	✓	✗	✓
SL	✗	✗	✗
VH	✓	✓	✓

**Table 3.7:** Categories of algorithm that are applicable for object class 6.

### 3.7.1 VH

The non-Lambertian property makes any stereo based methods unaviable while the dark surface, or low reflectance makes active lighting methods unsuitable. Thus only visual hull fits in this case, the summary of VH algorithms are listed in Table 3.7.

Technique	Representation	Algorithm
Szeliski [52]	3D grids	Voxel-based
Tarini [54]	3D rays	MI-based
Matusik [39]	Polygonal mesh	Exact polyhedral methods

**Table 3.8:** Summary of VH representations and reconstruction approach.

## 3.8 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 3.9.

<b>Technique</b>	Translucency	Texture	Lightness	Reflectance	Roughness	Concavity	Class
Horn [28]	Opaque	Textureless	Bright	Lambertian	N/A	Convex	Class 1
Woodham [60]	Opaque	N/A	Bright	Lambertian	N/A	Convex	Class 1, 3
Hayakawa [25]	Opaque	N/A	Bright	Lambertian	N/A	Convex	Class 1, 3
Belhumeur [10]	Opaque	N/A	Bright	Lambertian	N/A	Convex	Class 1, 3
Coleman [16]	Opaque	N/A	Bright	Non-Lambertian	N/A	Convex	Class 2, 5
Barsky [9]	Opaque	N/A	Bright	Non-Lambertian	N/A	Convex	Class 2, 5
Schluns [47]	Opaque	N/A	Bright	Non-Lambertian	N/A	Convex	Class 2, 5
Sato [46]	Opaque	N/A	Bright	Non-Lambertian	N/A	Convex	Class 2, 5
Mallick [36]	Opaque	N/A	Bright	Non-Lambertian	N/A	Convex	Class 2, 5
Alldrain [5]	Opaque	N/A	Bright	Non-Lambertian	N/A	Convex	Class 2, 5
Goldman [24]	Opaque	N/A	Bright	Non-Lambertian	N/A	Convex	Class 2, 5
Silver [51]	Opaque	N/A	Bright	Non-Lambertian	N/A	Convex	Class 1, 2
Hertzmann [26]	Opaque	N/A	Bright	Non-Lambertian	N/A	Convex	Class 1, 2, 3, 5
Zickler [62]	Opaque	N/A	Bright	Non-Lambertian	N/A	Convex	Class 3, 5
Furukawa [20]	Opaque	Textured	N/A	Lambertian	N/A	Convex	Class 3, 4
Goesele [23]	Opaque	Textured	N/A	Lambertian	N/A	Convex	Class 3, 4
Vogiatzis [58]	Opaque	Textured	N/A	Lambertian	N/A	Convex	Class 3, 4
Szeliski [52]	Opaque	N/A	N/A	N/A	N/A	Convex	Class 1-6
Tarini [54]	Opaque	N/A	N/A	N/A	N/A	Convex	Class 1-6
Matusik [39]	Opaque	N/A	N/A	N/A	N/A	Convex	Class 1-6

**Table 3.9:** 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 properties of the object. In this chapter, we attempt to extend this mapping by providing a model of 3D reconstruction 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 reconstruction.

We first propose a formal definition of the 3D reconstruction problem in Section 4.1. Section 4.2 discusses various key *properties* that can be used to describe the appearance of the object. Section 4.3 explores the inputs and outputs used in 3D reconstruction problems. Section 4.4 provides the mapping of the representations and properties into a formal model via which 3D reconstruction problem can be expressed. These 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 closely related to the reconstruction problem before a formal defini-

tion 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.

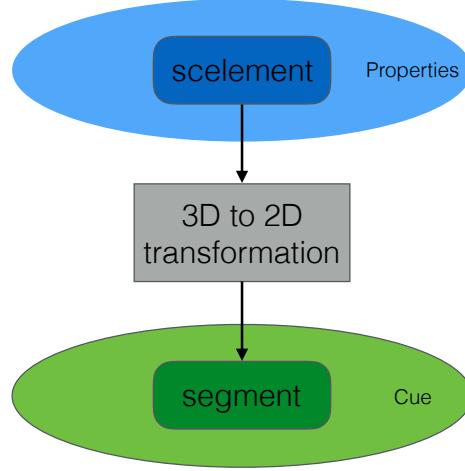
A scell can be considered as a generalization of a voxel. However, a scell is not necessarily distinct since

**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 an 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  $seg$ .

$$consist(prop(sc), cue(seg)) = x, x \in [0, 1]$$

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

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

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

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

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

**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  $\{\text{seg}_{(i,n)}\}_{i=0,j=0}^{M-1,N-1}$  are maximized.

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

## 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 [14]. For instance, surface orientation is one component of surface geometry model, and the corresponding representaion 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 4.1.

Component	Texture	Lightness	Reflectance	Roughness	Concavity
-----------	---------	-----------	-------------	-----------	-----------

**Table 4.1:** Model of the 3D reconstruction problem.

## 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 informa-

tion 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*.

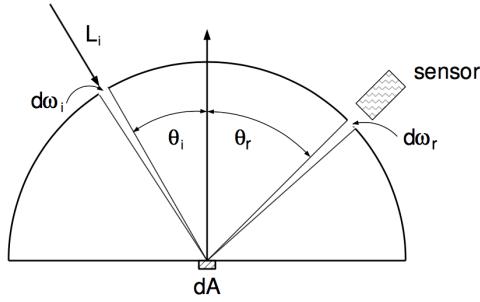
### Definition of Radiometric Terms

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  $(\theta_i, \phi_i)$  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,  $\mathbf{E}_i(\theta_i, \phi_i) = \int_{\Omega_i} L_i(\theta_i, \phi_i) d\Omega_i (\text{watt}/\text{m}^2)$ .
- Surface radiance ( $\mathbf{L}_r(\theta_r, \phi_r)$ ): light flux emmited from a unit surface area in the direction  $(\theta_r, \phi_r)$ , unit ( $\text{watt} \cdot \text{m}^{-2} \cdot \text{steradian}^{-1}$ ).

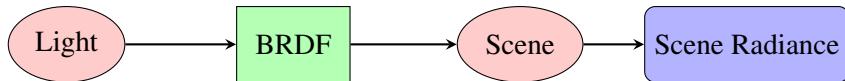
### Light-matter interaction

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



**Figure 4.2:** Light-matter interaction

**Definition (BRDF)** the ratio of the surface radiance  $\mathbf{L}_r(\theta_r, \phi_r)$  to the irradiance  $\mathbf{E}_i(\theta_i, \phi_i)$ , i.e.,  $f(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{\mathbf{E}_{surface}(\theta_i, \phi_i)}{\mathbf{L}_{surface}(\theta_r, \phi_r)}$ .

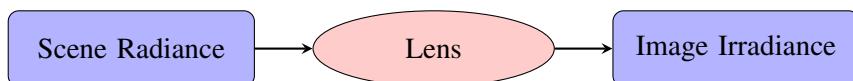


**Figure 4.3:** The light-matter interaction.

**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 [40]. 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.



**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.

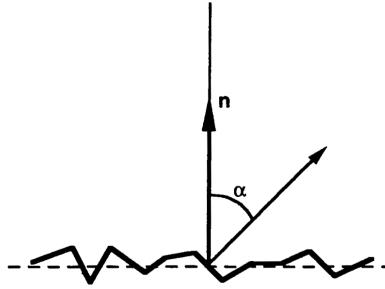
#### 4.3.4 Roughness

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 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.



**Figure 4.7:** Surface Slope Distribution Model

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  $\alpha$ .

We will use the parameter  $\alpha$  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 for the facet slope  $\alpha$ , with mean value  $\bar{\alpha} = 0$  and standard deviation  $\sigma_\alpha$ , and larger  $\sigma_\alpha$  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 4.2.

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 4.2:** Expression of the reconstruction problem for the object class 1, 2, 3, 5.

## Chapter 5

# A Mapping of 3D Reconstruction Techniques

Most of the vision work focus 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 problem or application domain or the conditions. This section build on the 3D description proposed in Chapter 4, and attempts to find out the optimal algorithms for all possible conditions.

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 of the online dataset. To the best of our knowledge, current existing 3D benchmarks focus on one specific class of algorithms, for example, the Middlebury dataset is targeted to MVS algorithms, and the ‘DiLiGenT’ dataset is for Photometric Stereo algorithms. This makes them suitable only to the evaluation of algorithms within the same category. There is no dataset that evaluates 3D reconstruction across differ categories, let alone one that covers a range of properties and their combinations. The reasons for the lack of such dataset is: 1). it’s tedious to create a real-world dataset for a specific category of algorithm, it would be more challenging to create datasets for a range of categories with the ground truth; 2). it’s practically impossible to make one property (e.g., , noise level, lighting configuration, material, etc) varied while fixing the other in order to conduct a thorough evaluation.

We propose a synthetic but realistic (physically-based) dataset for evaluation of 3D reconstruction algorithms. The dataset includes a collection of images of a scene under different material 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 physical-based renderer Cycles in Blender to generate the synthetic dataset. For each technique, the configuration of the camera remains fixed. The image resolution is  $1280 \times 720$ , with a focal length of  $35mm$ .

For MVS, 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  $360^\circ$ . Thus there are in total  $12 + 12 + 8 + 8 + 1 = 41$  cameras.

For photometric stereo, according to [11], 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. 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  $45^\circ$ . The azimuth angle between two neighbouring light sources is  $45^\circ$ .

For the structured light, the baseline angle between the camera and the projector is  $10^\circ$ , and only one camera is used, thus only a portion of the object is invisible. The resolution of the projector is  $1024 \times 768$ , 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 the combinations of properties. For if we have  $N$  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). first we investigate the *depen-*

dency between any two properties, if these two properties are independent, there is no need to render all their combinations whereas it's necessary to do so if they are dependent; 2). render all the dependent properties and their combinations.

### 5.3 Selected methods

We have selected one representative algorithm from each class of algorithms: the PMVS proposed in [20], the example-based photometric stereo proposed in [26], and the Gray-encoded structured light technique. 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.

### 5.4 Evaluation metrics

We use the metric proposed in [49] 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. The completeness measures 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 and the completeness value goes up. To make it more consistent to the natural context, we say the accuracy goes up when the accuracy ‘value’ goes down.

For photometric stereo, we employ another evaluation criteria, which is based on the statistics of angular error. For each pixel, the angular error is calculated as  $\arccos(n_g^T n)$  in degrees, where  $n_g$  and  $n$  are ground truth and estimated normals respectively. In addition to the mean angular error, we also calculate the minimum, maximum, median, the first quartile, and the third quartile of angular errors for each estimated normal map.

## 5.5 Dependency Check

Part of the difficulty in establishing a comprehensive set of experiments for such an evaluation is the large variability of shapes and material properties. We conduct a dependency check that evaluates the performance of the algorithm by changing the values of two properties at a time while fixating the others. The goal is to identify the dependent properties so that the dimension of the problem domain could become more manageable.

### 5.5.1 PMVS

We evaluate the performance of PMVS in terms of accuracy and completeness under varied combinations of properties, the settings of the properties and all their combinations are listed in Table 5.1.

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)	0.8	[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 5.1:** Parameter settings of dependency check of MVS.

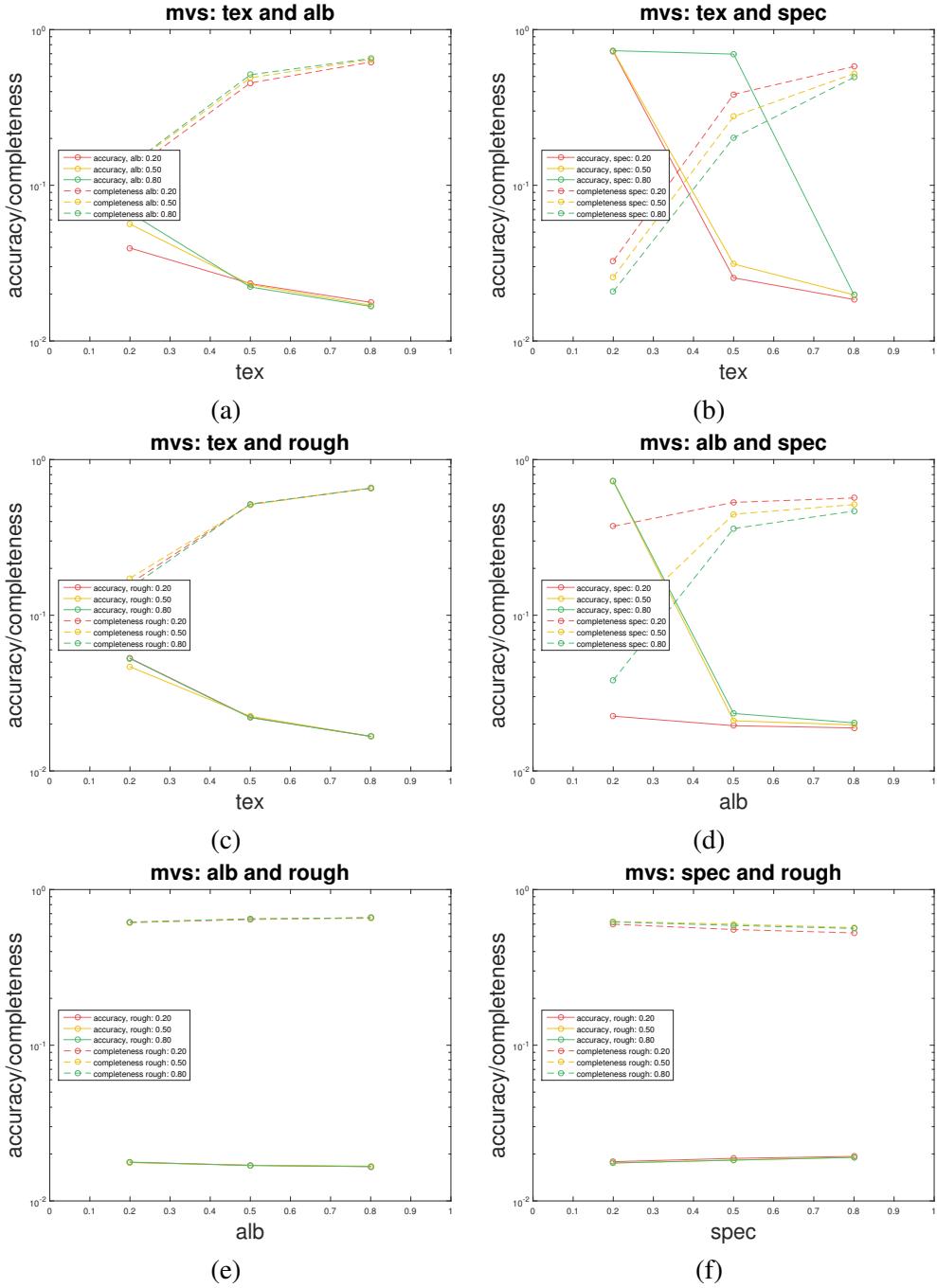
Now let's investigate the correlation between each property or each pair of properties and the chosen metrics, as shown in Table

#### Single property

**Texture** - (a), (b), (c). As the texture level increases, the accuracy and completeness both increase. Thus texture has a positive correlation with the accuracy and completeness of the reconstruction.

**Albedo** - (a), (d), (e). we can tell that albedo doesn't have an effect when texture or roughness changes. But when the specular level changes, albedo has an effect on both accuracy and completeness of the reconstruction.

**Specular** - (b), (d), (f). As the specular level goes up, the accuracy and com-



**Figure 5.1:** Performance of MVS with varied properties

pleteness of the reconstruction gets worse. Thus specular has an negative correlation with the accuracy and completeness of the reconstruction.

**Roughness** - (c), (e), (f). We can see that roughness doesn't have a influence on the reconstruction.

### Pairwise properties

Now let's investigate the influence of pairwise properties. We observed that (b), and (d) has significant changes in terms of accuracy and completeness when one property changes while the other is fixed.

**(b) Texture and Specularity** For a fixed texture, as the specularity goes up, the accuracy and the completeness goes up, which is consistent to previous observations. Besides, for a lower value texture, the effect of specular is more substantial than that for a higher value texture.

**(d) Albedo and Specularity** For a fixed albedo, as the specular goes up, the accuracy and completeness both goes down, which is consistent to previous observations. However, the effect of specular is more substantial for a lower value albedo than that for a higher value albedo.

**Conclusion** the properties that have an effect on the MVS are: texture, albedo, and specularity, as shown in Table 5.2. Thus, we will only consider these three properties for all forthcoming discussion of MVS.

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

**Table 5.2:** The correlation between each property and the metrics *accuracy* and *completeness*.

### 5.5.2 Example-based PS

We evaluate the performance of example-based PS in terms of angular difference under varied combinations of properties, The statistical measures that we used include median, mean, first and third quartile of the angular difference. We investigate two properties at a time. The settings of the properties and all their combina-

tions are listed in Table 5.3.

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

**Table 5.3:** Parameter settings of dependency check of example-based PS.

### 5.5.3 Single property

**Texture** - (a), (b), (c). As the texture level increases, all statistic measures of the angular difference remain almost the same. Thus texture doesn't affect the reconstruction of the chosen PS algorithm.

**Albedo** - (a), (d), (e). As the albedo level increases, all statistic measures of the angular difference decreases. Thus the albedo has a positive correlation to the reconstruction.

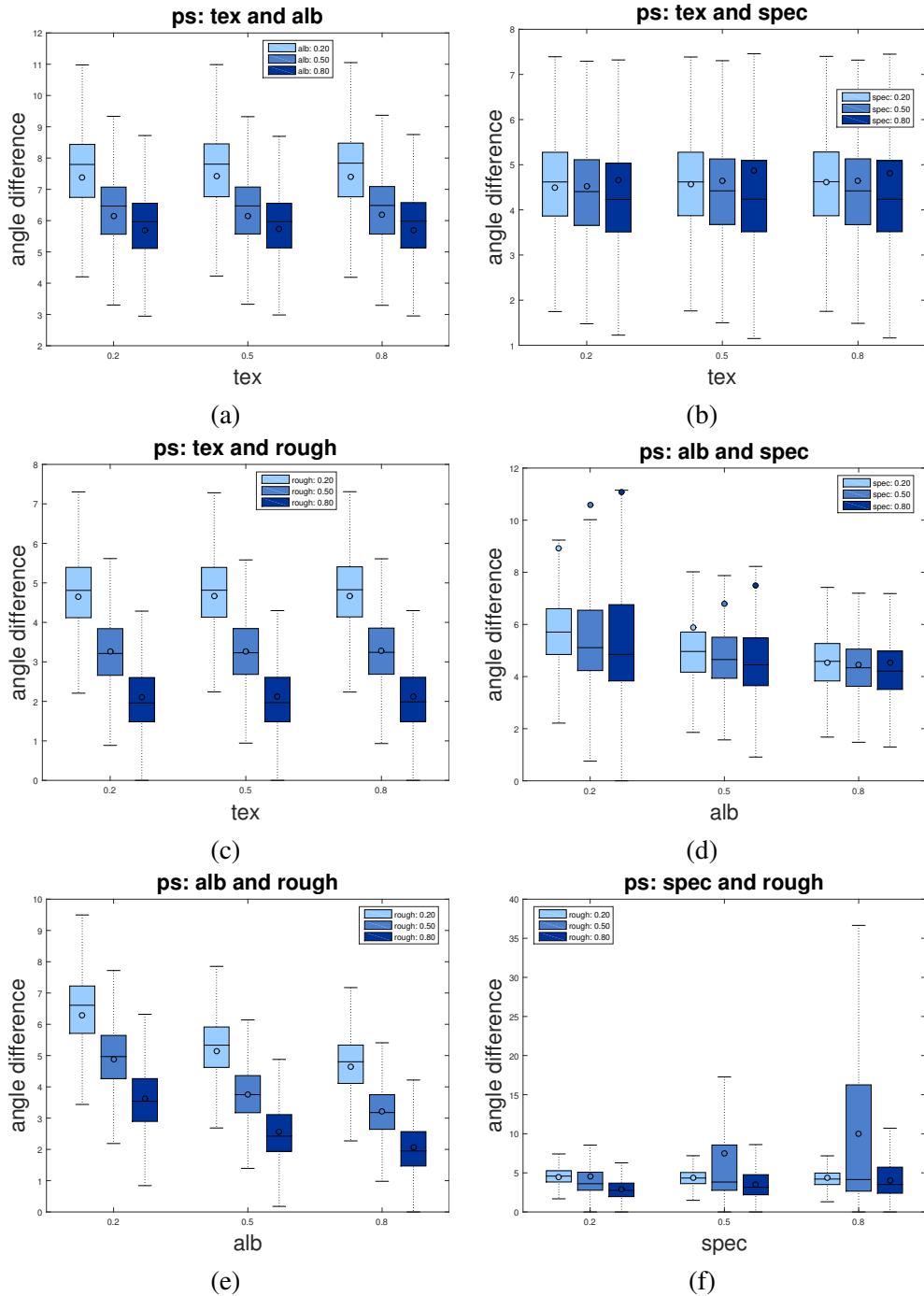
**Specular** - (b), (d), (f). As the specular level goes up, all statistic measures of the angular difference increases. Thus specular has an negative impact on the reconstruction.

**Roughness** - (c), (e), (f). The effect of roughness is a bit complicated. Generally, it will improve the reconstruction as the roughness goes higher. However, as shown in Figure 5.2 (f), the roughness will cause worse reconstruction for medium high value, which will be discussed more later.

### 5.5.4 Pairwise properties

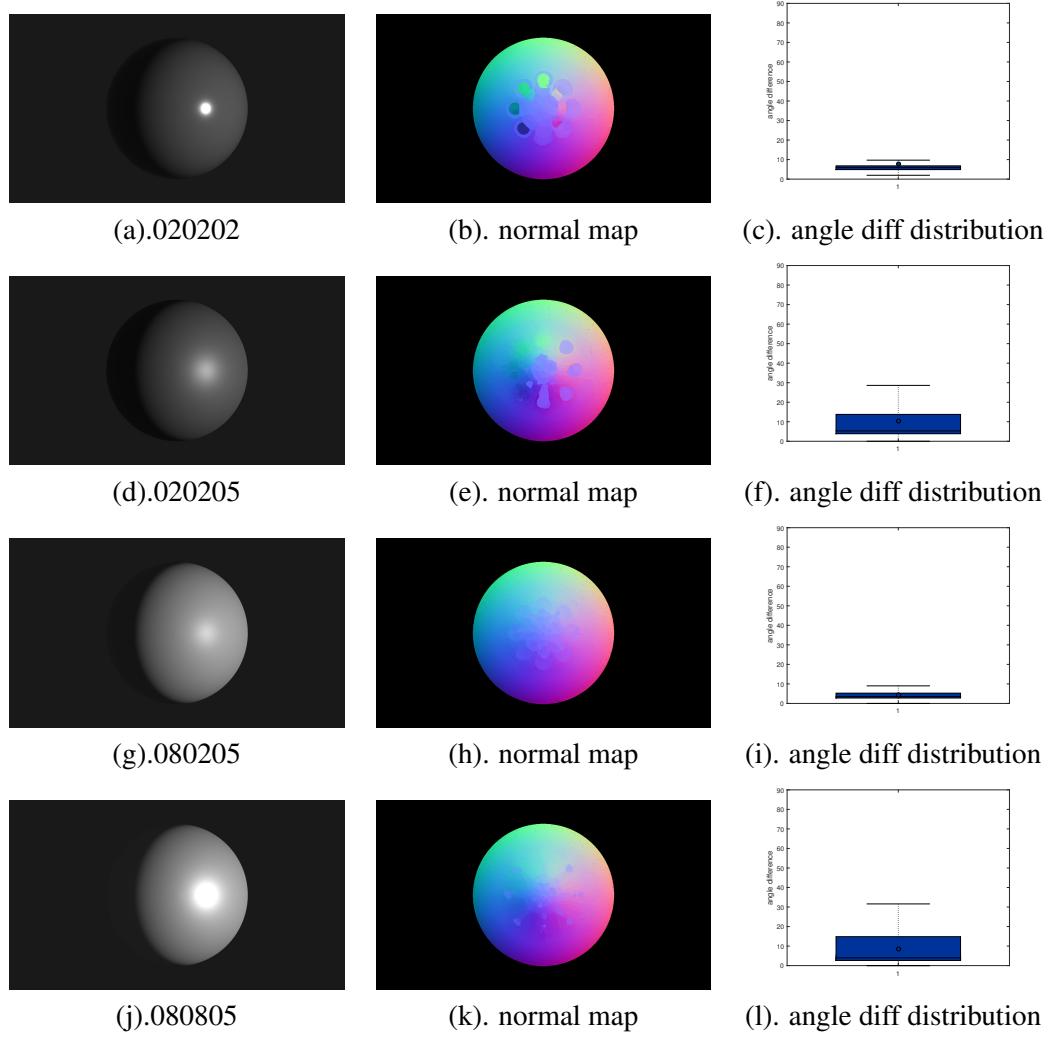
All the pairwise relations are straightforward except for (f).

**(f) Specularity and Roughness** For a fixed specularity, if the specularity is lower, the effect of roughness is less noticeable, whereas if the specularity is higher, the effect of roughness becomes more substantial. We've also noticed a 'peculiar' case when roughness is 0.5, it makes the reconstruction worse, which is counter-intuitive. However, we argue that it's because the roughness effect is not strong



**Figure 5.2:** Performance of PS with varied properties

enough to cancel out the specularity, thus causing a much larger area of ‘blurred’ specularity, which makes the reconstruction worse. This effect is also demonstrated in the training stage, see Figure 5.3 for some visual examples.



**Figure 5.3:** The ‘peculiar’ effect of roughness on PS. The order of the property is: albedo, specular, and roughness, thus 080205 means albedo: 0.8, specular: 0.2, and roughness: 0.5

**Conclusion** the properties that have an effect on the PS are: albedo, specularity,

and roughness, as shown in Table 5.4. Therefore, we will only consider these three properties for all forthcoming discussion of PS.

Metric	Texture	Albedo	Specular	Roughness
Angle difference	✗	✓	✓	✓

**Table 5.4:** The correlation between each property and the metric *angular difference*.

### 5.5.5 Gray-code SL

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

Property	Texture coverage	Albedo	Specular/Diffuse ratio	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 5.5:** Parameter settings of dependency check of Gray code SL.

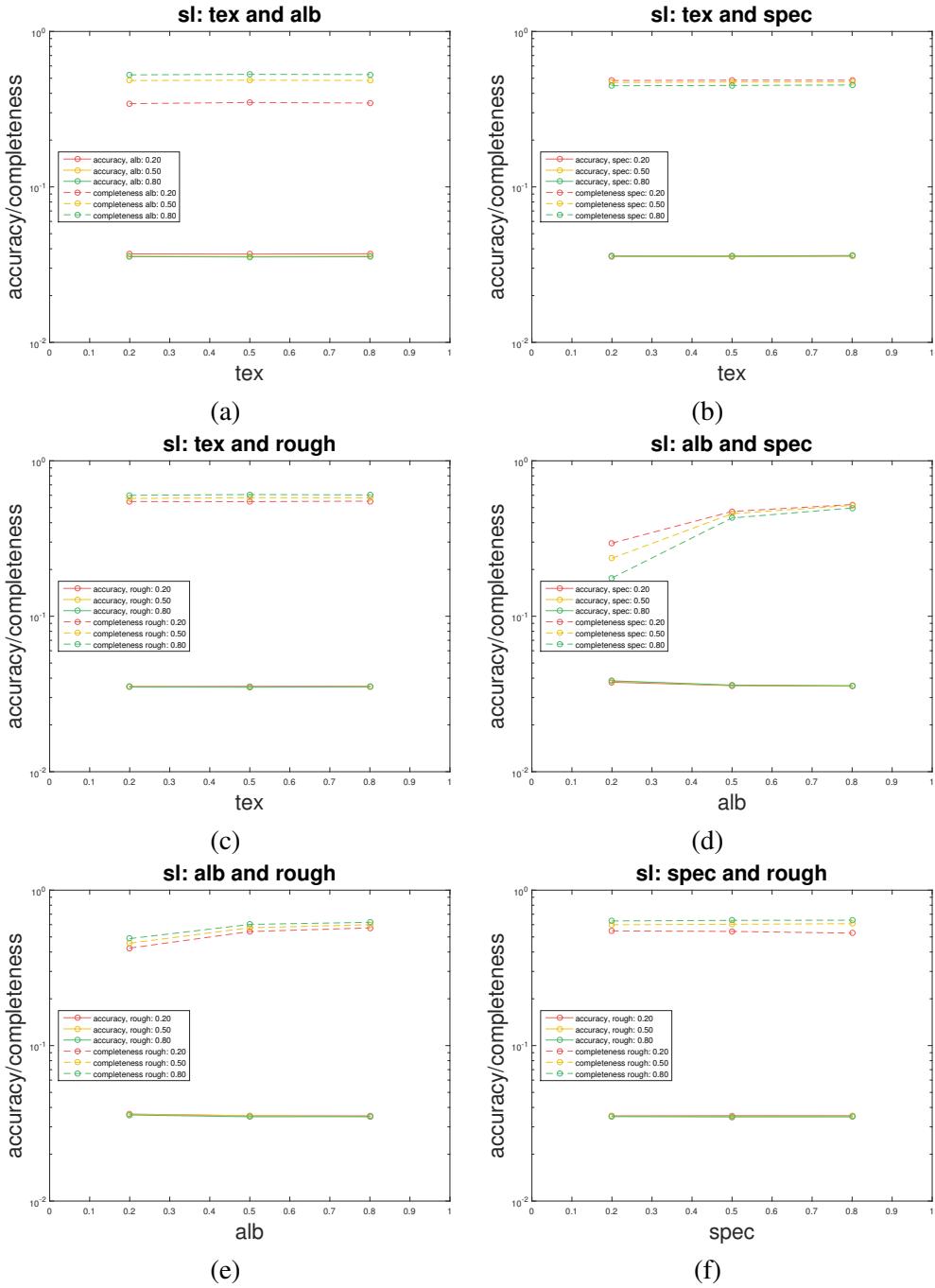
### 5.5.6 Single property

A depth check step is performed to remove erroneous depth, thus the accuracy remain almost constant across all cases.

**Texture** - (a), (b), (c). As the texture level increases, the accuracy and completeness remain almost constant. Thus texture doesn't affect the reconstruction of the chosen SL algorithm.

**Albedo** - (a), (d), (e). As the albedo level increases, the accuracy remain almost constant while the completeness increases. Thus albedo has a positive correlation with completeness.

**Specular** - (b), (d), (f). As the specular level goes up, the accuracy remain al-



**Figure 5.4:** Performance of SL with varied properties

most constant while the completeness of the reconstruction decreases. Thus specular has an negative correlation with the completeness of the reconstruction.

**Roughness - (c), (e), (f).** As the roughness level increases, the accuracy remain almost constant while the completeness increases. Thus roughness has a positive correlation with completeness.

### 5.5.7 Pairwise properties

**(d) Albedo and Specularity** For a fixed albedo, the completeness goes down as the specularity goes up for low albedo surface, this effect becomes less substantial when the albedo increases. Thus we conclude that the effect of specular is most significant when the albedo is low.

**Conclusion** the properties that have an effect on the SL are: texture, albedo, specularity, as shown in Table. Therefore, we will only consider these three properties for all forthcoming discussion of SL.

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

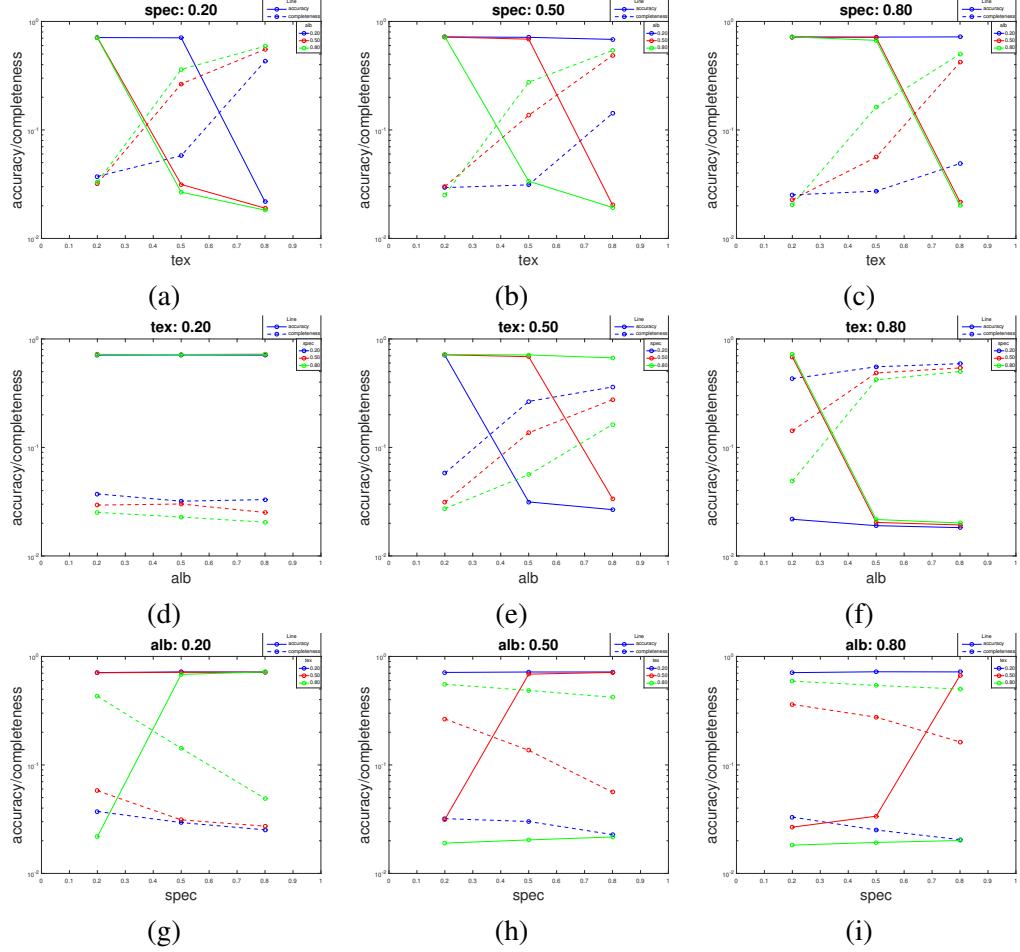
**Table 5.6:** The correlation between each property and the metrics *accuracy* and *completeness*.

## 5.6 Training and mapping

For each technique, we generate the synthetic dataset using only the dependent properties, thus there are  $L \times L \times L$  different combinations for each technique, where  $L$  is the number of levels for each property.

### 5.6.1 PMVS

The performance of PMVS under difference combinations of properties is shown in Figure 5.5. The conditions that PMVS works well is listed in Table ??.



**Figure 5.5:** Performance of MVS with varied properties.

### 5.6.2 Example-based PS

The performance of example-based PS under difference combinations of properties is shown in Figure 5.6. The conditions that example-based PS works well is listed in Table ??.

### 5.6.3 Gray-code SL

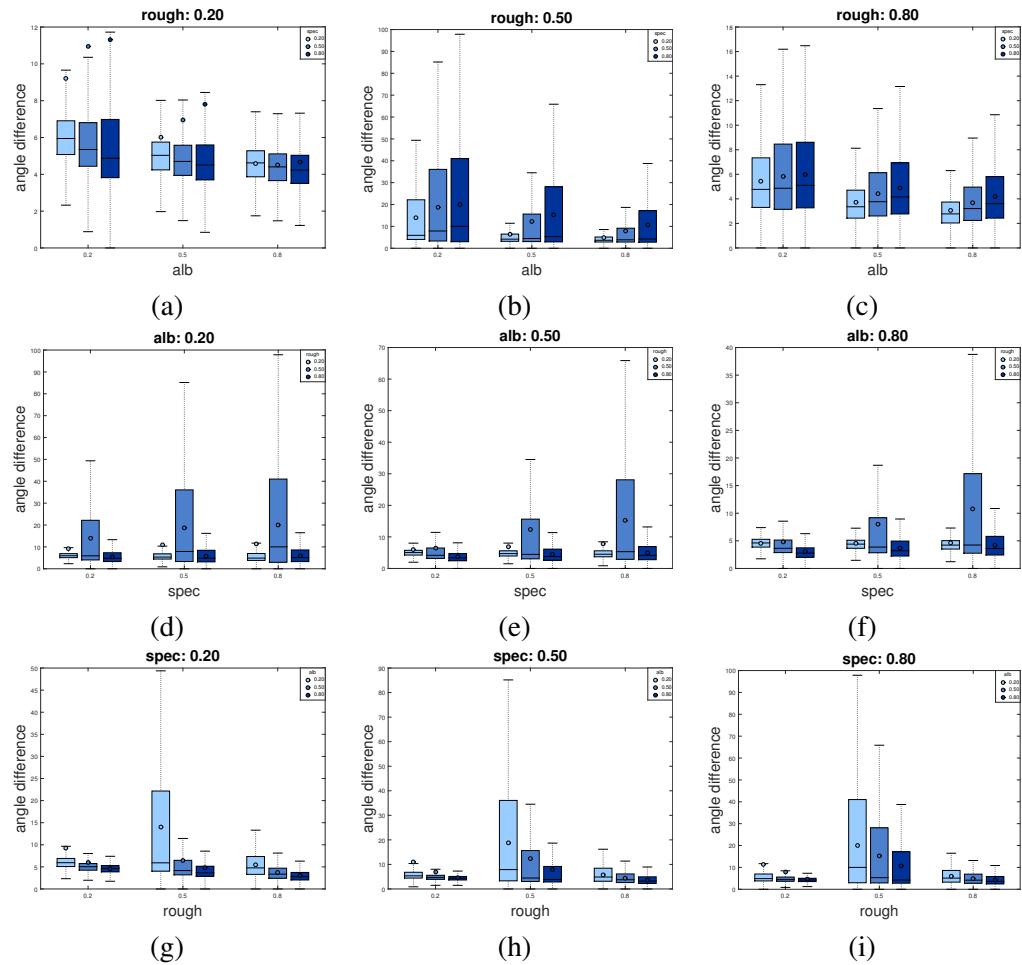
The performance of Gray code SL under difference combinations of properties is shown in Figure 5.7. The conditions that PMVS works well is listed in Table ??.

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

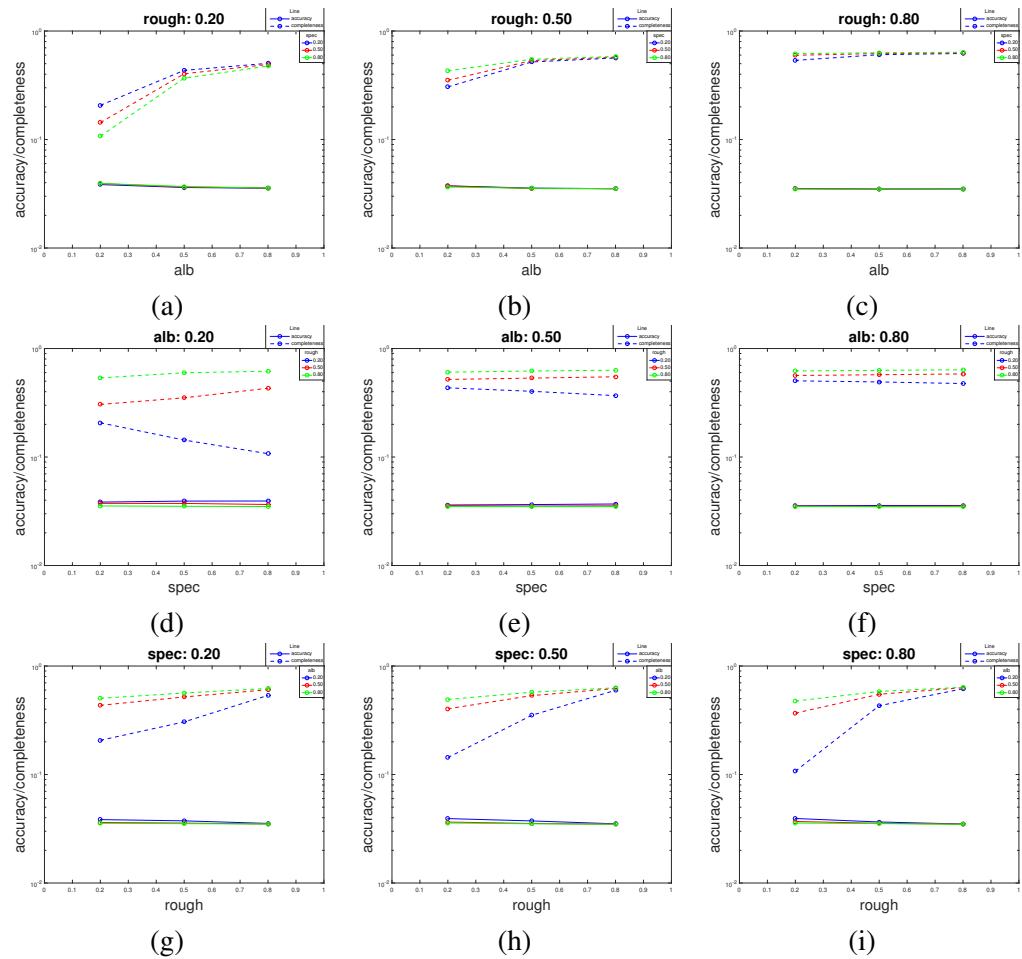
**Table 5.7:** The conditions under which PMVS works well in terms of the two metrics *accuracy* and *completeness*.

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

**Table 5.8:** The conditions under which example-based PS works well in terms of the two metric *angular difference*.



**Figure 5.6:** Performance of PS with varied properties.



**Figure 5.7:** Performance of SL with varied properties.

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

**Table 5.9:** The conditions under which Gray code SL works well in terms of the two metrics *accuracy* and *completeness*.

# **Chapter 6**

## **Interpretation of 3D Reconstruction Model**

In order to validate the 3D reconstruction mapping derived from the Chapter 5, evaluation of the object centric model into appropriate solutions must be shown. Our interpreter is based on the direct evaluation of the performance of each 3D reconstruction algorithm under different conditions presented in Chapter 5. From this analysis of how algorithms perform on objects which have different visual and geometric properties, an algorithm(s) can be definitively chosen based on which performed best on the training images.

Although only three algorithms selected, all of which are the top performers in the corresponding field, and are sufficient to demonstrate the framework's ability to translate the descriptive model into a reconstruction. Furthermore, the integration of a new algorithm requires the same process presented in Chapter 5, allowing researchers to contribute novel algorithms to the framework.

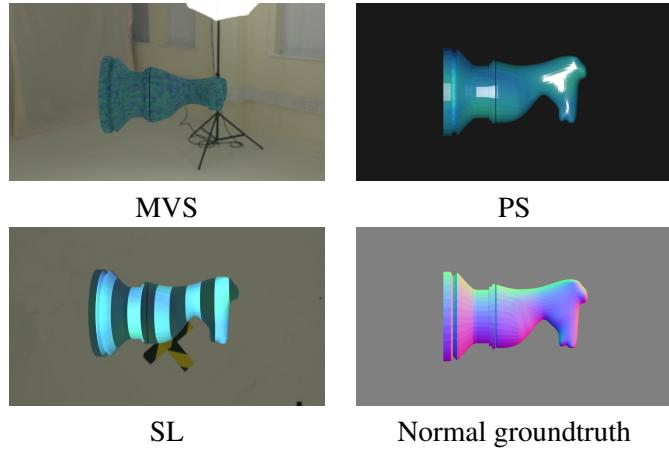
### **6.1 Synthetic Datasets**

We use one object shown in Figure 6.1, and four property settings in Table 6.1 to test the validity of the abstraction. Those four settings represent four classes of objects discussed in Chapter 4. The best suited algorithm as suggested by the mapping derived from Chapter 5 is included.

Property	Texture	Albedo	Specular	Roughness	Best-suited techniques
(a)	0.2	0.8	0.2	0.8	PS, SL
(b)	0.2	0.8	0.5	0.2	PS, SL
(c)	0.8	0.8	0.2	0.8	MVS, PS, SL
(d)	0.8	0.8	0.5	0.2	MVS, PS, SL

**Table 6.1:** Property lists of the test objects. Link to the labels in Figure ??,

(a): dark blue rectangle, (b): dark green rectangle, (c): light blud rectangle, (d): light green rectangle.

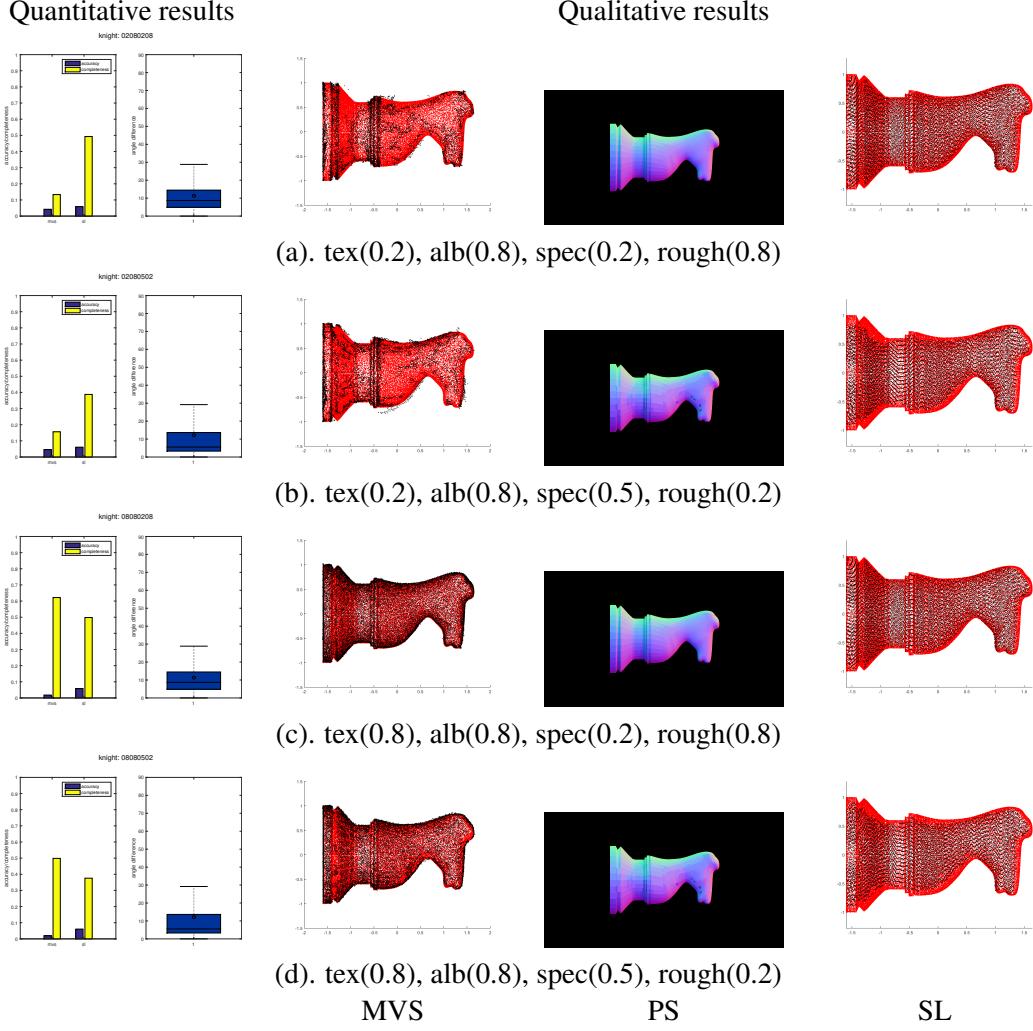


**Figure 6.1:** The synthetic datasets and groundtruth for the evaluation

Now we show both the quantitative results and qualitative results of the test objects, and see if the results is consistent with the techniques selected by our abstraction.

**Case 1&2** Both example-based PS and Gray coded SL perform relatively well, as suggested by the mapping. The completeness of the PMVS is low due to the lack of texture. The results are consistent to those returned by the mapping which is shown in Table ?? (a) and (b).

**Case 3&4** All three algorithms perform relatively well, which is also consistent to the results returned by the mapping, as shown in Table 6.1 (c) - (d).



**Figure 6.2:** 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.

## 6.2 Real-world Datasets

We used a similar setup to the synthetic settings and captured a real world dataset for 11 objects. The property of these objects are listed in Table 6.2.

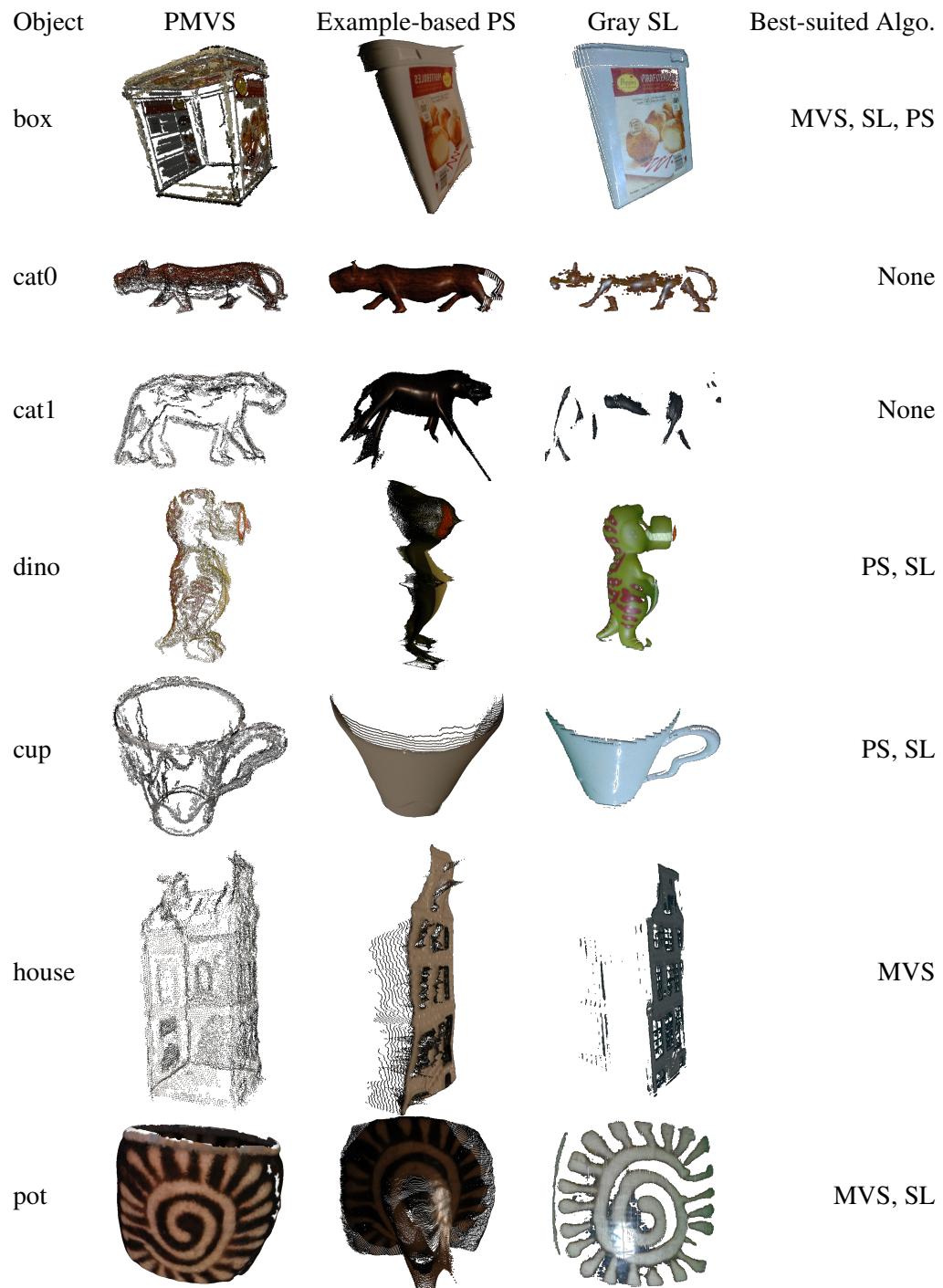
In Figure 6.4, we show the reconstructions of the real-world dataset. Since we

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

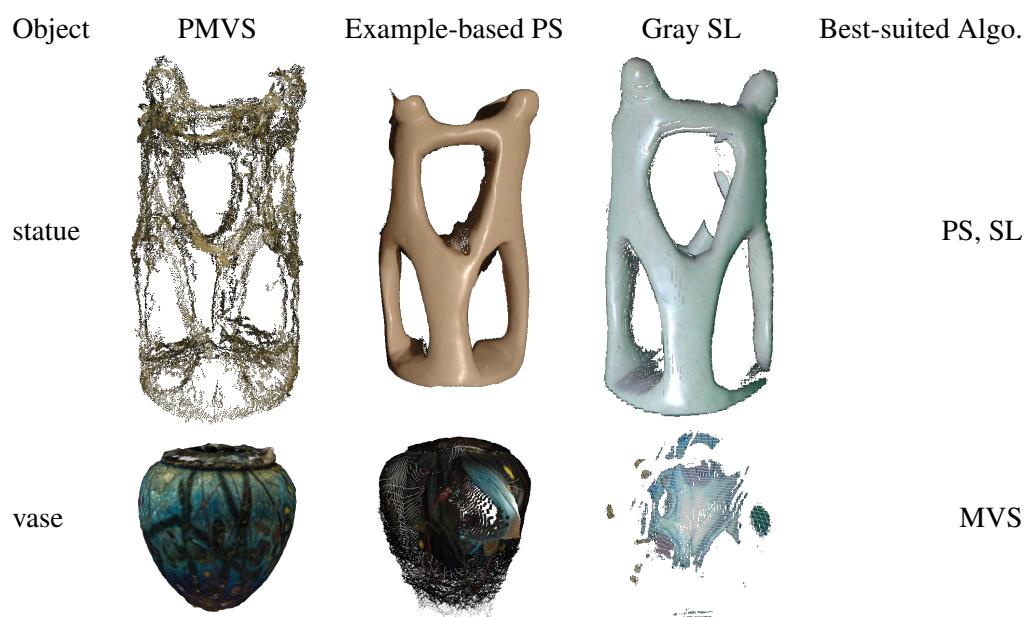
**Table 6.2:** Property list for the real-world objects

don't have the ground truth, no quantitative measures are available, only visual inspection is used. We can show that the qualitative results match the results returned by the mapping.

### 6.3 Summary



**Figure 6.3:** Reconstruction results of MVS, PS, SL



**Figure 6.4:** Reconstruction results of MVS, PS, SL (cont'd)

# Bibliography

- [1] Autodesk. URL <http://en.wikipedia.org/wiki/Autodesk>. → pages 1
- [2] Lidar. URL <http://en.wikipedia.org/wiki/Lidar>. → pages 1
- [3] Kinect. URL <http://en.wikipedia.org/wiki/Kinect>. → pages 1
- [4] Vxl c++ libraries for computer vision research and implementation.  
<http://vxl.sourceforge.net>. → pages 8
- [5] 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 23, 30, 32
- [6] 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 16, 30
- [7] 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 23
- [8] 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
- [9] 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, 23, 30, 32

- [10] 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 23, 28, 32
- [11] S. Berkiten and S. Rusinkiewicz. An RGBN benchmark. Technical report, Technical Report TR-977-16, Princeton University, Feb. 2016. → pages 46
- [12] 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
- [13] F. Blais. Review of 20 years of range sensor development. *Journal of Electronic Imaging*, 13(1), 2004. → pages 10
- [14] 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 37
- [15] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library.* ” O’Reilly Media, Inc.”, 2008. → pages 8
- [16] 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, 23, 30, 32
- [17] 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, 23, 27
- [18] O. Faugeras and R. Keriven. *Variational principles, surface evolution, pde’s, level set methods and the stereo problem*. IEEE, 2002. → pages 1, 10, 23, 27
- [19] Y. Furukawa. *High-fidelity image-based modeling*. University of Illinois at Urbana-Champaign, 2008. → pages 27
- [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, 23, 32, 47
- [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 22
- [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, 23, 27, 32
- [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 16, 23, 30, 32
- [25] H. Hayakawa. Photometric stereo under a light source with arbitrary motion. *JOSA A*, 11(11):3079–3089, 1994. → pages 15, 23, 28, 32
- [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, 23, 27, 29, 32, 47
- [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, 23, 25, 32
- [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 23
- [30] S. Inokuchi. Range-imaging system for 3d object recognition. In *Proc. of 7th International Conference on Pattern Recognition, 1984*, 1984. → pages 23
- [31] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. Available from: <<http://www.peterkovesi.com/matlabfns/>>. → pages 8
- [32] 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

- [33] 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
- [34] 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
- [35] 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
- [36] 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, 23, 30, 32
- [37] 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
- [38] D. Marr. Vision: A computational investigation into the human representation and processing of visual information. 1982. → pages 10
- [39] 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 23, 31, 32
- [40] S. K. Nayar, K. Ikeuchi, and T. Kanade. Surface reflection: physical and geometrical perspectives. Technical report, DTIC Document, 1989. → pages 41
- [41] G. P. Otto and T. K. Chau. region-growingalgorithm for matching of terrain images. *Image and vision computing*, 7(2):83–94, 1989. → pages 10
- [42] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(6035):314–319, 1985. → pages 9
- [43] 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
- [44] 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
  - [45] J. Salvi, J. Pages, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern recognition*, 37(4):827–849, 2004. → pages 12, 22
  - [46] Y. Sato and K. Ikeuchi. Temporal-color space analysis of reflection. *JOSA A*, 11(11):2990–3002, 1994. → pages 16, 23, 30, 32
  - [47] 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, 23, 30, 32
  - [48] 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
  - [49] 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, 22, 47
  - [50] 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 22
  - [51] W. M. Silver. *Determining shape and reflectance using multiple images*. PhD thesis, Massachusetts Institute of Technology, 1980. → pages 16, 23, 25, 27, 28, 32
  - [52] R. Szeliski. Rapid octree construction from image sequences. *CVGIP: Image understanding*, 58(1):23–32, 1993. → pages 23, 31, 32
  - [53] 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

- [54] 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 23, 31, 32
- [55] 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
- [56] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008. → pages 8
- [57] 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
- [58] 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, 23, 28, 32
- [59] 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
- [60] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):191139–191139, 1980. → pages 1, 14, 23, 25, 28, 32
- [61] E. Zheng, E. Dunn, V. Jojic, 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
- [62] 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 16, 23, 30, 32

## **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