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

Abstract

This document provides brief instructions for using the `ubcdiss` class to write a UBC-conformant dissertation in LaTeX. This document is itself written using the `ubcdiss` class and is intended to serve as an example of writing a dissertation in LaTeX. 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

List of Tables

List of Figures

Glossary

This glossary uses the handy `acroynym` 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, 8, 27] and passive image-based methods [16, 17, 20, 26] applied successfully to some sub-domains of the problem. Among the existing techniques, active techniques such as laser scanner [27], structured light system (SL) [8], and Photometric Stereo (PS) [45], and passive method such as Multi-view Stereo (MVS) [40] 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 calibration, 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.

Possibly more???

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, specularity,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.

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.

Model 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.

Benchmark 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:

- A new taxonomy of 3D reconstruction problem from object-centered perspective;
- A model of 3D reconstruction that can describe 3D reconstruction problem distincitvely without loss of generability;
- An abstraction of 3D reconstruction that maps problem model to a suite of algorithms.

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 model 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 abstraction 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 abstraction.

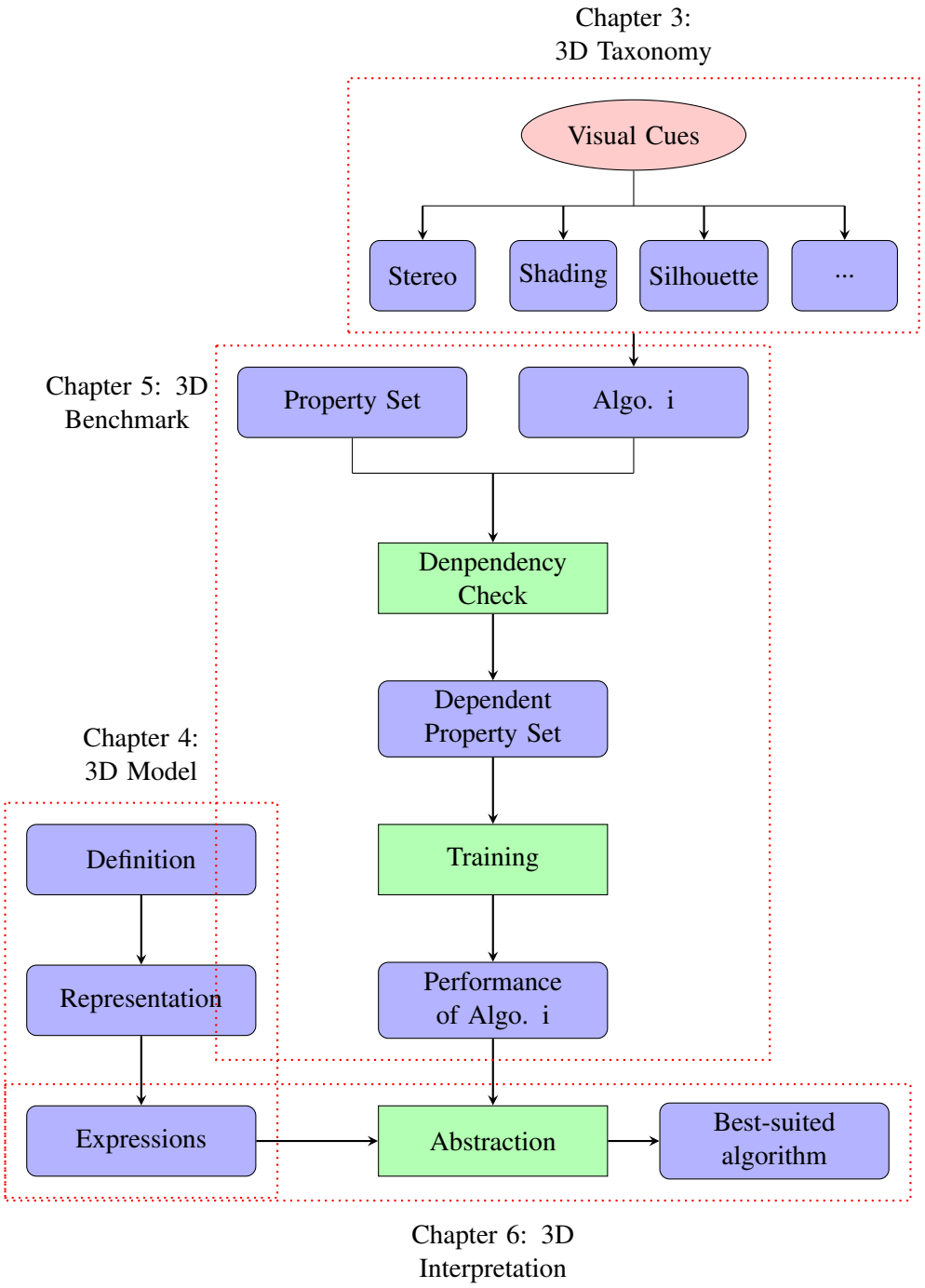


Figure 1.1: System overview. Rectangles denote process. Rounded rectangles represents data.

Chapter 2

Related Work

We present in this chapter the related work on appearance modelling and 3D reconstruction techniques. In section 2.2, we provide a comprehensive review of the field of image-based 3D reconstruction based on varied visual cues. In section **??**, we investigate the contributing factors to object appearance, which serves to choose the best-suited appearance model and reconstruction technique.

2.1 Software

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 [11], VLFeat [42], and multiple Matlab libraries [25, 30]. 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 of the object or scene from images under different viewpoints or illuminations. The goal of image-based 3D reconstruction 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 invaild, this becomes an ill-posed problem since multiple combinations of geometry, viewpoint and illumination can produce exactly the same images [35]. Traditionally, the most common way of dealing with this ambiguity has been to apply smoothness heuristics and regularization techniques [35] to obtain reconstructions that are as smooth as possible. A drawback of this type of approach is that it typically penalizes discontinuities and sharp edges, features that are very common in real scenes.

The 3D reconstruction techniques are typically categorized as passive and active methods depending on whether the controlled illumination is required. Passive methods do not require controlled light and can work with ambient light whereas active methods require some form of temporal or spatial modulation of the illumination. We can approach the categorization based on the image cues used to reconstruct the geometry: stereo, shading, contours, texture, defocus, etc. We present different techniques based on the cues exploited in this review.

Three-dimensional model acquisition has always been one of the fundamental research topics in computer vision. Active 3D scanners are currently the dominant technology for capturing digital object models for applications. Their geometric accuracy has continually improved. But they remain expensive, and, more importantly, they suffer from a number of technical limitations. They are invasive and some materials such as hair can not be scanned. They are also not "scalable" to objects of different sizes, especially large ones and outdoor scenes. In comparison, passive image-based modeling from collections of images captured by handheld cameras offers several advantages. It needs only low-cost hardware, it can be applied to objects of any size, and also it preserves the appearance information from original photographs while maintaining perfect geometric alignment.

2.2.1 Stereo cue

Stereo is one of the most widely used visual cues, and is used in stereoscopy. Stereoscopy estimates the point of a 3D point by triangulation: 1). the corresponding 2D image points are detected and matched across difference views, 2). 3D line containing the center of projection and 2D projection is obtained with known camera parameters, 3). the intersection of all 3D lines is used to recover the 3D point. Trinocular and Multi-View Stereo have been introduced to improve the accuracy and robustness. However these passive approaches suffer from uniform or periodic surfaces. Active approaches overcome this problem using controlled illumination. 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. We discuss various MVS and SL techniques in the current literature.

Multi-View Stereo algorithms can be roughly categorized into four classes: volumetric based, surface evolution based, region growing based, and depthmap based methods [40].

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. Another thread of work formulates the problem in the Markov Random Field (MRF) framework and extracts the optimal surface by Graph-Cut algorithms.

Seitz et al. proposed a voxel coloring technique that traverses a discretized 3D space in depth- order to identify voxels that have a unique coloring, con- stant across all possible interpretations of the scene.

Surface Evolution

The second class works by iteratively evolving a surface to minimize a cost function. The representations include voxels, level set, and surface meshes. *Space*

Carving technique works by iteratively remove inconsisteny voxels from the scene. *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. Other approaches represent the scene as surface meshes that moves as a function of internal and external forces. (Read Hernandez's [15])

The N-view reconstruction problem is generally an ill-posed problem, which means there exists an infinite number of photo-consistent scenes. Kutulakos and Seitz introduced the notion of the *photo hull* and the Space Carving algorithm that computes this least-commitment shape [31]. They can avoid to performing regularization, also ensures that the recovered 3D shape can serve as a description for the entire equivalence class of photo-consistent shapes.

Level-set based techniques minimize a set of partial differential equations defined in a volume. Like space carving methods, level-set methods typically start from a initial volume and shrink inward, or outward if the cost function is minimized. Faugeras and Keriven proposed a novel geometric approach based on variational principle, from which a set of PDE's can be deduced. The level set method is used to deform an initial set of surfaces towards the objects to be detected. However, level-set is no long a popular MVS technique, because high quailty models with correct topology can be directly computed from photo-consistency functions without the refinement steps.

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 propagate these points to spatial neighbours and refine the cost function with respect to position and orientation. Furukawa and Ponce starts from sparse, reliable seed points, and iteratively expand and filter the set of points to obtain a quasi-dense point cloud. PatchMatch Stereo and the variants start with a randomly initialized 3D volume, and make the assumption that one of the initial patch is close to the "true" one. This true patch

can be propagated to spatial neighbours and gets refined to get closer to the optimal patch.

Otto and Chau proposed one of the first work on region growing 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. Since the stereo matching algorithm is applicable for planar surfaces, it doesn't make sense to match every pixel. Therefore, they first defined a regular grid on the left image, and then defined the "neighbourhood" as four nearest cells in the grid.

Lhuillier and Quan presented a robust two-view quasi-dense correspondence algorithm. They first sort the list of point correspondences using the correlation score, which is called seed points. Then at each step of the propagation, they choose the best corresponding pixels from the list of seed points. Lastly, in the immediate spatial neighbourhood of the selected seed point, they look for new matches and add the bests to the list of seed points according to a combination of local constraints such as correlation, gradient disparity, and confidence. Their approach is the so-call best-first strategy, which can drastically limit the possibility of bad matches and avoid bad initialization.

PMVS is one of the first open source MVS algorithm developed by Furukawa and Ponce. The goal of this method is to reconstruct at least an oriented patch at each grid cell. First, a sparse oriented patch cloud is obtained from triangulating corresponding feature points. At the expansion stage, the current patch with the best Normalized Cross Correlation score is selected and propagated to neighbouring empty cells. Lastly, two visibility-based filtering step are performed to remove erroneous patches lying outside or inside of the "true" surface.

PatchMatch Stereo proposed by Bleyer et al. overcomes a traditional bias that pixels within a support window have the same disparity, or fronto-parallel assumption. The method is inspired by PatchMatch, which is a randomized algorithm for finding approximate nearest neighbour matches between image patches [6]. 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. Each pixel is propagated to the left/top or right/bottom pixels, or corresponding pixel

in the second view, or a preceding or consecutive frame for stereo videos. This method can achieve sub-pixel accuracy, but is computational heavy and difficult to parallelism.

There has been some effors extending PatchMatch Stereo to multi-view scenario or proposing new propagation scheme to increase the computational efficiency. A massively parallel method using a diffusion-like propagation scheme was proposed by Galliani et al..

Depthmap Merging

The fourth class works on the image space instead of the scene space, 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.

This method takes a set of images with camera parameters, discretizes the depth range into a finite set of depth values, then select one with maximum photo-consistency score. 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.

Winner-Takes-All Depthmaps This simple depthmap reconstruction algorithm is to evaluate photo-consistency value throughout the depth range, and pick the depth value with the highest photo-consistency score for each pixel independently. This process is call "Winner-Takes-All" strategy.

In addition to the depth value with the highest photo-consistency score, the algorithm often evaluates a confidence measure so that low-confidence depth values can be ignored or down-weighted in the merging step [24]. This algorithm is first proposed in by Esteban and Schmitt.

Though this simplistic approach can in general achieve good enough results, it's still problematic as occlusion or non-Lambertian effects might add noise to the photo-consistency score. Therefore, a larger window size is more likely leads to a stabler match. However, the associated peak will become broader and less well localized, reducing the accuracy of the depth estimate. Vogiatzis et al. proposed a robust photo-consistency function to overcome this problem. The basic idea is that all potential causes of mismatches like occlusion, image noise, lack of texture,

or highlights are treated as outliers. Matching is treated as a problem of robust model fitting to data containing outliers. More explicitly, for each pixel in the reference image, a photo-consistency curve $S_j(d)$ is computed for each visible view $j(j \in \mathcal{N}(i))$. Since simpling avering the photo-consistency scores across various views cannot handle outliers, they build a new $\mathscr{C}$ by detecting all the local maxima d_k of S_j, and using a Parzen window with a kernel W as follows:

$$\mathscr{C}(d) = \sum_{j \in \mathcal{N}(i)} \sum_{k} S_j(d_k) W(d - d_k) \tag{2.1}$$

This robust photo-consistency score can surpress local mixima, while simple averaging leads to erroneous results.

Goesele et al. proposed a simpler yet effective approach, which is to compute the average of pairwise photo-consistency scores after ignoring those below a certain threshold.

MRF Depthmaps In the case of severe occlusion, there may not exist a correspondence in the other images. 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 is the cost of assigning a depth label k_p from a label set to the pixel p, whereas the pairwise term is the cost of assigning depth label k_p, k_q to a pair of neighbouring pixels p and q, respectively.

$$E(k_p) = \sum_{p} \Phi(k_p) + \sum_{(p,q) \in \mathcal{N}} \Psi(k_p, k_q) \tag{2.2}$$

The unary cost reflects the photo-consistency score, which in this case, is the inversely proportional to the photo-consistency score. The pairwise term enforces the spatial regularization, thus is proportional to the amount of depth discrepancy at neighbouring pixels.

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 represented (change of word) by using grey level, colour or geometric representations. Structured light is classified based on the coding strategy: time-multiplex, neighbourhood codification and direct codification [37]. Time-multiplexing techniques generate the codeword by projecting a sequence of patterns. Neighbourhood codification represents the codewords in a unique pattern. Direct codification techniques define a codeword for every pixel, which is equal to its grey level or colour.

Time-multiplexing A sequence of patterns are successively projected onto the surface, the codeword for a given pixel is formed by the sequence of illuminaiton 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 Neighbourhood 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 codification 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 intrisic clour 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 cue

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 orientaiton 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 [44]. 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 continous, the integrated height map can reach an accuracy that cannot be achieve by any triangulation methods. Therefore, photometric stereo is more desirable if the intrisic geometric details are of great importance.

Photometric Stereo

Despite the superior results achieved by Photometric Stereo, traditional photometric stereo generally makes the following assumptions:

- Camera: orthographic projection, linear radiometric response
- Reflectance: known reflectance properties, e.g., Lambertian in [45], specular in [].
- Illumination: the lighting conditions are parallel rays with known directions and intensities.
- Others: shadow, interreflection, and other global light transportation are neglected

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 arbitray or fixed position, orientation, and intensity. Therefore the research on Photometric Stereo are generally on three directions: 1). traditional photometric stereo with known reflectance and lighting conditions; 2). generalization of reflectance; 3). generalization of lighting conditions.

Traditional case The photometric stereo is first proposed in [45]. In this work, the Lambertian model is used, which is constant, and independent to incident and emit light direction.

Generalization of Reflectance refer to [4]

Parametric reflectance model The reflectance is characterized by Bidirectinoal Reflectance Distribution Function (BRDF). Most BRDFs are more complicated than the Lambertian since relative few objects are either ideal diffuse or perfectly specular. The BRDF of many surfaces can be approximated as a combination of a Lambertian component and a specular component. This reflectance model has motivated a line of research [7, 12, 33]. Coleman and Jain and Barsky and Petrou who treat specular pixels as outliers, and Schlüns, 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.

Non-parametric reflectance model While parametric reflectance models are very good at reducing the complexity of BRDFs, they are usually only vaid for a imited class of materials. An alternative is to exploit physical properties common to a large classes of BRDFs. Typical properties include energy conservation, non-negativity, Helmholtz reciprocity, isotropy, etc. Helmholtz stereopsis, introduced by Zickler et al., is on such technique, exploiting reciprocity to obtain surface reconstruction with no dependence to the BRDF. 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.

Generalization of Lighting

The generalized lighting condition is anything other than the 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.

Shape from Shading

The problem of recovering the shape of a surface from the intensity variation is first proposed by Horn. Most shape from shading algorithms assume that the surface under consideration is of a uniform albedo and reflectance, and that the light source directions are either known or can be calibrated by the use of a reference object. Under the assumption of distant light sources and viewer, the variation in intensity (irradiance equation) become purely a function of the local surface orientation

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

where $(p,q) = (z_x, z_y)$ are the depth map derivatives, and $R(p,q)$ is the reflectance map, which is often obtained from measuring or theoretical analysis.

Since there are more unknowns, additional constraints such as smoothness or integrability is required to estimate (p,q).

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

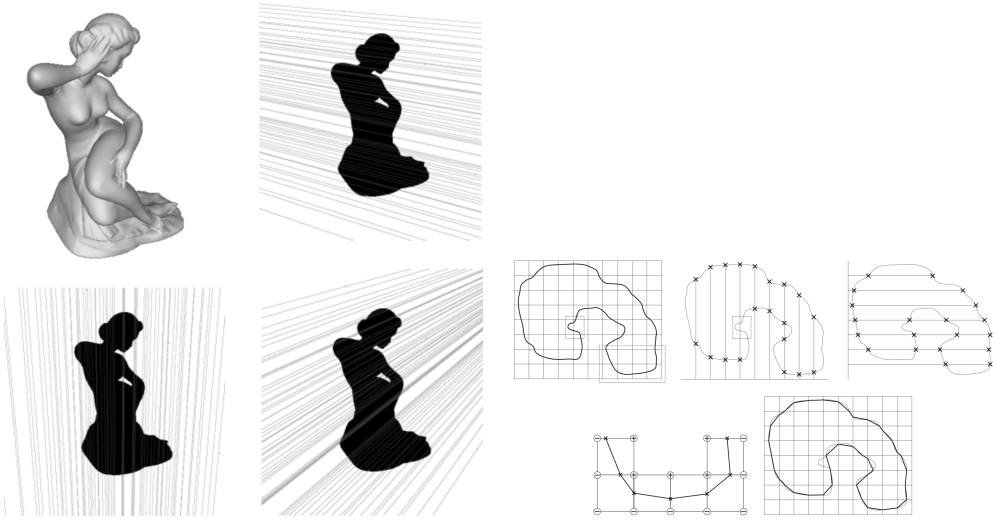
First the object space is split up into a 3D grid of voxels; each voxel is intersected with each silhouette volume; only voxels that lie inside all silhouette volumes remain part of the final shape.

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 unprojected intersections on the same ray are merged by the boolean AND operation.

Once the MI data structure representing the intersection of all extended sil-

houettes, a triangular mesh is extrated from it. This is done by the MI technique proposed in [36] 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.



(a) (b)

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 silouettes 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 *forshortening 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 foreshortending 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.

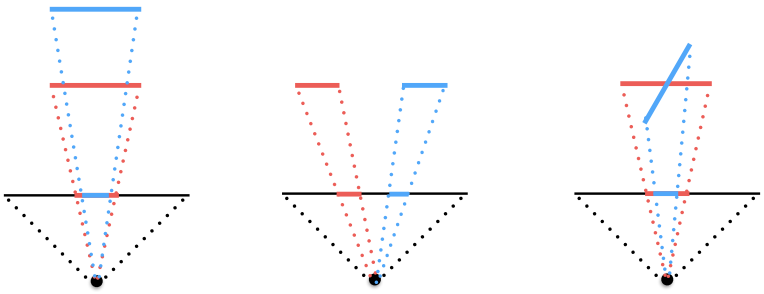


Figure 2.2: Three distortion effect: distance distortion, position distortion, and foreshortening distortion.

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.

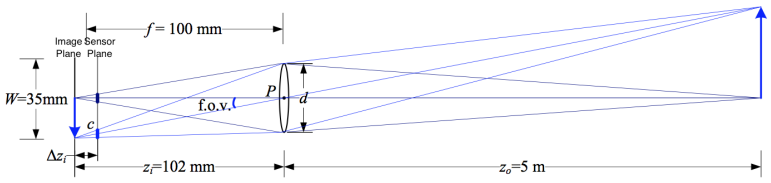


Figure 2.3: A thin lens of focal length f focuses the light from a plane a distance z_0 in front of the lens at a distance z_i behind the lens, where $\frac{1}{z_o}+\frac{1}{z_i}=\frac{1}{f}$. If the sensor plane moved forward Δz_i, the image are no longer in focus and the *circle of confusion* c depends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

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

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

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

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

The defocused images can be obtained in three ways: by displacing the sensor with respect to the image plane, by moving the lens, or by moving the object with respect to the object plane. The first two ways cab 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 increaments of Δd, and an image is captured at each stage position ($d = n\Delta d$). By studying the behaviour of the focus measure, an interpolation method is used to compute the accurate depth estimates from a small number of focus measures. An important feature of this method is the local nature, the depth estimate at an image point is computed only from focus measures recorded at that point.

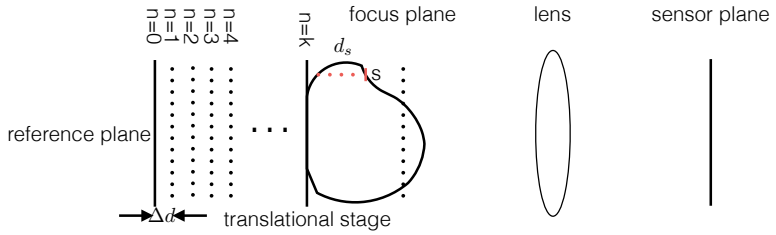


Figure 2.4: shape from focus

Chapter 3

A new taxonomy of 3D Reconstruction

Existing taxonomies of 3D reconstruction techniques generally only focus on one category of techniques: Seitz et al. proposed multiple means to classify Multi-view Stereo algorithms from various perspectives. Reviews [19, 37] of Structured Light techniques generally classify techniques based on the type of pattern used. Photometric Stereo algorithms are classified by the assumptions or generalizations made, for instance, calibrated/uncalibrated, unknown/known reflectance, unknown/known light conditions, etc. This framework provides a means to compare intra-category algorithms, but is unsuitable to evaluate the performance of each technique across object with a range of attributes.

To have a more comprehensive understanding of the strengths and weaknesses of different techniques, a more general taxonomy is need, and one of the most popular framework categorizes 3D reconstruction techniques into active and passive methods: if the controlled light condition is used, then it's active, otherwise, it's passive. Other notable taxonomy is the spacetime framework proposed in [14], which categorizes depth from triangulation techniques based on the sources of information: temporal or spatial information. Though widely adopted, the mapping of the algorithm to the conditions that works the best is generally empirical.

In the previous taxonomies, algorithms of a certain category generally work well on limited conditions, and a it's crucial to understand where algorithms per-

form well and where they fail. Under the previous framework, this knowledge is largely empirical, with each algorithm roughly maps to a problem domain that is poorly defined.

The taxonomy proposed in this chapter defines the 3D reconstruction techniques based on the visual and geometric cues that techniques utilizes for reconstruction. 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.

3D reconstruction problem is classified into the following categories: stereo correspondence, shading, silhouette, texture, defocus.

We need to way general/univeral enough that incorporates any between-class methods, and distinctive enough that distinguish with-class methods. Thus we categorize our algorithms based on the *setup*, *information domain*, *cue*, *characteristics*, and *representation*.

3.1 Setup

We consider setup using off-the-shelf hardwares, including camera, light sources, and projectors. Therefore commercial products such as laser scanner is not included.

Camera is the most basic component in any setup for reconstruction.

Light source is needed for any photometric stereo algorithms, the types of light sources include, but not limited to point light source, distant/directional light, and embient light.

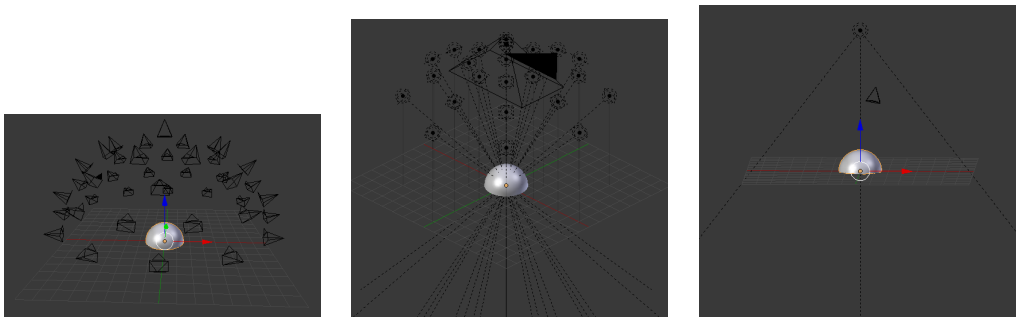
Point light typically models nearby light source

Directional light assumes that all points in space share the same lighting direction which is only true when the light source is far away from the scene.

Ambient light model allows PS to work in less constrained lighting conditions. Ambient lighting could be considered as a spherical function.

For some methods, there is a need to project certain patterns onto the scene to help find correspondence. Therefore a projector is needed.

The typically setup for is shown in Figure **??**.



(MVS) (PS) (SL)

Figure 3.1: Typical setup of MVS, PS, and SL

3.2 Information domain

The information that is utilized for reconstruction can come from different domains. Techniques such as laser scanner and passive stereo typically utilize spatial information whereas methods such as structured light and temporal laser scanner make use of information across time. This criterion can bring together techniques that are considered separately by traditional active/passive taxonomy. We characterize the *spacetime* domain in which the cues are located. The information can come from *spatial domain*, or from *temporal domain*.

3.3 Cue

We discussed different visual/geometric cues that can be used by reconstruction algorithms in Chapter 2, we classify them based on the spacetime domain that they occupy.

3.3.1 Spatial cue

Texture is the mostly used as spatial cue.

Silhouette is used by shape from silhouette and some stereo algorithms as an complementary cue, and is regarded as a spatial cue.

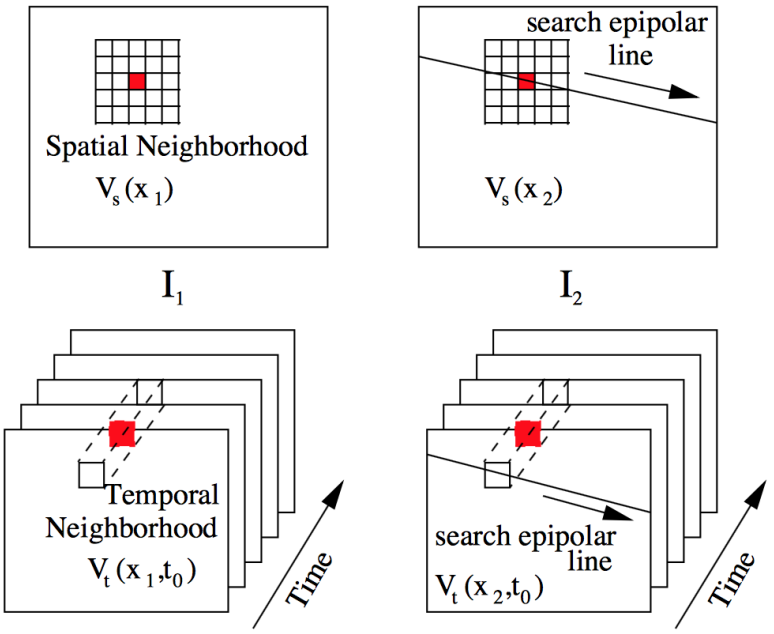


Figure 3.2: Comparison of spatial (top) and temporal (bottom) stereo. In spatial stereo, the epipolar line is searched for similar spatial neighbourhoods. In temporal stereo, teh search is for similar temporal variation.

3.3.2 Temporal cue

Intensity variation of one pixel is used widely as the visual cue for photometric stereo. It is also the basis for temporal binary-encoded structured light techniques.

3.4 Characteristics

This is where we distinguish within-class methods.

3.4.1 MVS

As discussed in Chapter 2, the MVS methods can generally be classified into four classes: 1). volumetric based methods compute a cost function in a 3D volume, and extracts a surface from this volume; 2). surface evolution based methods iteratively evolve a surface/volume to minimize a cost function; 3). seed propagation based methods start with a sparse set of scene points, and perform multiple iterations of

propagations and refinements; and 4) depth-map based methods compute a per-view depth map and merge multiple depth maps into a complete 3D point cloud. We use the notation below to denote each class:

- Reconstruction algorithm: **V**: volumetric based methods; **E**: surface evolution based methods; **P**: seed propagation based methods; **D**: depth-map based methods.

3.4.2 SL

Structured light system overcomes the correspondence problem faced by any stereo technique by projecting a coded pattern onto the surface. The patterns are specifically designed so that codewords are assigned to a set of pixels, thus there is a direct mapping from the codewords to the coordinates of the corresponding pixel in the pattern. Based on the type of codeword used, the projection patterns are generally classified as: temporal encoding, spatial encoding, and direct encoding, refer to [37] for more details.

For temporal encoding, a set of patterns are successively projected onto the surface. The codeword is formed by a sequence of illumination for a specific pixel across the projected patterns. In the case of spatial encoding, the codeword of a point is obtained by considering the neighbourhood of the points around it. In the case of direct encoding, each pixel is labeled by the information representing it, which can be intensity or colour information. We use the notation below to denote each class:

- **Temporal encoding**: **B**: binary encoding; **N**: $n-$ary encoding; **BPS**: binary with phase shift; **H**: hybrid methods combining temporal and spatial encoding;
- **Spatial encoding**: **NF**: non-formal codification; **DB**: methods based on De Bruijn sequences; **M**: $M-$array;

3.4.3 PS

Almost all Photometric Stereo techniques make the following assumptions:

- **Camera**: orthographic projection, linear radiometric response
- **Reflectance**: known reflectance property
- **Illumination**: known light direction and intensity
- **Others**: shadows, inter-reflection, and other global light transportation are neglected or considered outliers.

Thus our notation is also based on the assumptions made:

- Reflectance model: **L**: Lambertian model, **M**: Mixture of BRDFs
- Illumination: **U**: uncalibrated lighting, **C**: calibrated lighting; **D**: Directional lighting, **P**: Point lighting, **E**: Environmental lighting.
- Number of images: **S**: Small, at least three and typically 10 - 20, **M**: Medium, typically 50 - 100, and **L**: Large, typically 500 - 1000.

3.5 Representation

We choose the scene representation as another axis of taxonomy as it often determines the range of possible applications. The most popular scene representation for a 3D model is: a point/patch cloud, a depth map, a volumetric grid, surface mesh, and a normal map.

3.5.1 Depth map

Depth map is one of the most popular choice due to its flexibility and scalability. Given a set of images, one can compute a per-view depth map for each input images once a set of neighbouring images were found. Multiple depth maps can be merged together to form a point cloud.

The estimation of depth map is similar to that of binocular stereo. A typical process discretizes the depth range into a finite set of depth values, then an optimal depth value is chosen based on photo-consistency measure. Uniform depth sampling may suffice for simple can compact objects. However, a non-uniform sampling is a prerequisite to achieve efficiency and quality for complicated scenes.

3.5.2 3D point/patch cloud

The biggest flaw of a depth map is that a post-processing step is needed to convert multiple depth maps into a 3D scene model. A point cloud or a patch cloud overcomes this issue since, a patch is a point with surface normal estimation. A common feature of point cloud based algorithms is that they utilize spatial consistency assumption and grow or expand a set of seed points into a dense point cloud.

3.5.3 Volumetric grid

In the previous two representation, a reference view needs to be determined to compute the photo-consistency measure, thus are view-dependent. A view independent representation called volumetric grid can overcome this issue. Voxel grid can also

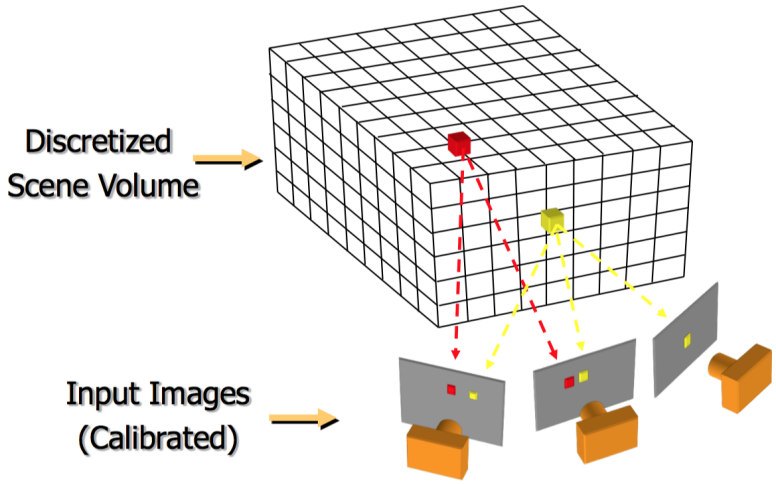


Figure 3.3: Voxel grid

be used for surface mesh extraction. Volemetric surface extraction accumulate information from depth maps, or laser scanned 3D points, and extracts a iso-surface using Marching Cube algorithm, or as a 3D binary(inside/outside) segmentation problem.

3.5.4 Mesh

One of the most popular surface mesh is triangular mesh.

3.5.5 Normal map

Normal map is typically visualized as a colour image, with each pixel colour coded as the normal. Each colour channel ranges from $[0,255]$, and n_x, n_y ranges from [-1, 1] while n_z rangs from [0, 1]. The transformation between a normal and a colour is shown in Figure 3.4.

$$n = \frac{c}{128} - 1$$
$$c = 128 \cdot (n+1)$$

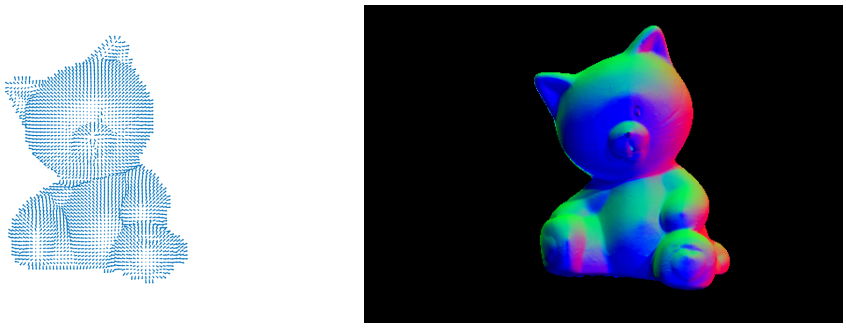


Figure 3.4: Representation of normal map

3.6 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 **??**.

Technique	Setup	Dom	Cue	Charact	Rep
PMVS	C_n	*S*	*T*	*P*	*P*
Goesele	C_n	*S*	*T*	*D*	*D*
Woodham	C_1L_n	*T*	*I*	*LDS*	*N*
Hertzemann	C_1L_n	*T*	*I*	*MDM*	*N*
Gray code	C_1P	*T*	*I*	*B*	*P*

Table 3.1: Algorithm classification based on the new taxonomy

Chapter 4

Model of 3D Reconstruction

In Chapter 3, we introduce a taxonomy of 3D reconstruction which map algorithms according to the main visual cues used for reconstruction. 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.

The goal when providing an abstraction to 3D reconstruction is that with better description should lead to better result. To order to achieve this, the visual and geometric properties of an object that can affect the visual cues should be examined in depth so that important aspects of the problem can be described.

A key requirement of the 3D reconstruction model is that it should be interpretable. Thus the components of this model must be well defined. We first propose a formal definition of the 3D reconstruction problem in Section 4.1. Section 4.2 explores the inputs and outputs used in 3D reconstruction problems. Section **??** discusses various *properties* that can be used to describe the appearance of the object. Section 4.3 provides the mapping of the representations and properties into a formal model via which 3D reconstruction problem can be expressed. These layers: Definition, Representation, Conditions, and Expression represent out framework of accessible 3D reconstruction.

4.1 Definition

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

4.1.1 Basic notations

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

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

Definition 2 (Image) The transformation of the scene S onto the image plane of camera C_i at time t_0, which can be modelled as: $I_i = T(S, C_i, L_0, t_0)$, or the transformation of the scene S onto 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 transformation.

The transformation can be a geometric one which determines the 2D coordinates from a 3D position, 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 Scelement

Segment is the lowest level element in the image, can be considered as a generalized pixel.

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

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 (Scelement) A Scelement (scene element) is a distinct volume in the scene which corresponds to at least one segment, can be considered as a generalization of a voxel.

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

The property of the scelment can be the visual texture, diffuse albedo, surface orientation, roughness, convacity, etc.

The relation between these notions is shown in Figure **??**.

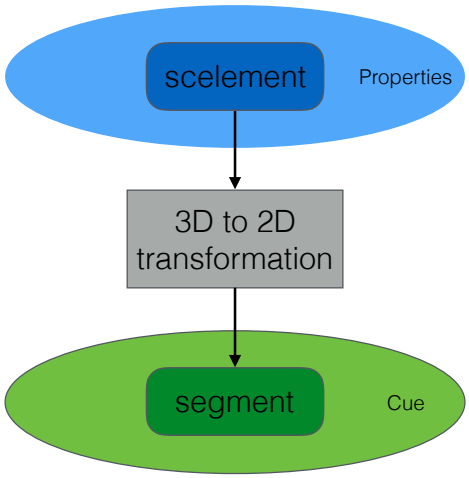


Figure 4.1: Relation between a scelement and a segment

Definition (Representation) The scelement can be represented as a voxel, a depth value, a 3D point/patch, or a surface normal, etc, which is denoted as $rep(slmt)$.

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 scelement *slmt* can produce the cues observed in the corresponding segment *seg*.

$$consist(rep(slmt), prop(slmt), cue(seg)) = 1 \Rightarrow photo\ consistent$$
$$consist(rep(slmt), prop(slmt), cue(seg)) = 0 \Rightarrow not\ photo\ consistent$$

Definition (Segment photo-consistency) Let S be the scene. A scelement $s \in S$ 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 photo-consistent with image I_i if for all scelements $\forall s \in S$ visible from the camera C_i are 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 scelements $\{slmt_n\}_{n=0}^{M-1}$ such that any scelement is photo-consistent with the image set $\{I_n\}_{n=0}^{N-1}$, i.e., $\forall slmt_i \in \{slmt_n\}_{n=0}^{M-1}$, we have $consist(rep(slmt_i), prop(slmt_i), cue(seg_{(i,n)})) = 1$.

where $seg_{(i,n)}$ is the corresponding segment of $slmt_i$ in camera C_n. Alternatively, 3D reconstruction tries to find a set of scelments $\{slmt_n\}_{n=0}^{M-1}$ 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 definitive definition of the problem of 3D reconstruction, it does so in a purely theoretical way which 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 similariy betweem a scelement *slmt* and the corresponding segment *seg*.

$$consist(rep(slmt), prop(slmt), cue(seg)) = x,\ x \in [0,1]$$
$$consist(rep(slmt), prop(slmt), cue(seg)) = 1 \Rightarrow photo\ consistent$$
$$consist(rep(slmt), prop(slmt), cue(seg)) = 0 \Rightarrow not\ photo\ consistent$$

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

$$consist(rep(slmt), prop(slmt), cue(seg)) > \varepsilon$$

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 scelements $\{slmt_n\}_{n=0}^{M-1}$ such that the photo-consistency measure between the set of scelements and their corresponding segments $\{seg_{(i,n)}\}_{i=0,j=0}^{M-1,N-1}$ are maximized.

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

4.2 Representation

Based on the proposed definitions of 3D reconstruction problem, we need to further define the representations so that any developers can express their problem based on our proposed model. We look at the *cues* that are utilized by 3D reconstruction techniques and their corresponding contributing properties. In Chapter 3, we explored a new taxonomy of 3D reconstruction based visual/geometric cues. Now we need to investigate the visual and geometric properties of the object that can affect those cues. This section is organized by the visual/geomtric cues, and the visual/geomtric properties are investigated in each section.

4.2.1 Segment and scelement

As defined in section 4.1, a segment is the 3D to 2D transformation of a scelement. Here we discuss concrete examples of segment and scelement.

Pixel and voxel

In the image plane, a pixel is a square of size 1×1. In the matrix representation of an image I, a pixel is an entry of the matrix, $I(x,y)$. A voxel is a 3D regular cube, and the center of which is projected to the center of the pixel.

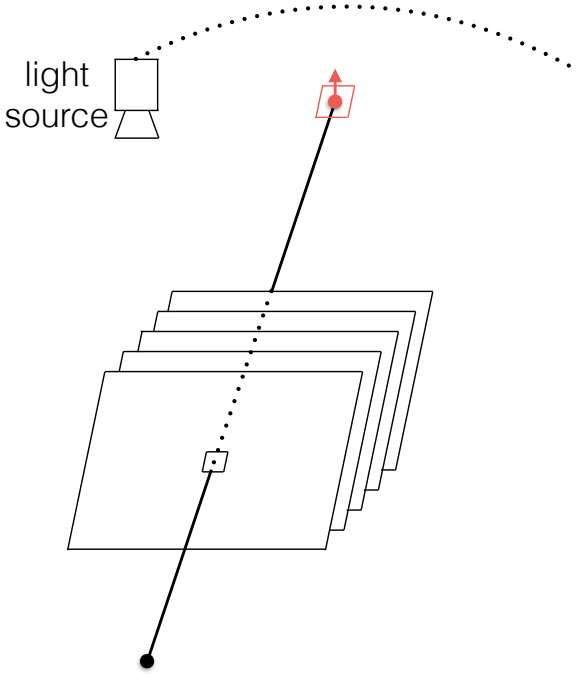


Figure 4.2: Pixel and voxel

Silhouette and bounding edge

Window area and patch

A window area is contained in a $w \times w$ regular square, and the surface patch is a 3D point of $p \times p$ with a normal vector.

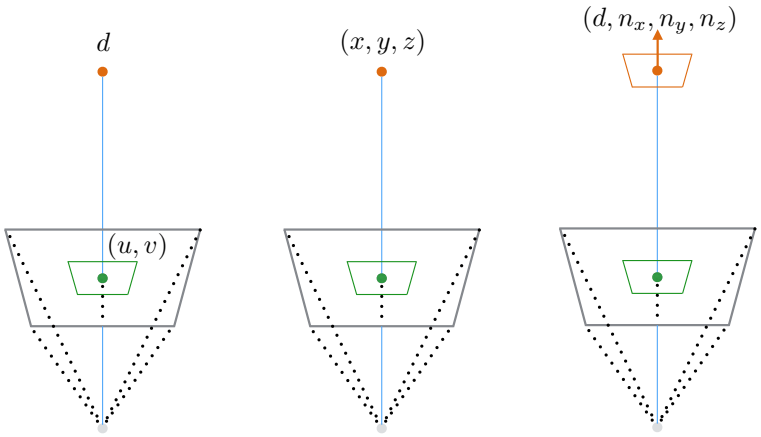


Figure 4.3: a window area and a surface patch

4.2.2 Cues and properties

As defined in Chapter 4.1, cue is the characteristics of the segment while property is that of the scelement. For each cue observed in a segment, we discuss the underlying properties that have an impact on it.

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. Haralick considers a texture as an "organized area phenomenon" which can be decomposed into 'primitives' having specific spatial distributions [21]. This

definition, also known as structural approach, comes directly from human visual experience of texture. These primitives are organized in a particular spatial structure indicating certain underlying placement rules. Alternatively, as Cross and Jain suggested, a texture is "a stochastic, possibly periodic, two-dimensional image field" [13], which is also known as *stochastic approach*.

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.

Most texture synthetic research has focused on data-driven or statistical approaches. For the data-driven approach, the generated texture is not general enough whereas it's not intuitive enough for the statistical approach. Thus we turn to an approach that is more tailored to the stereo vision problem. Based on the observations from practical tests, stereo algorithms work well under the condition of non-uniform texture, even for textures caused by shadow. This is theoretically plausible as stereo vision tries to find the correspondence based on the 'distinctiveness' of the texture. Therefore, as long as the surface is covered by distinct texture, it doesn't matter what the basic texture element is. Thus the most significant attributes of the texture is coverage, i.e., the percent of the surface that is covered, and it's the focus of this thesis.

Intensity variation

When light strikes a surface, it may be reflected, transmitted, absorbed, or scattered; usually, a combination of these effects occur. The intensity/colour information received by the sensor is thus determined, among other factors, the amount of light after these interaction. We consider intensity caused solely by reflection as it

is the most common phenomenon and the easiest to analyze. 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**.

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

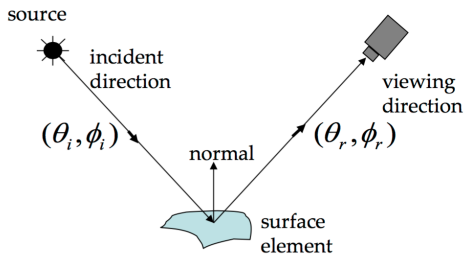


Figure 4.4: Surface reflection, image courtesy of Srinivasa Narasimhan

$\mathbf{E}^{\mathbf{surface}}(\theta_\mathbf{i}, \phi_\mathbf{i})$: irradiance at surface in direction (θ_i, ϕ_i).

$\mathbf{L}^{\mathbf{surface}}(\theta_\mathbf{r}, \phi_\mathbf{r})$: irradiance at surface in direction (θ_r, ϕ_r).

Definition (BRDF) the ratio of the radiance of the outgoing direction to the incident irradiance, i.e., $f(\theta_i, \phi_i, \theta_r, \phi_r) = \frac{E^{surface}(\theta_i, \phi_i)}{L^{surface}(\theta_r, \phi_r)}$.

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.

The reflectance of light is dependent on the spectrum of the light, which means that the reflectance of the light is dependent on the light frequency, see Figure **??**. We consider the reflectance across all spectrum, meaning only intensity albedo is considered.

The reflectance of light also depends on the incident direction. Specifically, light that lands on a surface at a grazing angle will be much more likely to reflect, see Figure **??**. We take into account the Fresnel effect in the synthetic stage, thus we consider the albedo with small incident angle.

It 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

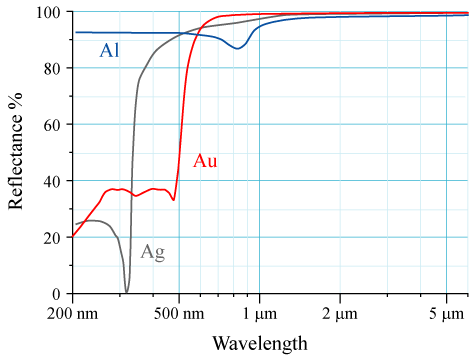
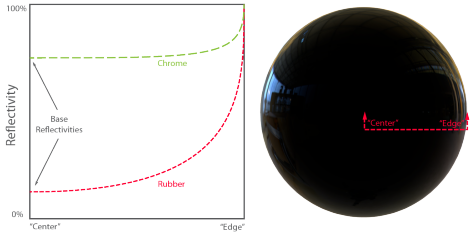


Figure 4.5: Spectral reflectance curves for aluminium (Al), silver (Ag), and gold (Au) metal mirrors at normal incidence.



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 incident light and intensity value.

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 [32]. 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 word, specular reflections are simply images of the environment, or the illumination field, distorted by the geometry of the reflecting surface. See Figure **??**, 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.

Another observation is that from the image changes as the viewer changes. This can be derived directly from specular reflection, which make stereo correspondence searching extremely challenging.

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, its 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 also think of a surface as a collection of planar micro-facets.

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

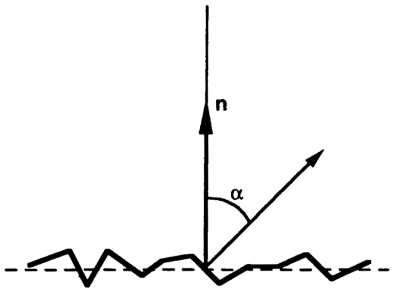


Figure 4.6: Surface Slope Distribution Model

We will use the parameter α to represent the slope of individual facets. Surfaces can be modeled by a statistical distribution of the micro-facet slopes. If the surface is isotropic, the probability distribution of the micro-facet slopes can be assumed to be rotationally symmetric w.r.t the mean surface normal $\vec{n}$. Therefore, facet slopes can be described by a one-dimensional probability distribution function. For instance, the surface may be modeled by assuming a normal distribution for the facet slope α, with mean value $\bar{\alpha} = 0$ and standard deviation σ_α:

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

The surface model is determined by a single parameter σ_α, and larger σ_α can be used to model rougher surfaces. While autocorrelation coefficient is important, the concept of slope correlation is more difficult to interpret and is not that useful in the generation of surface, which results in a weaker model compared to the height model. However, slope distribution model is popular in the analysis of surface reflection, as the scattering of light rays is dependent on the local slope of the surface and not the local height of the surface.

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*.

Silhouette

Concavity is not shown in the silhouette image, thus methods that utilize silhouette information may fail to reconstruct concavities. Concavity is measured by *surface curvature*.

4.3 Expression

Now with the proposed definition and representation of 3D reconstruction problem, we can express some existing 3D reconstruction algorithms under this framework. We present four examples as shown in Figure 4.7.

The expression of the reconstruction problem is shown in table **??**.

object	Texture coverage	Albedo	Specular	Roughness	Concavity
Temple	0.8	0.8	0.2	0.8	0.5
Dino	0.2	0.8	0.2	0.2	0.2
Cat	0.2	0.8	0.2	0.2	0.2
Statue	0.2	0.8	0.2	0.2	0.2

Table 4.1: Expression of the reconstruction problem for the “temple”, “dino”, “cat”, and “statue” datasets.

Temple Dino

Cat Statue

Figure 4.7: Image of “temple”, “dino”, “cat”, and “statue” datasets

Chapter 5

A benchmark of 3D Reconstruction Techniques

Current existing 3D benchmarks all 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) benchmark for evaluation of 3D reconstruction algorithms. Each benchmark dataset includes a collection of images of a scene under different material or lighting conditions, together with ground-truth point cloud, and surface normals. The datasets are organized into 'depend_check' and 'training' in which one property of the object is varied while others are kept constant.

5.1 Synthetic setup

We use the physically-based renderer Cycles in Blender. For each technique, the configuration of the camera remains fixed. The image resolution is 1280×720. For MVS, there are five rings of camera, of which the elevation angle is 15°, 30°, 45°, 60°, 90°. The angle between two neighbouring camera in the first four rings is 30°, 30°, 45°, and 45°. Thus there are in total $12+12+8+8+1=41$ cameras.

For photometric stereo, according to [**?**], 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°, 85°, 60°, and 45°. The azimuth angle between two neighbouring light sources is 45°.

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

5.2 Structure of Datasets

Due to the number of properties and number of levels for each property, it would be unrealistic to render all 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 dependency 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 combinations for dependent properties.

The camera/projector intrinsic and extrinsic parameters are computed directly from the positions and orientations of the synthetic setup, and the ground truth including the 3D model and normal map are generated directly from Blender.

5.3 Selected methods

We have selected three algorithms: the PMVS proposed by Furukawa and Ponce ($C_n - S - T - P - P$), the example-based photometric stereo proposed by Hertzmann and Seitz ($C_1 L_n - T - I - MDS - N$), and the Gray-encoded structured light technique ($C_1 P - T - I - B - P$).

5.4 Evaluation metrics

We use the metric proposed by Seitz et al. to evaluate MVS and SL. 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.

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.

5.5.1 $C_n - S - T - P - P$

We evaluate the performance of MVS in terms of accuracy and completeness under varied combination of properties.

(a) Texture and Albedo For a fixed texture, the accuracy and completeness doesn't change much as the albedo changes, which shows that the influence of the texture on the performance is not impacted by albedo.

Property	Texture coverage	Albedo	Specular/Diffuse ratio	Roughness
Value	0.2-0.8	0.2-0.8	0.0	0.0
	0.2-0.8	1	0.2-0.8	0.0
	0.2-0.8	1	0.0	0.2-0.8
	1.0	0.2-0.8	0.2-0.8	0.0
	1.0	0.2-0.8	0.0	0.2-0.8
	1.0	1.0	0.2-0.8	0.2-0.8

Table 5.1: Parameter of MVS with varied texture and albedo

For a fixed albedo, the accuracy remains almost the same and completeness goes up a little bit as texture level goes up, which demonstrates that the texture level has a larger influence on the completeness instead of the accuracy, which is consistent with the real-world data.

(b) Texture and Specularity For a fixed texture, as the specularity goes up, the accuracy value of MVS goes up, and the completeness goes down, meaning the reconstruction gets worse as specularity goes up.

For a fixed specularity, the accuracy goes down as the texture level goes up, and the completeness goes up as the texture goes up, which means the reconstruction gets more accurate and more complete, which is consistent with the results obtained from real-world data. But for lower specularity, the impact of texture is more substantial, thus these two properties are dependent to each other.

(c) Texture and Roughness For a fixed texture, the accuracy and completeness doesn't change much as the roughness changes, which shows that the influence of the texture on the performance is not impacted by roughness.

For a fixed roughness, the accuracy remain almost the same and completeness goes up as texture level goes up, which demonstrate again that the texture level has a larger influence on the completeness instead of the accuracy, which is consistent with the real-world data.

(d) Albedo and Specularity For a fixed albedo, the accuracy increases and the completeness decreases as the specularity increases, which demonstrates the effect of specularity. Since we're using a physical-based rendering engine (PBR), the diffuse decrease as the specularity increases.

for a fixed specularity, the accuracy increase and the completeness decreases

as the albedo increases, but this effect is more noticeable for lower albedo, highly specular surface, which shows that albedo and specularity are two dependent properties.

(e) Albedo and Roughness For a fixed albedo, the accuracy and completeness remain almost the same as the roughness changes.

For a fixed roughness, the accuracy and completeness remain also almost the same as the albedo changes, which is also consistent with the real-world scenario. Thus these two properties are independent to each other.

(f) Specularity and Roughness For a fixed specularity, the accuracy and consistency doesn't change much as the roughness changes especially when specularity is low.

For a fixed roughness, the accuracy value increases and completeness value decreases, which again shows that the specularity can affect the MVS.

Therefore, specularity has an impact on MVS, but its effect won't be interfered by roughness, thus those two properties are independent.

Conclusion the properties that have an effect on the MVS are: texture, albedo, and specularity. Therefore, we will only consider these three properties for all forthcoming discussion of MVS.

5.5.2 $C_1L_n - T - I - MDS - N$

We evaluate the performance of PS in terms of angle difference under varied combinations of properties. The statistical measures that we used include median, mean, first and third quartile. We investigate two properties at a time.

Property	Texture coverage	Albedo	Specular/Diffuse ratio	Roughness
Value	0.2-0.8	0.2-0.8	0.0	0.0
	0.2-0.8	1	0.2-0.8	0.0
	0.2-0.8	1	0.0	0.2-0.8
	0.0	0.2-0.8	0.2-0.8	0.0
	0.0	0.2-0.8	0.0	0.2-0.8
	0.0	1.0	0.2-0.8	0.2-0.8

Table 5.2: Parameter of PS with varied properties

(a) Texture and Albedo For a fixed texture, as the albedo level goes up, all

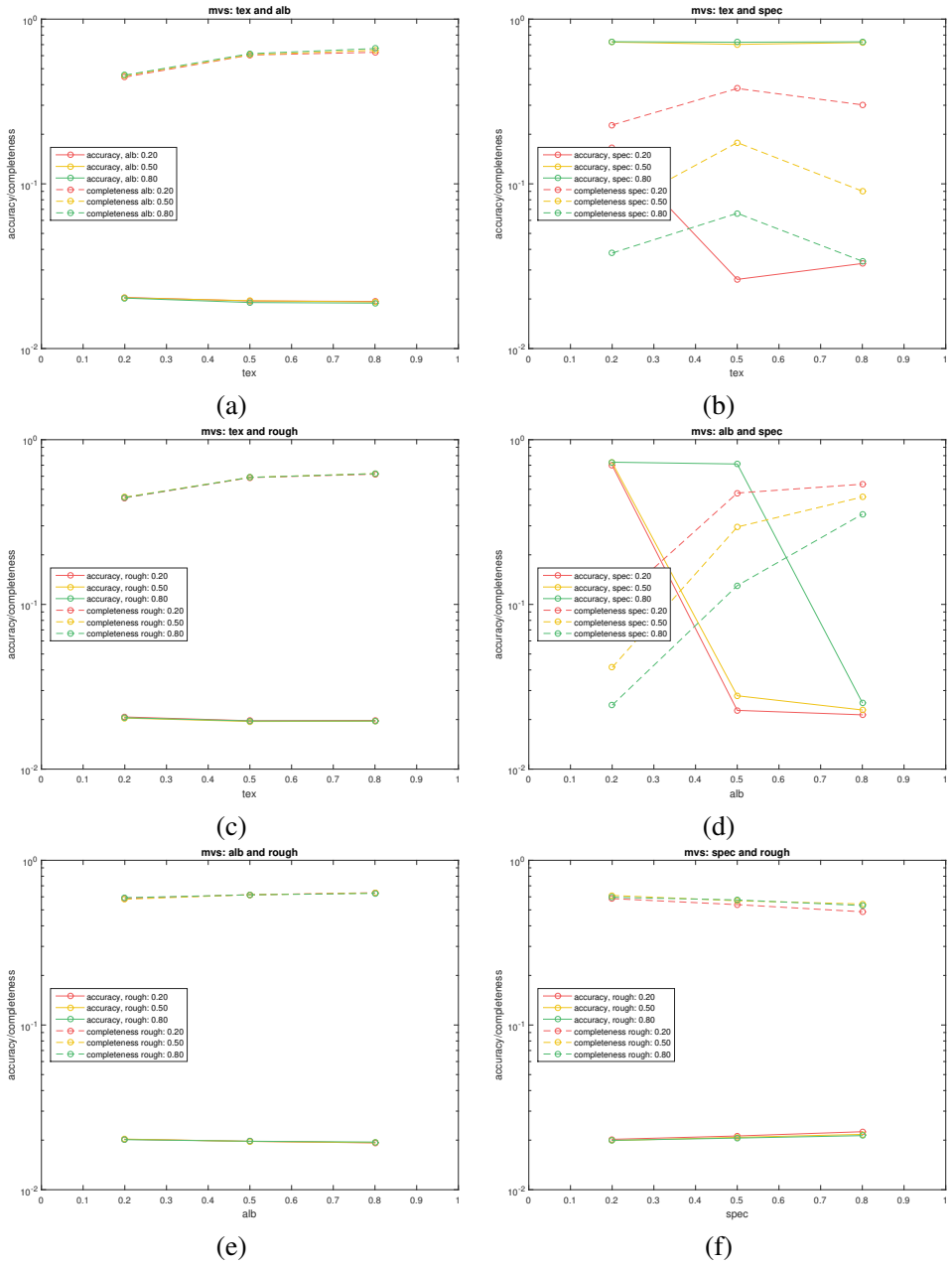


Figure 5.1: Performance of MVS with varied properties

the statistic measures go down, which means that the reconstruction gets better as albedo level goes up, which is consistent to the real-world scenario.

For a fixed albedo, the angle difference doesn't change much as the texture level changes, which shows that texture doesn't interfere with albedo, and these two properties are thus independent.

(b) Texture and Specularity For a fixed texture, as the specularity goes up, all the statistic measures go up, which means that the reconstruction gets worse as specularity level goes up, which is consistent to the real-world scenario.

For a fixed specularity, the angle difference doesn't change much as the texture level changes, which shows that texture doesn't interfere with specularity, and these two properties are independent.

(c) Texture and Roughness For a fixed texture, as the roughness goes up, all the statistic measures go down, which means that the reconstruction gets better as roughness level goes up.

For a fixed roughness, the angle difference doesn't change much as the texture level changes, which shows that texture doesn't interfere with roughness, and these two properties are independent.

(d) Albedo and Specularity We're using a physically-based renderer, thus the higher the specularity, the less the diffusion would be. Thus rising specularity would 'darken' the diffuse areas.

For a fixed albedo, the angle difference goes up as the specularity rises, which demonstrate that PS can't deal with high specularity, and it's worse for lower albedo surfaces than that for the higher albedo surfaces, which is consistent to real-world scenario.

For a fixed specularity, the angle difference goes down as the albedo rises, which is consistent to the real-world scenario since high albedo would make the intensity variation more distinctive.

(e) Albedo and Roughness For a fixed albedo, as the roughness goes up, all the statistic measures go down, which means that the reconstruction gets better as roughness level goes up.

For a fixed roughness, the angle difference also goes down as the albedo level goes up, which shows that albedo does interfere with roughness, and these two properties are dependent.

(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 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 **??** for some visual examples.

For a fixed roughness, increasing the specularity would make the angle difference worse. The effect is less substantial when the roughness is higher or when the specularity is lower.

Therefore, the specularity and roughness cancels each other's effect, thus they are dependent properties, which is consistent to visual inspection.

Conclusion the properties that have an effect on the PS are: albedo, specularity, and roughness. Therefore, we will only consider these three properties for all forthcoming discussion of PS.

5.5.3 $C_1P-T-I-B-P$

We evaluate the performance of SL in terms of accuracy and completeness under varied combination of properties.

Property	Texture coverage	Albedo	Specular/Diffuse ratio	Roughness
Value	0.2-0.8	0.2-0.8	0.0	0.0
	0.2-0.8	1	0.2-0.8	0.0
	0.2-0.8	1	0.0	0.2-0.8
	0.0	0.2-0.8	0.2-0.8	0.0
	0.0	0.2-0.8	0.0	0.2-0.8
	0.0	1.0	0.2-0.8	0.2-0.8

Table 5.3: Parameter of SL with varied properties

Our current implementation of SL projects column patterns and a row patterns, and compute depth values using images captured using these two kinds of patterns individually. A depth consistency checking step is performed to reject erreneous triangulations, thus the accuracy remains almost the same across all cases.

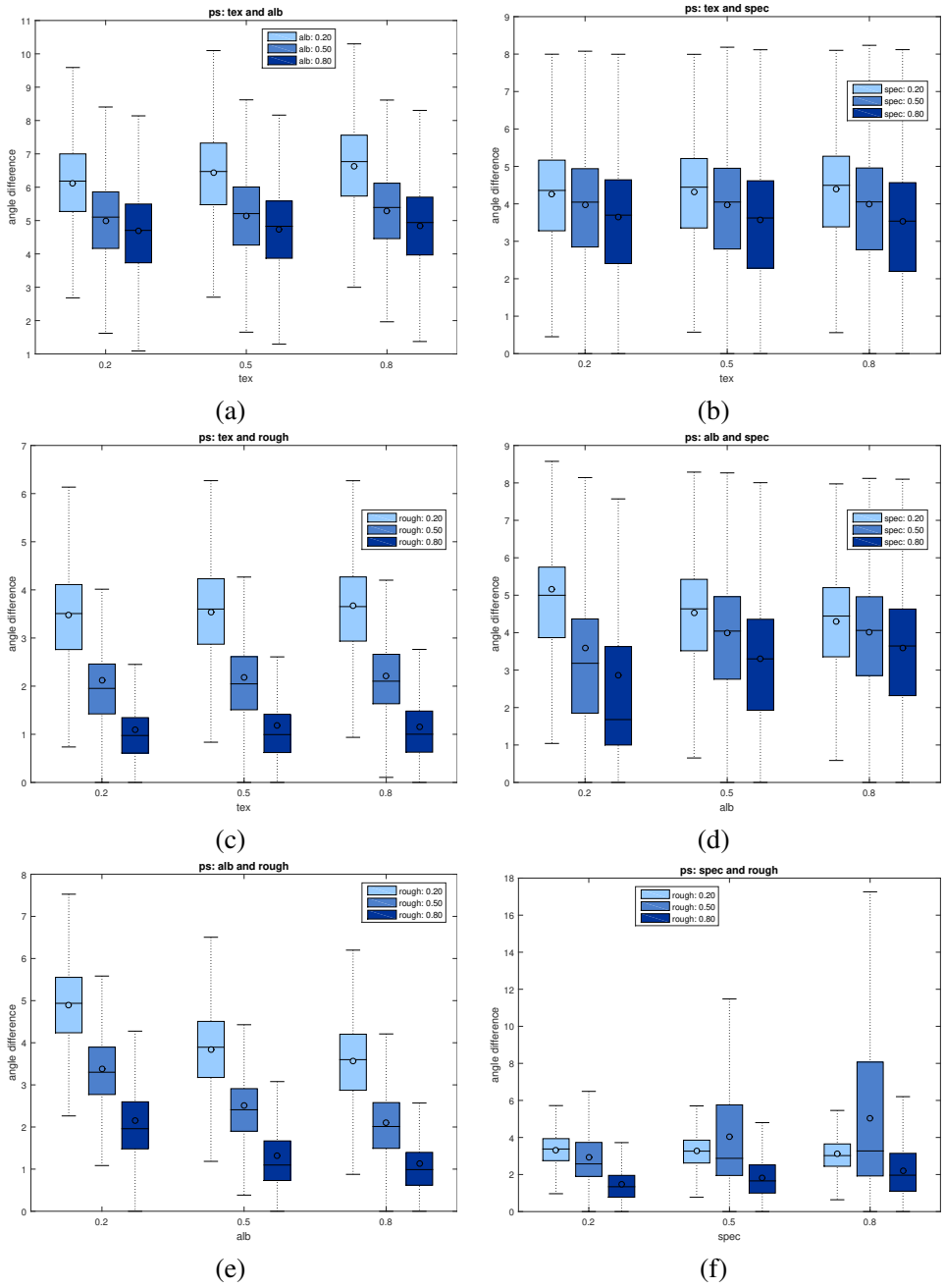


Figure 5.2: Performance of PS with varied properties

(a) Texture and Albedo For a fixed texture, as the albedo goes up, the accuracy value of SL remain almost the same, whereas the completeness goes up, meaning that the reconstruction gets more dense as albedo goes up, which is consistent to real-world scenario.

For a fixed albedo, the accuracy remains almost the same as the texture level goes up, and the completeness goes down a little as the texture goes up, which demonstrate the real-world observation that surface texture would interfere with some SL techniques.

(b) Texture and Specularity No substantial changes when either of the two properties changes.

(c) Texture and Roughness No substantial changes when either of the two properties changes.

(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 when the albedo increases. Thus these two properties are dependent

(e) Albedo and Roughness No substantial changes when either of the two properties changes.

(f) Specularity and Roughness No substantial changes when either of the two properties changes.

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

5.6 Training

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. We show the performance of each technique w.r.t one property in Figure 5.4, note that column 2, 3 uses the exactly same data as column 1.

5.7 Summary

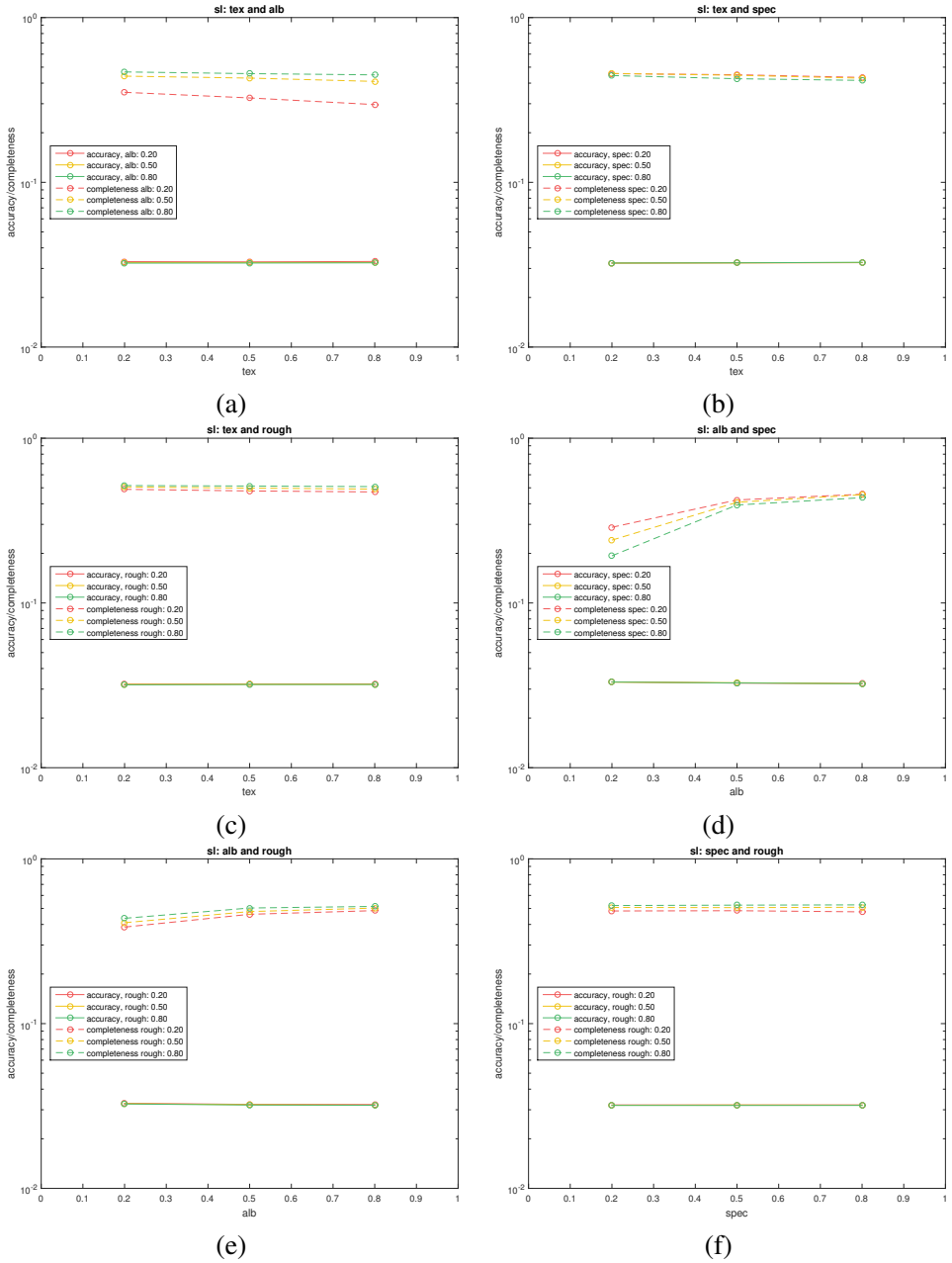


Figure 5.3: Performance of SL with varied properties

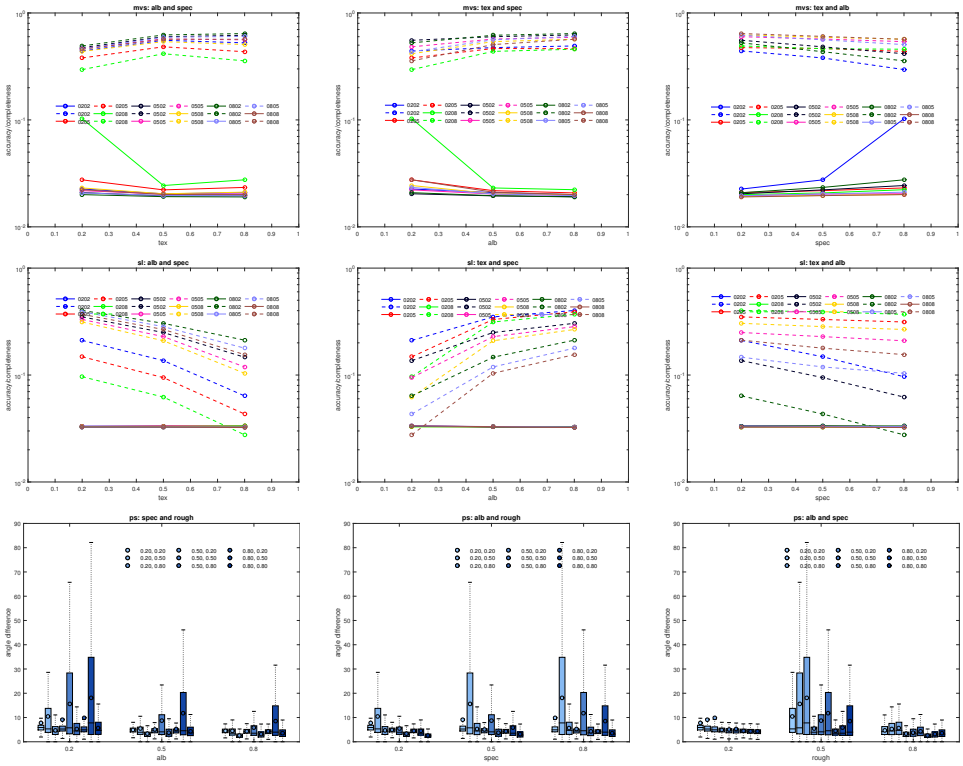


Figure 5.4: Performance of MVS, SL and PS with varied properties. Each each column, we fix one property while changing the others, thus the second and the third columns are essentially the same as the first column, they are just different point of views of looking at those relations. Each line/boxplot represents a different combinations of property values: 0202, 0205, 0208, 0502, ..., 0808. Beware that we consider {tex, alb, spec} for MVS and SL, and {alb, spec, rough} for SL.

Chapter 6

Interpretation of 3D Reconstruction Model

In order to validate the 3D reconstruction taxonomy and the model derived from it, interpretability from 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.

The three algorithms introduced in our test bench are: the PMVS proposed by Furukawa and Ponce, the example-based Photometric Stereo proposed by Hertzmann and Seitz, and a standard gray-coded Structured Light technique with error rejection.

Although there are only three algorithms selected, all of them 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. The integration of a new algorithm requires only that they be evaluated with a similar procedure and images presented in Chapter 5, allowing researchers to contribute novel algorithms to the framework. The source code and blender files used to generate the images are available online to encourage the testing of additional algorithm, and incorporation of additional properties.

6.1 Synthetic Datasets

We use three objects shown in Figure 6.2, and four property lists in Table **??** to test the validity of the abstraction. All four cases are labeled in Figure 6.1 so that it would be easier to check which technique(s) give a good reconstruction based on our abstraction.

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

Table 6.1: Property lists of the test objects. Link to the labels in Figure 6.1, (a): dark blue rectangle, (b): dark green rectangle, (c): light blud rectangle, (d): light green rectangle.

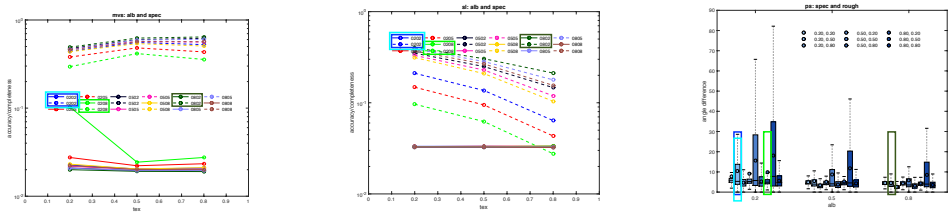


Figure 6.1: Performance of MVS, Sl and PS with varied properties.

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 Both MVS and SL perform relatively well, but we can see that it has a bigger impact on the completeness, and less of an impact on the accuracy of MVS. This is consistent to the results shown in Table **??** (a). If it weren't for this abstraction, it would be hard to imagine that MVS actually works decently with relatively low textured surface. PS performs poorly as suggested by the abstraction, see Figure 6.3.

Case 2 All three techniques perform well in this case, , see Figure 6.4, which is consistent to the result returned by the abstraction which is shown in Table **??**

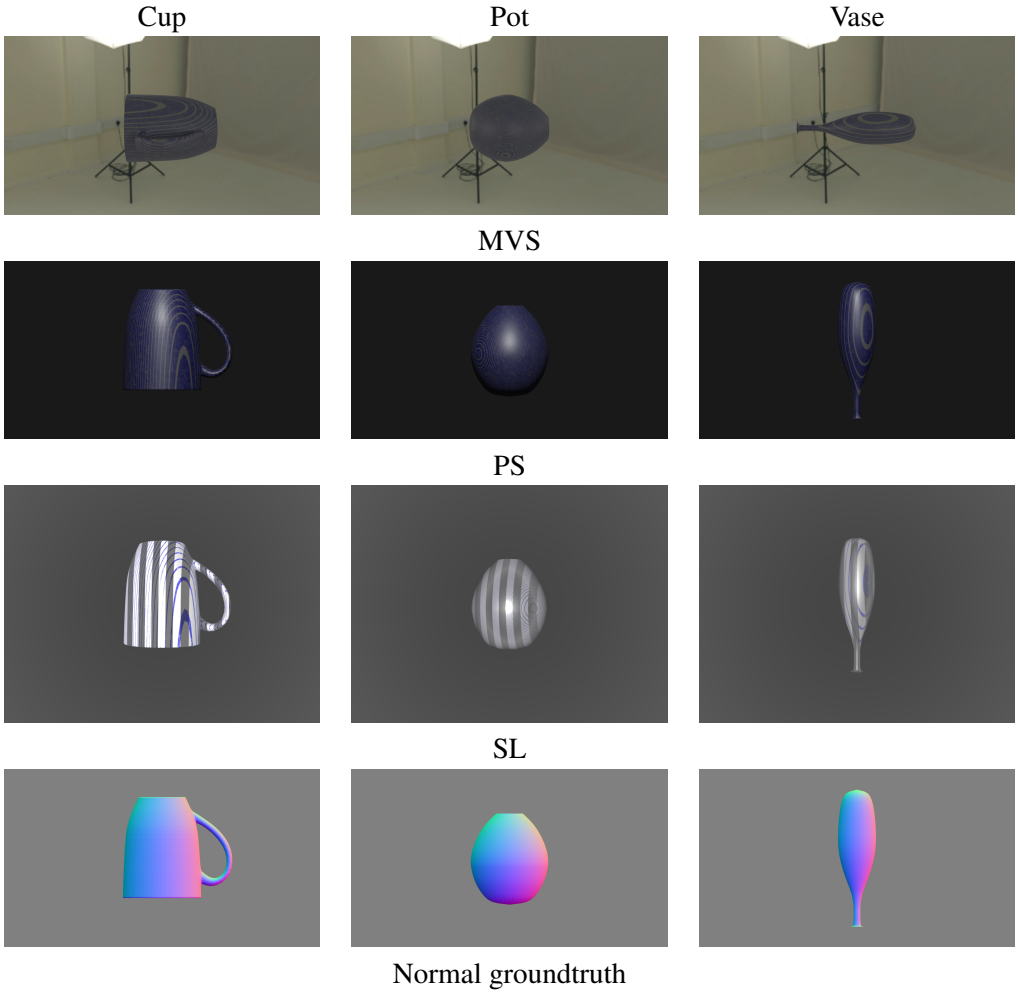


Figure 6.2: The synthetic datasets and groundtruth for the evaluation

(b).

Case 3 MVS performs well, and the completeness of MVS boosted compared to case 1, while the completeness of SL declines from case 1, see Figure 6.5, which are consistent to the result returned by the abstraction as shown in Table **??** (c).

Case 4 Both MVS and SL perform poorly in this case in terms of completeness. The accuracy of 'cup' and 'vase' are pretty good, which is not consistent to the abstraction, we argue that it's because their structure is too thin, and both

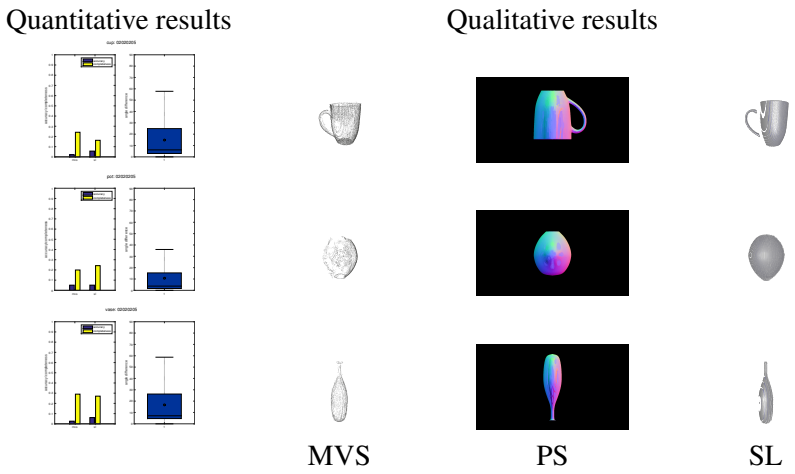


Figure 6.3: Property list: {tex:0.2, alb:0.2, spec:0.2, rough: 0.5}. The quantitative and qualitative performance of each technique on three test objects

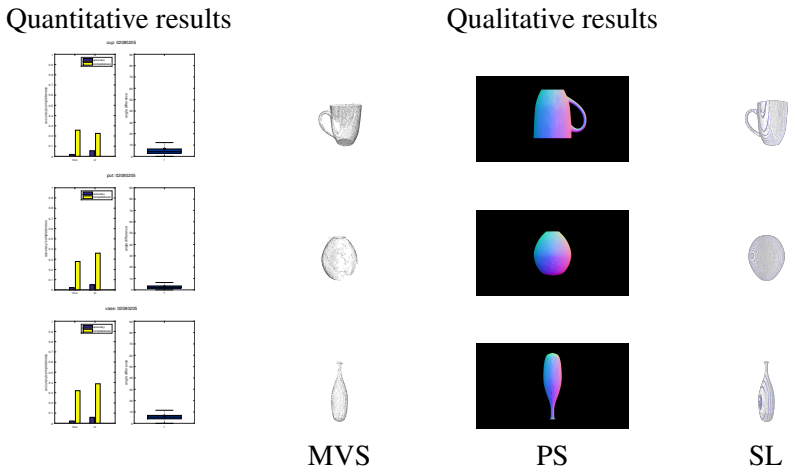


Figure 6.4: Property list: {tex:0.2, alb:0.8, spec:0.2, rough: 0.5}. The quantitative and qualitative performance of each technique on three test objects

the interior and outside of the cup is textured, which helps improve the accuracy. This shows the need to incorporate more visual and geometric properties to make the abstraction more robust. The PS performs the best among these techniques,

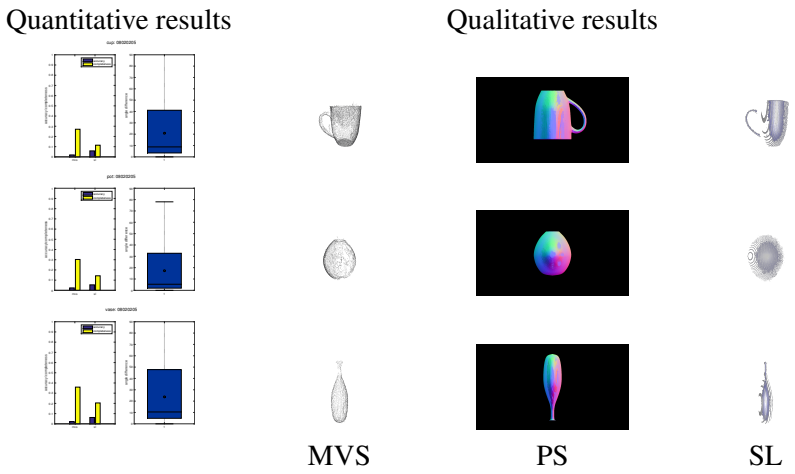


Figure 6.5: Property list: {tex:0.8, alb:0.2, spec:0.2, rough: 0.5}. The quantitative and qualitative performance of each technique on three test objects

which is still consistent to the abstraction as shown in Table **??** (d). Please refer to Figure 6.6.

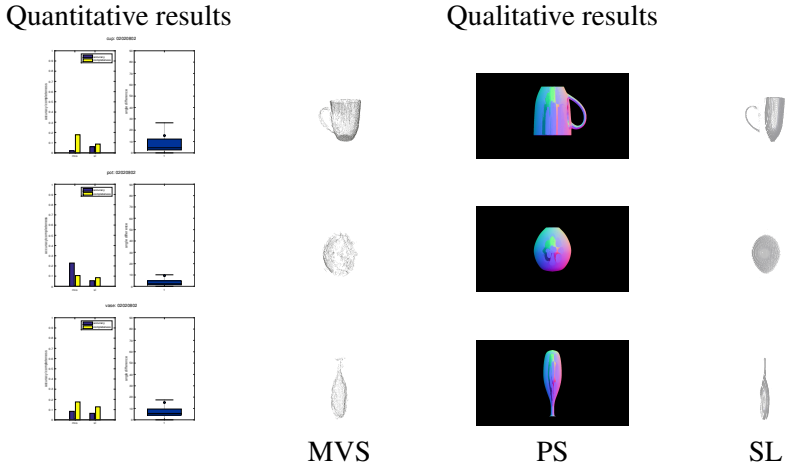


Figure 6.6: Property list: {tex:0.2, alb:0.2, spec:0.8, rough: 0.2}. The quantitative and qualitative performance of each technique on three test objects

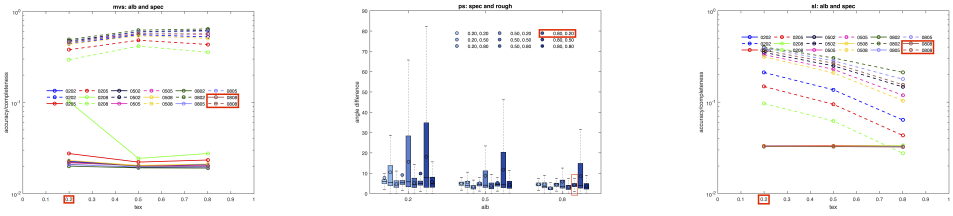
6.2 Real-world Datasets

We use the dataset ‘cup’ as an example. The property of the ‘cup’ is listed in Table **??**.

Property	Texture coverage	Albedo	Specularity	Roughness
cup	0.2	0.8	0.8	0.2

Table 6.2: Property list for the real-world objects

From the trained performance of each technique as shown in Figure 6.7



(MVS) (PS) (SL)

Figure 6.7: Performance of MVS, PS, and SL with varied properties

From the trained performance, we can clearly see that MVS performs poorly, as it ranks fifth among all 9 combinations, after 020802, 020502, 020805, 020505 (order of property as previously stated), thus we conclude that MVS is not suitable for ‘cup’.

From the performance of PS, we can see that it performs well in this case, as the mean and median angle different is below 10°, and mean and median are not far apart, suggesting that there is no spikes.

From the performance of SL, we can see that it ranks top 3 among all 9 combinations in terms of completeness, thus SL also does a decent job reconstructing ‘cup’.

Following the same methods, we obtain the best-suited algorithm(s) for all the other objects as shown in Table **??**.

Here we show the reconstruction of the real-world datasets. Since we don’t have the ground truth, visual

Property	Texture coverage	Albedo	Specularity	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	0.5	PS, SL
house	0.8	0.2,0.8	0.8	0.2	MVS
pot	0.5	0.2,0.5	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	MVS

Table 6.3: Property list for the real-world objects

6.3 Observations

- roughness on ps
- low albedo, high specularity on SL
- low albedo, high specularity, low roughness, high spikes

6.4 Summary

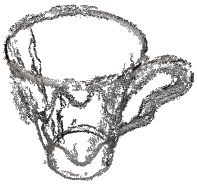
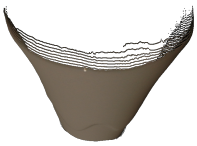
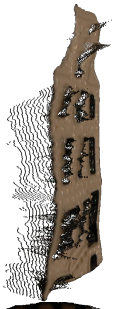
Object	$C_n - S - T - P - P$	$C_1L_n - T - I - MDS - N$	$C_1P - T - I - B - P$	Best-suited Algo.
box				MVS, SL, PS
cat0				None
cat1				None
dino				PS, SL
cup				PS, SL
house				MVS
pot				MVS, SL

Figure 6.8: Reconstruction results of MVS, PS, SL

Object	$C_n - S - T - P - P$	$C_1L_n - T - I - MDS - N$	$C_1P - T - I - B - P$	Best-suited Algo.
statue				PS, SL
vase				MVS

Figure 6.9: 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] 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 17

[5] 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 18

[6] 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 12

[7] 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 17

[8] 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

[9] F. Blais. Review of 20 years of range sensor development. *Journal of Electronic Imaging*, 13(1), 2004. → pages 10

[10] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo-stereo matching with slanted support windows. In *BMVC*, volume 11, pages 1–11, 2011. → pages 12

[11] G. Bradski and A. Kaehler. *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc.", 2008. → pages 8

[12] 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 17

[13] G. R. Cross and A. K. Jain. Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (1):25–39, 1983. → pages 40

[14] J. Davis, R. Ramamoorthi, and S. Rusinkiewicz. Spacetime stereo: A unifying framework for depth from triangulation. In *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, volume 2, pages II–359. IEEE, 2003. → pages 24

[15] 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 11, 13

[16] O. Faugeras and R. Keriven. *Variational principles, surface evolution, pde's, level set methods and the stereo problem*. IEEE, 2002. → pages 1, 11

[17] 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, 12, 49, 59

[18] 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 13

[19] J. Geng. Structured-light 3d surface imaging: a tutorial. *Advances in Optics and Photonics*, 3(2):128–160, 2011. → pages 24

[20] 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, 14

[21] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67(5):786–804, 1979. → pages 39

[22] 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 49, 59

[23] 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 11

[24] X. Hu and P. Mordohai. A quantitative evaluation of confidence measures for stereo vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2121–2133, 2012. → pages 13

[25] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. Available from: <http://www.peterkovesi.com/matlabfns/>. → pages 8

[26] 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, 11

[27] 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

[28] 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 12

[29] 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 17

[30] 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

[31] D. Marr. Vision: A computational investigation into the human representation and processing of visual information. 1982. → pages 11

[32] S. K. Nayar, K. Ikeuchi, and T. Kanade. Surface reflection: physical and geometrical perspectives. Technical report, DTIC Document, 1989. → pages 42

[33] S. K. Nayar, K. Ikeuchi, and T. Kanade. Determining shape and reflectance of hybrid surfaces by photometric sampling. *IEEE Transactions on Robotics and Automation*, 6(4):418–431, 1990. → pages 17

[34] G. P. Otto and T. K. Chau. region-growingalgorithm for matching of terrain images. *Image and vision computing*, 7(2):83–94, 1989. → pages 12

[35] T. Poggio, V. Torre, and C. Koch. Computational vision and regularization theory. *Nature*, 317(6035):314–319, 1985. → pages 9

[36] 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 20

[37] J. Salvi, J. Pages, and J. Batlle. Pattern codification strategies in structured light systems. *Pattern recognition*, 37(4):827–849, 2004. → pages 15, 24, 28

[38] Y. Sato and K. Ikeuchi. Temporal-color space analysis of reflection. *JOSA A*, 11(11):2990–3002, 1994. → pages 17

[39] K. Schlüns. 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 17

[40] 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, 10, 24, 49

[41] 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 17

[42] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008. → pages 8

[43] 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 13

[44] 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 16

[45] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):191139–191139, 1980. → pages 1, 16, 17

[46] T. E. Zickler, P. N. Belhumeur, and D. J. Kriegman. Helmholtz stereopsis: Exploiting reciprocity for surface reconstruction. *International Journal of Computer Vision*, 49(2-3):215–227, 2002. → pages 17

Appendix A

Supporting Materials

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

- additional details of methodology and/or data;
- diagrams of specialized equipment developed.;
- copies of questionnaires and survey instruments.