Robust Surface Normal Estimation via Greedy Sparse Regression

by

Mingjing Zhang

B.Sc., China University of Geosciences (Wuhan), 2006

a Thesis submitted in partial fulfillment
of the requirements for the degree of

Master of Science

in the
School of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Spring 2014

APPROVAL

Name:	Mingjing Zhang
Degree:	Master of Science
Title of Thesis:	Robust Surface Normal Estimation via Greedy Sparse Regression

Examining Committee: Dr. Jiangchuan Liu
Chair

Dr. Mark S. Drew,
Professor, Senior Supervisor

Dr. Torsten Möller,
Professor, Supervisor

Dr. Ze-Nian Li,
Professor, SFU Examiner

Date Approved: January 7th, 2014

Partial Copyright Licence

Abstract

Photometric Stereo (PST) is a widely used technique of estimating surface normals from an image set. However, it often produces inaccurate results for non-Lambertian surface reflectance. In this study, PST is reformulated as a sparse recovery problem where non-Lambertian errors are explicitly identified and corrected. We show that such a problem can be accurately solved via a greedy algorithm called Orthogonal Matching Pursuit (OMP). Furthermore, we introduce a smoothness constraint by expanding the pixel-wise sparse PST into a joint sparse recovery problem where several adjacent pixels are processed simultaneously, and employ a Sequential Compressive - Multiple Signal Classification (SeqCS-MUSIC) algorithm based on Simultaneous Orthogonal Matching Pursuit (S-OMP) to reach a robust solution. The performance of OMP and SeqCS-MUSIC is evaluated on synthesized and real-world datasets, and we found that these greedy algorithms are overall more robust to non-Lambertian errors than other state-of-the-art sparse approaches with little loss of efficiency.

Keywords: Photometric stereo; robust regression; sparse recovery; orthogonal matching pursuit; sequential compressive MUSIC

To my parents, Xiang Zhang and Xiuqin Gao,
and my little sister, Mingzhu Zhang.

Acknowledgments

First and foremost, I would like to express my most sincere gratitude to my Senior Supervisor, Dr. Mark Drew, for the full support and guidance throughout my entire Master's degree run. He always gives prompt answers and suggestions to my problems, but leaves enough space for me to learn and think autonomously. He is passionate, rigorous and very open to new ideas. As his student, I feel well-instructed and constantly inspired. I am lucky to have him as my Senior Supervisor and I can hardly imagine myself doing research with someone who is even better.

I am obliged to Carla Shroer and Mark Mudge at Cultural Heritage Imaging, San Francisco, for their inspiration and generous assistance. The gestation of the research idea employed in this thesis owes a lot to their insightful comments and suggestions during my collaboration with them. I am also very grateful for the dataset they provided for the pilot test of my algorithms.

My thanks also go to all my present and former colleagues in Vision and Media Laboratory at Simon Fraser University. This thesis is not possible without the accumulation of the encouragement and help from each and every one of them.

Last but not least, I wish to thank my parents Xiang Zhang and Xiuqin Gao, and my little sister Mingzhu Zhang, for their deepest, trans-Pacific love and trust from my home country. I know they are always there for me no matter what happens, and because of that, I know I am neither alone nor afraid.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

1.1.1 Photometric stereo

Shading in 2D images provides valuable visual cues for understanding the spatial structure of objects. Photometric Stereo (PST) is a powerful technique that exploits the shading information to directly estimate the 3D surface orientation, i.e. normal vectors. In the classical PST problem, the input is a set of n images captured from a fixed viewpoint under n different calibrated lighting conditions; hence there are n observations of luminance at each pixel location. Under the assumption of a Lambertian reflectance model, where the observed luminance is proportional to the cosine of the incident angle and remains constant regardless of the viewing angle, the relationship between n observations $\mathbf{y} \in \mathbb{R}^n$ at each pixel and the collection of n lighting directions $\mathbf{L} \in \mathbb{R}^{n\times 3}$ is formulated as a linear equation group with respect to the normal vector $\mathbf{n} \in \mathbb{R}^3$, i.e.:

$$\mathbf{y} = \mathbf{L}\mathbf{n} \tag{1.1}$$

Then the linear system Eq. 1.1 is solved via ordinary Least Squares (LS). The advantage of PST over 3D laser scanning is that the former provides a very high resolution (depending on the actual resolution of the camera) and therefore could capture the fine details of the surface that may not show up in the scanned model. In addition, PST only requires a simple and inexpensive hardware setup whereas 3D scanning devices are usually costly and less portable.

Although the classical PST method almost always guarantees a visually plausible normal map, it in fact suffers from a serious accuracy problem: The simple Lambertian reflectance model adopted in PST does not strictly apply to most real-world textures, which exhibit specular reflection properties to various degrees. Even if the surface is indeed approximately Lambertian, other non-Lambertian errors can be introduced by the interaction of the light and the objects' geometry, resulting in cast shadows and interreflections. These non-Lambertian observations, regarded as "outliers" in a Lambertian-based linear model, may severely reduce the accuracy of LS results. Hence, a PST method that is robust to such non-Lambertian effects is needed in order to generate a high quality normal map.

Many improved PST methods have been proposed since the original PST in an attempt to minimize the effect of non-Lambertian components. These methods either adopt a more sophisticated reflectance model to accommodate non-Lambertian observations as "inliers" (e.g. [32, 5, 4]), or rather, keep the Lambertian model but use robust statistical methods to rule out or reduce the effect of non-Lambertian outliers (e.g. [64, 31, 26]). A typical example of the second category is the Least Median of Squares (LMS) approach used in our previous study [72] (and see [26, 25]), in which the observations outside a certain confidence band are deemed to be outliers. In this study, we also adopt the Lambertian model, but solve for the normal vectors via a sparse representation framework that estimates both the normals and non-Lambertian errors at the same time. This sparse method is more closely related to the statistical-based methods. However, instead of rejecting the errant observations, our method attempts to "correct" them so that they would follow a Lambertian model as closely as possible. Eventually, the normal vector that requires the least amount of correction is taken as the final solution.

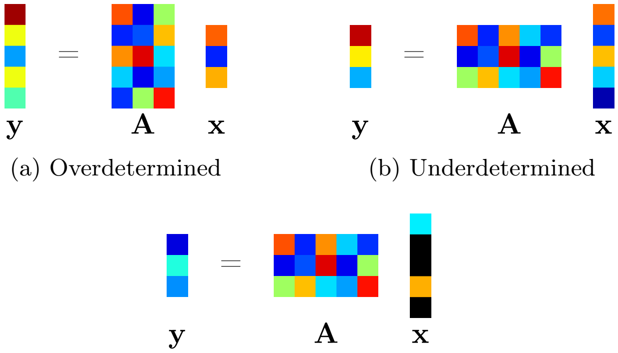
1.1.2 Sparse representation and recovery

It is well understood that ordinary LS fails to unambiguously reconstruct a signal that is passed through an underdetermined linear system, where the number of unknown variables exceeds that of linear equations (Fig. 1.1b). However, it has been shown that if the signal to be recovered is sparse – having a considerable amount of zero or nearly zero entries (Fig. 1.1c), then an accurate reconstruction of the signal is still possible via a sparse recovery scheme [23].

The canonical form of a sparse recovery problem can be stated as follows: Given an underdetermined linear model $\mathbf{y} = \mathbf{A}\mathbf{x}$, where $\mathbf{A} \in \mathbb{R}^{n\times p}$ is the so-called "dictionary" matrix ($n < p$), and $\mathbf{y} \in \mathbb{R}^{n\times 1}$ is the vector consisting of n scalar observations, find the unknown sparse signal $\mathbf{x} \in \mathbb{R}^{p\times 1}$ such that:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \qquad \text{s.t.} \quad \mathbf{y} = \mathbf{A}\mathbf{x} \tag{1.2}$$

where $\|\cdot\|_0$ represents ℓ_0 pseudo-norm, or the number of non-zero entries.



(a) Overdetermined

(b) Underdetermined

(c) Sparse

Figure 1.1: Stylized visualization of three examples of linear equation systems. $\mathbf{A}$ and $\mathbf{y}$ represent the design matrix and observations, respectively; $\mathbf{x}$ is the unknown signal to be recovered. Positive and negative values are shown as coloured blocks, and zero entries are represented by black blocks. (a) Overdetermined system, where there are more observations (5) than unknowns (3). (b) Underdetermined system, where there are more unknowns (5) than observations (3). The signal $\mathbf{x}$ cannot be uniquely determined from such a system. (c) Underdetermined system with sparse signal. It is possible to recover $\mathbf{x}$ using sparse recovery methods as long as we know that $\mathbf{x}$ is sparse, even though the system is underdetermined and the exact positions of non-zero entries are not known *a priori*.

Eq. 1.2 is generally an NP-hard combinatorial problem [47]. In practice, it is more feasible to solve a relaxed form thereof. We will briefly discuss various alternative formulations and corresponding solvers in Section 2.2.

Photometric stereo and sparse recovery

PST is often formulated as an overdetermined regression problem. The classical PST adopts three lights (hence three observations of luminance at each pixel location) [67] to solve for the 3D normal vectors. Later methods use more lights ranging from four to hundreds [7, 54, 26, 5]. Recently, a few attempts have been made to represent PST as an underdetermined system firstly in the case of calibrated lighting directions [69, 33], as addressed here, and for the case of unknown lighting conditions [27, 6], not studied in this report. Reconfiguring PST as an underdetermined system means explicitly modelling the non-Lambertian components as additional unknowns. Suppose there are n lights (hence n equations for each pixel), the total number of unknowns would be $n + 3$ (3 normal vector components and n non-Lambertian error terms). As was already pointed out, such a system cannot be unambiguously solved through ordinary LS. Fortunately, if we make an assumption that the majority of luminance observations are approximately Lambertian, which is true for many real-world materials, then the error vector is essentially a sparse vector with a large number of zero or approximately zero entries. Now that we have a sparse representation of the PST problem, we can solve it using a sparse recovery algorithm.

It has been shown by Wu et al. [69] and Ikehata et al. [33] that sparse PST behaves significantly more robustly than the classical PST method. However, the accuracy is contingent on the solver. As of now, the most accurate solver for the sparse formulation is Sparse Bayesian Learning (SBL) as tested by Ikehata et al. in [33]. In the current study, we employ a modified form of the sparse representation given in [33], and solve it via a different type of approach – greedy sparse recovery.

1.2 Contribution

The main contributions of the current study are fourfold:

1. We propose an alternative sparse formulation for PST (Eq. 4.6 and Eq. 4.7). In previous sparse PST studies [69, 33], the surface normal vector and the error vector are treated as two entities and are solved independently. In this study, we convert their formulations into a new canonical form of the sparse recovery problem by combining the two vectors into one large vector. Although such a "stacked" formulation is not novel (e.g. [68]), it is used in the context of surface normal estimation for the first time. The advantage of this formulation is that it allows for a large repertoire of existing sparse recovery algorithms to be more straightforwardly applied to the PST problem.

2. We apply a greedy algorithm called Orthogonal Matching Pursuit (OMP) [52, 61, 11], from Information Theory, to solve the PST problem. It has been previously demonstrated in [69, 33] that PST can be solved by several sparse recovery algorithms that fall into different categories, including augmented Lagrangian rank-minimization [69],

ℓ_1 optimization approaches and probability-based methods [33]. However, the possibility of applying greedy solvers, an important category of sparse recovery algorithms, to the PST problem has never been explored. To the best of our knowledge, this study is the first such attempt at employing greedy approaches to estimate surface normals within the framework of PST.

3. We employ a smoothness constraint by recovering several neighbouring pixel locations at once. Specifically, we reformulate PST as a joint sparse problem, and solve it using an extension of OMP – Simultaneous Orthogonal Matching Pursuit (S-OMP) [62, 63] from Signal Processing. In order to further improve the results, another refining approach, again from Signal Processing, named Sequential Compressive - Multiple Signal Classification (SeqCS-MUSIC) [39] is also applied following the initial greedy estimation. As of yet, no previous attempts have been made at introducing a joint sparse representation or any MUSIC-like approach into PST.

4. We numerically compare the performance of several normal vector recovery methods. Most notably, it is the first time that the normal estimation accuracy of our previously proposed method – LMS – has been tested and quantitatively demonstrated on complex models.

1.3 Thesis Overview

This thesis is organized as follows: Chapter 2 provides a short survey on recent robust PST and sparse recovery methods. Chapter 3 briefly recapitulates the mathematical details of the classical PST, as well as one of our previous state-of-the-art robust PST methods: Least Median of Squares (LMS) regression. To strike a balance between efficiency and accuracy, we also introduce a 1D version of LMS (a.k.a. “mode-finder”) in Section 3.2 which is significantly faster than regular LMS [72]. In Chapter 4, we provide a detailed description of our sparse formulation and the OMP algorithm. Then in Chapter 5, we further extend PST into a joint sparse problem, and attempt to solve it via S-OMP and SeqCS-MUSIC. Experimental results and discussions are presented in Chapter 6, followed by several possible future research directions discussed in Chapter 7.

Chapter 2

Related Work

2.1 Robust Photometric Stereo

This section presents a brief overview of current robust Photometric Stereo (PST) methods. Since the original non-robust Lambertian-based PST [67], many methods have been proposed in an attempt to address non-Lambertian effects such as specularities and shadows. These approaches usually adopt a ***robust statistical method*** and/or an improved ***non-Lambertian reflectance model***.

2.1.1 Statistics-based methods

In statistics-based methods, a robust statistical algorithm is employed to detect the non-Lambertian observations as outliers, and exclude them from the estimation process in order to minimize their influence on the final result. Early examples include a 4-light PST approach in which the values yielding significantly differing albedos are excluded [19, 57, 7]. In a similar 5-light PST method [54], the highest and the lowest values, presumably corresponding to highlights and shadows, are simply discarded.

Another 4-light method [71] explicitly includes ambient illumination and surface integrability, and adopts an iterative strategy, using current surface estimates to accept or reject each additional light based on a threshold indicating a shadowed value. The problem with these methods is that they rely on throwing away a small number of outlier observation values, whereas our robust sparse methods in the current study reach the solution based on all observations, by correcting the non-Lambertian error of the outliers.

Willems et al. [65] apply an iterative method to estimate normals. First, the pixel values within a certain range (10–240 out of 255) are used to estimate an initial normal map. In each of the following iterations, error residuals of normals for all lighting directions are computed and the normals are updated based only on those directions with small residuals. Sun et al. [58] show that at least six light sources are needed to guarantee that every location on the surface is illuminated by at least three lights. They propose a decision algorithm to discard only doubtful pixels, rather than throwing away all pixel values that lie outside a certain range. However, the validity of their method is based on the assumption that out of the six values for each pixel, there are at most one highlight and two shadow observations. Julià et al. [35] utilize a factorization technique to decompose the luminance matrix into surface and light source matrices, in which the shadow and highlight pixels are considered as missing data.

Some recent studies introduce probability models as a mechanism to incorporate the handling of shadows and highlights into the PST formulation. Tang et al. [60] model normal orientations and discontinuities with two coupled Markov Random Fields (MRF). They propose a tensorial belief propagation method to solve the *Maximum A Posteriori* (MAP) problem in the Markov network. Chandraker et al. [17] formulate PST as a shadow labelling problem where the labels of each pixel's neighbours are taken into consideration, enforcing the smoothness of the shadowed region, and approximate the solution via a fast iterative graph-cut method. Another study [64] employs a Maximum-Likelihood (ML) imaging model for PST. In this method, an inlier map modelled via MRF is included in the ML model. However, the initial estimation of the inlier map would directly influence the final result, whereas our sparse methods do not depend on the choice of any prior.

A few other studies employ random sampling based methods. Using 3-light datasets, Mukaigawa et al. [46] adopt a Random Sample Consensus (RANSAC) based approach to iteratively select random groups of pixels from different regions of the image, and the sampled groups whose pixels are all taken from diffuse regions are used to calculate the coefficients in the linear equation. RANSAC is also used in a multiview context [31] as a robust fitting approach to select the points on a certain 3D curve. Drew et al. [26, 25] and Zhang and Drew [72] employ a Least Median of Squares (LMS) method. Instead of taking samples from different regions on the image, they use a denser image set (50 lights) and sample only from the observations at each pixel location. Non-Lambertian observations are rejected as outliers and excluded from the following LS step. Based on [17], Miyazaki et al. [45] use a median filtering approach similar to LMS but also considering neighbouring pixels. Instead of taking random samples, they simply compare all the 3-combinations of observations, which is feasible for the small number of lights used in their study. Although guaranteeing a high statistical robustness, these methods are computationally heavy since they usually rely on a large number of samples to take effect.

2.1.2 Non-Lambertian reflectance modelling

Instead of statistically rejecting non-Lambertian observations as outliers, another way to minimize their negative influence on surface normal recovery is to incorporate a more sophisticated reflectance model to directly account for the non-Lambertian components.

Tagare and de Figueiredo [59] construct an m-lobed reflectance map model to approximate diffuse non-Lambertian surface-light interactions. In [57], a Torrance-Sparrow model is employed to estimate the roughness of the surface that is divided into different areas. Similarly, Nayar et al. [48] adopt a Torrance-Sparrow and Beckmann-Spizzichino hybrid reflectance model. Other mathematical models to encode surface reflectance include a polynomial model as applied in Polynomial Texture Mapping (PTM) [44], and Spherical Harmonics (SH) [8]. Drew et al. [26] propose a Radial Basis Function (RBF) interpolation to handle the rendering of specularities and shadows. These models, however, generally require a large number of parameters to be estimated.

Other studies use reference objects to facilitate the estimation of surface properties. In [56], an object with simple, known 3D geometry and approximately Lambertian reflectance (for instance, a white matte sphere) is present in the captured images. A look-up table is established that relates luminance observations at each pixel location and the surface orientation. Then the surface properties of other objects with similar reflectance as the reference object can simply be inferred from the look-up table. This method, however, only applies to isotropic materials. Hertzmann and Seitz [32] later revisited the idea of including reference material. By adopting an orientation-consistency cue assumption that two points on the surface with the same orientation have the same observed light intensity, they effectively cast PST as a stereoptic correspondence problem. This approach is capable of handling a wider range of anisotropic materials with a small number of reference objects, usually one or two. Similar to [32], an appearance clustering method proposed by Koppal and Narasimhan [40], also adopting the orientation consistency cue, focuses on finding isonormals across frames in a captured image sequence, and a classical PST approach is applied later to obtain the actual values of the surface normals. Although their method does not rely on the presence of a reference object, it does require the image sequence to be densely captured on a continuous path.

Recent studies attempt to solve a more complicated problem where neither shape nor material information of the object surface is available. Goldman et al. [30] employ an objective function that contains terms for both shape and material, and apply an iterative approach where the reflectance and shape are alternately optimized. The estimation of the material is an inseparable part of the reconstruction process so an explicit reference object is no longer needed. Alldrin et al. [5] also adopt a similar iterative approach that updates shapes and materials alternately. Their formulation is non-parametric and data-driven, and as such is capable of capturing an even wider range of reflectance materials. Ackermann et al. [4] propose an example-based multi-view PST method which uses the captured object's own geometry as reference.

Yang and Ahuja [70] include a dichromatic reflection model into PST for both estimating surface normals as well as separating the diffuse and specular components, based on a surface chromaticity invariant. Their method is able to reduce the specular effect even when the specular-free observability assumption (that is, each pixel is diffuse in at least one input image) is violated. However, this method does not address shadows and fails on surfaces that mix their own colours into the reflected highlights, such as metallic materials. Moreover, their method also requires knowledge of the lighting chromaticity – they suggest a simple white-patch estimator – whereas in our method we have no such requirement. Kherada et al. [37] propose a component-based mapping (CBM) method. They decompose the captured images into direct components (single bounce of light from a surface) and global components (illumination onto a point that is interreflected from all other points in the scene), and then model matte, shadow and specularity separately within each component. This method depends on a training phase and requires accurate disambiguation of direct and global contributions, and has a high computational load.

The problems with these methods are that they either adopt a restricted assumption of the surface reflectance, or require a certain type of reference, either by using an explicit reference object or through self-referencing. Moreover, they usually do not work well against non-Lambertian effects that are not accounted for by the surface reflectance alone, such as cast shadows. In our current sparse method, we make no assumption of the surface reflectance property and treat all non-Lambertian effects equally.

2.1.3 Sparse formulation

Sparse representation has been incorporated into PST in a few recent studies. Wu et al. [69] model the matrix of all luminance observations as a linear combination of Lambertian and non-Lambertian components, and represent the non-Lambertian error as an additive sparse noise matrix. Under the assumption that most pixel observations approximately follow the Lambertian reflectance model, they obtain the solution by finding a sparse error matrix such that the rank of the Lambertian component matrix is minimized. The formulation is known as Robust Principal Component Analysis (R-PCA) in the field of sparse recovery. Specifically, they adopt a fast and scalable algorithm suitable for handling a large amount of data points, i.e. the Augmented Lagrange Multiplier method [41]. However, this method requires a shadow mask to be specified explicitly. Ikehata et al. [33] reconsider PST as a pure sparse regression problem and aim to minimize the number of non-zero entries (i.e. the ℓ_0 pseudo norm) in the error matrix. They also add an ℓ_2 relaxation term to account for cases when the sparse assumption is violated. In order to avoid the difficult combinatorial problem involved in the minimization of ℓ_0 norm, they introduce two approximation algorithms. One is to relax the ℓ_0 pseudo norm into ℓ_1 norm, as justified in [23, 12], and the solution is obtained via Iteratively Reweighted L1 minimization (IRL1) [15]. The other method is a hierarchical Bayesian approach called Sparse Bayesian Learning (SBL) [66]. It has been shown that SBL has an improved accuracy over IRL1 at the expense of lower efficiency, and both IRL1 and SBL perform better than R-PCA [33].

Sparse methods have also found their use in uncalibrated PST, where the lighting directions are not known (but note that in this study we do assume known lighting directions so that these works are somewhat peripheral). Favaro and Papadhimitri [27] incorporate the rank-minimization algorithm proposed in [69] into the uncalibrated PST problem as a pre-processing step to remove shadow and specularity effects. Argyriou et al. [6] recently also adopt a sparse representation framework to decide the weights for finding the best illuminants to use, again with the lighting directions unknown.

2.2 Sparse Recovery Methods

As was pointed out in Section 1.1.2, the canonical form of the sparse recovery problem (Eq. 1.2) is NP-hard [47] and cannot be solved efficiently as-is. In this section, we summarize alternative formulations to Eq. 1.2 and several types of solvers.

The first type of approach is convex ℓ_1 relaxation. It has been shown that for a dictionary matrix $\mathbf{A}$ that satisfies a certain restriction, Eq. 1.2 is likely to be equivalent to an ℓ_1 minimization problem [23, 12]:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \qquad \text{s.t.} \quad \mathbf{y} = \mathbf{A}\mathbf{x} \tag{2.1}$$

which can be solved via convex optimization techniques such as Interior-Point (IP) methods [10], Gradient Projection [28], Iteratively Reweighted L1 (IRL1) [15], and so forth.

Alternatively, sparse recovery can be achieved via greedy algorithms. The basic idea is to find the non-zero entries, one at a time, based on a certain criterion, and then recover $\mathbf{x}$ via LS using only the observations in the support.

One of the most notable greedy algorithms is Orthogonal Matching Pursuit (OMP) [52, 61, 11], an improvement over the simple Matching Pursuit (MP) algorithm [43]. In OMP, the column of the dictionary matrix $\mathbf{A}$ that has the strongest correlation with the current residual $\mathbf{r}$ is selected at every iteration, and $\mathbf{r}$ in turn is iteratively updated based on the previously selected columns. The algorithm is terminated as a fixed number of non-zero entries are recovered or other stopping criteria are met. Then, a simple LS is performed only on a submatrix of $\mathbf{A}$ consisting of the columns chosen by OMP, and the regressed result will be assigned only to the signal entries corresponding to the selected columns. The columns that are not selected by OMP, on the other hand, will not be used in the final LS step, and their corresponding signal entries are simply set to zero.

In fact, OMP approximately solves the following k-sparse recovery problem:

$$\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \qquad s.t. \quad \|\mathbf{x}\|_0 \leq k \tag{2.2}$$

Many state-of-the-art greedy algorithms nowadays are based on OMP. Examples include Regularized OMP (ROMP) [51, 49], Stagewise OMP (StOMP) [24], and Compressive Sampling Matching Pursuit (CoSaMP) [50], Probability OMP (PrOMP) [21], Look ahead OMP [18], OMP with Replacement (OMPR) [34], A* OMP [36], etc.

Some solvers employ a thresholding step to iteratively refine the recovered support, i.e. the selection/rejection of an entry at each step is decided by whether the value of a certain function dependent on this entry falls below a given threshold. Algorithms in this category include Iterative Hard Thresholding (IHT) [9], Subspace Pursuit(SP) [20], Approximate Message Passing (AMP) [22], Two-Stage Thresholding (TST) [42], Algebraic Pursuit (ALPS) [16], etc.

The fourth category is probability-based algorithms. These methods assume the signal to be recovered follows a specific probability distribution and solve the sparse recovery problem with statistical methods such as Maximum Likelihood (ML) or Maximum A Posterior (MAP) estimation. Sparse Bayesian Learning (SBL) [66] is one of the major algorithms in this category, and has already been applied in the context of PST [33].

Chapter 3

Robust Estimation of Surface Normal Vectors

3.1 Naive Lambertian-based Photometric Stereo

The classical Photometric Stereo (PST) method of estimating surface normals from multiple images was proposed by Woodham [67]. PST usually assumes the object surface being inspected follows a Lambertian reflectance model; i.e. the observed luminance y at each point on the surface is proportional to the cosine of the incident light angle at that point, or in vector form, the dot product of the surface normal $\mathbf{n} \in \mathbb{R}^3$ and lighting direction $\boldsymbol{l} \in \mathbb{R}^3$:

$$y = \alpha \boldsymbol{l} \cdot \mathbf{n} \tag{3.1}$$

where α is a scalar constant representing *albedo*.

Now suppose we have a set of n images of an object surface taken at a fixed viewpoint with constant exposure but illuminated by n different distant parallel light sources, respectively. Let $\boldsymbol{l}_j$ denote the jth lighting direction, and $\mathbf{L} = (\boldsymbol{l}_1, \boldsymbol{l}_2, ...\boldsymbol{l}_n)^T \in \mathbb{R}^{n\times 3}$ the collection of all the lighting directions. For each pixel k, there are n observed luminances $\mathbf{y}^k = (y_1^k, y_2^k, ...y_n^k)^T \in \mathbb{R}^n$. Then we have n instances of Eq. 3.1 that can be collectively expressed as:

$$\mathbf{y}^k = \alpha^k \mathbf{L} \cdot \mathbf{n}^k \tag{3.2}$$

where again α^k is the albedo at pixel k.

Then the surface normal $\mathbf{n}^k$ can be simply solved via Least Squares (LS):

$$\mathbf{n}^k = \mathbf{L}^{\dagger}\mathbf{y}^k/\alpha^k \tag{3.3}$$

where

$$\alpha^k = \|\mathbf{L}^{\dagger}\mathbf{y}^k\| \tag{3.4}$$

and

$$\mathbf{L}^{\dagger} = (\mathbf{L}^T\mathbf{L})^{-1}\mathbf{L}^T \tag{3.5}$$

is the Moore-Penrose Pseudoinverse.

Eq. 3.3 can be extended to handle all pixels simultaneously. For an image with K pixels in total, let us construct a matrix $\mathbf{Y} \in \mathbb{R}^{n\times K}$ of all the $n \times K$ luminance observations as:

$$\mathbf{Y} = \left(\mathbf{y}^1, \mathbf{y}^2, ...\mathbf{y}^K\right) \tag{3.6}$$

Then, similar to Eq. 3.3, the unnormalized surface normal of all pixels can be solved via Least Squares:

$$\mathbf{N} = \mathbf{L}^{\dagger}\mathbf{Y} \tag{3.7}$$

where $\mathbf{N} = (\mathbf{n}^1, \mathbf{n}^2, ...\mathbf{n}^K) \in \mathbb{R}^{3\times K}$ whose K columns represent surface normals at K pixel locations, respectively. The albedo can be computed as the norm of each column vector in $\mathbf{N}$ as in Eq. 3.4. Note that in the current work, we only focus on the accurate estimation of surface normals, although our methods can potentially be used to recover albedo as well.

Eq. 3.7 is capable of producing a smooth and visually consistent estimation of surface normal vectors (for instance, Fig. 6.2c), provided $\mathbf{L}$ has a rank of 3; i.e. there are at least 3 lighting directions that do not lie on the same line. Moreover, such an LS solver has a very low computational cost. However, the LS result is usually erroneous (see the angular error map shown in Fig. 6.2d) due to the prevalence of non-Lambertian properties in real-world images, such as specularities, shadows, interreflections, etc. Hence, an accurate recovery of surface normal based on the Lambertian reflectance model must involve a robust method that is resistant to the non-Lambertian components.

3.2 Least Median of Squares

One way to deal with non-Lambertian effects in images is to treat the affected observations as outliers, and regress only on the inlier pixels. In our previous studies, we have adopted

Least Median of Squares (LMS) regression proposed by Rousseeuw [53]. For each pixel k, we have n given lighting directions $\mathbf{L} = (\boldsymbol{l}_1, \boldsymbol{l}_2, ...\boldsymbol{l}_n)^T \in \mathbb{R}^{n\times 3}$ and n corresponding observations of luminance $\mathbf{y}^k = (y_1^k, y_2^k, ...y_n^k)^T \in \mathbb{R}^n$. The LMS algorithm as applied in the context of PST can be outlined as follows:

1. Initialize the iteration counter $q = 1$.

2. Randomly sample a subset of d indices $J_q \subset \{1, 2, ..., n\}$ ($|J_q| = d$). Here we choose $d = 3$ since it is the smallest number required to unambiguously define a normal vector. Take the subset of lighting directions indexed by J_q, denoted by $\mathbf{L}_{J_q}$, and their corresponding observations $\mathbf{Y}_{J_q}$.

3. Perform Ordinary Least Squares on this subset. Obtain an estimation of normal vector $\mathbf{N}_{J_q}$:
$$\mathbf{N}_{J_q} = \mathbf{L}_{J_q}^{\dagger}\mathbf{Y}_{J_q} \tag{3.8}$$

4. Use the current estimation $\mathbf{N}_{J_q}$ to approximate $\mathbf{y}$, and calculate the squared residuals $r_1^2, r_2^2, ..., r_n^2$ for n observations, collectively stored in vector $\mathbf{R}^2 = (r_1^2, r_2^2, ..., r_n^2)$:
$$\mathbf{R} = \mathbf{y} - \mathbf{L}\mathbf{N}_{J_q} \tag{3.9}$$
$$\mathbf{R}^2 = \mathbf{R}^T\mathbf{R} \tag{3.10}$$

5. Find the median M_{J_q} of the n residuals $r_1^2, \cdots, r_n^2$ in $\mathbf{R}^2$:
$$M_{J_q} = \underset{i=1,\cdots,n}{\text{median}}\ r_i^2 \tag{3.11}$$

6. Increment q by 1, and go back to Step 2 until $q > m$. For our typical datasets where $n = 50$ and $d = 3$, we choose $m = 1500$.

7. Find the smallest median of squared residuals M_{min} such that:
$$M_{min} = \min_{q=1,\cdots,m} M_{J_q} \tag{3.12}$$
and keep the estimation of normal vector $\mathbf{n}$ that gives rise to M_{min}. Let the former be denoted by $\mathbf{n}_{min}$.

8. Estimate the robust standard deviation σ:

$$\sigma = 1.4826 \left(1 + \frac{5}{n-d}\right) \sqrt{M_{min}} \tag{3.13}$$

9. Obtain the squared residuals r_i^2 $(i = 1, \cdots, n)$ with respect to $\mathbf{n}_{min}$ for each of the n observations. Assign a binary weight w_i to each observation y_i^k based on σ such that:

$$w_i = \begin{cases} 1 & \text{if } r_i^2 \leq (2.5\sigma)^2 \\ 0 & \text{otherwise} \end{cases} \tag{3.14}$$

10. Obtain the LMS estimation for the kth pixel by performing LS only on observations with $w_i = 1$, i.e.:

$$\mathbf{n}_{LMS}^k = (\mathbf{wL})^\dagger (\mathbf{wy}^k) \tag{3.15}$$

where $\mathbf{w} = \text{diag}(w_1, w_2, \cdots, w_n) \in \mathbb{R}^{n \times n}$.

LMS has a 50% breakdown point, meaning it can tolerate up to 50% outliers. It has been found to be able to provide very robust normal estimates for test image sets [26]. However, its use is limited by its high computational cost. The time complexity of LMS is $O(n^d \log n)$, where $n = 50$ for a typical dataset and $d = 3$ (for PST) or higher (for more sophisticated reflectance models, such as Polynomial Texture Mapping [44, 26]). Note that the surface normal map recovered with LMS, though numerically more accurate than the LS result, does appear to be less smooth than the latter (for example, compare Fig. 6.8B6 and Fig. 6.8B1).

3.3 Mode Finder

It is desirable to have a robust method that runs faster than LMS while providing a smooth result. In this study, we have experimented with an alternative method known as LMS mode finder [72]. It simplifies the underlying general linear regression model of LMS:

$$y_i = x_{i1}\theta_1 + ... + x_{in}\theta_n + e_i \tag{3.16}$$

to a constant model:

$$y_i = \theta + e_i \tag{3.17}$$

where y_i represents the ith observation, x_{ik} and $\theta_{(k)}$ the independent variables and linear coefficients, respectively, and e_i the error term.

The idea is to classify the observations that are close enough to the robust mode as inliers, where the mode is identified using robust statistics. Mode finder is a special case of the general LMS, so the algorithm in Section 3.2 still applies with a slight modification:

ALGORITHM LMS Mode Finder

1. Initialize the iteration counter $q = 1$.
2. Calculate the n squared residuals r_i^2 $(i = 1, \cdots, n)$ with respect to the observation indexed by q:
$$r_i^2 = (y_i - y_q)^2 \tag{3.18}$$
3. Find the median M_q of all the squared residuals r_i^2:
$$M_q = \underset{i=1,\cdots,n}{\text{median}}\ r_i^2 \tag{3.19}$$
4. Increment q by 1, and go back to Step 2 until $q > n$.
5. Find the smallest median of squared residuals M_{min} such that:
$$M_{min} = \min_{q=1,\cdots,n} M_q \tag{3.20}$$
and keep the estimation of $\mathbf{n}$ that gives rise to M_{min}, denoted by $\mathbf{n}_{min}$.
6. Same as Step 8 – 10 in LMS.

LMS mode finder is essentially selecting a hyperplane that is perpendicular to the y direction that has the smallest median of squared residuals with respect to all the observations. Then LS is performed only on the data points that fall in a range around the hyperplane. In other words, we find a value θ that contains the greatest number of observations in the confidence band around it. This measurement of central tendency is similar to *mode*, except that the former decides a band rather than a single value, providing a degree of "tolerance". Since we only need to estimate one coefficient for the model, the mode-finder algorithm can be implemented much more efficiently than LMS.

In the context of PST, such a simplification is equivalent to a constant reflectance model where the observed luminance is invariant to the direction of the incoming lights. Notice that in our image datasets, shadowed and highlight pixels usually takes extreme brightness values (close to 0 and 1, respectively), whereas Lambertian observations tend to take intermediate values. Thus, although this model does not physically agree with most real-world surfaces, it is still helpful in rejecting the non-Lambertian observations that often deviate greatly from a centre value. We will show in Chapter 6 that LMS mode finder does exhibit an extent of robustness compared to LS.

Chapter 4

Sparse Regression

4.1 Sparse Formulation for Photometric Stereo

In this section, we explore the possibility of formulating and solving PST as a sparse regression problem. Since this study focuses only on the recovery of normal vectors, we omit the albedo term α from all equations in this and the following sections for simplicity, and always use $\mathbf{n}$ to represent the unnormalized surface normal vector unless otherwise specified.

Since most real-world objects do not have a strict Lambertian reflectance, it would be more accurate to introduce an additional term $e \in \mathbb{R}$ into Eq. 3.1 to account for the non-Lambertian error. Hence, Eq. 3.1 can be extended to:

$$y = \boldsymbol{l} \cdot \mathbf{n} + e \tag{4.1}$$

For each pixel k with n observations, we would have an n-vector of error $\mathbf{e} = (e_1, e_2, ...e_n)^T \in \mathbb{R}^n$. Now let us write Eq. 4.1 in vector form (similar to Eq. 3.2):

$$\mathbf{y} = \mathbf{L}\mathbf{n} + \mathbf{e} \tag{4.2}$$

Eq. 4.2, containing n linear equations but $n + 3$ unknowns (n components in $\mathbf{e}$ and 3 components in $\mathbf{n}$), is effectively an underdetermined problem, and as such cannot be solved unambiguously. However, if the error $\mathbf{e}$ is a sparse vector, i.e. most or at least a large percentage of its elements are zero, then it is still possible to recover $\mathbf{e}$ exactly or almost exactly using sparse methods. Fortunately, in many real-world images, often only a small number of pixels are heavily contaminated by shadows or highlights. For those image sets,

$\mathbf{e}$ is essentially sparse. This observation allows us to recover surface normal $\mathbf{n}$ and the non-Lambertian $\mathbf{e}$ simultaneously from the following sparse regression problem:

$$\min_{\mathbf{n},\mathbf{e}} \|\mathbf{e}\|_0 \quad s.t. \quad \mathbf{y} = \mathbf{L}\mathbf{n} + \mathbf{e} \tag{4.3}$$

In Eq. 4.3, $\|\cdot\|_0$ represents the ℓ_0 pseudo-norm, or the number of non-zero elements. This formulation, however, has two major issues: 1) It is an NP-hard combinatorial problem; 2) In real-world scenes, although many pixels are largely Lambertian, they may contain a small error due to ambient lighting, interreflections and so forth, and therefore $\mathbf{e}$ is usually not strictly sparse; there are many small elements that are close to but not exactly 0. Thus the equality constraint is very hard to be satisfied. Instead, it is more realistic to use an inequality constraint with a user-defined error tolerance ϵ:

$$\min_{\mathbf{n},\mathbf{e}} \|\mathbf{e}\|_0 \quad s.t. \quad \|\mathbf{y} - \mathbf{L}\mathbf{n} - \mathbf{e}\|_2 \leq \epsilon \tag{4.4}$$

Alternatively, if we care more about how much the reconstructed luminance approximates the real observations rather than the sparsity of $\mathbf{e}$, then it would be more natural to reformulate Eq. 4.4 as:

$$\min_{\mathbf{n},\mathbf{e}} \|\mathbf{y} - \mathbf{L}\mathbf{n} - \mathbf{e}\|_2 \quad s.t. \quad \|\mathbf{e}\|_0 \leq s \tag{4.5}$$

where the scalar s is the sparsity of vector $\mathbf{e}$. To further simplify Eq. 4.5, we propose merging $\mathbf{n}$ and $\mathbf{e}$ into one large vector and treating them as one entity, i.e.:

$$\begin{aligned} \mathbf{y} &= \mathbf{L}\mathbf{n} + \mathbf{e} \\ &= \mathbf{L}\mathbf{n} + \mathbf{I}\mathbf{e} \\ &= (\mathbf{L}, \mathbf{I}) \begin{pmatrix} \mathbf{n} \\ \mathbf{e} \end{pmatrix} \\ &= \mathbf{A}\mathbf{x} \end{aligned} \tag{4.6}$$

where $\mathbf{I} \in \mathbb{R}^{n \times n}$ is an $n \times n$ identity matrix, $\mathbf{A} = (\mathbf{L}, \mathbf{I}) \in \mathbb{R}^{n \times (n+3)}$ is the new merged dictionary matrix, and $\mathbf{x} = (\mathbf{n}^{\mathbf{T}}, \mathbf{e}^{\mathbf{T}})^{\mathbf{T}} \in \mathbb{R}^{(n+3) \times 1}$ is the combined vector of all the unknown variables. Thus Eq. 4.5 can be rewritten as:

$$\min_{\mathbf{n},\mathbf{e}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2 \quad s.t. \quad \|\mathbf{x}\|_0 \leq s \tag{4.7}$$

The stacked formulation was inspired by the work of Wright et al. [68], eq.(20). However, in [68], both the signal and the noise are assumed sparse, whereas in our case, the signal (normal vector) has only 3 components and is not at all sparse.

By formulating our problem in the form of Eq. 4.7, we can now take advantage of existing algorithms to efficiently achieve an accurate solution. One such solver is a greedy algorithm called Orthogonal Matching Pursuit (OMP) [52, 61, 11], which is known for its high accuracy and low time-complexity. We will describe this algorithm in Section 4.2 in detail.

Previously, Ikehata et al. [33] proposed a different formulation to Eq. 4.7. They expressed the PST problem in a so-called Lagrangian form, i.e.:

$$\min_{\mathbf{n},\mathbf{e}} \|\mathbf{y} - \mathbf{L}\mathbf{n} - \mathbf{e}\|_2^2 + \lambda\|\mathbf{e}\|_1 \tag{4.8}$$

and applied two solvers: Iteratively Reweighted L1 (IRL1) minimization and Sparse Bayesian Learning (SBL). They showed that SBL provides a more accurate estimation but is more computationally expensive. Later, in Section 6, we will show that our OMP solver produces a more accurate result than SBL with comparable efficiency to IRL1.

4.2 Orthogonal Matching Pursuit

Sparse recovery problems like Eq. 4.7 can be solved via many different methods (see Section 2.2 for a brief review). Here we choose the classical greedy Orthogonal Matching Pursuit (OMP) for our surface normal recovery problem. Given the linear model in Eq. 4.6, the basic idea of OMP is to iteratively select columns of the dictionary matrix $\mathbf{A}$ that are most closely correlated with the current residuals, then project the observation $\mathbf{y}$ to the linear subspace spanned by the columns selected as of the current iteration. We denote each column of $\mathbf{A}$ as $\mathbf{A}_j$, and assume each $\mathbf{A}_j$ is normalized, i.e. $\|\mathbf{A}_j\|_2 = 1$ for $j = 1, 2, \ldots, p$. Let i be the current number of iterations, $\mathbf{r}_i$ and c_i the residuals and the subset of selected columns in $\mathbf{A}$ at the ith iteration, respectively. Let $\mathbf{A}(c_i)$ and $\mathbf{x}(c_i)$ represent the columns indexed by c_i in $\mathbf{A}$, and the entries indexed by c_i in the signal $\mathbf{x}$ to be recovered, respectively. The OMP algorithm [11], as applied to our PST problem, can be summarized as follows:

ALGORITHM Orthogonal Matching Pursuit

1. Initialize the iteration counter $i = 1$, residual $\mathbf{r}_0 = \mathbf{y}$ and $c_0 = \varnothing$.

2. Find a column $\mathbf{A}_t$ ($t \in \{1, 2, \ldots, p\} - c_{i-1}$) that is most closely correlated with the current residual vector. Equivalently, solve the following maximization problem:

$$t = \underset{j}{\operatorname{argmax}} \|\mathbf{A}_j^T \mathbf{r}_{i-1}\| \tag{4.9}$$

3. Add t to the selected set of columns, i.e. update $c_i = c_{i-1} \cup t$, and use $\mathbf{A}(c_i)$ as the current selected subset of $\mathbf{A}$.

4. Project the observation $\mathbf{y}$ onto the linear space spanned by $\mathbf{A}(c_i)$. The projection matrix is calculated as:

$$\mathbf{P} = \mathbf{A}(c_i)(\mathbf{A}(c_i)^T \mathbf{A}(c_i))^{(-1)} \mathbf{A}(c_i)^T \tag{4.10}$$

5. Update the residuals with respect to the new projected observations:

$$\mathbf{r}_i = \mathbf{y} - \mathbf{P}\mathbf{y} \tag{4.11}$$

6. Increment i by 1. If $i > s = n/2 + 3$ (=28 for our typical datasets of 50 images), then proceed to Step 7, otherwise go back to Step 2.

7. Solve only for the entries indexed by c_i in signal $\mathbf{x}$, and simply set the rest of the entries to 0, i.e.:

$$\mathbf{x}(c_i) = \mathbf{A}(c_i)^\dagger \mathbf{y} \tag{4.12}$$

and

$$x_j = 0 \qquad \text{for each} \quad j \notin c_i \tag{4.13}$$

8. Take the first three entries in $\mathbf{x}$ as the solutions for the x, y, and z component of the normal vector, respectively:

$$\mathbf{n} = (x_1, x_2, x_3) \tag{4.14}$$

In our formulation Eq. 4.6, we merge the normal and the errors into a large vector, so the components of the two vectors are treated equally by OMP. In each iteration, which column in the dictionary matrix is to be chosen purely depends on its correlation with the current residuals. Thus there is no strict mathematical guarantee that the normal vector components will be selected in the first s iterations. Indeed, this failure could happen if the non-Lambertian error vector accounts for most of the observations. However, since most of the pixels in real-world images contain (at least partially) an underlying matte model, it is expected that the normal vector components are more closely correlated to the observations than the sporadic non-Lambertian errors. In our experiments, all normal vector components are usually selected within the first few iterations (< 10). On the other hand, if one or two components of the surface normal are rather small, then they might not be selected by our algorithm. However, since they are very close to zero anyway, simply treating them as zero would not negatively impact the accuracy of our estimation.

OMP has a low computational cost and can be implemented straightforwardly. We have found that it is significantly faster than LMS as well as many other state-of-the-art robust regression methods that have been applied in the context of PST. Note that for our particular choice of the design matrix $\mathbf{A}$, the correlation between any column in the identity matrix and the residual $\mathbf{r}$ can be simply represented by one element in $\mathbf{r}$. Therefore, the inner product in Eq. 4.9 may be reduced to finding the maximum entry in $\mathbf{r}$. This observation allows for an even more efficient implementation. In this work, however, we still implement OMP according to Eq. 4.9 for generality.

Stopping criterion

It is possible to choose a stopping criterion based on the convergence of the residual $\mathbf{r}$. For instance, we may terminate the iteration as the residuals in $\mathbf{r}$ fall below a given threshold, or alternatively, when recently selected observations do not introduce any significant change to $\mathbf{r}$. However, OMP is not guaranteed to make a correct selection at every iteration, especially at the heavily shadowed pixels (see Section 6.2.2). As a greedy algorithm, the error introduced by the incorrect selection at one step will irreversibly accumulate to the following iterations. In other words, $\mathbf{r}$ is not guaranteed to converge for these pixels. As a result, all the observations at these bad pixels may be eventually selected before any stopping criterion based on $\mathbf{r}$ is met, and therefore these pixels must be specially handled.

For simplicity, here we set the stopping criterion as a fixed number (s) of iterations. We make a conservative assumption that 50% of the observed pixels are polluted by non-Lambertian noise. Thus, for our typical datasets of $n = 50$ images and normal vectors with 3 components, the stopping criterion is $i > s = n/2 + 3 = 28$. This criterion ensures that there is always a moderate number of observations ($n/2 = 25$) available for regression, and that the accumulation of error will be stopped prematurely for bad pixels.

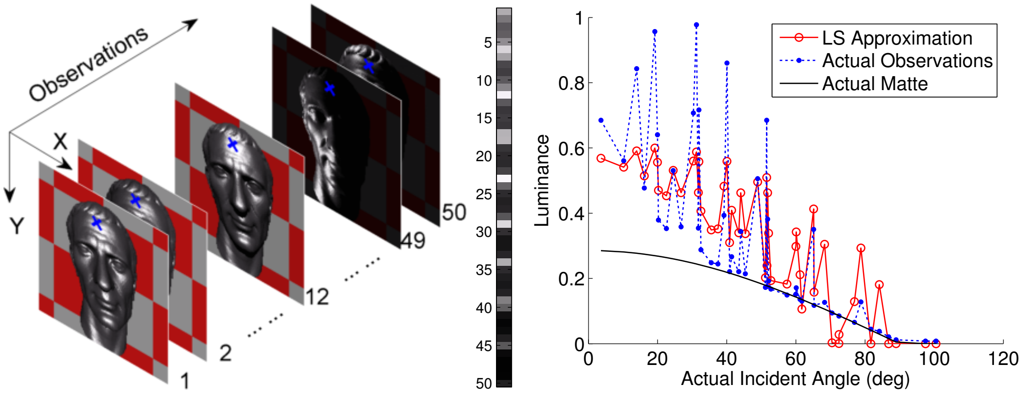
Normalization and orthogonality

OMP requires that every column in **A** be normalized. One may argue that doing so would change the relative ratio of the components in each row, and as a result, the first three columns in the column-normalized **A** would appear to contain directions different from the actual lighting vectors. However, this does not negatively affect our results. The most essential part of OMP, the selection of a column in Step 2, is based on the inner product of each column in **A** and the residual vector. In other words, it is the relative ratio between the components within each column that matters, which is well preserved in the column-wise normalization.

Another issue worth noting is the orthogonality of the design matrix **A**. It has been shown that if **A** satisfies a Restricted Isometry Property(RIP), then the exact recovery of signal **x** may be possible [14, 13]. Essentially, RIP specifies a near-orthonormal condition for **A**. In our experimental setup, the lights are evenly distributed on the dome and the three components of their position vectors are uncorrelated. As such, the first three columns of **A**, corresponding to the light positions, are nearly orthogonal to each other. The rest of **A** is a large identity matrix **I**, which itself is orthonormal. Also, due to the large number of zeros in **I**, the dot product of any of the first three columns and any column in **I** would be a rather small number. Thus, although it is yet to be strictly proven, we speculate that the design matrix **A** in our formulation at least approximately satisfies RIP. This speculation can be indirectly justified by the outstanding accuracy of OMP (see Chapter 6).

4.3 Visual Demonstration

In this section, we demonstrate how OMP enforces robustness onto the normal recovery process by using a simple example. Particularly, we use a synthesized dataset *Caesar* (see Section 6.1.1 for more information) and study all the 50 observations of one pixel (marked by blue crosses in Fig. 4.1a) at location $(X = 90, Y = 39)$ where the ground truth normal vector is $\mathbf{n}_{gt} = (-0.0780, 0.1828, 0.9801)$. The luminance profile of these observations, sorted by the angle of incidence, is shown in Fig. 4.1c (blue dotted line), along with the actual matte model curve (black solid line), i.e. the theoretical values of luminance if the surface is purely Lambertian. It is obvious from Fig. 4.1c that due to the existence of specular reflection, a good percentage of observations (especially when the incident angle is small) deviate from the values predicted by a matte model.



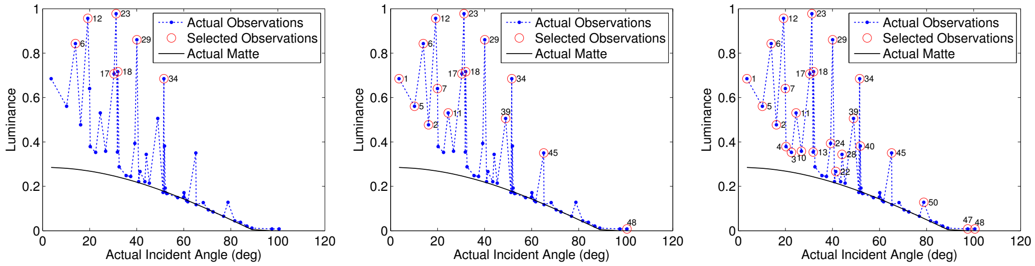
(a) (b) (c)

Figure 4.1: Visualization of all observations of one pixel from dataset *Caesar*. (a) The pixel studied is marked with blue crosses at the same location (X=90,Y=39) on all images numbered from 1 to 50. (b) Luminance observations arranged by the image index 1–50. (c) Luminance observations sorted by incident angle. Blue dotted line shows the actual 50 observations; red circled line shows the approximated luminance using Least Squares; black solid line represents the ground truth matte (Lambertian) luminance.

The naive LS regression (see Section 3.1), when applied to this pixel, attempts to approximate the values of all observations without taking the actual matte model into consideration (Fig. 4.1c, red line marked with circles). Naturally, the LS result $\mathbf{n}_{LS} = (-0.1578, 0.4852, 0.8600)$ deviates greatly from the ground truth $(-0.0780, 0.1828, 0.9801)$.

On the other hand, OMP first attempts to identify s entries, one in each iteration, from the stacked signal $\mathbf{x} = (x_1, x_2, ...x_{n+3})^T \in \mathbb{R}^{n+3}$ (see Eq. 4.6). Usually, these s entries include 3 components for normal vectors (x_1, x_2, x_3) and $(s-3)$ components from the remaining n entries (x_4, x_5, ..., x_{n+3}) that correspond to error values. When the OMP algorithm as described in Section 4.2 is applied to this pixel, it behaves as follows:

Iteration 1–3: The entries that correspond to normal components, x_3, x_2 and x_1, are selected in the order listed. This is not a coincidence since the first three columns of the dictionary matrix $\mathbf{A}$ are overall more strongly correlated to the observations than any of the rest of the columns that correspond to noise values. Also we noticed that these entries are in fact selected in order of the absolute value of their corresponding normal components. For instance, the third component of the ground truth normal (-0.0780, 0.1828, 0.9801) is greater than the other two components. Therefore x_3 gets



(a) (b) (c)

Figure 4.2: Outliers identified by orthogonal matching pursuit. (a) Outliers with large non-Lambertian error (red circles) detected in iteration 4 − 10. (b) Outliers with medium to large error (red circles) detected in iteration 4 − 18. (c) All outliers (red circles) detected as of iteration 28. Blue dotted lines show actual luminance observations in all three plots.

selected in the first iteration.

Iteration 4–10: Entries x_{26}, x_{15}, x_{32}, x_9, x_{37}, x_{21}, x_{20} are selected sequentially. These entries correspond to non-Lambertian errors at observation #23, #12, ... #17, respectively (marked with red circles in Fig. 4.2a). Note that the indices of observations mentioned here (23, 12, ...) are equal to the entry indices found (26, 15, ...) minus 3, since the first three elements in $\mathbf{x}$ do not represent errors. We notice that the corresponding observations of these selected entries all have very high error values. Same as in Iteration 1–3, the order of selection is also decided by the absolute values of the error entries. For instance, observation #23 (incident angle $\approx 32°$) has the largest non-Lambertian error value; therefore its corresponding error entry x_{26} is selected in Iteration 4, before other error entries.

Iteration 11–18: Another eight entries x_4, x_{10}, x_{42}, x_8, x_{14}, x_{51}, x_{48}, x_5 are selected sequentially. Their corresponding observations have medium error values (Fig. 4.2b).

Iteration 19–28: Select the rest of the error entries x_{43}, x_{27}, x_{50}, x_{31}, x_{16}, x_7, x_{13}, x_6, x_{53}, x_{25}. The corresponding observations have small error values (Fig. 4.2c).

Through the 28 iterations above, we have obtained 28 indices; 3 of them correspond to the normal vector components, and the remaining 25 represent the observations that have significant non-Lambertian effects, i.e. non-zero values in signal $\mathbf{x}$ in the sparse regression problem $\mathbf{y} = \mathbf{A}\mathbf{x}$ (Eq. 4.6).

These observations that do not follow the Lambertian model are called *outliers*. In many previous robust methods such as LMS [72], the outliers are simply discarded. Suppose the collection of outliers in 1..50 is represented by $c_{out} \subset \{1, 2, ..., 50\}$, and the inlier collection (observations that are approximately Lambertian) is $c_{in} = \{1, 2, ..., 50\} - c_{out}$. LMS only performs trimmed LS on the observations indexed by c_{in}:

$$\mathbf{y}(c_{in}) = \mathbf{L}(c_{in})\mathbf{n} \tag{4.15}$$

OMP differs from LMS in that OMP uses *all* observations in the LS step. Rather than excluding non-Lambertian outliers from the calculation, it simply marks them as "correctable" and attempts to correct their values with LS while keeping the inlier observations unchanged. Here, the word "correct" refers to the attempt of estimating the Lambertian luminance from the observations contaminated by non-Lambertian effects. Again, denote by $c_{out} \subset \{1, 2, ...50\}$ the indices of non-Lambertian outliers. For our sample pixel, c_{out} consists of the 25 indices of the non-Lambertian observations selected through Iteration 4 – 28 (i.e.: $c_{out} = \{23, 12, ..., 22\}$). We mean to solve the following equation for normal vector $\mathbf{n}$:

$$\mathbf{y} = (\mathbf{L}, \mathbf{I}(c_{out})) \begin{pmatrix} \mathbf{n} \\ \mathbf{e}(c_{out}) \end{pmatrix} \tag{4.16}$$

For a given collection of outliers c_{out}, Eq. 4.15 and Eq. 4.16 are expected to produce similar results when outliers consist of only a moderate portion of all observations. As a matter of fact, using the outliers identified with OMP, these two equations give the same result for the sample pixel: $\mathbf{n}_{OMP} = (-0.0877, 0.2282, 0.9697)$, which well approximates the ground truth $\mathbf{n}_{gt} = (-0.078, 0.1828, 0.9801)$ compared to the naive LS result $\mathbf{n}_{LS} = (-0.1578, 0.4852, 0.8600)$. However, we still favour Eq. 4.16 in this thesis for the following three reasons:

1. When the number of outliers is large, Eq. 4.16 is likely to be more numerically stable than Eq. 4.15. In the OMP algorithm, if we, as discussed in Section 4.2, adopt a stopping criterion based on residual $\mathbf{r}$, then for some heavily shadowed pixels, a large portion of or even all observations may be selected as outliers, leaving only a small number of inliers. As a result, Eq. 4.15 is performed on very few observations, which would possibly lead to erroneous solutions. On the other hand, Eq. 4.16 in that situation would effectively fall back to naive LS, still guaranteeing a reasonable solution.

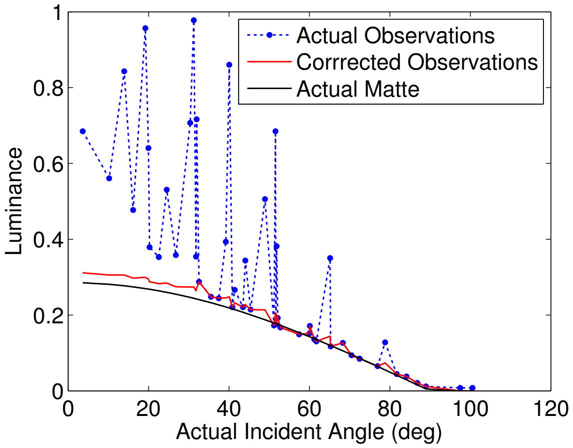


Figure 4.3: Observations corrected by orthogonal matching pursuit. Blue dotted line represents actual luminance observations; black solid line shows the actual matte model; red solid line shows the luminance corrected by OMP.

2. If one or two out of the first three entries in $\mathbf{x}$ (i.e. normal vector components) are not selected (for example, when the actual normal is parallel to the viewing vector, meaning small x and y normal-vector components), then Eq. 4.16 guarantees that those components are indeed zero, whereas Eq. 4.15 usually does not make such a guarantee.

3. Eq. 4.16 explicitly produces an error vector $\mathbf{e}$, and thus the corrected observations $\mathbf{y} - \mathbf{e}$ (Fig. 4.3, red solid line). Note that the luminance observations after OMP correction almost coincide with the ground truth matte model, exhibiting a high degree of robustness.

Chapter 5

Constraint via Joint Sparse Regression

In Chapter 4, we discussed formulating PST as a pixel-wise sparse regression problem. In this formulation, the estimation of the surface normal at one pixel is entirely independent from the observations at other pixels. If the solver fails at some pixel locations, then a noticeable artifact would appear, rendering the recovered surface normal map less smooth. In this chapter, we attempt to alleviate this problem by extending our aforementioned sparse formulation into a *joint sparse* framework, which allows for simultaneous processing of several adjacent pixels at once.

5.1 Formulation

In our regular sparse recovery model:

$$\mathbf{y} = \mathbf{A}\mathbf{x} \tag{5.1}$$

only one sparse signal vector $\mathbf{x} \in \mathbb{R}^{n\times 1}$ is recovered from the observation vector $\mathbf{y} \in \mathbb{R}^{n\times 1}$ and the design matrix $\mathbf{A} \in \mathbb{R}^{n\times p}$ $(n < p)$. However, it might happen that we want to recover multiple signals that have the same support (i.e. locations of non-zero rows) and are modulated by the same design matrix $\mathbf{A}$ simultaneously. This problem can be expressed as follows (see [38]):

$$\mathbf{Y} = \mathbf{A}\mathbf{X} \tag{5.2}$$

In Eq. 5.2, $\mathbf{X} = (\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N) \in \mathbb{R}^{p \times N}$ is now a matrix with N columns, and each column represents one signal to be recovered. Accordingly, $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N) \in \mathbb{R}^{n \times N}$ contains the corresponding observations. Similar to Eq. 1.2, the canonical form of joint sparse recovery aims to minimize the ℓ_0 pseudo norm of $\mathbf{X}$, i.e. the number of non-zero rows in $\mathbf{X}$:

$$\begin{aligned} \min_{\mathbf{X}} \|\mathbf{X}\|_0 \qquad s.t. \quad & \mathbf{Y} = \mathbf{A}\mathbf{X} \\ & \|\mathbf{X}\|_0 = \|\mathbf{x}_1\|_0 = \cdots = \|\mathbf{x}_N\|_0 \end{aligned} \tag{5.3}$$

The rationale for assuming equal ℓ_0 norms is that we wish to enforce each signal (in our case, pixel) having the same support.

The joint sparse recovery formulation Eq. 5.3 allows us to take advantage of the correlation between signals and therefore obtain a more robust result with fewer measurements. We found it very suitable for the purpose of surface normal reconstruction. Note that a key feature of joint sparse recovery is that all column vectors in $\mathbf{X}$ to be recovered have the same support. In PST, the physical surface of interest is usually smooth (at least locally), thus adjacent pixel locations are expected to have similar normal directions. Furthermore, these pixels tend to be contaminated by the same non-Lambertian effect under a given illumination condition. For instance, two neighbouring pixels are very likely to be both located inside or outside a cast shadow. Hence, when we estimate the normal of a given pixel k_0, it would provide us with more information if we also consider its $N-1$ neighbours $k_1, k_2, \cdots, k_{N-1}$ and process the ensemble of N normal vectors all at once. To the best of our knowledge, no one has used the joint sparse recovery formulation, as set out here, in PST.

We can express this idea in a form similar to Eq. 4.2:

$$\tilde{\mathbf{Y}}^{k_0} = \mathbf{L}\tilde{\mathbf{N}}^{k_0} + \tilde{\mathbf{E}}^{k_0} \tag{5.4}$$

where $\tilde{\mathbf{N}}^{k_0} = (\mathbf{n}^{k_0}, \mathbf{n}^{k_1}, \cdots, \mathbf{n}^{k_{N-1}}) \in \mathbb{R}^{3 \times N}$ is the ensemble of normal vectors of pixel k_0 and its $N-1$ neighbours, $\tilde{\mathbf{Y}}^{k_0} = (\mathbf{y}^{k_0}, \mathbf{y}^{k_1} \cdots, \mathbf{y}^{k_{N-1}}) \in \mathbb{R}^{n \times N}$ contains n observations at each of the N pixels, and $\tilde{\mathbf{E}}^{k_0} = (\mathbf{e}^{k_0}, \mathbf{e}^{k_1} \cdots, \mathbf{e}^{k_{N-1}}) \in \mathbb{R}^{n \times N}$ is the sparse error matrix for the N pixels.

We can further rewrite Eq. 5.4 in the standard form of sparse recovery as in Eq. 5.2:

$$\begin{aligned}\tilde{\mathbf{Y}}^{k_0} &= (\mathbf{L}, \mathbf{I}) \begin{pmatrix} \tilde{\mathbf{N}}^{k_0} \\ \tilde{\mathbf{E}}^{k_0} \end{pmatrix} \\ &= \mathbf{A}\tilde{\mathbf{X}}^{k_0} \end{aligned} \tag{5.5}$$

where $\tilde{\mathbf{X}}^{k_0} = (\mathbf{x}^{k_0}, \mathbf{x}^{k_1}, \cdots, \mathbf{x}^{k_{N-1}}) \in \mathbb{R}^{(n+3)\times N}$. Its every column $\mathbf{x}^{k_i}$ is a combination of $\mathbf{n}^{k_i}$ and $\mathbf{e}^{k_i}$, i.e.:

$$\mathbf{x}^{k_i} = \begin{pmatrix} \mathbf{n}^{k_i} \\ \mathbf{e}^{k_i} \end{pmatrix} \tag{5.6}$$

Now that the surface-light interaction is expressed with the joint linear model that has the same form as Eq. 5.2, we could approximate the sparse vector $\tilde{\mathbf{X}}^{k_0}$ by finding the solution to Eq. 5.3. In practice, we choose to circumvent this NP-hard combinatorial problem by solving a relaxed form of Eq. 5.3:

$$\begin{aligned} &\min_{\tilde{\mathbf{X}}^{k_0}} \sum_{i=0}^{N-1} \|\mathbf{y}^{k_i} - \mathbf{A}\mathbf{x}^{k_i}\|_2 \\ s.t. \quad &\|\tilde{\mathbf{X}}^{k_0}\|_0 = \|\mathbf{x}^{k_0}\|_0 = \|\mathbf{x}^{k_1}\|_0 = \cdots = \|\mathbf{x}^{k_{N-1}}\|_0 \leq s \end{aligned} \tag{5.7}$$

In order to enforce the smoothness constraint on the surface normal recovery process, we employ a joint sparse regression algorithm to solve Eq. 5.7 for each pixel and its $N-1$ neighbours simultaneously. This technique would eventually result in N normal estimates for each pixel (except for the pixels on the boundaries of the image, which may have fewer neighbours). Then the "centre" of all these estimated vectors is taken as the final result. Here we define each component n_x, n_y, n_z of the "centre" vector $\mathbf{n}$ as the median of the corresponding components across all the N *normalized* estimated vectors $\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_N$, i.e.:

$$\begin{aligned} n_x &= \text{median}(v_{1x}, v_{2x}, \cdots, v_{Nx}) \\ n_y &= \text{median}(v_{1y}, v_{2y}, \cdots, v_{Ny}) \\ n_z &= \text{median}(v_{1z}, v_{2z}, \cdots, v_{Nz}) \end{aligned} \tag{5.8}$$

where $(v_{ix}, v_{iy}, v_{iz}) = \mathbf{v}_i$. Note that this is an approximate method that we choose due to its simplicity. A more rigorous estimation may involve interpolation on the spherical coordinate system.

5.2 Simultaneous Orthogonal Matching Pursuit

It has been previously shown in Signal Processing that joint sparse regression problems can be solved via a generalized form of OMP – Simultaneous Orthogonal Matching Pursuit [62, 63]. Note that "Simultaneous" means the same, here, as what we have termed "Joint". Similar to OMP, S-OMP selects the basis that is best correlated with the current residuals at each iteration step. For the following linear model:

$$\mathbf{Y} = \mathbf{A}\mathbf{X} \tag{5.9}$$

where $\mathbf{Y} \in \mathbb{R}^{n \times N}$ is the observation matrix, $\mathbf{A} \in \mathbb{R}^{n \times p}$ is the dictionary matrix ($n < p$), and $\mathbf{X} \in \mathbb{R}^{p \times N}$ is a matrix of the N signals to be recovered. Let $\mathbf{R}_i \in \mathbb{R}^{n \times N}$ and $c_i \in \mathbb{R}^i$ be the residual matrix and the subset of selected column indices in the dictionary $\mathbf{A}$ as of the ith iteration, respectively. We also denote the submatrix of $\mathbf{A}$ that comprises the columns specified by an index set c as $\mathbf{A}(c)$. Naturally, the jth column of $\mathbf{A}$ is represented by $\mathbf{A}(j)$. In addition, let us denote the residual vector for the dth signal ($d \in \{1, ..., N\}$), i.e. the dth column in the residual matrix, by $\mathbf{R}_i(d) \in \mathbb{R}^n$. The S-OMP algorithm as proposed in [62] is summarized as follows:

1. Initialize the iteration counter $i = 1$, residual $\mathbf{R}_0 = \mathbf{Y}$ and index set $c_0 = \varnothing$.

2. Find a column $\mathbf{A}(t_i)$ ($t_i \in \{1, 2, \ldots, p\} - c_{i-1}$) that is best correlated with the current residual matrix, or equivalently, solve the following maximization problem:

$$t_i = \underset{j}{\operatorname{argmax}} \sum_{d=1}^{N} |\langle \mathbf{R}_{i-1}(d), \mathbf{A}(j) \rangle| \tag{5.10}$$

   where $\langle \cdot, \cdot \rangle$ represents the inner product.

3. Add t_i to the selected set of columns, i.e. update $c_i = c_{i-1} \cup t_i$.

4. Project the observation $\mathbf{Y}$ onto the linear space spanned by $\mathbf{A}(c_i)$. The projection matrix is calculated as:

$$\mathbf{P}_i = \mathbf{A}(c_i)(\mathbf{A}(c_i)^T \mathbf{A}(c_i))^{(-1)} \mathbf{A}(c_i)^T \tag{5.11}$$

5. Update the residuals with respect to the new projected observations:

$$\mathbf{R}_i = \mathbf{Y} - \mathbf{P}_i \mathbf{Y} \tag{5.12}$$

6. Update $i = i + 1$. Go back to Step 2 until the stopping criterion is met.

Again, assuming that at most 50% of the observations are heavily non-Lambertian, we choose the stopping criterion to be:

$$i > n/2 + 3 \tag{5.13}$$

5.3 Sequential CS-MUSIC

MUltiple SIgnal Classification (MUSIC) is a signal processing algorithm, initially developed by Schmidt [55] for estimating the direction of arrival (DOA) of signals received by sensor arrays. It has been found that MUSIC-like criteria can be applied to any joint sparse regression algorithms to refine the recovered support (i.e. collection of non-zero rows) [39, 38]. In this study, we employ the Sequential Compressive MUSIC algorithm (SeqCS-MUSIC) proposed in [39] to improve the accuracy of S-OMP. SeqCS-MUSIC consists of two major steps: **Sequential Subspace Estimation** and **Support Filtering**.

5.3.1 Notation

Consider the joint linear model in Eq. 5.9. Let $c_k \subset \{1, 2, ...p\} \in \mathbb{R}^k$ be a set of k indices of rows in the signal matrix $\mathbf{X}$, and let $\mathbf{A}(c_k)$ denote a submatrix of $\mathbf{A}$ that consists of the k columns indexed by c_k. For any sparse signal in $\mathbf{X}$, we call the set of non-zero rows its **support**, denoted by $supp\mathbf{X}$. Naturally we have $|supp\mathbf{X}| = k$ for a k-sparse signal. In addition, we denote the range space of $\mathbf{A}$ as $R(\mathbf{A})$, and the projection on the orthogonal complement of the subspace spanned by columns in $\mathbf{A}$ as $P^{\perp}_{R(\mathbf{A})}$.

5.3.2 Support filtering: backward greedy step

The support recovered by a joint sparse recovery algorithm such as S-OMP may only be partially correct. Let k denote the user specified sparsity of $\mathbf{X}$, and r the rank of the signal subspace of the observation matrix $\mathbf{Y}$. The purpose of the Backward Greedy step is to revisit each of the k indices in the recovered support and select $k - r$ of them that are most likely to be correct. The algorithm of this step is as follows:

Algorithm: $c_{k-r} = \text{SupportFiltering}(\mathbf{A}, \mathbf{Y}, c_k)$

1. Calculate $\eta(j) = \|P_{R([\mathbf{A}(c_k \backslash \{j\})\ \mathbf{Y}])}\mathbf{A}(j)\|^2$ for all $j \in c_k$.
2. Choose those j's that give rise to the smallest $k - r$ values of $\eta(j)$, and put them into c_{k-r}.
3. Return c_{k-r}.

5.3.3 Sequential subspace estimation: forward greedy step

The MUSIC criterion [39] states that once $k - r$ non-zero entries are correctly found, the remaining r entries can be determined unambiguously. In SeqCS-MUSIC, this is done through the following Forward Greedy step:

Algorithm: $c_k = \text{SequentialSubsapce}(\mathbf{A}, \mathbf{Y}, c_{k-r})$

1. Initialize the iteration counter $d = 1$, and let $c = c_{k-r}$.
2. Apply singular value decomposition on $[\mathbf{A}(c), \mathbf{Y}] = [U_1, U_0]\text{diag}[S_1, S_0][V_1, V_0]$ where $S_1 = \text{diag}[s_1, ..., s_k]$ and $S_0 = \text{diag}[s_{k+1}, ..., s_{k+d}]$.
3. Solve the following optimization problem:

$$j_d = \underset{j \notin c}{\text{argmin}}\, \|P^{\perp}_{R(U_1)}\mathbf{A}(j)\|^2 \tag{5.14}$$

4. Add j_d to c. Let $d = d + 1$.
5. Return to Step 2 if $d \leq r$, otherwise let $c_k = c$ and terminate.

5.3.4 Full Sequential CS-MUSIC algorithm

Combining the forward and backward greedy steps together, we now have the full SeqCS-MUSIC algorithm:

ALGORITHM: $\mathbf{X} = \text{SeqCS-MUSIC}(\mathbf{A}, \mathbf{Y})$

1. Estimate the initial k-support c_k using S-OMP.
2. Obtain $\mathbf{U}$ which is the rank-r signal subspace estimate of $R(\mathbf{Y})$.

3. Execute $c_{k-r} = \text{SupportFiltering}(\mathbf{A}, \mathbf{U}, c_k)$ to refine the first $k - r$ indices.

4. Execute $c_k = \text{SequentialSubspace}(\mathbf{A}, \mathbf{U}, c_{k-r})$ to recalculate the last r indices.

5. Perform LS only on the columns indexed by c_k in $\mathbf{A}$. Return the regressed result $\mathbf{X}$.

5.4 Joint Sparse Solution for Photometric Stereo

Based on the SeqCS-MUSIC algorithm described in Section 5.3.4, our joint sparse PST algorithm can be stated as follows:

ALGORITHM: Joint Sparse PST

1. For each pixel i in the image, create an empty queue $\mathbf{Q}^i$ that stores the normal vector estimates for pixel i. Thus, each element in $\mathbf{Q}^i$ is a 3-vector.

2. Combine the n observations for pixel i and its 8-neighbours $j_1(i), j_2(i), \cdots, j_8(i)$ to form an $n \times 9$ joint observation matrix $\mathbf{Y}^i$. For consistency in notation, we define $j_0(i) \equiv i$:

$$\mathbf{Y}^i = \begin{bmatrix} y_1^{j_0(i)} & y_1^{j_1(i)} & y_1^{j_2(i)} & \cdots & y_1^{j_8(i)} \\ y_2^{j_0(i)} & y_2^{j_1(i)} & y_2^{j_2(i)} & \cdots & y_2^{j_8(i)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ y_n^{j_0(i)} & y_n^{j_1(i)} & y_n^{j_2(i)} & \cdots & y_n^{j_8(i)} \end{bmatrix} \tag{5.15}$$

3. Solve the joint sparse recovery problem using SeqCS-MUSIC:

$$\mathbf{X}^i = \text{SeqCS-MUSIC}(\mathbf{A}, \mathbf{Y}^i) \tag{5.16}$$

4. Take the first three elements of each column $q \in \{0, 1, ..., 8\}$ in $\mathbf{X}^i$ as an estimate $\mathbf{v}^{j_q(i)}$ of the normal vector at pixel $j_q(i)$. Push each $\mathbf{v}^{j_q(i)}$ to the end of its corresponding queue $\mathbf{Q}^{j_q(i)}$.

5. Repeat Step 2–4 for all pixels.

6. For each pixel i, take all the vectors stored in $\mathbf{Q}^i$ and obtain the final estimate via Eq. 5.8.

Chapter 6

Results and Discussion

In this chapter, we present our experimental results and observations on synthesized and real datasets. All experiments were carried out on a Dell Optiplex 755 computer equipped with an Intel Core Duo E6550 CPU and 4GB RAM, running Windows 7 Enterprise (64-bit). All algorithms were implemented in MATLAB R2012a (64-bit).

6.1 Normal Map Recovery

We first examine the angular error of normal maps recovered by different methods on both synthesized and real datasets. For synthesized datasets, we quantitatively inspect the difference between the ground truth normal map and the recovered normal maps. For real datasets without a ground truth map, on the other hand, the recovered normal maps are examined visually and qualitatively.

6.1.1 Synthesized datasets

Four 3D objects are used for our synthesized datasets in this study: *Sphere*, *Caesar*, *Buddha* and *Venus*. All 3D models are either created programmtically as geometrical primitives (*Sphere*), or downloaded from the AIM@SHAPE Shape Repository (*Caesar*, *Buddha*) [2] and the INRIA Gamma research database (*Venus*) [1]. For each object, 50 images are rendered under various illumination conditions using raytracing software (POV-Ray 3.6) at a resolution of 200 × 200 (except for *Venus*, whose resolution is 150 × 250). All scenes feature significant specularity and large areas of cast shadow. *Caesar*, *Buddha* and *Venus* are rendered with the Specular Highlight shading model provided by POV-Ray (a modified version of the Phong model) [3], and *Sphere* is rendered with a pure Phong model. A checkered plane is intentionally included in the rendered scenes as background to (1) make the cast shadow visible and (2) add further challenges to the algorithms since it introduces local fluctuation in luminance while the surface normals remain constant. Sample images from these datasets are shown in Fig. 6.1.

For each image set, the normal map is estimated using regular OMP and SeqCS-MUSIC methods as proposed in this study. For comparison, we show the results given by two other state-of-the-art sparse recovery methods – Iteratively Reweighted L1 optimization (IRL1) and Sparse Bayesian Learning (SBL). Two other outlier detection based methods, LMS and LMS Mode finder, are also applied and compared. Then, the angular error between the normal map recovered using each method and the ground truth is quantitatively measured. Note that only results for *Caesar*, *Sphere* and *Buddha* are shown in this section. The fourth dataset *Venus* is reserved for later in Section 6.2.2 as a failure case.

We found these methods exhibit similar relative performance to each other on all three datasets tested in this section – *Caesar*, *Sphere*, and *Buddha.* In *Caesar*, the normal maps recovered using OMP (Fig. 6.2e) and SeqCS-MUSIC (Fig. 6.2i) have a higher quality than those by IRL1 (Fig. 6.2g) and SBL (Fig. 6.2k) both qualitatively and quantitatively. We observe that IRL1 and SBL, although much more robust than LS, still produce a considerable error at highly specular regions, most notably the cheek and the forehead. As a result, the faces on IRL1 and SBL normal maps appear to be more protruding than the ground truth. Also, some fine details on these two normal maps, such as the wrinkles on the forehead, are not well preserved. In addition, IRL1 and SBL fail to handle the regions right beside the neck which are heavily shadowed.

On the other hand, OMP and SeqCS-MUSIC show a higher degree of robustness than previous sparse methods at specularity-affected regions (cheek, forehead and nose) as well as shadowed regions (areas around the neck on the checkered background), resulting in a normal map closer to the original. For example, the forehead appears flat on OMP and SeqCS-MUSIC normal maps, closely resembling the ground truth. The wrinkles are almost perfectly recovered. However, OMP appears to be confused by the checkered pattern of the background, producing a small angular error in these flat regions. This error is eliminated by SeqCS-MUSIC. Overall, the SeqCS-MUSIC result is visually smoother and statistically more robust than OMP.

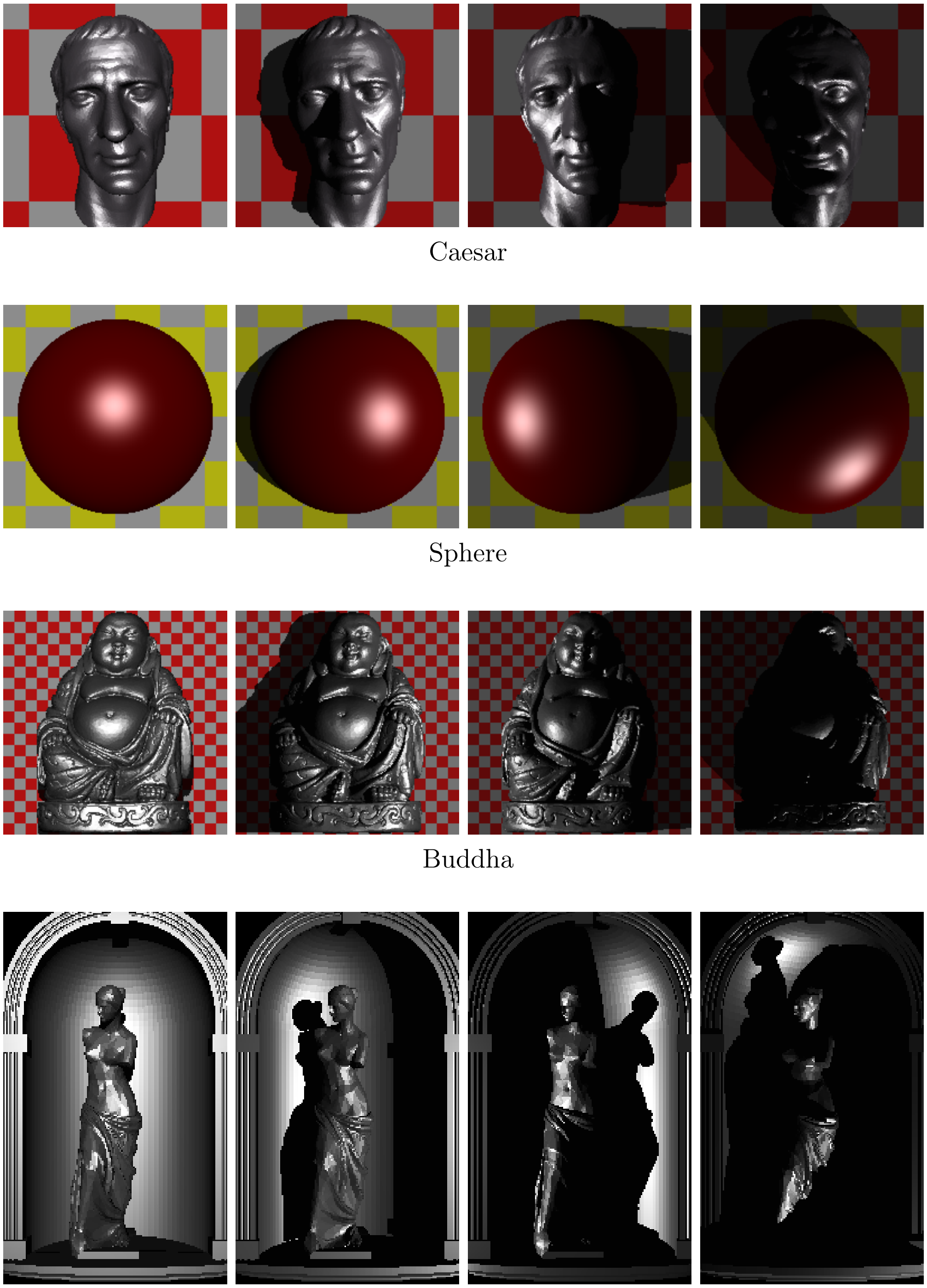
The LMS result (Fig. 6.2m) is better than IRL1 and SBL, but worse than OMP and SeqCS-MUSIC. The 1D version of LMS – LMS Mode finder – produces a poorer visual result (Fig. 6.2o) compared to the other robust methods, although it does give a statistically more reliable result than LS. We will exclude this method from future discussion but still show its result for reference.

The effect of specularity on normal map recovery can be further seen from the results for the *Sphere* dataset in Fig. 6.3. Again, IRL1 and SBL results are noisy in the specularity-affected areas, whereas OMP and SeqCS-MUSIC give much cleaner results. LMS performs similarly to OMP and SeqCS-MUSIC. Interestingly, a pentagon-shaped pattern is visible on each error map because there are exactly five lights at each elevation angle.

For *Buddha*, OMP and SeqCS-MUSIC again produce more accurate results than IRL1 and SBL. The angular error distribution of the LMS result is similar to that of OMP and SeqCS-MUSIC, though with a slightly larger overall error.

From the normal map recovery results obtained on the three datasets, we can see that OMP and SeqCS-MUSIC generally perform better than IRL1 and SBL on convex objects and are more resistant to specularities and cast shadows. The statistical result for the angular error of the normal maps recovered with different methods are listed in Table 6.1. The SeqCS-MUSIC results have the lowest mean, median, 25% and 75% quantiles, as well as standard deviation for all three datasets. The performance of OMP is comparable to SeqCS-MUSIC. The LMS result is better than IRL1 and SBL, but worse than OMP and SeqCS-MUSIC. These results are also depicted in Fig. 6.5. Curiously, we notice that the estimation accuracy is generally lower on *Sphere* than *Caesar*, despite the simple geometry of the former. This may be jointly caused by the unique lighting model, surface colour and material that *Sphere* is rendered with. The exact explanation for this observation requires further investigation in the future.

As is witnessed on the *Buddha* dataset, both OMP and SeqCS-MUSIC perform less optimally than IRL1 and SBL on small concave regions that are rarely illuminated, such as the creases on the clothes. This problem also occurs for *Caesar* on the medial side of the eyes and under the eyebrows. It is a lesser concern for objects that are generally convex such as *Caesar* and *Buddha*, but may exert a strong negative influence on a scene that contains large concave areas. We will demonstrate the result for such a scene using *Venus* in Section 6.2.2.



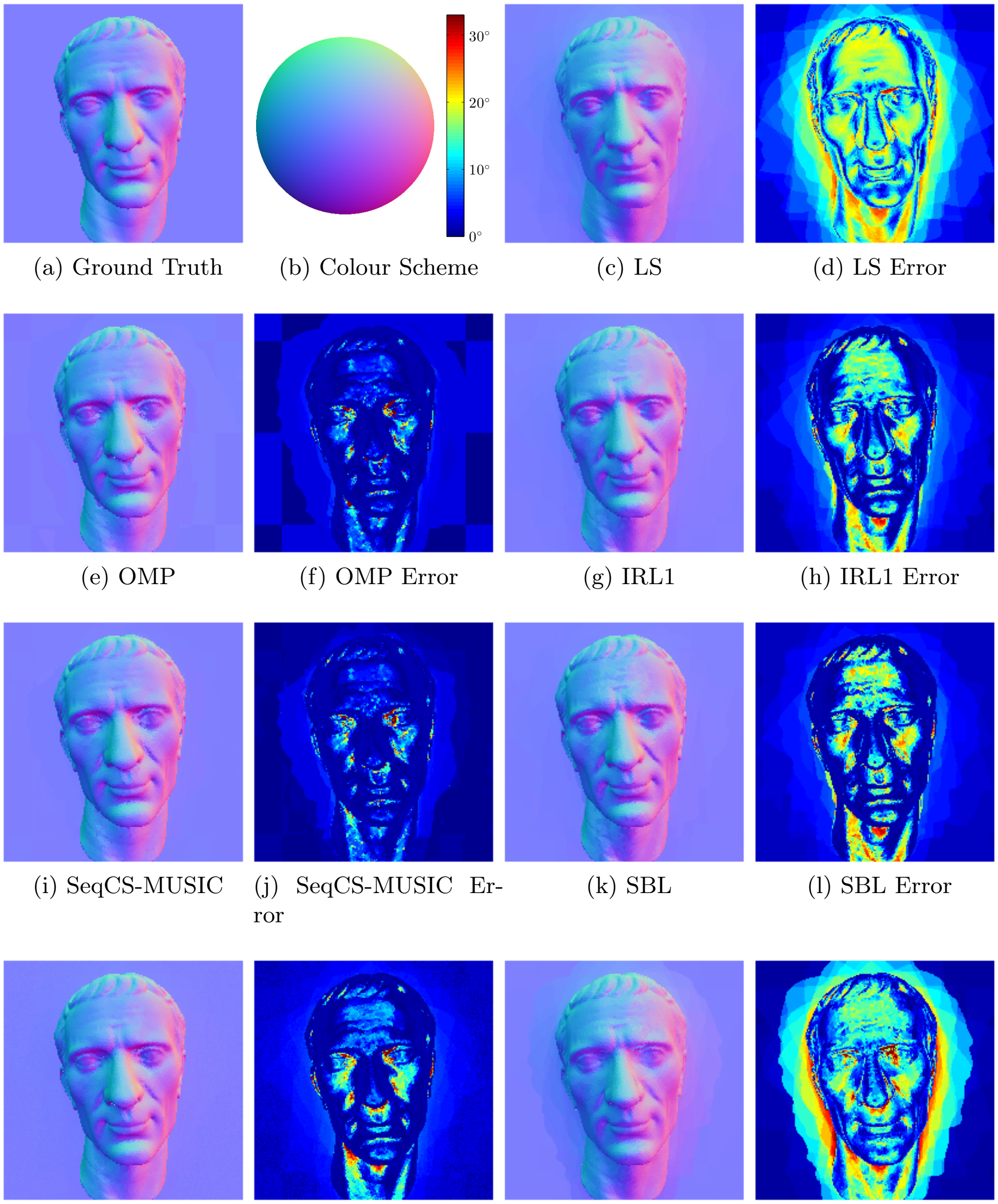
Caesar

Sphere

Buddha

Venus

Figure 6.1: Sample images from four synthesized datasets rendered with POV-Ray. From top row to bottom row: *Caesar*, *Sphere*, *Buddha* and *Venus*.



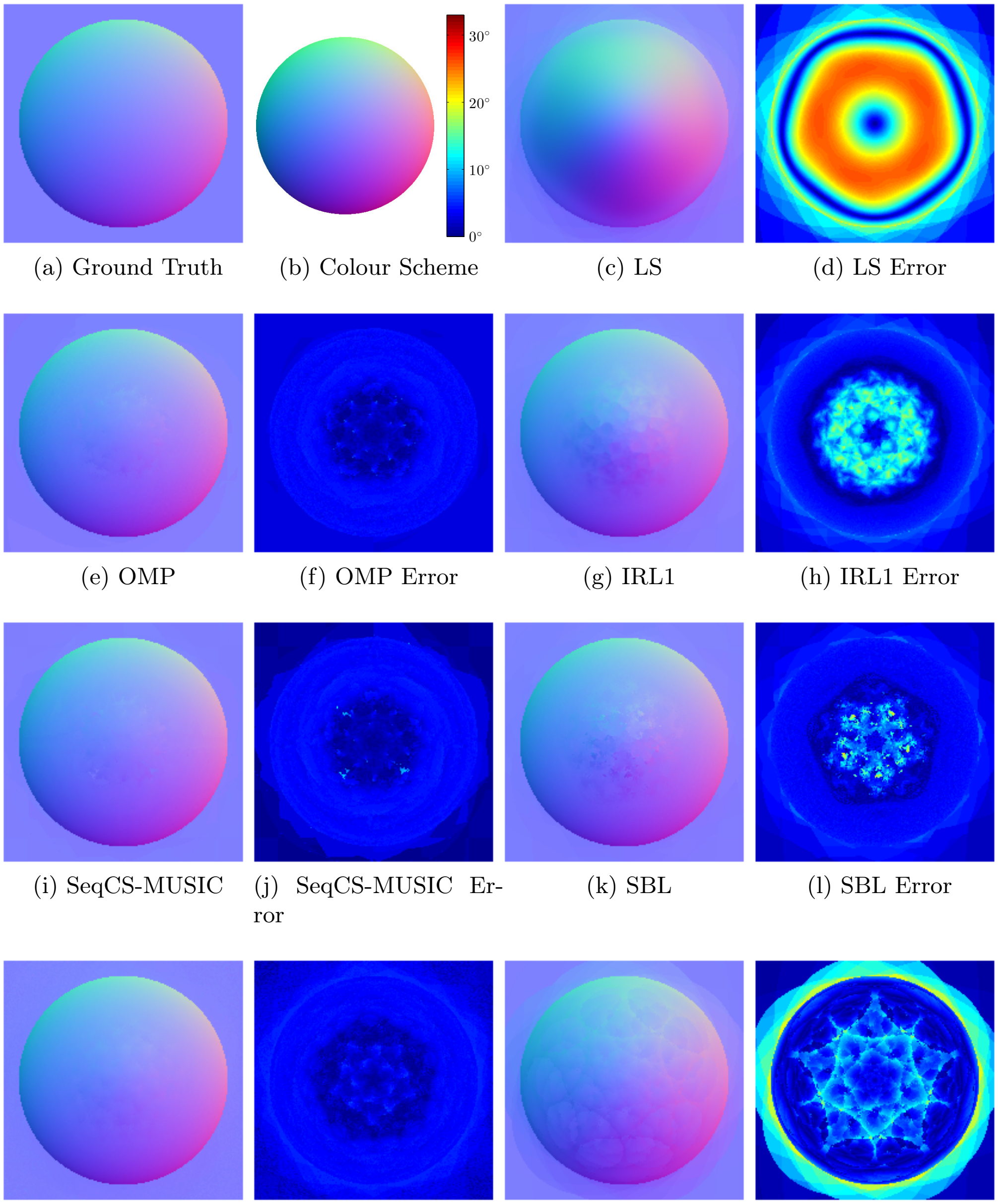
(a) Ground Truth (b) Colour Scheme (c) LS (d) LS Error

(e) OMP (f) OMP Error (g) IRL1 (h) IRL1 Error

(i) SeqCS-MUSIC (j) SeqCS-MUSIC Error (k) SBL (l) SBL Error

(m) LMS (n) LMS Error (o) LMS Mode (p) LMS Mode Error

Figure 6.2: Normal maps of a head statue of Caesar recovered using seven methods. (a) Ground truth normal map. (b) Colour wheel and colour bar used for normal and angular error visualization, respectively. Angular error is measured in degrees. (c,e,g,i,k,m,o) Normal maps recovered using LS, OMP, IRL1, SeqCS-MUSIC, SBL, LMS and LMS-Mode, respectively. (d,f,h,j,l,n,p) Angular error of normal maps recovered using the aforementioned methods.



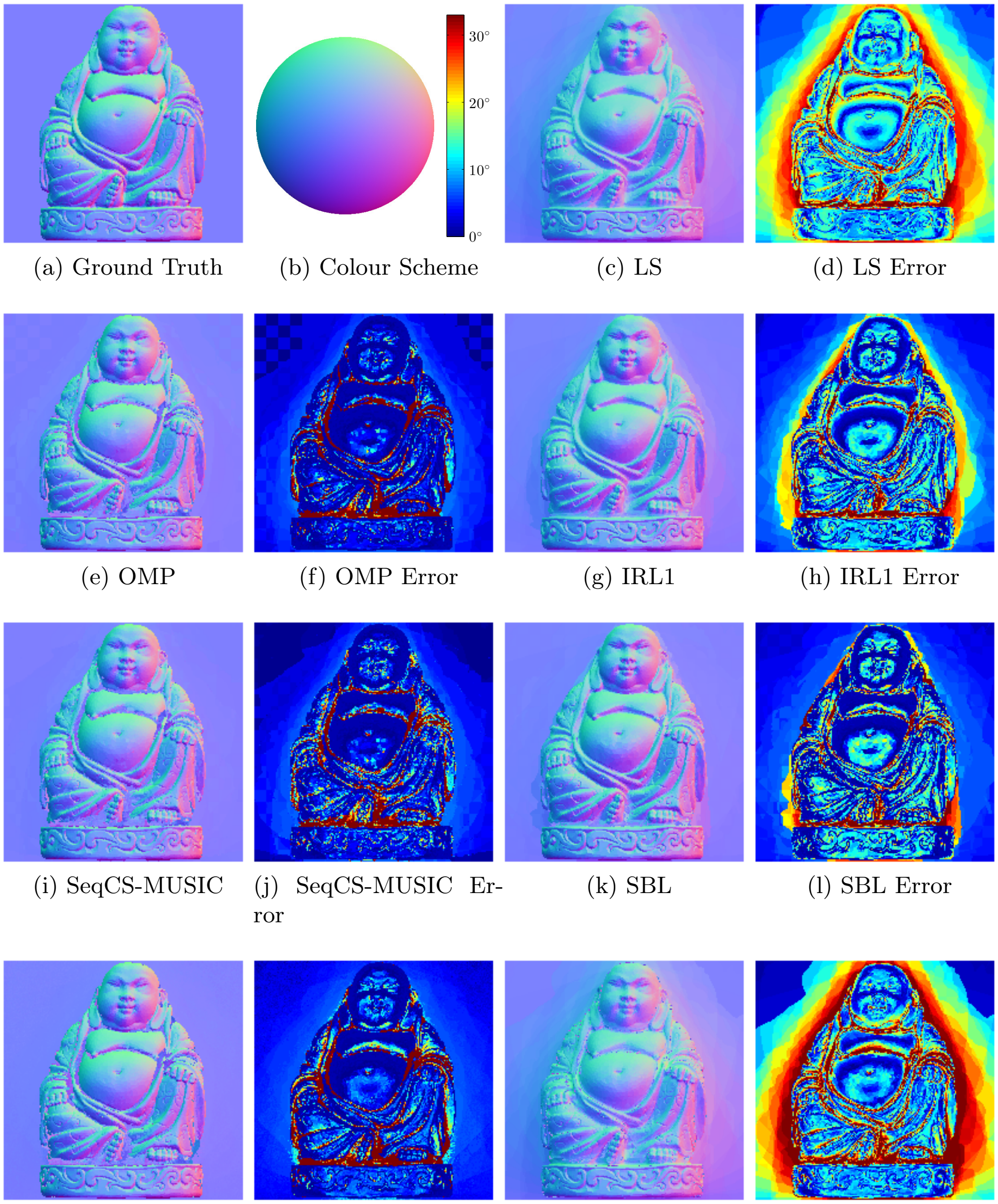
(a) Ground Truth (b) Colour Scheme (c) LS (d) LS Error

(e) OMP (f) OMP Error (g) IRL1 (h) IRL1 Error

(i) SeqCS-MUSIC (j) SeqCS-MUSIC Error (k) SBL (l) SBL Error

(m) LMS (n) LMS Error (o) LMS Mode (p) LMS Mode Error

Figure 6.3: Normal maps of an ideal sphere recovered using seven methods. (a) Ground truth normal map. (b) Colour wheel and colour bar used for normal and angular error visualization, respectively. Angular error is measured in degrees. (c,e,g,i,k,m,o) Normal maps recovered using LS, OMP, IRL1, SeqCS-MUSIC, SBL, LMS and LMS-Mode, respectively. (d,f,h,j,l,n,p) Angular error of normal maps recovered using the aforementioned methods.



(a) Ground Truth (b) Colour Scheme (c) LS (d) LS Error

(e) OMP (f) OMP Error (g) IRL1 (h) IRL1 Error

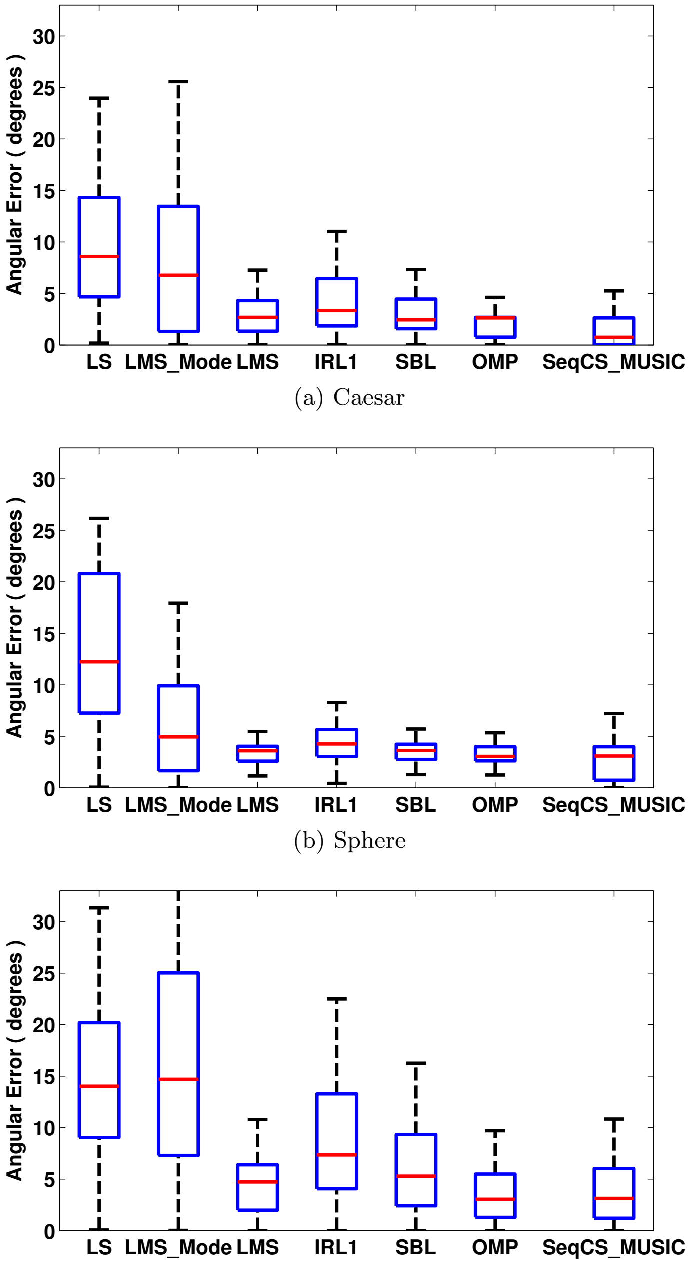
(i) SeqCS-MUSIC (j) SeqCS-MUSIC Error (k) SBL (l) SBL Error

(m) LMS (n) LMS Error (o) LMS Mode (p) LMS Mode Error

Figure 6.4: Normal maps of a Buddha statue recovered using seven methods. (a) Ground truth normal map. (b) Colour wheel and colour bar used for normal and angular error visualization, respectively. Angular error is measured in degrees. (c,e,g,i,k,m,o) Normal maps recovered using LS, OMP, IRL1, SeqCS-MUSIC, SBL, LMS and LMS-Mode, respectively. (d,f,h,j,l,n,p) Angular error of normal maps recovered using the aforementioned methods.

	25% Quantile	Median	Mean	75% Quantile	STD
Dataset Caesar					
LS	4.669	8.586	9.672	14.32	5.875
LMS Mode	1.313	6.769	8.148	13.46	7.620
LMS	1.335	2.692	3.700	4.305	4.242
IRL1	1.854	3.339	5.276	6.440	5.080
SBL	1.576	2.441	4.487	4.488	5.158
OMP	0.7571	2.624	2.424	2.690	3.332
SeqCS-MUSIC	0.000	0.7532	1.817	2.624	3.297
Dataset Sphere					
LS	7.255	12.23	13.55	20.79	7.566
LMS Mode	1.658	4.939	6.211	9.907	4.901
LMS	2.588	3.601	3.261	4.030	1.103
IRL1	3.045	4.253	4.960	5.662	3.119
SBL	2.762	3.620	3.671	4.239	1.822
OMP	2.617	3.051	3.179	3.991	0.9324
SeqCS-MUSIC	0.7381	3.089	2.584	3.987	1.603
Dataset Buddha					
LS	9.039	14.03	15.29	20.19	8.25
LMS Mode	7.304	14.70	17.42	25.03	13.79
LMS	1.985	4.736	8.454	6.391	15.35
IRL1	4.066	7.354	10.08	13.28	9.178
SBL	2.412	5.295	8.034	9.333	9.479
OMP	1.289	3.049	7.065	5.495	14.17
SeqCS-MUSIC	1.206	3.135	7.047	6.023	13.19

Table 6.1: Statistics for the angular error between the normal maps recovered for seven methods and the ground truth, for three synthesized datasets. All numbers are shown in degrees.



(a) Caesar

(b) Sphere

(c) Buddha

Figure 6.5: Angular error of normal maps recovered using seven methods. Red horizontal lines indicate medians. Upper and lower border of blue boxes represent third (Q3) and first quartile (Q1), respectively. Upper and lower whiskers show 1.0 Interquartile Range (the difference between the upper and lower quartiles, the IQR) extended from Q3 and Q1, respectively.

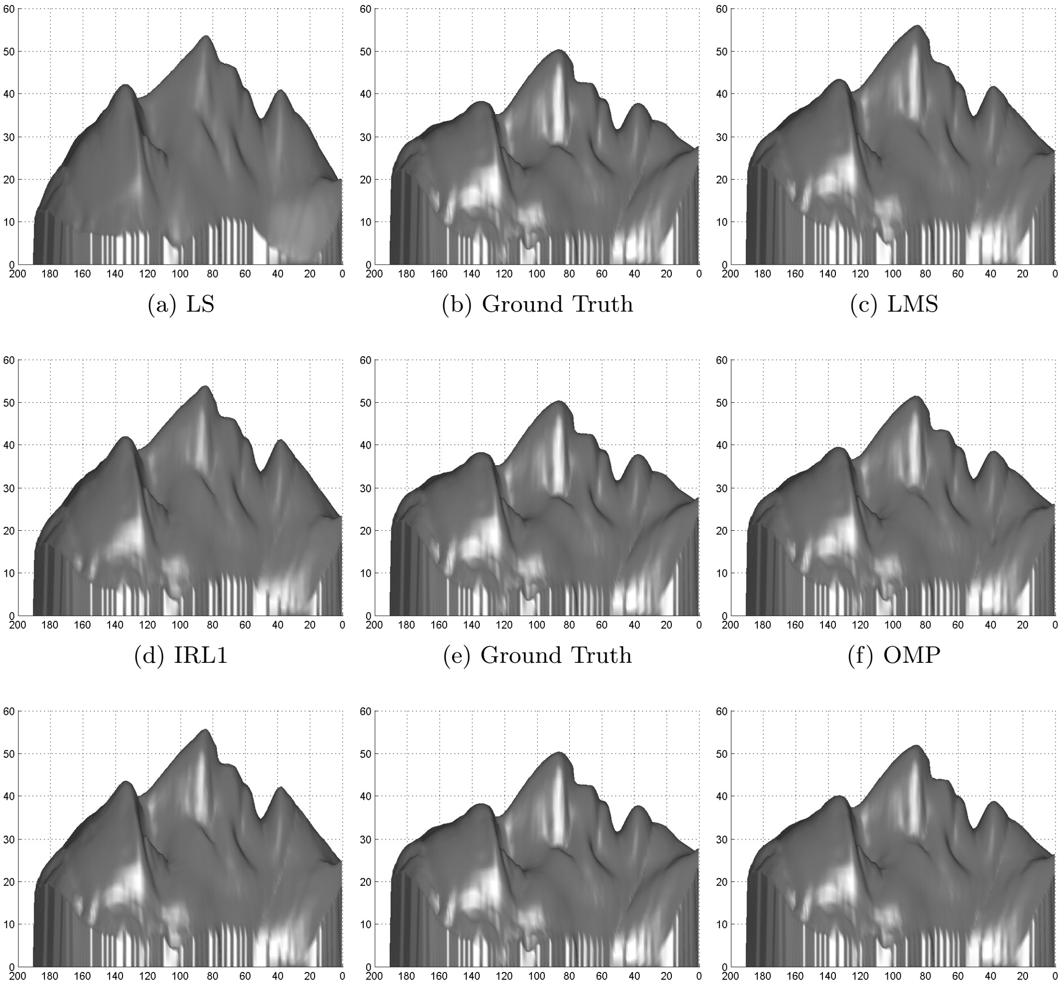
6.1.2 Comparison via reconstructed surfaces

Using the normal maps recovered with various methods, we also reconstruct 3D surfaces with the Frankot-Chellapa method [29] to directly compare the shapes. Here only the reconstruction result for *Caesar* is used for demonstration. It is apparent from Fig. 6.6 that in the LS, IRL1 and SBL results, the overall shape of the face appears to be more protruding then it actually is, especially at the eyebrow ridge and the nose, whereas the OMP and SeqCS-MUSIC methods manage to preserve the shape accurately. Again, the LMS result appears to be less protruding than IRL1 and SBL results, although still not as accurate as OMP and SeqCS-MUSIC. We speculate that the exaggerated convexity originates from the inaccurately estimated normal vectors at highlight areas, such as the forehead and the nose. Since our greedy algorithms generally provide a better recovery in those regions, they naturally yield a more accurate shape recovery.

6.1.3 Real datasets

Three datasets of real-world scenes are tested in this study: *Gold*, an ancient golden coin, *Elba*, an Italian relief sculpture, and *Frag*, a much-decorated golden frame (that surrounds a painting by Fragonard). Sample images of the three datasets are shown in Fig. 6.7.

The advantages and disadvantages of the methods we found using synthesized datasets are also observed in the real datasets. Most images in dataset *Gold* have a large area of cast shadow. The influence of shadow can be clearly seen on the normal maps recovered by LS, IRL1 and SBL (Fig. 6.8A1, A2 and A3, respectively), but is completely eliminated by OMP and SeqCS-MUSIC (Fig. 6.8A4 and A5, respectively). As for *Elba*, the scene contains a large number of small concave regions such as the pleats on the curtain. As expected, greedy algorithms fail at these regions. Again we notice that the LS, IRL1 and SBL results are more protruding than the greedy results, for both *Gold* and *Elba* (Fig. 6.8A1–A5 and B1–B5). Although there is not a ground truth normal map to support our speculation, it is reasonable to argue that the non-greedy algorithms exaggerate the convexity of *Elba*, as was the case for *Caesar* (Fig. 6.6). The complex geometry of the object in our third dataset – *Frag* – accounts for the noisy estimates observed in concave regions in the greedy normal maps (Fig. 6.8C4 and C5). Note that the non-greedy results also show a large degree of inaccuracy in these regions (Fig. 6.8C1–C3), but in a less noticeable manner since these artifacts are usually smoothly blended into less-affected areas.

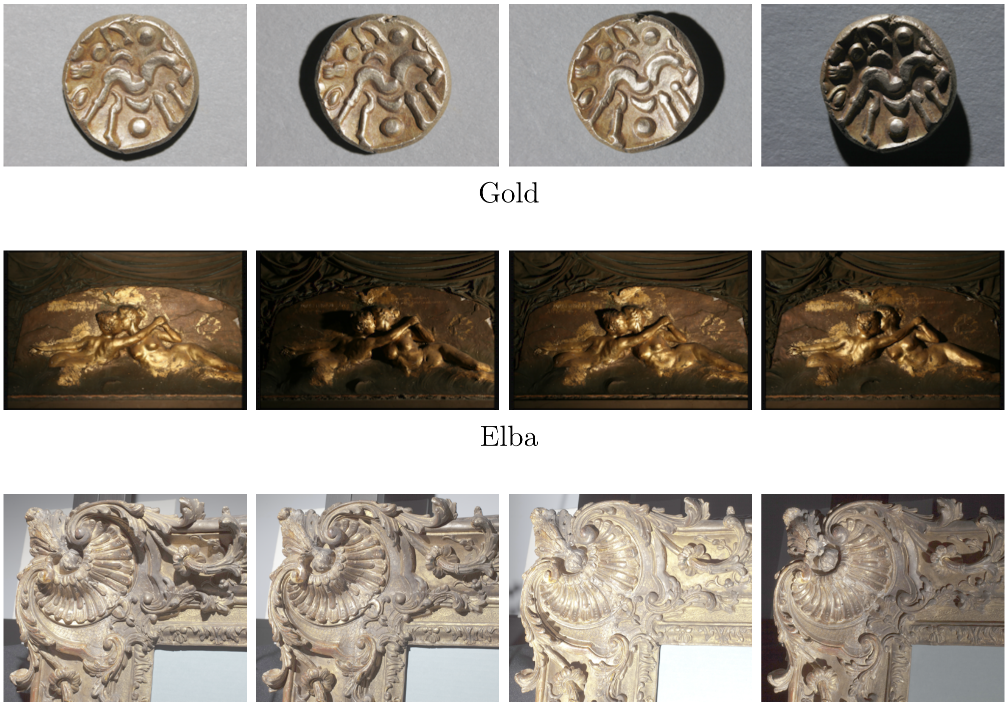


(a) LS (b) Ground Truth (c) LMS

(d) IRL1 (e) Ground Truth (f) OMP

(g) SBL (h) Ground Truth (i) SeqCS-MUSIC

Figure 6.6: Three-dimensional surfaces reconstructed from normal maps. (b, e, h) Depth map recovered with the ground truth normal map. The three ground truth depth maps are identical and are intentionally duplicated for the convenience of visual comparison. (a, d, g, c, f, i) Depth maps recovered using LS, IRL1, SBL, LMS, OMP, and SeqCS-MUSIC, respectively.



Gold

Elba

Frag

Figure 6.7: Sample images from three datasets of real-world scenes. From top row to bottom row: *Gold*, *Elba*, and *Frag*.

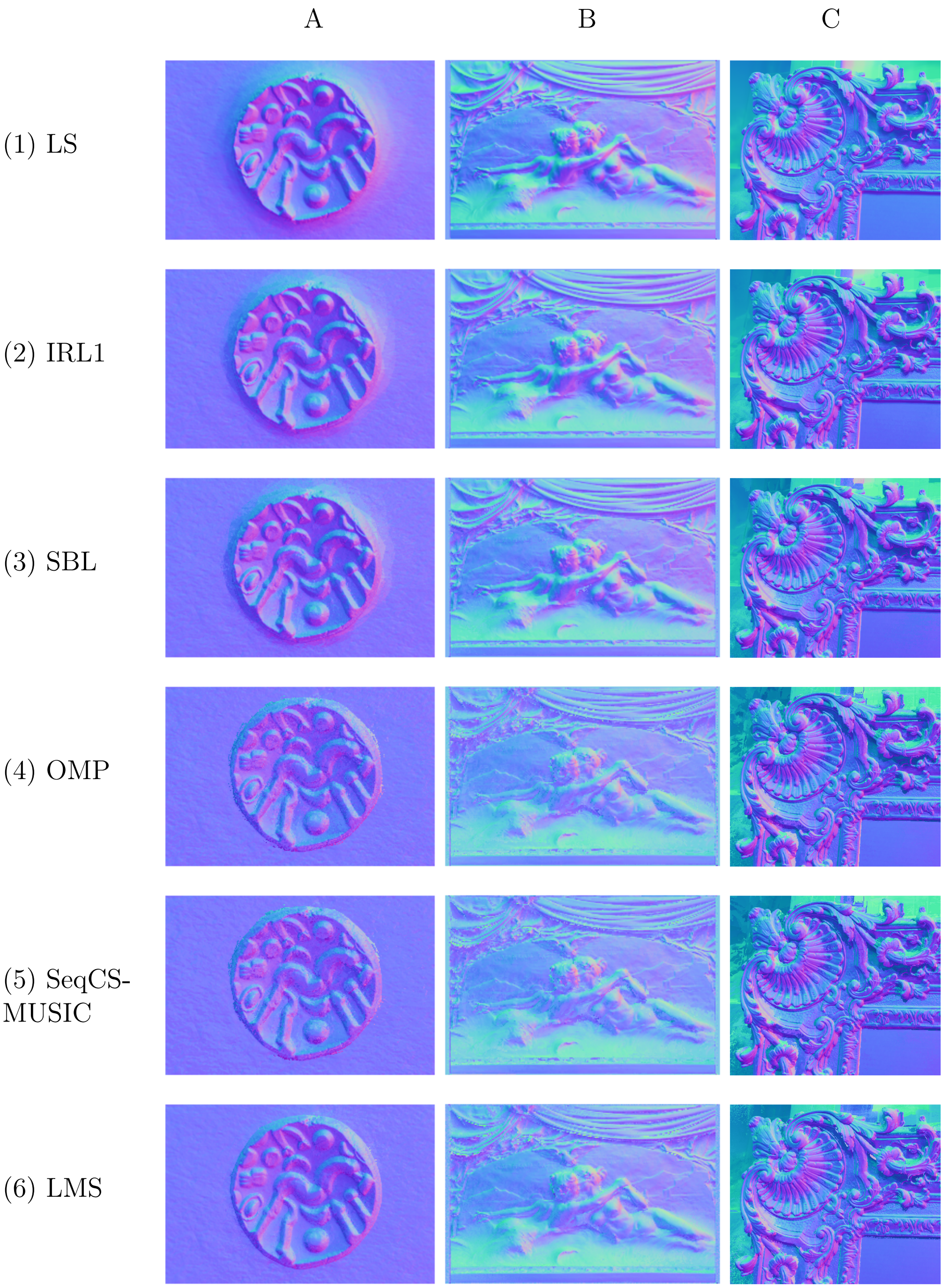


Figure 6.8: Normal maps for three real-world datasets recovered using various methods. Column A–C represent dataset *Gold*, *Elba* and *Frag*, respectively. Row (1)–(6) show results using LS, IRL1, SBL, OMP, SeqCS-MUSIC and LMS, respectively.

6.2 Robustness

For a better understanding of how well these methods behave in the presence of non-Lambertian effects, we tested their performance on *Sphere* with varying degrees of specularity, and on *Venus* where a large portion of the scene is concave and as such, is heavily polluted by cast shadow. To find out the robustness of these methods against external error introduced by the experimental setup, we also tested the methods with additive image noise and light calibration error.

6.2.1 Specularity

We rendered five datasets of the same object *Sphere* with highlight areas of various sizes (Fig. 6.9, top row), and tested how the size of the specular region affects the performance of our sparse regression methods. The size of the highlight is controlled by the **phong_size** parameter in POV-Ray [3]. We found that although the accuracy of all four methods compared (IRL1, SBL, OMP, SeqCS-MUSIC) decreases as the specular size increases, the greedy algorithms are less affected (Fig. 6.9, middle and bottom figures).

6.2.2 Shadow and concavity: a failure case

In Section 6.1.1, we have already noticed the possibility that the performance of our greedy algorithms may be negatively affected at shadowed concave regions. Here, we use the *Venus* dataset to further demonstrate this observation. In *Venus* (Fig. 6.1 bottom row), the convex foreground (the Venus statue) and the concave background (the dome) are well separated, allowing us to clearly inspect the performance of algorithms on different regions.

The result is shown in Fig. 6.10. As speculated, greedy methods show robustness in shiny, convex regions such as the outer rim of the dome, and on the statue itself, but fail on the heavily shadowed background. The other three methods (LS, IRL1 and SBL), on the contrary, suffer from noticeable angular error in convex areas. However, they are less severely affected by shadow and concavity on the background than greedy methods. Overall, the normal map recovered with greedy approaches are less smooth for *Venus* due to the inaccurate estimation of normal vectors in the concave regions.

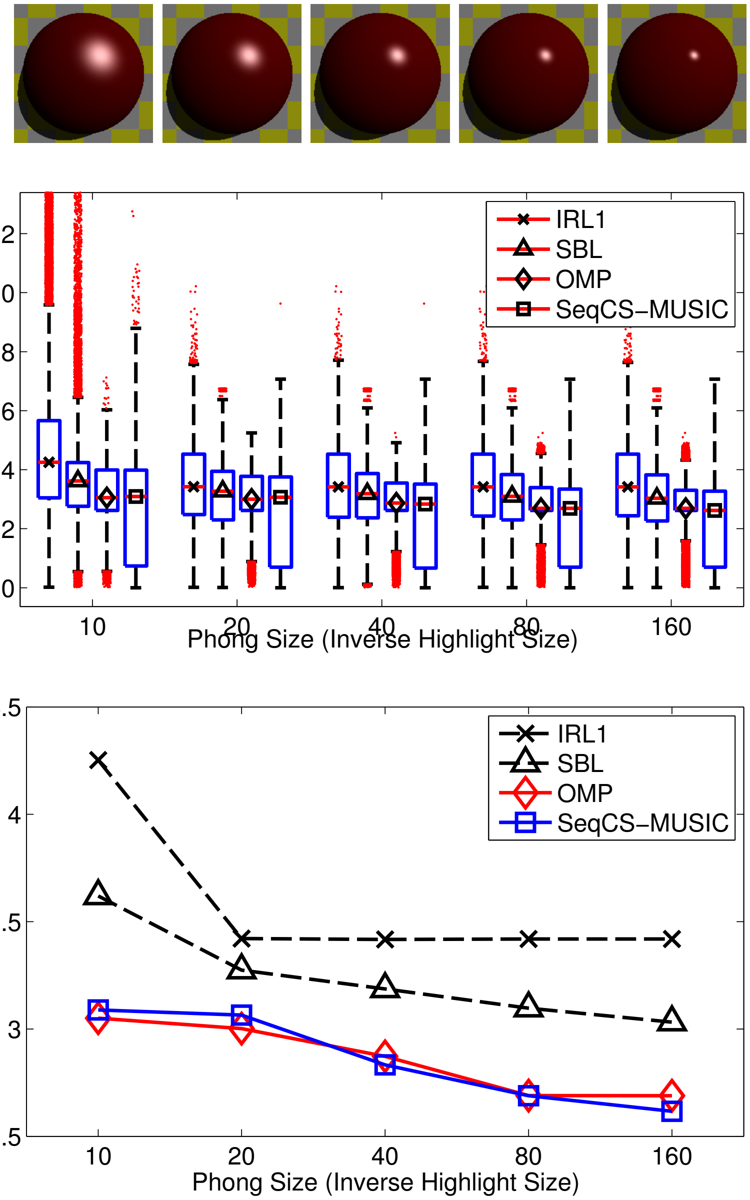
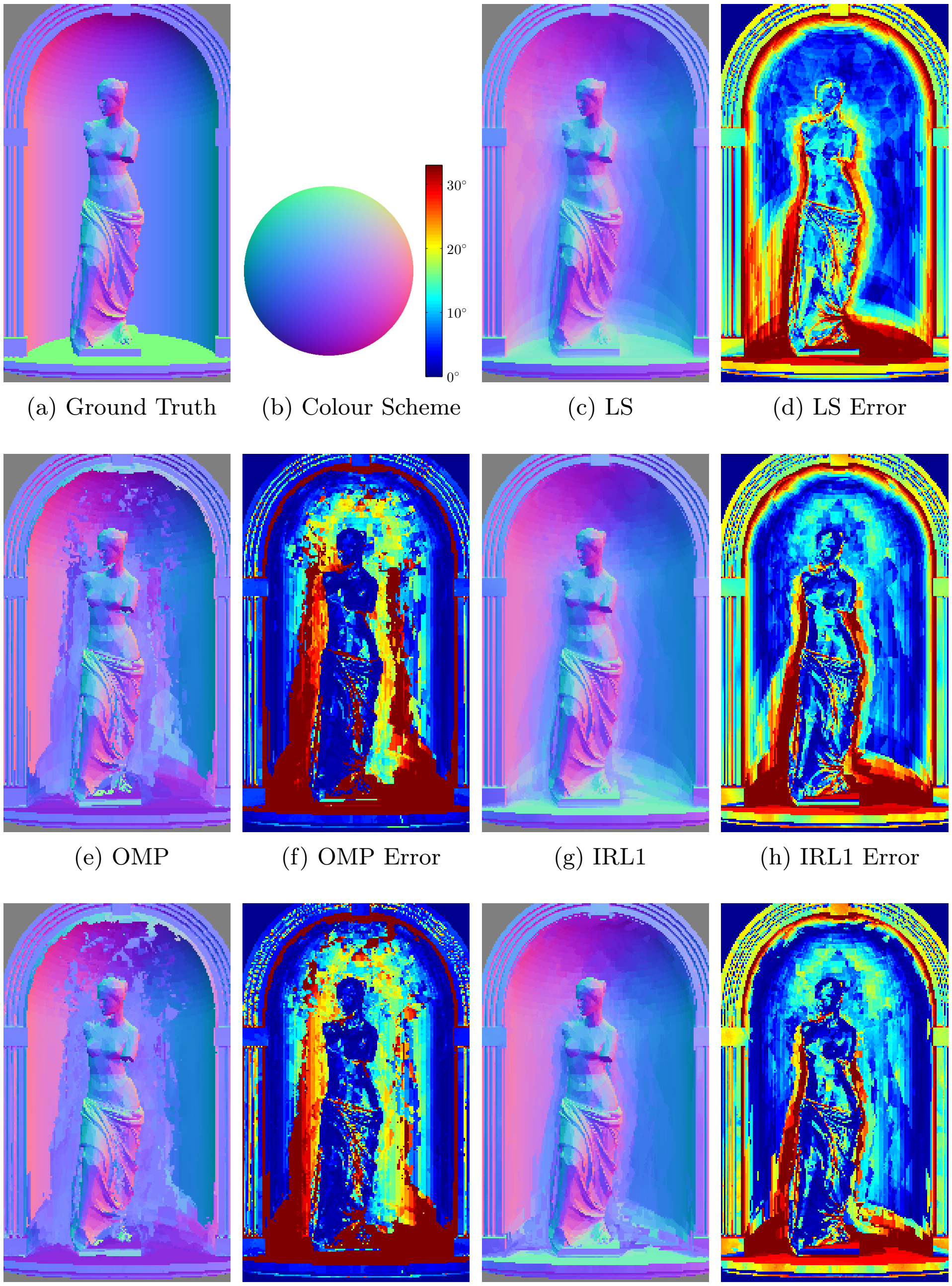


Figure 6.9: Robustness of sparse recovery methods against specularity. Upper row: Sample rendered images with **phong_size** 10–160, respectively from left to right. Middle figure: Boxplot of the angular error of normal maps under various degrees of specularity. Upper and lower border of blue boxes represent third (Q3) and first quartile (Q1), respectively. Upper and lower whiskers show 1.0 Interquartile Range (IQR) extended from Q3 and Q1, respectively. Red bars in blue boxes represent medians. At each specularity level, four sparse recovery methods are compared, symbolized by different markers on the median bar: IRL1 (cross), SBL (triangle), OMP (diamond), and SeqCS-MUSIC (square). Red dots are outliers distributed out of the error bar range. Lower figure: the medians of the angular error.



(a) Ground Truth (b) Colour Scheme (c) LS (d) LS Error

(e) OMP (f) OMP Error (g) IRL1 (h) IRL1 Error

(i) SeqCS-MUSIC (j) SeqCS-MUSIC Error (k) SBL (l) SBL Error

Figure 6.10: Normal maps of the *Venus* dataset recovered using various methods. (a) Ground truth normal map. (b) Colour wheel and colour bar used for normal and angular error visualization, respectively. Angular error is measured in degrees. (c,e,g,i,k) Normal maps recovered using LS, OMP, IRL1, SeqCS-MUSIC and SBL, respectively. (d,f,h,j,l) Angular error of normal maps recovered using the aforementioned methods.

6.2.3 Image noise

We tested four sparse algorithms (IRL1, SBL, OMP and SeqCS-MUSIC) against Gaussian noise as well as salt and pepper noise. For Gaussian noise (Fig. 6.11), the accuracy of all four methods drastically decreases as the noise level increases. Although greedy methods appear to be slightly more adversely affected by noise, the joint recovery method SeqCS-MUSIC still outperforms IRL1 and SBL while the noise distribution σ is within the range of $[0, 0.016]$. On the other hand, all four methods are quite insensitive to salt and pepper noise (Fig. 6.12). Although the performance of SeqCS-MUSIC remains the best among the four methods across all experimental conditions, it does show a slight decrease in performance at high levels of salt and pepper noise. This may indicate that the improvement introduced by SeqCS-MUSIC is more likely to be accounted for by the MUSIC step on a per pixel level, rather than the joint sparse recovery scheme.

6.2.4 Light calibration error

There might be cases when the lighting directions are not properly calibrated. That is, the assumed lighting directions deviate from their actual values. In this test, we introduce for every assumed lighting vector a fixed angular perturbation, ranging from 2° to 32°, at a random direction, while keeping the actual arrangement of lights unchanged.

We tested the performance of the sparse methods under various degrees of light calibration error on the *Caesar* dataset. The actual arrangement of lights is displayed in Fig. 6.13 (leftmost plot on the top row). As an increasingly greater random perturbation is added to the assumed lighting directions, the angular error gradually increases for all four sparse methods. Note that OMP appears to be most susceptible to the random calibration error, especially when the perturbation reaches 32°. This problem is partially eliminated by SeqCS-MUSIC.

Also note that in Fig. 6.13 (bottom), the median of the angular error produced by OMP slightly decreases at 16° compared to previous conditions. We believe that this is a fluctuation caused by the particular arrangement of lights at this condition. Despite this decrease in the median, the widths of the error distributions steadily increase at 16° for all four methods, as can be clearly seen from Fig. 6.13 (middle).

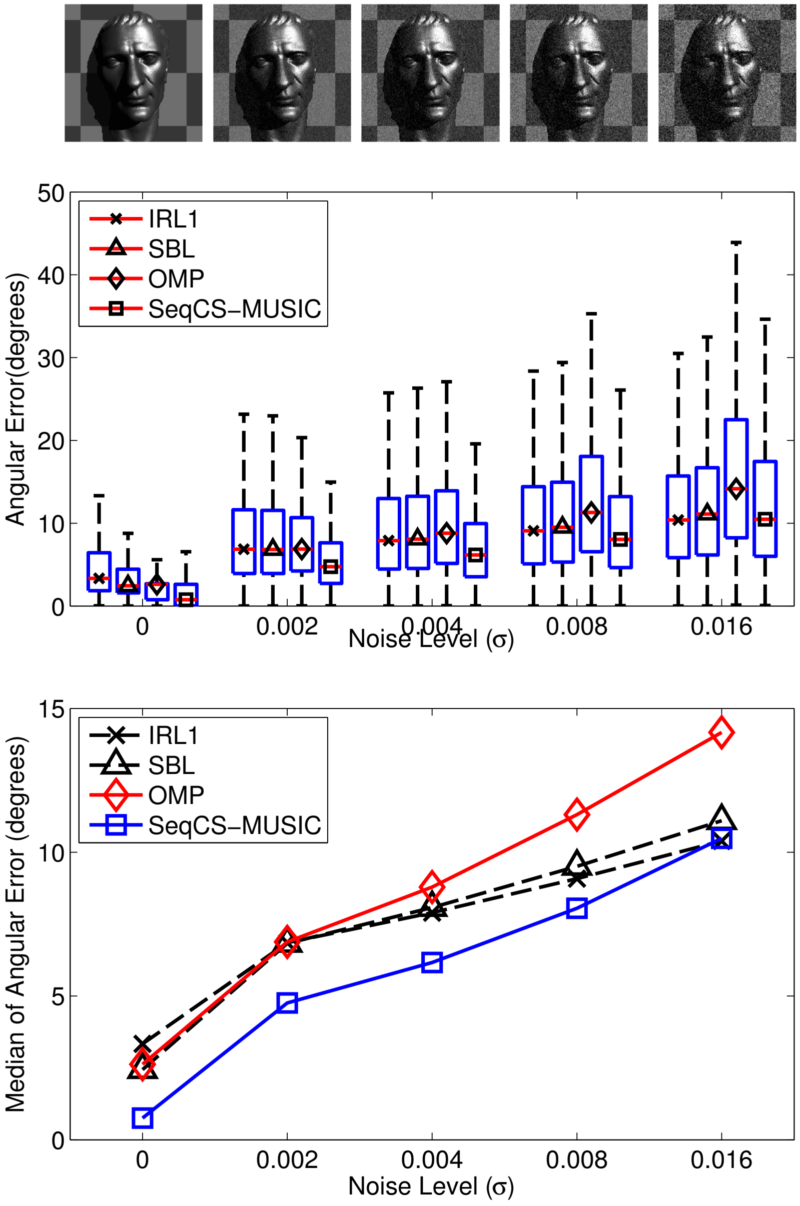


Figure 6.11: Robustness of sparse recovery methods against Gaussian noise. Upper row: Sample images rendered with Gaussian noise of Mean = 0.5, and STD = 0 − 0.016 , respectively from left to right. Middle figure: Boxplot of the angular error of normal maps under various degrees of Gaussian noise. Upper and lower border of blue boxes represent third (Q3) and first quartile (Q1), respectively. Upper and lower whiskers show 1.0 Interquartile Range (IQR) extended from Q3 and Q1, respectively. Red bars in blue boxes represent medians. At each noise level, four sparse recovery methods are compared, symbolized by different markers on the median bar: IRL1 (cross), SBL (triangle), OMP (diamond), and SeqCS-MUSIC (square). Lower figure: the medians of angular error.

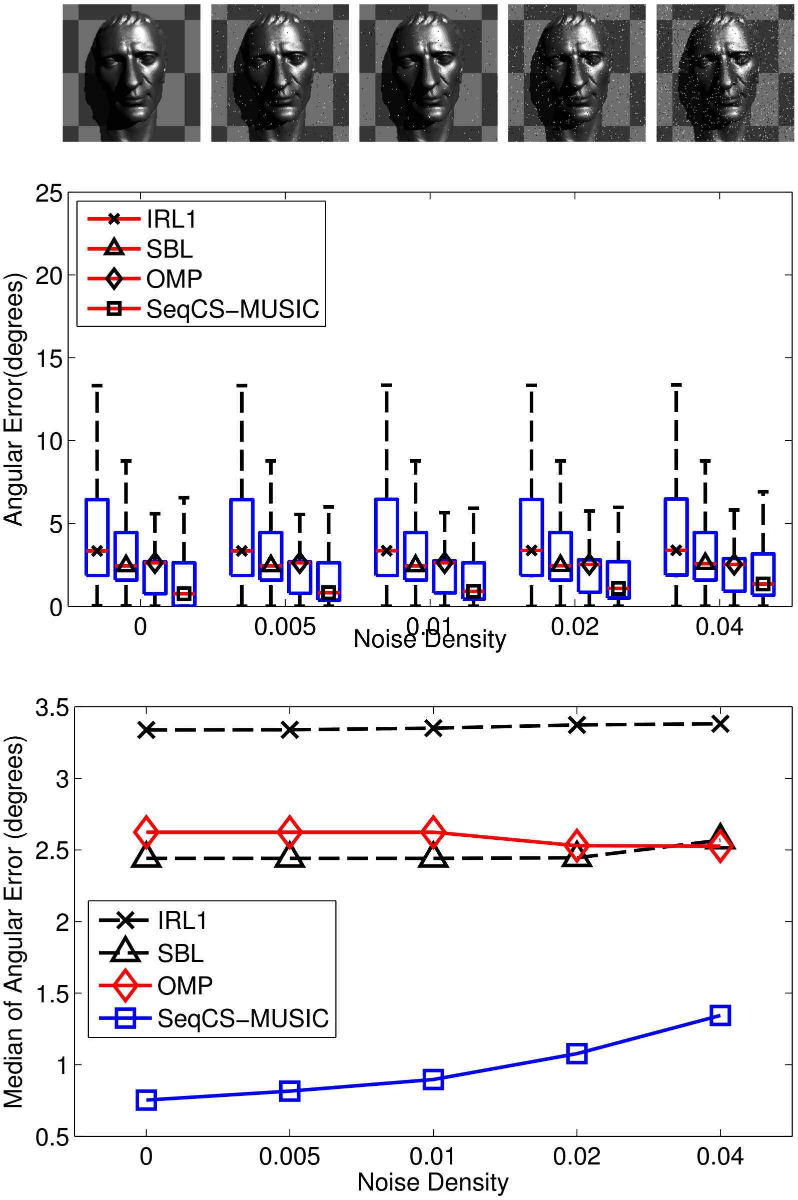


Figure 6.12: Robustness of sparse recovery methods against salt and pepper noise. Upper row: Sample images rendered with salt and pepper noise of density = 0 − 4%, respectively from left to right. Middle figure: Boxplot of the angular error of normal maps under various degrees of salt and pepper noise. Upper and lower border of blue boxes represent third (Q3) and first quartile (Q1), respectively. Upper and lower whiskers show 1.0 Interquartile Range (IQR) extended from Q3 and Q1, respectively. Red bars in blue boxes represent medians. At each noise level, four sparse recovery methods are compared, symbolized by different markers on the median bar: IRL1 (cross), SBL (triangle), OMP (diamond), and SeqCS-MUSIC (square). Lower figure: the medians of angular error.

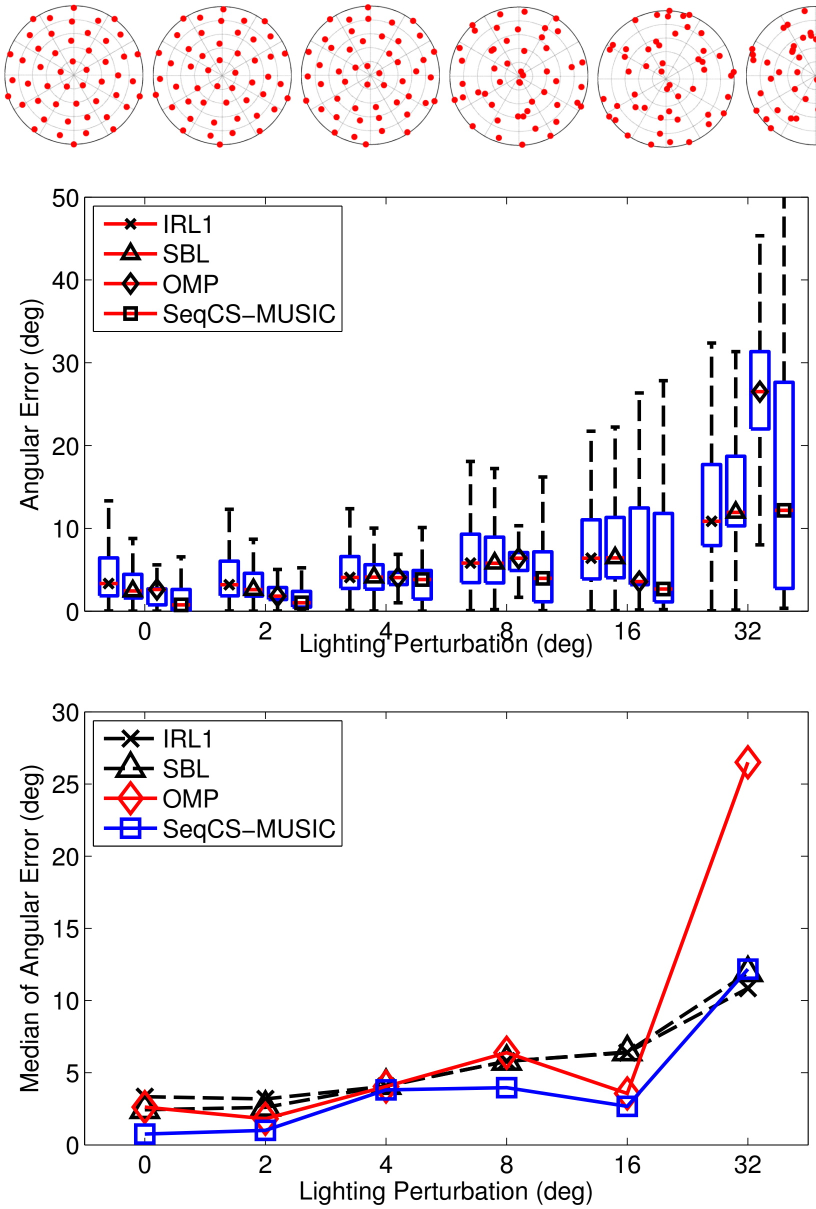


Figure 6.13: Robustness of sparse recovery methods against light calibration error. Top row: Leftmost plot shows actual light positions under which the dataset is generated. Remaining plots show miscalibrated light positions with angular perturbations ($2° - 32°$, from left to right) from the actual light positions at random directions. Middle figure: Boxplot of the angular error of normal maps under various degrees of light calibration error. Upper and lower border of blue boxes represent third (Q3) and first quartile (Q1), respectively. Upper and lower whiskers show 1.0 Interquartile Range (IQR) extended from Q3 and Q1, respectively. Red bars in blue boxes represent medians. At each angular perturbation level, four sparse recovery methods are compared, symbolized by different markers on the median bar: IRL1 (cross), SBL (triangle), OMP (diamond), and SeqCS-MUSIC (square). Bottom figure: the medians of angular error.

6.3 Efficiency

The actual per-pixel processing time for the MATLAB implementation of the seven algorithms tested in this study are reported in Fig. 6.14. The maximum number of iterations for IRL1 and SBL are set to 100 although the iteration will be terminated as soon as another stopping criterion is met; OMP and SeqCS-MUSIC always terminate after exactly 28 iterations for our datasets of 50 images; for LMS, the number of iterations is fixed at 1500 and this number is simply reduced to 50 for the simpler 1D LMS Mode finder.

In our current implementation, the running time of OMP (4.823 ms/pixel) and SeqCS-MUSIC (19.45 ms/pixel) are comparable to IRL1 (3.338 ms/pixel) and SBL (15.19 ms/pixel), respectively. LMS is the slowest (57.48 ms/pixel), though it can be made faster with fewer iterations at the expense of accuracy.

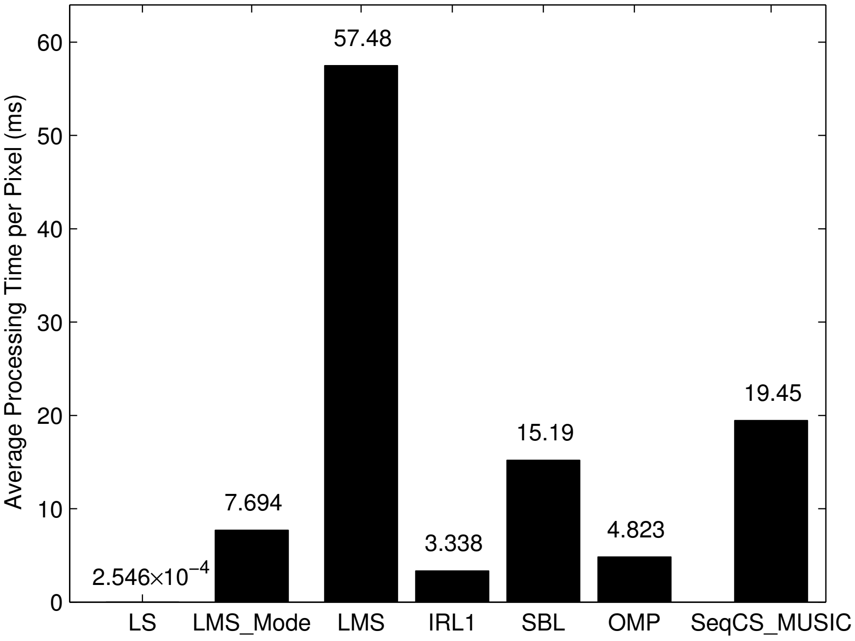


Figure 6.14: Average running time (per pixel) of photometric stereo algorithms measured in milliseconds.

6.4 Summary

Based on the experimental results above, we have come to the conclusion that greedy algorithms overall have a higher accuracy than L1 minimization and SBL with a comparable efficiency, though greedy approaches may be less robust in poorly illuminated regions. LMS is close to greedy algorithms in accuracy, despite its low efficiency. The algorithms tested in this chapter are summarized and compared in Table 6.2.

	Robustness				Smoothness	Efficiency
	Overall	Highlight	Shadow	Concavity		
LS	Very Low	Very Low	Very Low	Very Low	Very High	Very High
LMS	Very High	Very High	Very High	– –	Low	Very Low
LMS Mode	Low	Low	Low	– –	Medium	High
IRL1	High	High	High	Medium	High	High
SBL	High	High	High	Medium	High	Low
OMP	Very High	Very High	Very High	Low	Medium	High
SeqCS-MUSIC	Very High	Very High	Very High	Low	Medium	Low

Table 6.2: Qualitative comparison of photometric stereo algorithms. The performance is evaluated on a five-level scale: “Very Low”, “Low”, “Medium”, “High”, and “Very High”. Fields that are not available are indicated by a “– –” sign.

Chapter 7

Conclusion and Future Work

In this study, the classical PST is reformulated as the canonical form of sparse recovery, and a greedy algorithm – Orthogonal Matching Pursuit (OMP) – is applied to solve the problem. Our formulation is different from previous ones [69, 33] in that the former incorporates normal vector components and non-Lambertian errors in one combined vector, allowing for the straightforward application of OMP. In order for OMP to obtain normal estimations, the normal vector components have to be selected before the iteration stops. Although it is not theoretically guaranteed, we observed that the normal components are always selected within the first few iterations in the datasets we tested, unless some components are indeed zero or very close to zero. We also speculate that the dictionary matrix in our formulation is near-orthonormal and satisfies the conditions required by OMP to achieve exact recovery.

As an extension to the pixel-wise estimation scheme, a joint sparse recovery formulation is also introduced. In this framework, many adjacent pixels are estimated simultaneously through joint sparse recovery and up to 9 estimates of normal vectors will be obtained for each pixel. Then the centre vector of these estimates is deemed to be the final result. The Simultaneous Orthogonal Matching Pursuit (S-OMP) algorithm is employed as the base solver, and Sequential Compressive MUSIC (SeqCS-MUSIC) is applied to refine the support recovery result of OMP.

We found that our greedy methods – OMP and SeqCS-MUSIC – in general outperform other state-of-the-art sparse solvers such as IRL1 and SBL [33] with little loss of efficiency. In particular, the greedy methods provide a more numerically accurate estimation of normal vectors in the presence of common non-Lambertian effects such as highlights and cast shadows, although they may occasionally fail at concave areas that are poorly illuminated. In addition, all sparse methods tested are reasonably robust against additive image noise and lighting calibration error.

Another two outlier-removal based methods – LMS and LMS Mode finder – are also tested in this study for comparison. LMS results are overall statistically more accurate than IRL1 and SBL, but less so than OMP and SeqCS-MUSIC. LMS Mode finder, the 1D simplification of LMS, shows some robustness against non-Lambertian errors, especially highlights, but performs poorly against cast shadows.

This study opens up many possible directions for future research. First, a great number of sparse recovery algorithms have already been proposed in the past few decades, each designed for a specific formulation. Even within the domain of greedy algorithms, there are many potential candidates aside from OMP that may be directly applied to PST. It would be interesting to explore this large repertoire of sparse formulations and recovery algorithms to find an optimal method for the PST problem.

Another future research topic is on enforcing a better smoothness constraint onto the pixel-wise sparse regression. One may notice that the normal maps recovered with LS, IRL1 and SBL are usually visually smooth, even though they may greatly deviate from the ground truth. On the other hand, smoothness cannot be guaranteed by a pure greedy algorithm, especially in shadowed concave regions. The joint sparse formulation and SeqCS-MUSIC relieve this problem with moderate success. However, in large regions where a pixel and all its neighbours are inadequately illuminated, the improvement on smoothness introduced by the joint sparse framework is limited. Moreover, the backward and forward greedy steps in SeqCS-MUSIC are very time-consuming. Thus, more work needs to be done in order to construct an efficient and strong smoothness constraint that addresses shadows and concavity gracefully.

It has been shown that sparse methods such as IRL1 and SBL can be used to estimate the lighting directions in the context of uncalibrated PST [6]. It is highly possible that greedy algorithms can also be applied to this problem. Future studies may reveal more applications of greedy algorithms in different aspects of the PST framework.

Bibliography

[1] 3D meshes research databse by INRIA Gamma Group. http://www-roc.inria.fr/gamma/gamma/download/download.php. 36

[2] AIM@SHAPE shape repository. http://shapes.aimatshape.net/. 36

[3] POV-Ray 3.6.0 documentation online. http://www.povray.org/documentation/view/3.6.0/347/. 37, 49

[4] J. Ackermann, M. Ritz, A. Stork, and M. Goesele. Removing the example from example-based photometric stereo. In *Trends and Topics in Computer Vision*, volume 6554 of *Lecture Notes in Computer Science*, pages 197–210. Springer Berlin Heidelberg, January 2012. 2, 9

[5] N. Alldrin, T. Zickler, and D. Kriegman. Photometric stereo with non-parametric and spatially-varying reflectance. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2008. 2, 3, 9

[6] V. Argyriou, S. Zafeiriou, B. Villarini, and M. Petrou. A sparse representation method for determining the optimal illumination directions in photometric stereo. *Signal Processing*, 93(11):3027–3038, 2013. 3, 10, 59

[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. 3, 6

[8] R. Basri, D. Jacobs, and I. Kemelmacher. Photometric stereo with general, unknown lighting. *International Journal of Computer Vision*, 72:239–257, 2007. 8

[9] T. Blumensath and M. E. Davies. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis*, 27(3):265–274, November 2009. 11

[10] S. P. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004. 11

[11] T. T. Cai and L. Wang. Orthogonal Matching Pursuit for Sparse Signal Recovery With Noise. *IEEE Transactions on Information Theory*, 57(7):4680–4688, July 2011. 4, 11, 21

[12] E. J. Candès and J. Romberg. L1-magic: recovery of sparse signals via convex programming. *URL: www. acm. caltech. edu/l1magic/downloads/l1magic. pdf*, 4, 2005. 10, 11

[13] E. J. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006. 24

[14] E. J. Candès and T. Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005. 24

[15] E. J. Candès, M. B. Wakin, and S. P. Boyd. Enhancing sparsity by reweighted L1 minimization. *Journal of Fourier Analysis and Applications*, 14(5-6):877–905, December 2008. 10, 11

[16] V. Cevher. An ALPS view of sparse recovery. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5808–5811, 2011. 11

[17] M. Chandraker, S. Agarwal, and D. Kriegman. ShadowCuts: photometric stereo with shadows. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 7

[18] S. Chatterjee, D. Sundman, and M. Skoglund. Look ahead orthogonal matching pursuit. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 4024–4027, 2011. 11

[19] E. N. Coleman Jr. and R. Jain. Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry. *Computer Graphics and Image Processing*, 18:309–328, 1982. 6

[20] W. Dai and O. Milenkovic. Subspace pursuit for compressive sensing signal reconstruction. *IEEE Transactions on Information Theory*, 55(5):2230–2249, 2009. 11

[21] A. Divekar and O. Ersoy. Probabilistic matching pursuit for compressive sensing. Technical report, Purdue University, West Lafayette, IN, USA, 2010. 11

[22] D. Donoho, A. Maleki, and A. Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, November 2009. 11

[23] D. Donoho. Compressed sensing. *IEEE Transactions on Information Theory*, 52(4):1289–1306, 2006. 2, 10, 11

[24] D. Donoho, Y. Tsaig, I. Drori, and J.-L. Starck. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 58(2):1094–1121, 2012. 11

[25] M. S. Drew, N. Hajari, Y. Hel-Or, and T. Malzbender. Specularity and shadow interpolation via robust polynomial texture maps. In *Proceedings of the British Machine Vision Conference*, pages 114.1–114.11, 2009. 2, 7

[26] M. S. Drew, Y. Hel-Or, T. Malzbender, and N. Hajari. Robust estimation of surface properties and interpolation of shadow/specularity components. *Image and Vision Computing*, 30(4-5):317–331, 2012. 2, 3, 7, 8, 16

[27] P. Favaro and T. Papadhimitri. A closed-form solution to uncalibrated photometric stereo via diffuse maxima. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 821–828, 2012. 3, 10

[28] M. A. T. Figueiredo, R. Nowak, and S. Wright. Gradient projection for sparse reconstruction: application to compressed sensing and other inverse problems. *IEEE Journal of Selected Topics in Signal Processing*, 1(4):586–597, 2007. 11

[29] R. T. Frankot and R. Chellapa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):439–451, 1988. 45

[30] D. Goldman, B. Curless, A. Hertzmann, and S. Seitz. Shape and spatially-varying BRDFs from photometric stereo. In *IEEE International Conference on Computer Vision*, volume 1, pages 341–348, 2005. 9

[31] C. Hernandez, G. Vogiatzis, and R. Cipolla. Multiview photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(3):548–554, 2008. 2, 7

[32] A. Hertzmann and S. M. Seitz. Shape and materials by example: a photometric stereo approach. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages I533–I540, 2003. 2, 8

[33] S. Ikehata, D. Wipf, Y. Matsushita, and K. Aizawa. Robust photometric stereo using sparse regression. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 318–325, June 2012. 3, 4, 5, 10, 12, 21, 58

[34] P. Jain, A. Tewari, and I. S. Dhillon. Orthogonal matching pursuit with replacement. In *Proceedings of Neural Information Processing Systems*, pages 1215–1223, June 2011. 11

[35] C. Julià, F. Lumbreras, and A. D. Sappa. A factorization-based approach to photometric stereo. *International Journal of Imaging Systems and Technology*, 21(1):115–119, 2011. 7

[36] N. B. Karahanoglu and H. Erdogan. A* orthogonal matching pursuit: Best-first search for compressed sensing signal recovery. *Digital Signal Processing*, 22(4):555–568, July 2012. 11

[37] S. Kherada, P. Pandey, and A. Namboodiri. Improving realism of 3D texture using component based modeling. In *IEEE Workshop on Applications of Computer Vision*, pages 41–47, 2012. 9

[38] J. M. Kim, O. K. Lee, and J.-C. Ye. Compressive MUSIC: Revisiting the link between compressive sensing and array signal processing. *IEEE Transactions on Information Theory*, 58(1):278–301, 2012. 29, 33

[39] J. M. Kim, O. K. Lee, and J.-C. Ye. Improving noise robustness in subspace-based joint sparse recovery. *IEEE Transactions on Signal Processing*, 60(11):5799–5809, 2012. 5, 33, 34

[40] S. Koppal and S. Narasimhan. Clustering Appearance for Scene Analysis. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1323–1330, 2006. 8

[41] Z. Lin, M. Chen, and Y. Ma. The augmented Lagrange multiplier method for exact recovery of corrupted low-rank matrices. Technical report, UIUC (UILU-ENG-09-2215), September 2009. 10

[42] A. Maleki and D. Donoho. Optimally tuned iterative reconstruction algorithms for compressed sensing. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):330–341, 2010. 11

[43] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. 11

[44] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, pages 519–528, 2001. 8, 16

[45] D. Miyazaki, K. Hara, and K. Ikeuchi. Median photometric stereo as applied to the Segonko Tumulus and museum objects. *International Journal of Computer Vision*, 86(2-3):229–242, January 2010. 7

[46] Y. Mukaigawa, Y. Ishii, and T. Shakunaga. Analysis of photometric factors based on photometric linearization. *Journal of the Optical Society of America A*, 24(10):3326–3334, October 2007. 7

[47] B. K. Natarajan. Sparse approximation solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995. 3, 10

[48] S. 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. 8

[49] D. Needell and R. Vershynin. Signal recovery from incomplete and inaccurate measurements via regularized orthogonal matching pursuit. *IEEE Journal of Selected Topics in Signal Processing*, 4(2):310–316, 2010. 11

[50] D. Needell and J. A. Tropp. CoSaMP: Iterative signal recovery from incomplete and inaccurate samples. *Communications of the ACM*, 53(12):93–100, December 2010. 11

[51] D. Needell and R. Vershynin. Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit. *Foundations of Computational Mathematics*, 9(3):317–334, April 2009. 11

[52] Y. Pati, R. Rezaiifar, and P. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Proceedings of the 27th Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 40–44, 1993. 4, 11, 21

[53] P. J. Rousseeuw. Least median of squares regression. *Journal of the American Statistical Association*, 79(388):871–880, 1984. 15

[54] H. Rushmeier, G. Taubin, and A. Guéziec. Applying shape from lighting variation to bump map capture. In *Proceedings of Eurographics Workshop on Rendering*, pages 35–44. 1997. 3, 6

[55] R. Schmidt. Multiple emitter location and signal parameter estimation. *IEEE Transactions on Antennas and Propagation*, 34(3):276–280, 1986. 33

[56] W. M. Silver. *Determining shape and reflectance using multiple images.* Master thesis, Massachusetts Institute of Technology, 1980. 8

[57] F. Solomon and K. Ikeuchi. Extracting the shape and roughness of specular lobe objects using four light photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:449–454, 1996. 6, 8

[58] J. Sun, M. Smith, L. Smith, S. Midha, and J. Bamber. Object surface recovery using a multi-light photometric stereo technique for non-Lambertian surfaces subject to shadows and specularities. *Image and Vision Computing*, 25:1050–1057, 2007. 7

[59] H. D. Tagare and R. J. P. de Figueiredo. A theory of photometric stereo for a class of diffuse non-Lambertian surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):133–152, 1991. 8

[60] K.-L. Tang, C.-K. Tang, and T.-T. Wong. Dense photometric stereo using tensorial belief propagation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 132–139, 2005. 7

[61] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Transactions on Information Theory*, 53(12):4655–4666, December 2007. 4, 11, 21

[62] J. A. Tropp, A. C. Gilbert, and M. J. Strauss. Simultaneous sparse approximation via greedy pursuit. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages v/721–v/724, 2005. 5, 32

[63] J. A. Tropp, A. C. Gilbert, and M. J. Strauss. Algorithms for simultaneous sparse approximation. Part I: Greedy pursuit. *Signal Processing*, 86(3):572–588, March 2006. 5, 32

[64] F. Verbiest and L. Van Gool. Photometric stereo with coherent outlier handling and confidence estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 2, 7

[65] G. Willems, F. Verbiest, W. Moreau, H. Hameeuw, K. Van Lerberghe, and L. Van Gool. Easy and cost-effective cuneiform digitizing. In *Short and Project Papers Proceedings of 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage*, pages 73–80, 2005. 7

[66] D. Wipf and B. Rao. Sparse Bayesian learning for basis selection. *IEEE Transactions on Signal Processing*, 52(8):2153–2164, 2004. 10, 12

[67] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):139–144, 1980. 3, 6, 13

[68] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma. Robust face recognition via sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(2):210–227, 2009. 4, 21

[69] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma. Robust photometric stereo via low-rank matrix completion and recovery. In *Proceedings of Asian Conference of Computer Vision*, pages 703–717, 2010. 3, 4, 10, 58

[70] Q. Yang and N. Ahuja. Surface reflectance and normal estimation from photometric stereo. *Computer Vision and Image Understanding*, 116(7):793–802, July 2012. 9

[71] A. Yuille and D. Snow. Shape and albedo from multiple images using integrability. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 158–164, 1997. 6

[72] M. Zhang and M. S. Drew. Robust luminance and chromaticity for matte regression in polynomial texture mapping. In *European Conference on Computer Vision Workshop on Color and Photometry in Computer Vision*, volume 7584/2012 of *Lecture Notes in Computer Science*, pages 360–369, October 2012. 2, 5, 7, 16, 27