ROBUST POLYNOMIAL TEXTURE MAPPING FOR SHADOW AND SPECULARITY INTERPOLATION

by

Nasim Hajari

B.Sc., University of Tehran, 2006

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Computing Science

SIMON FRASER UNIVERSITY
Fall 2009

APPROVAL

Name: Nasim Hajari

Degree: Master of Science

Title of Thesis: Robust Polynomial Texture Mapping for Shadow and Specularity Interpolation

Examining Committee: Dr. Richard Zhang
Chair

Dr. Mark Drew, Senior Supervisor

Dr. Greg Mori, Supervisor

Dr. Torsten Möller, Examiner

Date Approved: 5 Nov 2009

Declaration of Partial Copyright Licence

Abstract

PTM forms an alternative method for apprehending surface properties that extends a simple model of image formation from the Lambertian variant of PST to more general reflectances. Here we consider solving such a model in a robust version using either LMS or LTS, which is useful in the identification of matte, shadow and specularity automatically. Identifying the matte contribution, we can then estimate chromaticity and surface properties accurately. We can model specular and non-specular contributions using two sets of RBF regression over specular and non-specular pixels. Then for a new lighting direction, we can interpolate both specularity and shadows. Finally, to generate the full color output we can either utilize the estimated chromaticity along with specular chromaticity, or model specularity and shadow for each color channel separately. The latter is found to generate qualitatively and quantitatively better results. Overall, the proposed approach generates convincing interpolations of both specularities and shadows.

To my family!

"Don't worry, Gromit. Everything's under control!"

— *The Wrong Trousers,* AARDMAN ANIMATIONS, *1993*

Acknowledgments

I am deeply indebted to my senior supervisor, Dr. Mark Drew, for his continued support, encouragement and guidance through my research. This thesis would not have been possible without him.

I would also like to thank my supervisor, Dr. Greg Mori, and my examiner, Dr. Torsten Möller, for being on my committee and reviewing this thesis.

I would really like to thank my mother for her care and support throughout my life, and last but not least, I would like to thank my husband Hassan for his love and encouragement in every step that I take.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Texture mapping adds realism details to raster images with relatively small increase in computation and has long been used in computer graphics. Texture can be defined in the usual sense (such as cloth, wood, brick and so on), more specifically a detailed pattern or a multidimensional image that is mapped into a multidimensional space. In the latter form, a photograph is used to map onto a planar surface, which avoids modeling complex surface details. However this method fails when the lighting conditions of the synthetic environment are different from those of the texture image and the results will look unrealistic and flat [17].

Due to these problems, the field of image based modeling and rendering (IBMR) has attracted people's attention in recent decades. IBMR methods rely on a set of two dimensional images as inputs. In order to obtain photorealistic rendering, one should be able to characterize surface reflectance properties such as surface normal and albedo.

Photometric Stereo (PST) [31] is an image based modeling technique which solves the problem of surface normal recovery. PST uses at least three images from the same viewpoint but with different illumination directions as input. It then produces surface normals and albedos for each pixel by performing *linear regression* on illumination directions. However this method will fail in the presence of highlights and shadows or for non-Lambertian surfaces.

Polynomial Texture Maps (PTM) [17] form an alternative method for apprehending surface colour and albedo that extends a simple model of image formation from the Lambertian variant of Photometric Stereo (PST) to more general reflectance. PTM performs a *nonlinear polynomial regression* on lighting directions which can better model real radiance, and thus apprehend intricate dependencies due to self-shadowing and interreflections. The regression

produces six coefficients at each pixel. Subsequently, re-rendering can take place, e.g. by relighting images using a new lighting direction, by calculating surface normals and thence generating artificial specular highlights, by re-mapping color, by increasing directional sensitivity to lighting direction in order to enhance contrast, by light source extrapolation, or by artificially varying focus.

In this thesis, we are interested in using PTM as a vehicle to carry out *interpolation* of specularities and shadows. To the best of our knowledge robust methods have not to date been applied to PTM, and we use these to be able to accomplish interpolation. As well as using robust regression, this work moreover shows how outliers can be classified as belonging either to specular highlights or self- or cast-shadows. Knowledge of inlier pixel values means that recovered surface albedo and chromaticity is robust, in the sense of ignoring outlier contributions and thus more accurately mapping surface reflectance and color.

1.1 Approach

Our approach to solve the problem of shadow and specularity interpolation is based on robust regression applied to PTM. Robust regression helps in identification of outlier pixels(both shadow and specularity pixels) automatically without the need of any thresholds. It provides a tripartite set of weights for each pixel which are labeled as matte, shadow or specularity.

The next step is generating surface normals, albedo and chromaticity in order to recover the matte surface. The matte surface does not contain any shadow or specularity. PST and PTM is used to recover the desired quantities. However with inliers in hand, more accurate values for chromaticity, surface normals and albedo are generated.

With specular pixel values over the lights known, we can then model specularity using a radial basis function (RBF) regression. A second RBF regression is performed on non-specular pixel values to model shadow. It is then easy for a new lighting direction to interpolate both specular contents as well as shadows.

Finally the recovered chromaticity along with shadow and specularity is used to generate full color output results.

1.1.1 Highlights and Shadow Detection

The original version of PTM assumes that least squares regression will effectively be adequate for modeling a smooth dependence of images, with a fixed viewpoint, on lighting

direction. However least squares is very susceptible to outliers and its breakdown point is equal to zero. In other words if the dependency of the image on lighting direction is not smooth, which means there are shadows or highlights in the image, the PTM results will be affected. So it would be of great benefit if one can detect outliers, which are shadow and specularity in our problem.

To detect outliers and solve the PTM model we suggest using robust regression, which to the best of our knowledge has not to date been attempted for PTM. The main upshot of utilizing robust regression is in the identification of both shadows and specularities automatically, without the need for any thresholds.

In this thesis we applied two robust regression techniques to PTM and compare them according to some quantitative metrics as well as timing. We used *Least Median of Squares (LMS)* and *Least Trimmed Squares (LTS)* [25], each having a breakdown point of 50%, which means regression results will be unaffected even if half of the data points minus one are outliers.

1.1.2 Normal, Albedo and Color

Our method to solve the shadow and specularity interpolation problem can be considered as a three step method, where the first step is identification of shadows and specularities, the second step is modeling matte contribution and the last step is modeling shadow and highlight contribution.

In order to model matte pixels we first need to find surface normals and albedo. In this thesis three different methods (robust modified PTM, PST and robust PST) are used to recover surface normals and albedo. Although the result of robust PTM itself can be an alternative for generating matte surfaces, experimental results showed that matte surface recovery using surface normals is more accurate.

Also we propose two methods called Lum-PTM and RGB-PTM for dealing with color. The first method regresses the luminance at each pixel position; thus in order to add back color we need to recover chromaticity. The second methods regresses each color channel at each pixel position separately.

1.1.3 Shadow and Specularity Interpolation

PTM and robust PTM only captures the *smooth* dependence of image on lighting direction. In order to model the dependence of highlights and shadows on lighting direction, we make use of Radial Basis Function (RBF) regression.

Using the weights from robust regression, we then perform two separate sets of RBF regressions. The first RBF models the dependence of highlights on lighting direction, while the second one models the remaining non-matte contributions on the lighting direction.

After modelling matte, highlights and shadow contributions to lighting direction, we can interpolate both specular contents and shadow for a new lighting direction, and visualize how the image would look under that lighting.

1.2 Contribution

The main contributions in this thesis are:

1. application of robust regression to PTM.
2. separate modelling and thus better capturing of shadows and specularities.
3. specular and shadow interpolation.

Applying robust regression to PTM, provides us with a tripartite set of weights that labeles each pixel as matte, shadow or specularity. We then use the weights to separately model each contribution to the lighting direction. Finally with the models in hand we can interpolate shadow and specularity for a new lighting direction.

In this thesis we also *modify* PTM such that it can correctly generate surface normals and albedo if the surface is indeed Lambertian. The original PTM moves a linear regression involving lighting directions into a nonlinear, polynomial model of lighting dependence to elliminate the assumption of Lambertian model. However here we go back to a regression including a linear part corresponding to all three components of light direction, since if the surface is indeed Lambertian plus outliers due to highlighting and shadows, then a robust version of nonlinear regression will still pick up the correct, linear, Lambertian dependence exactly. Note that we do not make any assumption that the surface should be Lambertian and it can still work for any general surface.

1.3 Outline

The rest of this thesis is organized as follows: we will explain the previous work in the area of *robust photometric stereo* and *polynomial texture mapping* in Chapter 2. In Chapter 3 different robust regression techniques will be discussed and we will show how they can help us in shadows and specularities identification. Chapter 4 will discuss surface normal and surface albedo recovery as well as dealing with color images. In Chapter 5 we will explain shadow and highlight modeling on lighting direction and interpolation of these contributions for a new lighting direction. Chapter 6 presents experimental results and finally we will conclude the thesis and discuss future work in Chapter 7.

Chapter 2

Background and Related Work

Several powerful texture mapping methods have been proposed in the literature which avoid modeling of the complex surface details. Usually a photograph is used as a texture map on a planar surface. However this method fails if the lighting conditions of the synthetic environment are different from the lighting conditions of the texture image. To solve this problem, people have used image based modeling and rendering techniques. In these methods several images of the same object are taken under varying lighting directions and fixed viewpoint. These methods construct a surface reflectance model which characterizes surface appearance under different lighting directions. Polynomial Texture Mapping is an image based modeling method for a general surface. For a Lambertian surface people usually use a simple approach called Photometric Stereo to obtain surface normal and surface albedo. However Photometric Stereo is very prone to outliers (shadows and highlights), so some *robust* techniques have been proposed.

In this section some of the previous work in the fields of Robust Photometric Stereo and Polynomial Texture Mapping is presented.

2.1 Robust Photometric Stereo

Photometric Stereo (PST) was first introduced by Woodham [31]. This method provides an estimation for surface orientation (surface normal) and surface reflectance (albedo) at each pixel, using at least three images captured from a fixed viewpoint but under different illumination directions. In its most common variant, standard PST assumes the surface is Lambertian with no shadow or highlight present.

Suppose a Lambertian surface is illuminated in turn by three light sources with directions l_1, l_2 and l_3, where each light direction has the orientation $l = (l_x, l_y, l_z)^T$. Thus the luminance intensity at each pixel can be expressed as:

$$e_i = \alpha n \cdot l_i \ (i = 1, 2, 3) \tag{2.1}$$

where α is the albedo and n is the surface normal with orientation $n = (n_x, n_y, n_z)$ at each pixel. We can rewrite Eq. 2.1 in matrix form as:

$$E = \tilde{n} \cdot L \tag{2.2}$$

where L is lighting matrix ($L = (l_1, l_2, l_3)$), E is a vector of measured per-pixel luminance for each lighting direction ($E = (e_1, e_2, e_3)$) and $\tilde{n}$ is the unit normal scaled by albedo. Then the surface orientation for this pixel can be solved by:

$$\tilde{n} = L^+ \cdot E \tag{2.3}$$

where L^+ is the Moore-Penrose pseudoinverse of L. Albedo is the norm of $\tilde{n}$ and normalized $\tilde{n}$ is the surface orientation.

Surface recovery of PST will be affected in the presence of specularities and shadows. In this case the normal will be bent more toward the light direction which produces a highlight or away from the light direction which produces shadowing.

One of the earliest ideas for robust photometric stereo was increasing the number of captured images to more than three. Coleman and Jain [7], and Solomon and Ikeuchi [27], proposed a method to detect specularity using a series of four captured images.

Coleman and Jain [7] compared the recovered albedos from all four possible triplets of each pixel. If there is a highlight at that pixel then the recovered albedos would differ significantly. So the smallest albedo, containing only the Lambertian component, is used for recovery. However not all the surface pixels are illuminated by all four lights.

Solomon and Ikeuchi [27] solved this problem. They stated that the set of surface orientations that a light source can illuminate can be represented on the Gaussian sphere. If the surface is illuminated by four light sources, three illumination regions will be produced. Regions illuminated by all four light sources, by three light sources and by two light sources are shown in Figure 2.1. Region 1 is illuminated by all four light sources. Regions 2, 3, 4 and 5 are illuminated by three light sources and regions 6, 7, 8 and 9 are illuminated by two light sources respectively. Different strategies were suggested for detecting highlights and local

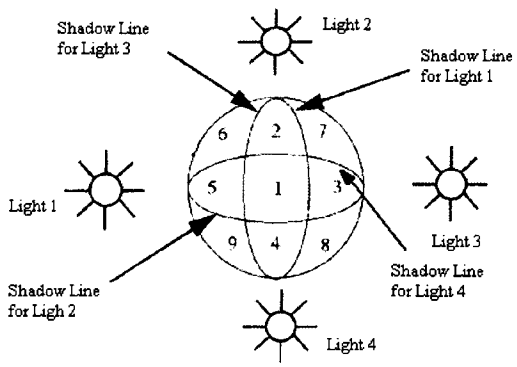


Figure 2.1: Three different illumination region

surface recovery for each of the regions. For regions illuminated by all four light sources, they used the method proposed in [7] to detect specular pixels and recover surface properties. To recover surface normal and surface albedo in the area illuminated by three light sources, they exclude the light source which may cause highlights and use the information of neighboring pixels and available light sources. For instance area 3 in Figure 2.1 is illuminated by light sources 2, 3 and 4 and is in the shadow region of light 1. Light 3 may cause highlight in this area, so they excluded this light and used Eq. 2.4 to get surface normals:

$$\frac{e_2}{\alpha} = l_2.n \quad \frac{e_4}{\alpha} = l_4.n \quad ||n|| = 1 \tag{2.4}$$

Luminance (e) and light direction (l) are known at each pixel. The albedo (α) can be calculated from the albedo of the Lambertian neighbor pixels. They later compared the luminance of light 3 (e_3) to the Lambertian luminance just calculated to find out whether light 3 causes highlight or not. For regions illuminated by two light sources, they assumed that no specularity would be present and used Eq. 2.4 to recover surface normals and surface albedo. However this method has several limitations. First of all they assumed that the albedo is spatially uniform. They also assumed that shadows are perfectly black and used a simple threshold to find shadow areas, while in real images there is always a range of shadow values.

The method proposed by Yuille and Snow [32] is another four source photometric stereo method which gives a solution for surface reconstruction in the presence of shadow. They explicitly include ambient illumination and surface integrability. They used robust regression

in an iterative fashion to eliminate shadows. Their paper didn't deal with specularities.

Another four source photometric stereo was proposed by Barsky and Petrou [4]. They stated that in a Lambertian model, a linear combination of intensities for any non-shadowed pixels should approximate to zero as shown in Eq. 2.5.

$$a_1e_1 + a_2e_2 + a_3e_3 + a_4e_4 = 0 \tag{2.5}$$

If Eq. 2.5 is not satisfied, there would be shadow or specularity at that pixel. Then they used a threshold to find out whether shadow or specularity existed. They recovered the surface normals using three darkest pixels in the case of specularity or three brightest pixels in the case of shadow.

Recently Argyriou et al. [1], and Argyriou and Petrou [2] generalized the method proposed by [4] into a multiple light sources technique. Their proposed method is a recursive algorithm which eliminates intensities affected by shadows or highlights, based on a least squares error scheme.

Miyazaki et al. [18] proposed a five light source photometric stereo which is suitable for images taken from a virtual museum, where objects are kept beyond glasses. It is aimed to detect specularities and exclude them from normal and albedo reconstruction. The main idea behind this algorithm is the same as [7]. Their method is robust to outliers under the condition that only a small number of images are supplied.

Rushmeier et al. [26] proposed a five light source photometric stereo system where the highest and lowest values in five components are discarded to avoid highlight and shadow. They used the three middle intensities to reconstruct surface normals.

Sun et al. [28] showed that the minimum number of lights that photometric stereo needs to illuminate the complete visible surface of any convex object is six. They argued that simply discarding highest or lowest intensity pixels may lead to some important information loss. So unlike [26], they only discard pixels with doubtful intensities. They argued that in their system there is at most one highlight among the six pixel intensities, but there is a possibility of two shadows occurring. So the highest and the two lowest intensities can be problematic. However they discarded these values if they were to be problematic. They used a threshold to determine if a pixel is problematic or not.

It is obvious that extra lights can eliminate the effects of shadows and specularity and make the system more robust. Wenger et al. [30] make use of a spherical device to capture 156 images. Since their system is very overdetermined, they simply discard the lowest 50%

and the highest 5% as possible shadow and specularity respectively to perform a robust photometric stereo. However the thresholding or arbitrary discarding depends strongly on surface shape and surface material, and these percentages cannot be the same for all objects.

Julia et al. [14] defined two thresholds to simply discard shadow and highlight pixels. The lower threshold, which corresponds to shadows, depends on the intensity values in each set of images, while they set the upper threshold, which is related to highlight, to 255. It is obvious that the second thresholding is not realistic.

Chandarker et al. [6] used a fast graph-cut based method for performing Lambertian PST in the presence of shadows. They labeled each pixel as shadow or non-shadow. So the problem is finding a labeling vector w_p for each pixel with length equal to the number of light sources with elements either 1 for non-shadow pixels or 0 for shadow pixels, such that the given energy function E is minimized:

$$E(W) = \sum_{p \in \mathcal{P}} D_p(w_p) + \sum_{(p,q) \in \mathcal{N}} (V_{p,q}(w_p, w_q)) \tag{2.6}$$

where $\mathcal{P}$ is the set of pixels and $\mathcal{N}$ is a neighborhood. The first term in the above expression measures the disagreement between a given labeling and the observed data and is called the data term. The second term imposes a penalty on variation of labeling within a neighborhood and is called the smoothness term. They used the fast graph-cut based algorithm of Boykov et al. [5] to minimize the above equation.

The data term is defined as Eq. 2.7

$$D_p(w_p) = ||w_p(n'_p \cdot L) - E_p|| \tag{2.7}$$

and the smoothness term is the Hamming distance between w_p and w_q, shown in Eq. 2.8

$$V_{p,q}(w_p, w_q) = ||w_p - w_q||_H \tag{2.8}$$

Verbiest and Van Gool [29] introduces a probabilistic imaging model which can detect outliers. They used a dense system which captures 169 images of an object. They formulate the problem as a Maximum Likelihood estimation problem and used an Expectation Maximization technique to solve it. However their proposed method needs some prior information to speed up convergence and lead to a more likely solution, so for each pixel they considered the 50% brightest intensity as inliers and then run the algorithm iteratively until it converges.

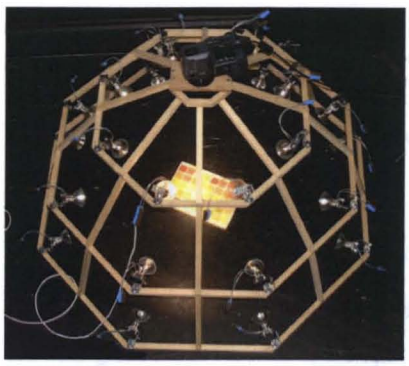


Figure 2.2: Hemispherical dome with multiple, identical lights

Hernández et al. [13] proposed a method to detect shadows in three-source photometric stereo. They used a simple Markov Random Field optimization scheme to identify and segment shadow regions in an image. Since one constraint has been lost due to shadow, they showed that integrability over two remaining constraints can still lead to surface geometry reconstruction in the shadow region.

2.2 Polynomial Texture Mapping

PTM proposed by Malzbender et al. [17] is formulated as a generalization of the simplest variant of PST [31]. It moves a linear regression involving lighting directions into a nonlinear, polynomial model of lighting dependence. The polynomial regression from lighting directions to observed image values can better model real radiance, and thus apprehend intricate dependencies due to self-shadowing and interreflections. In a typical setup as shown in Figure 2.2, a camera is mounted at the apex of a hemisphere of lights, and each light is then fired in turn, thus generating a sequence of images. Usually, some 40 to 50 images are used, with the larger the number of images the better.

Thus PTM is a pixel-based method for modeling dependency of luminance (or RGB in a different embodiment) on lighting with the objective being *relightable* images. At each pixel, a 6-vector of coefficients is calculated using nonlinear least squares over the 2-vector of lighting directions $u, v-$ which are projected components of normalized light vector into the local texture coordinate system. Figure 2.3 and Eq. 2.9 shows the calculation of the projected components.

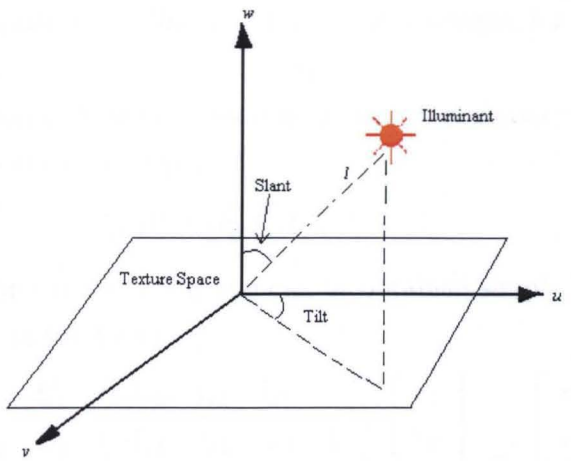


Figure 2.3: Projection of light vector into texture coordinate system

$$l = (l_u, l_v, l_w) \quad l_u = \sin\sigma \cos\tau \quad l_v = \sin\sigma \sin\tau \quad l_w = \cos\sigma \tag{2.9}$$

where l is normalized light vector, σ is slant angle and τ is tilt angle.

Suppose we have acquired n images of a scene, taken from $i = 1..n$ different lighting directions l_i. They determined that chromaticity of a particular pixel is fairly constant under varying lighting directions and it is luminance that varies. Since the reduced dimensional problem is faster, Malzbender et al. [17] preferred working on luminance PTM rather than RGB PTM, operating on R, G, B separately, although both of them are applicable. Malzbender et al. defined luminance at each pixel as a sum of three different channels R, G and B at that pixel, which is shown in Eq. 2.10:

$$e_i = R_i + G_i + B_i \tag{2.10}$$

Then a PTM model consists of a nonlinear regression from lighting to luminance via a vector of polynomial terms p, with p a function of lighting direction l, as follows:

$$\begin{bmatrix} p(l_1) \\ p(l_2) \\ \dots \\ p(l_n) \end{bmatrix} C = \begin{bmatrix} e_1 \\ e_2 \\ \dots \\ e_n \end{bmatrix}, \text{ or } \quad PC = E \tag{2.11}$$

where C is a vector of regression coefficients. Each pixel has its own C, and the E vector is the collection of all luminances at that pixel over the n images, for polynomials P for the known lighting directions.

They defined function of lighting direction, p, as a 6-component set p_0 of polynomial terms in $\{u, v\}$ space as shown in Eq. 2.12

$$p_0(l) = (l_u^2, l_v^2, l_u l_v, l_u, l_v, 1) \tag{2.12}$$

and the coefficient vector c is a 6-vector and can be obtained simply by least squares. Thus we can rewrite Eq. 2.11 in the form:

$$\begin{bmatrix} l_{u1}^2 & l_{v1}^2 & l_{u1}l_{v1} & l_{u1} & l_{v1} & 1 \\ l_{u2}^2 & l_{v2}^2 & l_{u2}l_{v2} & l_{u2} & l_{v2} & 1 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ l_{un}^2 & l_{vn}^2 & l_{un}l_{vn} & l_{un} & l_{vn} & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \cdots \\ c_6 \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \\ \cdots \\ e_n \end{bmatrix} \tag{2.13}$$

The use of a polynomial model is purely heuristic and other functions may be better; however a low-order polynomial guarantees a basically smooth model for matte reflectance and this is what we desire.

PTM has been used in several applications. Padfield et al. [21] employed PTM to monitor surface changes of paintings. They examined several paintings and showed that PTM can record surface features such as craquelure, planner distortion, wood grain, canvas wave and so on. They applied PTM to paintings before and after physical changes which enabled them to monitor surface changes. Hammer et al. [12] used PTM to visualize fossils. Many fossils cannot be fully illustrated using conventional photographic approaches. Thus PTM is used to resolve this problem.

Dong and Chantler [8] presented and compared five methods including PST, a gradient method, PTM, and eigenvector basis images for synthesizing 3D textures. They defined a gradient method as an over-constrained PST, where the number of images is more than three. This system may be solved using a least squares technique. In Eigen3 and Eigen6 methods, they used 3 or 6 base images in an eigenspace for surface texture synthesizing. They performed Singular Value Decomposition (SVD) on n sample images to get basis images. Suppose M is the sample space matrix where row i represents the pixels of image i. Thus M is an $m \times n$ matrix, where m is the number of images and n is number of pixels per image. By performing SVD on M they will end up with:

$$M = U\Sigma V^T \tag{2.14}$$

In Eq. 2.14, Σ is a diagonal matrix with elements being the singular value of matrix M, $\Sigma = diag(\sigma_1, \sigma_2, ..., \sigma_m)$, and $\sigma_i > \sigma_{i+1}$. Each row of matrix ΣV^T is a base image. They used the first 3 or first 6 base images in Eigen3 or Eigen6 respectively. To relight the surface they simply generate linear combination of the base images. Their results showed that PST and gradient methods failed in the presence of highlights and shadows as they are based on Lambertian assumption. The non-linear polynomial regression in PTM helps it to capture more complicated dependencies due to self-shadowing and interreflection, however it still cannot deal with specularities and cast-shadowing. The eigen-based methods also failed to model complex surfaces and they need more base images to represent these surfaces.

Finally Dong et al. [9] introduced a mathematical framework which is used to express three commonly used surface texture relighting representations: surface gradients (Gradient), Polynomial Texture Maps (PTM) and eigen base images (Eigen). The framework explicitly reveals the relations between the three methods, and a set of conversion methods.

Chapter 3

Highlight and Shadow Detection

The first step to perform interpolation on shadows and highlights is detecting them. In this thesis we used robust regression on a modified version of polynomial texture maps in order to find outliers. This altered version will be explained in details in Chapter 4.

Robust regression can find outliers automatically and propose the best fit for inliers. Our model can work on any arbitrary surface and the outliers are shadows and specularities. The input to the regression technique is a set of n intensities for each pixel, where n is the number of images. In this thesis we tried two different robust techniques (Least Median of Squares and Least Trimmed Squares), which are explained in this chapter.

Then we will show how robust regression can automatically detect shadows and specularities separately without the need of any threshold.

3.1 Synthetic Data

In this chapter and following chapters we used a synthetic sphere to illustrate different steps of our algorithm. Hence this section explains the sphere data set in more details.

A synthetic sphere with radius 1 and equation $z = \sqrt{1 - x^2 - y^2}$ is illuminated under 50 different lighting directions. Figure 3.1 shows the positions of these lights on the hemispherical dome. We assumed that the surface is Lambertian; thus the shading factor for a pixel i under light j is:

$$s = n^i \cdot l_j \tag{3.1}$$

where $n = (x, y, z)$ is the surface normal and $l = (l_u, l_v, l_w)^T$ is the lighting direction.

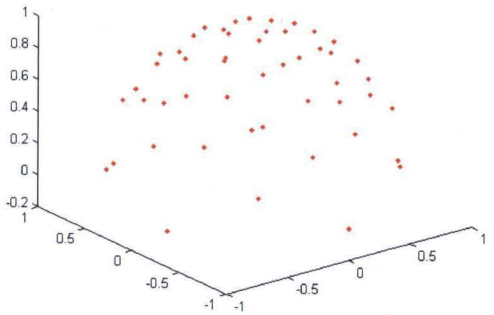


Figure 3.1: Lighting positions on a hemispherical dome, used for lighting the synthetic sphere; these directions are not arbitrarily chosen but in fact correspond to those experimentally measured for Figure 6.2 below.

Shading for all the pixels under light j will then looks like Eq. 3.2, where N is the surface normal matrix.

$$S = N \cdot l_j \tag{3.2}$$

The part of the sphere that cannot see the light is considered as shadow. In other words shadow is the part for which shading is equal to zero.

We can add color to the shading matrix and get a matte sphere. In order to add some highlights to the matte sphere, we used Phong illumination [22], with surface roughness power 20.

Figure 3.2 shows a synthetic sphere with Phong illumination, the matte image, surface normal, luminance and matte luminance. The lighting direction in these images is $l = (0.6396, -0.0046, 0.7687)^T$. In 3.2(c), the surface normal is shown in pseudocolor, with the x-component of the surface normal represented as red, the y-component as green, and the z-component as blue.

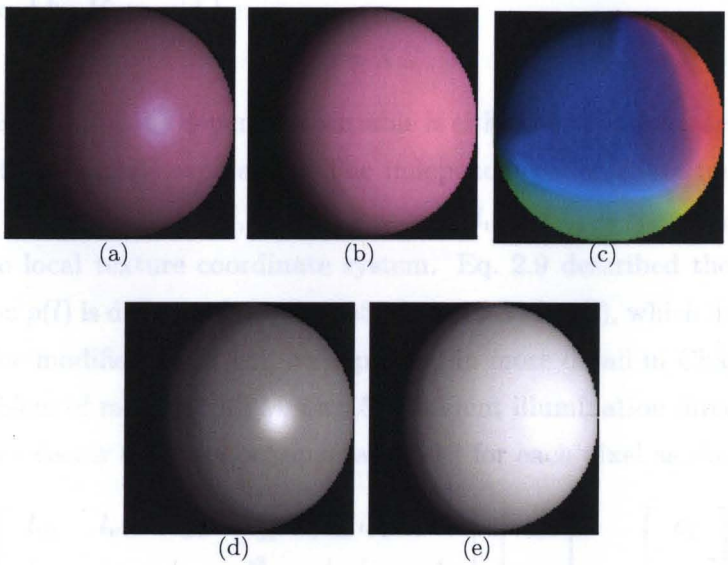


Figure 3.2: (a): Synthetic sphere with Phong illumination. (b): Matte version of the sphere. (c): Surface normals: x, y, and z components represented as R,G,B respectively. (d): Luminance of synthetic sphere with highlight. (e): Luminance of matte synthetic sphere.

3.2 Why Robust Regression

Regression analysis is used to fit an equation to n observed data values. Suppose a linear model as in Eq. 3.3:

$$y_i = c_1 x_{1i} + c_2 x_{2i} + ... + c_p x_{pi} \quad (i = 1..n) \quad (n \geq p) \tag{3.3}$$

Y is the dependent variable vector and X is the independent variable matrix. The goal of regression analysis is estimating the unknown parameter vector C. The estimated parameter vector shown as $\hat{C}$ called the regression coefficient vector. Thus the estimated or predicted value of Y, denoted by $\hat{Y}$, would be:

$$\hat{Y} = X\hat{C} \tag{3.4}$$

In the problem of PTM the dependent variable is either luminance as defined in Eq. 2.10, or each of the RGB channels separately. The independent vector is a function of lighting direction such as $p(l) = (l_u, l_v, l_w, l_u^2, l_u l_v, 1)$, where l_u, l_v and l_w is the projection of lighting direction into the local texture coordinate system. Eq. 2.9 described the light projection step. Our function $p(l)$ is different from that of original PTM $p_0(l)$, which has been described in section 2.2. The modified PTM will be explained in more detail in Chapter 4.

So in the problem of modified PTM with 50 different illumination directions, we should find the parameter vector C of the polynomial model for each pixel as shown in Eq. 3.5:

$$\begin{bmatrix} l_{u1} & l_{v1} & l_{w1} & l_{u1}^2 & l_{u1}l_{v1} & 1 \\ l_{u2} & l_{v2} & l_{w2} & l_{u2}^2 & l_{u2}l_{v2} & 1 \\ ... & ... & ... & ... & ... & ... \\ l_{u50} & l_{v50} & l_{w50} & l_{u50}^2 & l_{u50}l_{v50} & 1 \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ ... \\ c_6 \end{bmatrix} = \begin{bmatrix} e_1 \\ e_2 \\ ... \\ e_{50} \end{bmatrix} \tag{3.5}$$

The most popular and traditional regression technique is *least sum of squares (LS)*, in which the sum of squared residuals is minimized as shown in Eq. 3.6:

$$\min_{\hat{C}} \sum_{i=1}^{n} (r_i)^2 \tag{3.6}$$

The residual r_i of the ith observed variable is the difference between the actual observed value and the estimated value. Then solving Eq. 3.4, a LS estimation of the regression coefficient vector is given by:

$$\hat{C} = X^+ Y \tag{3.7}$$

where X^+ is the Moore-Penrose pseudoinverse of X.

Despite the popularity of least square regression, it is well known that it can be greatly affected by outliers, since its breakdown point is 0%. The breakdown point of an estimator is the proportion of outliers to the whole data that an estimator can handle before being affected. However [25] introduced regression techniques which are robust against outliers, with breakdown points of 50% such as, *least median of squares (LMS)* or *least trimmed squares (LTS).*

Figure 3.3 shows the result of applying robust and non-robust regression estimators on 50 data points. The robust regression technique is LMS. It can be seen that a non-robust regression estimator is affected by outliers, while robust regression produces a much better estimation of the underlying matte data and is less affected by shadows and specularities. In Figure 3.3, in fact, we reconstruct the correct underlying matte component of the Luminance essentially exactly. Note that in Figure 3.3 what is shown is regression results, both non-robust and robust, for data at a single pixel, indexed by 50 lighting positions. Here, we sort the data according to the overall Luminance observed.

For the input data itself (e.g., 50 different input images themselves), we shall indeed do very well at recovering the matte contribution; and for a new, non-measured lighting direction we expect to do well agaim although of course not as well as for the input data iteslf.

Any structure observed in the pattern of dots in Figure 3.3 is simply due to the arrangement of lights as in Figure 3.1.

The result of using LMS based modified PTM and LS based modified PTM on the synthetic sphere is shown in Figure 3.4. For visualization purposes we show the luminance and RGB version separately. The regeneration step of matte RGB will be explained in Section 4.3.1. There is a bright pink area in the recovered matte sphere using LS (Figure 3.4.b), and this shows that it was affected by outlier pixle values (highlights, in this case). As this figure depicts, it is obvious that using robust regression rather than non-robust regression will improve the recovered matte surface and recovered normal, where the latter is shown in pseudocolor with x, y, z mapped into R, G, B.

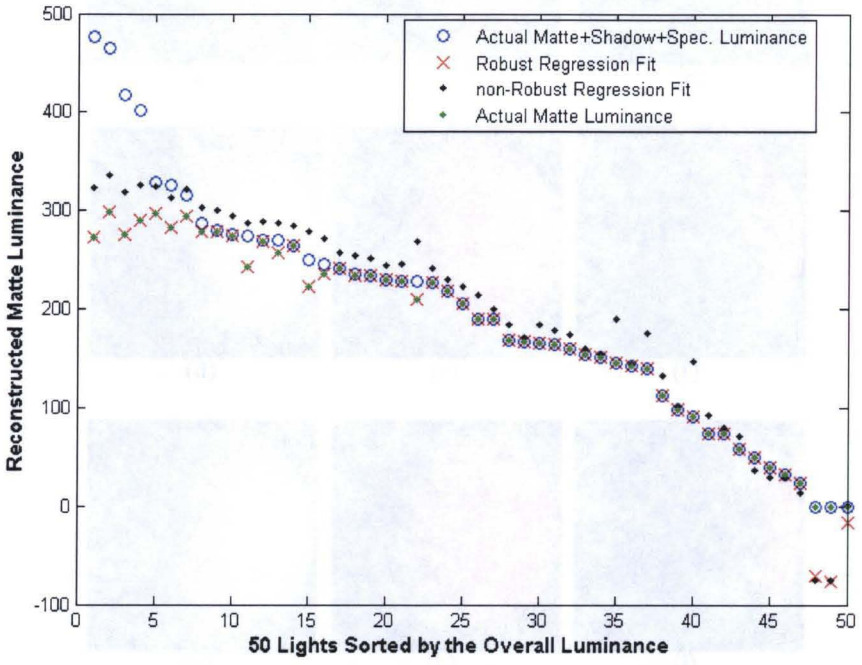


Figure 3.3: Robust regression versus non-robust regression for matte plus specular plus shadow data

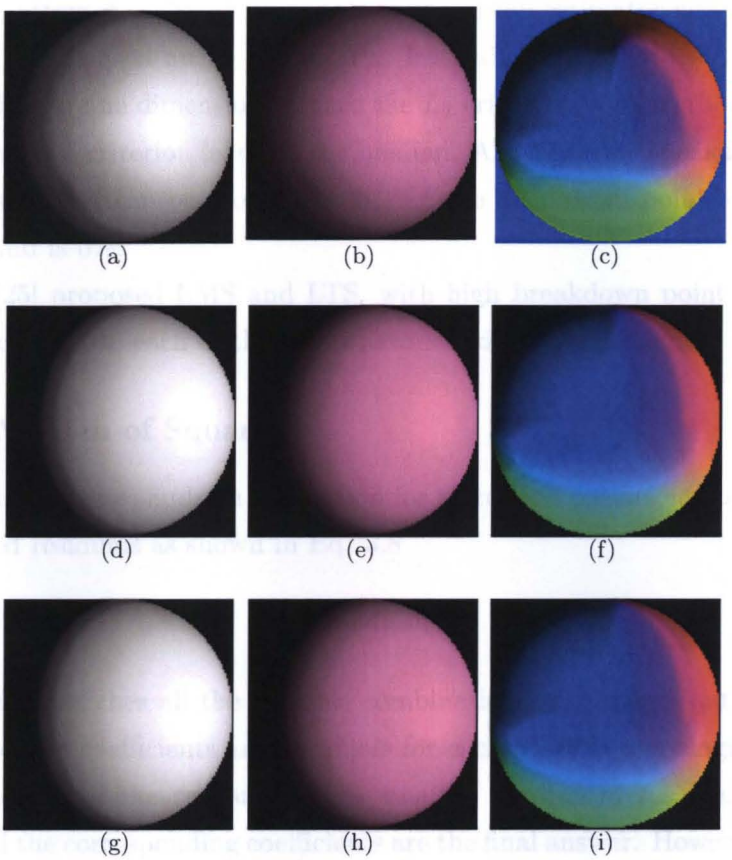


Figure 3.4: (a,b): Recovered luminance and RGB matte sphere using least squares. (c): Recovered surface normals using least squares: x, y, z are mapped into R, G, B. (d,e): Recovered luminance and RGB matte sphere using least median of squares. (f): Recovered surface normals using least median of squares. (g,h): Recovered luminance and RGB matte sphere using least trimmed squares. (i): Recovered surface normals using least trimmed squares.

3.3 Robust Regression Estimators

As it was mentioned earlier, robust regression estimators are those with breakdown point greater than 0%. The best possible breakdown point is 50%, since for larger amount of contamination the bad part will be considered as the good part [23].

The very first attempt toward a more robust regression estimator was proposed in 1887 by Edgeworth [11]. He used an L_1 criterion or least absolute value instead of L_2 or least square value. While in one dimensional space the L_2 criterion leads to the arithmetic mean of observations, the L_1 criterion leads to the median. Although the breakdown point of the sample median is 50%, it can be shown that in fact the breakdown point of L_1 regression is the same as LS and is 0%.

Rousseeuw [25] proposed LMS and LTS, with high breakdown point (50%). The following subsections explain each of these estimators and compare them.

3.3.1 Least Median of Squares

LMS, as the name suggests, finds an estimation for regression coefficients by minimizing the median of squared residuals as shown in Eq. 3.8

$$\min_{\hat{C}} \; median_i \; r_i^2 \tag{3.8}$$

The *exact* LMS searches all the possible combinations of p points out of the n points, calculates least square coefficients and residuals for each possible non-singular combination and finds the median of the residuals. The smallest median of residuals of all possible combinations and the corresponding coefficients are the final answer. However this method is very inefficient and time consuming, since the number of all possible combinations increases so fast, even with a small increase in number of parameters (p) or number of observed data (n). For example in our case $n = 50$ and $p = 6$ which will result in $\begin{pmatrix} 50 \\ 6 \end{pmatrix} = 15890700$ different combinations.

In order to implement the LMS algorithm, researchers usually considered only m sub-samples. The number of sub-samples should be such that the probability of at least one good sub-sample will be one. A good sub-sample consists of p good observations of the sample. Rousseeuw and Leroy [25] provide possible m values for different scenarios. In our problem $m = 3000$ is used.

A good point about LMS is its outlier detection. According to [25] a point is an inlier with weight one ($w = 1$) if it satisfies Eq. 3.9, otherwise it is an outlier and $w = 0$:

$$|\frac{r}{s_0}| <= 2.5 \tag{3.9}$$

where s_0 is the initial scale estimate which is given by:

$$s_0 = 1.4826 \times (1 + \frac{5}{(n-p)}) \times \sqrt{(min\ of\ med\ of\ r_i^2)} \tag{3.10}$$

Finally, a reweighted Least Squares regression is carried out to get a final scale and final weights, using only the accepted points with weights 1 as demonstrated in Eq. 3.11.

$$scale = \sqrt{\frac{\sum_{i=1} n(r_i^2 w_i)}{(\sum_{i=1} n w_i) - p}} \tag{3.11}$$

The final weights are obtained by replacing s_0 by *scale* in Eq. 3.9.

3.3.2 Least Trimmed Squares

The least trimmed squares estimator (LTS) is another robust regression estimator which was proposed by [23]. The objective of the LTS is:

$$\min_{\hat{C}} \sum_{i=1}^{h} (r^2)_{i:n} \tag{3.12}$$

where $(r^2)_{1:n} \leq (r^2)_{2:n} \leq ... \leq (r^2)_{n:n}$ are the ordered squared residuals. Eq. 3.12 is equivalent to finding an h-subset with smallest least sum of squares [24], where h is the coverage value and may be set between $\frac{n}{2}$ and n. Note that the calculation of LTS requires sorting the squared residuals which takes at least $O(n \log n)$ operations, compared to $O(n)$ operations for median in LMS. Therefore the overall LTS regression algorithm is more expensive than LMS. Ref. [25] showed that the best robustness properties are achieved when h is approximately $\frac{n}{2}$.

Since the exhaustive search for finding the h-subset with smallest least sum of squares is very time consuming, [24] proposed an approximation algorithm for LTS. It randomly draws 500 h-subsets H_{old} and carries out a "two C-step". The "two C-step" is as follow:

- compute c_{old} based on H_{old} using LS regression estimator
- compute $r'_{i:n}$ for all n observations

- compute $Q_{old} = \sum_{i \in H_{old}} (r'_{i:n})^2$
- sort the absolute value of the residuals
- take the permutation for h smallest residuals and call it H_{new}
- compute c_{new} based on H_{new} using LS regression estimator
- compute $r_{i:n}$ for $i \in H_{new}$
- compute $Q_{new} = \sum_{i \in H_{new}} (r_{i:n})^2$

For the 10 lowest Q carry out C-steps until convergence, when $Q_{new} = Q_{old}$. The lowest corresponding Q will give us coefficients (c). Once the final coefficients have been calculated, weights are obtained by the equations in the previous sub-section.

3.3.3 Comparison

We tried both LMS-based modified PTM and LTS-based modified PTM on the synthetic sphere. The qualitative results, are shown by Figure 3.4. However we preferred to use LMS, since LTS was much more time consuming due to sorting the residuals.

Table 3.1 presents the two estimators based on three quantitative metrics. It can be seen that both LMS and LTS are more or less the same. However LMS is more accurate in predicting surface normals and surface albedo. Section 4 describes the method for recovering surface normals and albedo.

In this thesis we used three quantitative metrics: *peak signal to noise ratio* (PSNR), *median normal angle error* (MNAE) and *median albedo percentage error* (MAE). Each of them is described as follows.

If image I is a monochrome $m \times n$ image, assumed to be the original image, and image J is the reconstructed one, then the PSNR of image J is:

$$PSNR = 20 \log_{10} \frac{max(I)}{\frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (I(i,j) - J(i,j))^2} \tag{3.13}$$

Another metric is MNAE, which is the deviation of the reconstructed surface normal angle from the actual surface normal angle. It computes the angle for each pixel position and the median of these angles is our MNAE metric.

MAE finds the albedo error at each pixel position as shown in Eq. 3.14:

$$MAE = 100 \times \frac{(actual\ albedo - recovered\ albedo)}{actual\ albedo} \tag{3.14}$$

	LTS-based modified PTM	LMS-based modified PTM
PSNR	22.23	22.14
MNAE	1.21×10^{-6}	8.54×10^{-7}
MAE	1.36×10^{-5}	6×10^{-7}

Table 3.1: PSNR, MNAE and MAE metrics for LMS-based modified PTM and LTS-based modified PTM

Note that the PSNR in this table is between original and approximated luminance sphere.

3.4 Highlights and Shadows

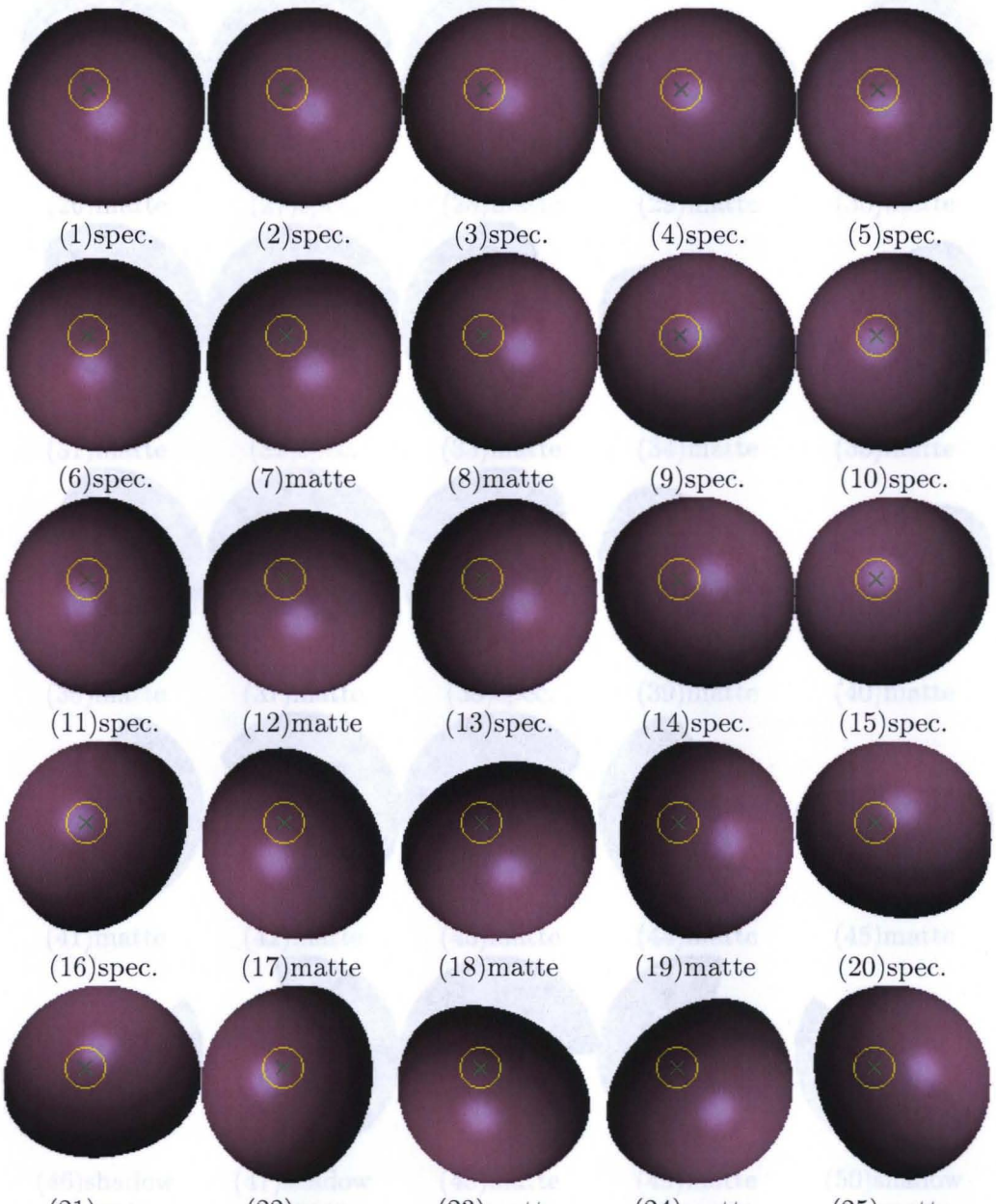
As explained in Section 3.3.1, outliers detected by LMS are the points with weights equal to zero. However in our problem domain, an outlier can be either shadows or highlights. However, we know that shadows are darker than matte, and specularities are brighter than matte. Thus if a pixel has been detected as an outlier and its residual is positive, which means that actual luminance is greater than approximated luminance, then it is labeled as specularity. Similarly an outlier pixel with negative residual is labeled as shadow. The inlier pixels with weights equal to one are labeled as matte.

Thus we modified the weights into a tripartite set with labels $\{w^-, w^0, w^+\}$, where w^- indicates shadow pixel, w^0 indicate matte pixel and w^+ indicate specularity.

Figure 3.5 shows one particular pixel position of a synthetic sphere under 50 different lighting directions. Even though we are interested in exactly the same pixel under all the 50 lights, we nevertheless indicate the pixel using a small yellow circle for ease of understanding. For visualization purposes, we show the color sphere as synthesized, but the parts that are in shadow have been shown in white rather then the black actually calculated.

The idea in this Figure is to display how the robust regression categorizes the single pixel we are interested in, as the light changes: is it identified as being matte, specular, or shadow? As it can be seen, the robust PTM model correctly found shadows and specularity in most of the cases. However in some cases, sphere 32 for instance, the pixel has been marked as specularity while it is not a specularity, but is bright enough to be considered

as specularity. It is important to note that the residual of this pixel (0.83) is much smaller than the residual of pixel marked as specularity in sphere 15 which is 205.48.


(1)spec.
(2)spec.
(3)spec.
(4)spec.
(5)spec.
(6)spec.
(7)matte
(8)matte
(9)spec.
(10)spec.
(11)spec.
(12)matte
(13)spec.
(14)spec.
(15)spec.
(16)spec.
(17)matte
(18)matte
(19)matte
(20)spec.
(21)spec.
(22)spec.
(23)matte
(24)matte
(25)matte

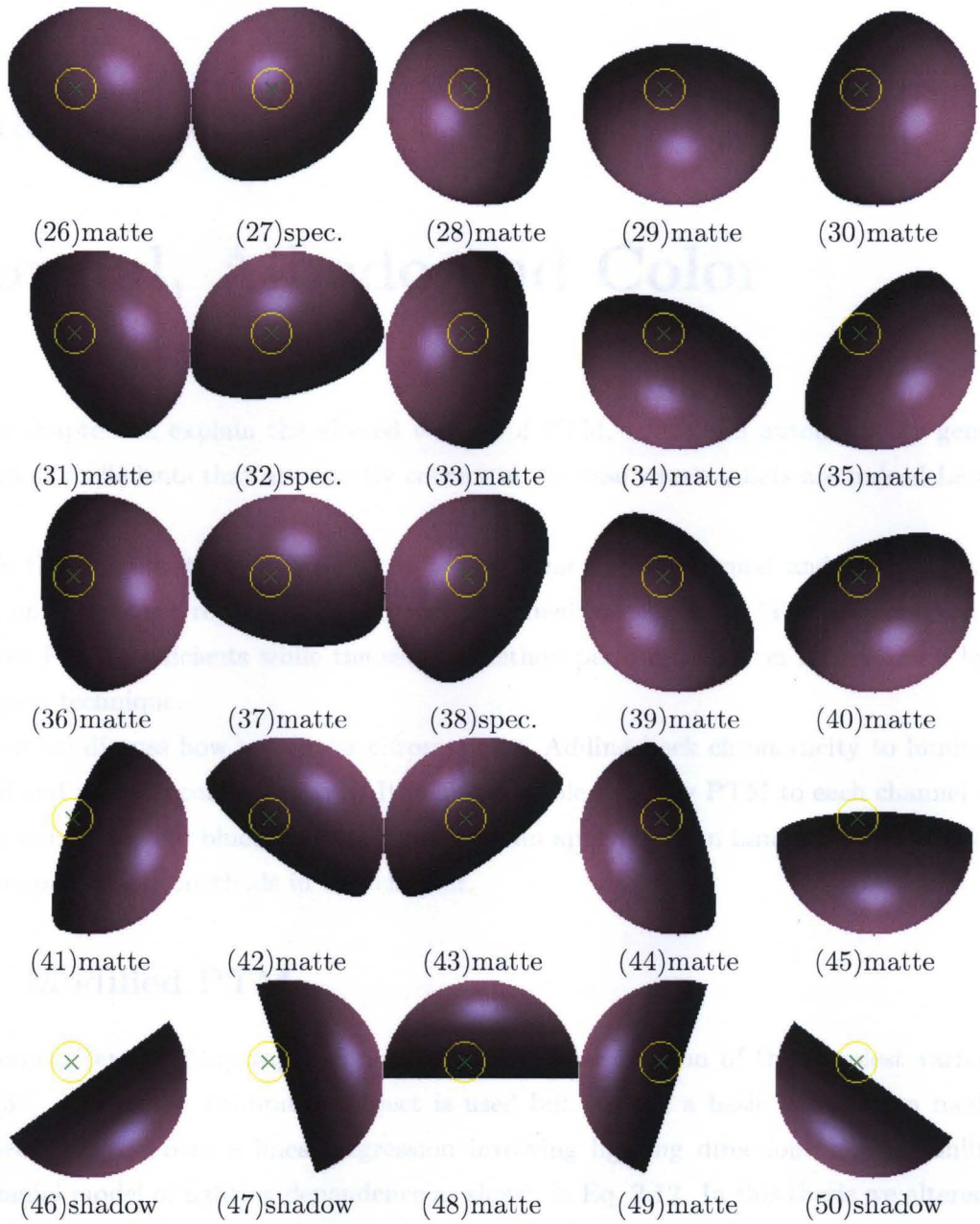


Figure 3.5: Pixel position $(53, 53)$ of synthetic sphere under different 50 lighting directions.

Chapter 4

Normal, Albedo and Color

In this chapter we explain the altered version of PTM, which can automatically generate regression coefficients that are exactly correct in the case where inliers are indeed Lambertian.

We then demonstrate two different methods for surface normal and albedo recovery based on the robust regression which was explained in chapter 3. The first method uses only the PTM coefficients while the second method performs PST on inliers found by the regression technique.

Next we discuss how to recover chromaticity. Adding back chromaticity to luminance, we will end up with colored images. It is also possible to apply PTM to each channel of an image, red, green and blue, separately, rather than applying it on luminance. We explained and compared both methods in this chapter.

4.1 Modified PTM

Polynomial Texture Mapping is formulated as a generalization of the simplest variant of PST [31], wherein no calibration object is used but instead a basic Lambertian model is assumed. PTM moves a linear regression involving lighting directions into a nonlinear, polynomial model of lighting dependence as shown in Eq. 2.12. In this thesis we altered the polynomial used in PTM and replacing it with Eq. 4.1 [10].

$$p(l) = (l_u, l_v, l_w, l_u^2, l_{uv}, 1)\,, \quad \text{where } l_w = \sqrt{1 - l_u^2 - l_v^2} \tag{4.1}$$

The rationale is as follows: Suppose we indeed have a Lambertian surface; since we

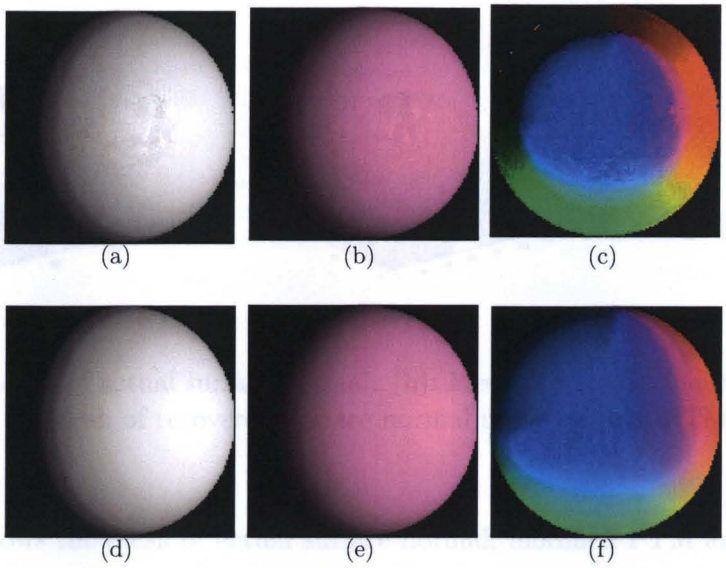


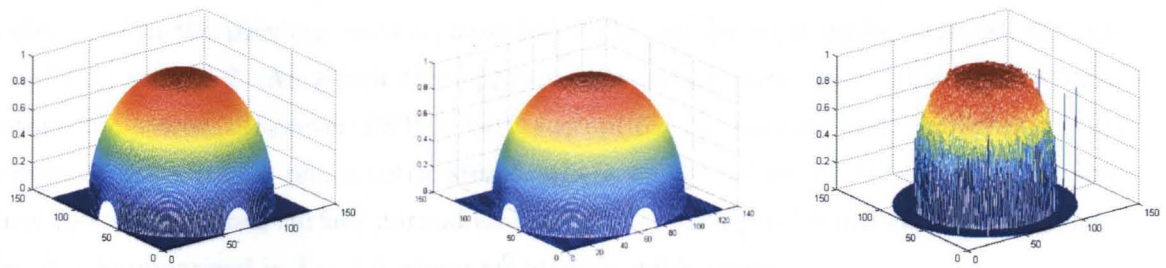
Figure 4.1: (a,b): Recovered luminance and matte sphere using original PTM. (c): Recovered Surface normals using original PTM. (d,e): Recovered luminance and matte sphere using modified PTM. (f): Recovered Surface normals using modified PTM.

mean to use a robust regression to solve for the regression coefficients c, then regardless of specularities or shadows, the regression will just generate the correct surface normal vector, multiplied by a surface albedo times a lighting strength factor. The regression will place zeros in c for the higher order terms. Nevertheless, it is useful to keep a polynomial description, to suit surfaces which are not Lambertian. Note that if we were to use instead $p(l) = (l_u, l_v, l_w)$ the method would simply reduce to PST. Here we regress making no assumption about a Lambertian character of the surface, and can reconstruct pixel values without making any such assumption. The reason for 6-D in Eq. (4.1) is simply to retain the same low dimensionality, rather than the 10-D possible for quadratic combinations of $\{l_u, l_v, l_w, 1\}$. (Also, 7-D, with a l_v^2 term in (4.1), performed about the same.)

Figure 4.1 depicts the difference between original PTM and modified PTM on the synthetic sphere — the new PTM polynomial performs much better.

Thus the regression involving Eq. 4.1 automatically generates an estimate of surface normal n and albedo α, by considering the contribution to the first three terms of $p(l)$:

$$\tilde{n} = \{c_1, c_2, c_3\}; \quad \alpha = \|\tilde{n}\|, \quad n = \tilde{n}/\alpha \tag{4.2}$$



(a) (b) (c)

Figure 4.2: (a): Mesh of actual surface normal. (b): Mesh of recovered surface normal using modified PTM. (c): Mesh of recovered surface normal using original PTM.

Figure 4.2 shows the mesh of actual surface normal, modified PTM and original PTM. The surface normal recovery of original PTM has been explained in [17]. For a diffuse surface, being photographed under varying lighting directions, the surface normal per pixel can be estimated by maximum luminance of 4.3.

$$e = c_1 l_u^2 + c_2 l_v^2 + c_3 l_u l_v + c_4 l_u + c_5 l_v + c_6 \tag{4.3}$$

Setting $\frac{\partial e}{\partial u} = \frac{\partial e}{\partial v} = 0$, we arrive at [17]:

$$\tilde{l_u} = \frac{c_3 c_5 - 2c_2 c_4}{4c_1 c_2 - c_3^2} \tag{4.4}$$

$$\tilde{l_v} = \frac{c_3 c_4 - 2c_1 c_5}{4c_1 c_2 - c_3^2} \tag{4.5}$$

and the surface normal will be $n = (\tilde{l_u}, \tilde{l_v}, \sqrt{1 - \tilde{l_u}^2 - \tilde{l_v}^2})$.

It can be seen that using our modification of the standard PTM improves the qualitative results as well as the quantitative results. Table 4.1 compares a robust version of the original PTM and our modified robust PTM comparing PSNR and MNAE.

	Robust Original PTM	Robust Modified PTM
PSNR	20.86	22.14
MNAE	10.64	8.54×10^{-7}

Table 4.1: Comparison table between original PTM and modified PTM

From now on by the term PTM, we mean our modified PTM unless stated otherwise.

4.2 Robust Photometric Stereo

As discussed in the previous section, modified PTM can generate surface normal and surface albedo using 4.2. An alternative approach for surface normal and albedo recovery is photometric stereo. However PST is very susceptible to shadows and highlights. Thus we propose using robust photometric stereo, where we use robust PTM regression weights. Hence robust PST finds surface normal and albedo, for each pixel position, based on inliers only. It is summarized in Eq. 4.6 where w^0 identify inlier pixels.

$$\tilde{n} = \left(\mathrm{L}(w^0)\right)^+ E(w^0); \quad \alpha = \|\tilde{n}\|, \quad n = \tilde{n}/\alpha \tag{4.6}$$

Figure 4.3 shows surface normal and albedo for robust PTM, robust PST and standard PST over increasing percent Gaussian noise, for the synthetic sphere plus noise. Here the solid lines show the results for error in normal vector direction (blue) and albedo (red). Unsurprisingly, if there is indeed no noise, and the base reflectance is Lambertian, the robust regression on robust PTM ignores the polynomial terms in the regression model and returns regression coefficients proportional to the surface normal since the shading model is normal dotted into light direction, and specularities and shadows do not distract the regression. The lowest-error estimations for surface normal and albedo in Figure 4.3 belongs to robust PST, which is shown as dot-dashed lines. This is because the extra polynomial terms in PTM will tend to over-fit the correct underlying noise-free values, whereas robust PST assumes a Lambertian model, correctly in this synthetic case. In comparison, standard PST, based on straightforward least squares including all pixel values, has poorer estimates – dashed lines in Figure 4.3 – because the effect of noise is swamped by the main problem, inclusion of shadow and specular values.

For real images, we therefore apply the robust PST, with matte weights generated by PTM, as our estimator of albedo and normal.

4.3 Color

The robust regression and normal recovery techniques we have discussed so far worked on luminance. We call this method Lum-PTM. In this thesis we have defined the luminance at each pixel position is simply the sum of red, green and blue channel at that pixel. In order to add back color to the results we need to somehow recover chromaticity (the color information

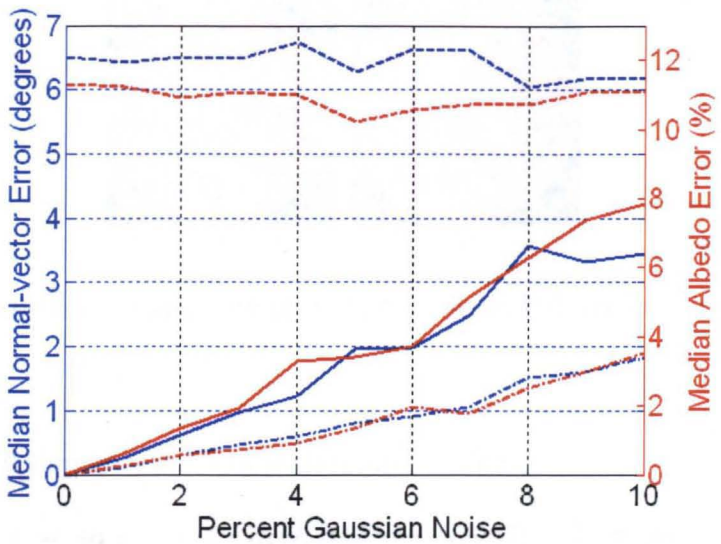


Figure 4.3: Solid lines: Noise sensitivity for robust PTM estimate of surface normal and albedo. Dot-dashed lines: robust PST – using weights w^0 generated by robust PTM. Dashed lines: standard PST.

not including magnitude). However we could take a different approach to deal with color. We can perform robust regression and normal recovery techniques on each colored channel separately. This method is called RGB-PTM.

Lum-PTM is less time consuming and needs less memory than RGB-PTM, and they each generate almost the same qualitative and quantitative results. In the next section these two methods are discussed.

4.3.1 Lum-PTM

This method, as its name suggests, regresses on luminance. At this point, we already have an advantage of applying a robust method to the problem at hand, *viz.* a more reliable calculation of coefficient c. But in fact we also have produced a better grasp of color, as well. Let us factor each RGB triple ρ into luminance $L = R + G + B$ times chromaticity χ ($\rho = L\chi$). Luminance will be composed of a scalar albedo α times lighting strength times shading factor s; since we have no way of disentangling lighting intensity from surface

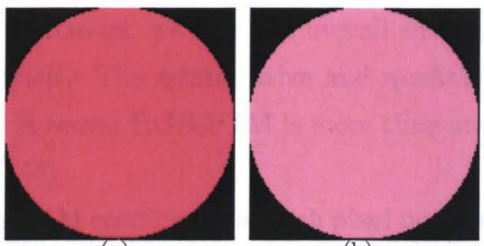


Figure 4.4: (a): Recovered chromaticity of synthetic sphere. (b): Intrinsic image of synthetic sphere.

reflectance, we shall simply lump both scalars into α. Thus,

$$\rho = s\alpha\chi, \quad \chi \equiv \{R,G,B\}/(R+G+B), \quad L = s\alpha \tag{4.7}$$

An intrinsic image [3] (for this lighting strength) — by intrinsic image we mean surface independent of shading — would then be

$$\rho_{intrinsic} = \alpha\chi \tag{4.8}$$

I.e., this is what the surface would look like under this light, with shading removed.

Since we have weights w^0 identifying pixels indeed corresponding to the PTM model, the robust regression delivers a reliable estimate of chromaticity. For suppose that a set of all chromaticities χ is given by $\chi_k(i) = \rho_k(i)/L(i)$, $k = 1..3$, for the ith light, $i = 1..n$ (with, say $n = 50$ lights), then we can identify a good estimate for the chromaticity independent of light direction via

$$\chi = median\left(\rho(w^0)/E(w^0)\right) \tag{4.9}$$

Figure 4.4 shows the chromaticity and intrinsic image of the synthetic sphere. Note that the estimated chromaticity we defined here is independent of illumination direction and does not contain any shading information — the color without magnitude χ shows correctly as a constant disk — and also the intrinsic image is correctly a constant disk.

4.3.2 RGB-PTM

The basic idea behind this method is performing robust regression on each color channel separately. Thus we will have three sets of results for red, green and blue. The results

contain: approximated red, green and blue, weights and PTM coefficients. Since there are surface normals for each channel, we take the overall surface normal to be the average of normals over three channels. The quantitative and qualitative results show the same accuracy as for Lum-PTM. However RGB-PTM is more time and space consuming (almost three times that of Lum-PTM).

Lum-PTM stores only 6 PTM coefficients at each pixel position plus any pair of the chromaticities, for instance red and blue chromaticity, since the other value, green chromaticity here, can be achieved by:

$$\chi_2 = 1 - (\chi_1 + \chi_3) \tag{4.10}$$

while RGB-PTM stores 6 PTM-coefficients for each color channel at each pixel position, which is 18 coefficients in total.

In fact we found that Lum-PTM and RGB-PTM showed nearly the same performance; since Lum-PTM is more efficient, we use Lum-PTM throughout this thesis.

Chapter 5

Shadow and Specularity

The model we have introduced so far will not account completely for the luminance including highlights and shadows, but only a basic matte reflectance, the result of the robust regression. Since we have in hand labels for specular w^+ and shadow w^- pixels (over lights $i = 1..n$ at each each x, y location), we can *model* these extra contributions, with a view to being able to interpolate them later, for new, unmeasured, lighting conditions.

In this section we will explain how to model shadow and specular contribution using *radial basis functions* and then utilize the model parameters for interpolating new lighting directions.

5.1 Radial Basis Function

Radial Basis Functions (RBF) are a nonparametric regression method used to interpolate scattered data, and does not require that data lie on any regular grid. RBF is based on linear combinations of a radial function, such as a Gaussian or Thin Plate Spline, centered at the origin so that $\phi(x) = \phi(\|x\|)$ or on the distance from some center point c so that $\phi(x, c) = \phi(\|x - c\|)$. Suppose we would like to interpolate a function with n observed points. Then we can use n radial basis functions centered at these points and the resulting interpolated function becomes:

$$f(x) = \sum_{i=1}^{n} \gamma_i \phi(\|x - x_i\|) \tag{5.1}$$

where x_i is the position of ith known point and γ_i is the weight of the radial basis function positioned at that point. In some cases a first-degree polynomial $P(x)$ is added to account for the linear and constant portions of f and ensure positive definiteness of the solution [19]. Suppose x is a three dimensional data point; then $P(x) = \alpha + \beta^1 x^1 + \beta^2 x^2 + \beta^3 x^3$ where β are the polynomial coefficients:

$$f(x) = \alpha + \beta^1 x^1 + \beta^2 x^2 + \beta^3 x^3 + \sum_{i=1}^{n} \gamma_i \phi(\|x - x_i\|) \tag{5.2}$$

Since the system defined by Eq. 5.2 is under-determined, one scalar and one vector are usually introduced into the system.

$$\sum_{i=1}^{n} \gamma_i = 0, \quad \sum_{i=1}^{n} \gamma_i x_i^k = 0, \quad k = 1, 2, 3 \tag{5.3}$$

5.2 Modeling Shadow and Specularity

We utilize RBF to model specularity and shadow, separately, since we may wish to interpolate these quantities plus a matte contribution separately. In principle we could also use RBF on the luminance directly, but each component that makes up the luminance — i.e. matte, specularity, and shadow — is of interest in its own right, for such tasks as matte relighting [17] or shape-from-specularity [20]. Moreover, since we are aware that the matte component is smooth, whereas the specular and shadow contributions are much more high-frequency, we risk over-fitting if we apply RBF as part of generating the matte luminance, whereas RBF can indeed handle the abrupt changes of the other two components.

In order to model specularity at each pixel, we first consider the extra value, over and above the approximated luminance vector $\hat{E}$ at each pixel, due to specularity. Let us call this highlight-driven value the "sheen", ς , defined as

$$\varsigma(w^+) \;=\; E(w^+) \;-\; \hat{E}(w^+), \quad \varsigma(\neg w^+) \equiv 0 \tag{5.4}$$

where approximated luminance $\hat{E}$ is the result of robust regression and contains only the matte contribution.

Then we can model the dependence of specularity on lighting direction using a set of (RBF) coefficients. I.e., supposing that the sheen is given in terms of 3-vector light direction

l as [16]

$$\hat{\varsigma}(l) \;=\; \alpha + \beta^1 l_u + \beta^2 l_v + \beta^3 l_w + \sum_{i=1}^{n} \gamma_i \phi(\|l - l_i\|) \tag{5.5}$$

In the case of our synthetic sphere n is 50 lights, at each pixel. Then Eq. 5.5 can be rewritten in matrix form as:

$$\begin{bmatrix} \phi_{11} & \phi_{12} & \dots & \phi_{1n} & l_{u1} & l_{v1} & l_{w1} & 1 \\ \phi_{21} & \phi_{22} & \dots & \phi_{2n} & l_{u2} & l_{v2} & l_{w2} & 1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \phi_{n1} & \phi_{n2} & \dots & \phi_{nn} & l_{un} & l_{vn} & l_{wn} & 1 \\ l_{u1} & l_{u2} & \dots & l_{un} & 0 & 0 & 0 & 0 \\ l_{v1} & l_{v2} & \dots & l_{vn} & 0 & 0 & 0 & 0 \\ l_{w1} & l_{w2} & \dots & l_{wn} & 0 & 0 & 0 & 0 \\ 1 & 1 & \dots & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \dots \\ \gamma_n \\ \beta^1 \\ \beta^2 \\ \beta^3 \\ \alpha \end{bmatrix} = \begin{bmatrix} \varsigma_1 \\ \varsigma_2 \\ \dots \\ \varsigma_n \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{5.6}$$

with nodes l_i and Gaussian radial base functions ϕ, where $\phi_{ij} = \phi(\|l_i - l_j\|)$. We solve for the $(n+4)$ coefficient values {scalar α, 3-vector β, n-vector γ}. For the Gaussian width s we take the approximate average distance between interpolation nodes [16]:

$$s = \left\{ \frac{1}{n} \Pi_{k=1}^{3} \left[\max_{i=1..n}(l_i^k) - \min_{i=1..n}(l_i^k) \right] \right\}^{1/3} \tag{5.7}$$

where l_i^1 is l_{ui}, l_i^2 is l_{vi} and l_i^3 is l_{wi}. I.e., s is $(1/n)^{1/3}$ times the geometric mean of the bounding-box widths enclosing the lights.

Interpolation nodes are mapped exactly, but the interpolated function might model the underlying dependency well or not, depending on the amount of smoothing applied. Here we use no smoothing, but this may be re-considered in future.

To model shadowing and any other contribution, all the remaining non-matte contribution the "shade", denoted σ:

$$\sigma(\neg w^+) \;=\; \hat{E}(\neg w^+) \;-\; E(\neg w^+), \qquad \sigma(w^+) \equiv 0 \tag{5.8}$$

Again, we model this contribution using a different set of RBF coefficients, separately. The reason for including all non-sheen pixels in the shade σ, and not just w^-, is that then the two RBF interpolations, plus matte, combine to exactly equal the input luminance images.

The model that describes luminance E' for any new lighting direction l, at each pixel position is then shown in Eq. 5.9.

$$E'(l) \;=\; \hat{E}(l) + \hat{\varsigma}(l) \;-\; \hat{\sigma}(l) \tag{5.9}$$

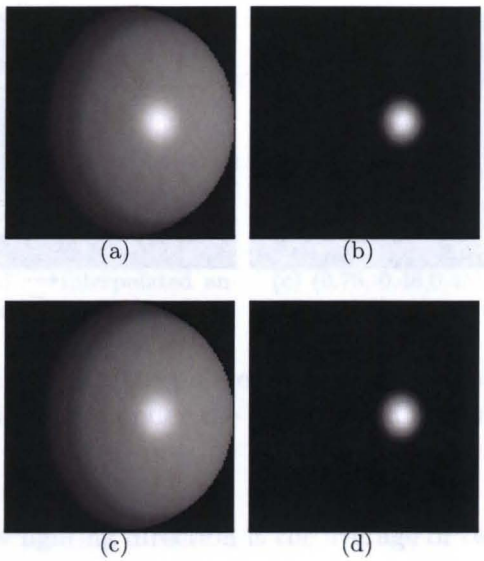


Figure 5.1: (a): Actual luminance image. (b): Actual highlight. (c): Regenerated luminance image. (d): Regenerated highlight.

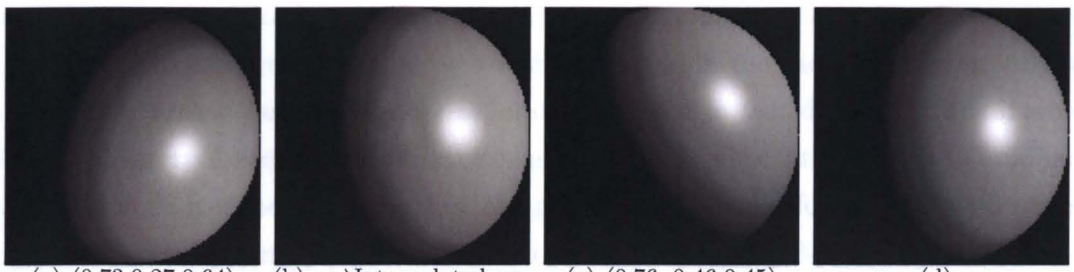
Again, note that the input luminance images themselves are essentially exactly regenerated, using this model. Figure 5.1 shows the original and regenerated input luminance and specularity images of a synthetic sphere for light direction $l = (0.6396, -0.0046, 0.7687)^T$. The PSNR of the regenerated luminance sphere compared to the actual luminance sphere is 47.82dB for this lighting direction.

5.3 Interpolating Shadow and Specularity

With shade and sheen RBF model parameters in hand, i.e. scalar α, 3-vector β, n-vector γ, we can interpolate shadow and specularity for any new light $\mathcal{L}$. We simply put the new lighting direction into Eq. 5.5 for shade and sheen separately.

We can use either PTM coefficients ($\hat{e} = p(l) \cdot c$) or robust PST ($\hat{e} = n \cdot l^T$) as the base contribution. Note that PTM approximated luminance $\hat{e}$ may include some measure of shadow or specular contribution, since in fact we have used a polynomial approximation, not simply a Lambertian one.

Figure 5.2 shows the interpolation result of a synthetic sphere under a new, unmeasured



(a) (0.73,0.27,0.64) (b) ⟶Interpolated angle ⟵ (c) (0.76,-0.46,0.45) (d)

Figure 5.2: (a,c): Two inputs. (b): Interpolated for a light between (a) and (c). (d): Actual appearance of the synthetic sphere under the interpolated lighting direction.

lighting direction. The new lighting direction is the average of two measured lighting directions. Figure 5.2(d) shows how the sphere should look like under the new lighting direction. The PSNR of the interpolated sphere is 41.26dB, which shows that our method works very well indeed.

5.4 Adding Back Color

Since we have performed regression on the luminance only, the output generated so far is the regressed luminance. However we would like to render full color RGB output and the naive approach to do so is using RGB-PTM and then performing RBF on each color channel separately. This approach is relatively time and space consuming which is not desirable. In this section we will discuss two simpler approaches to add back color. The first one is based on chromaticity information, and the second method models shade and sheen in each color channel separately.

5.4.1 Chromaticity based Method

We already have a good estimation of chromaticity, using Eq. 4.9, and the regressed luminance, using Eq. 5.9, at each pixel position. Thus one can simply say color value at each pixel position is chromaticity times luminance. However we know that the chromaticity in a highlight area is different than the estimated chromaticity, and the result of the above approach would not be accurate. Figure 5.3(b) demonstrates how such a result will look.

The highlight area displays as pink, which is not desirable, since for dielectric materials we expect the highlight color to approximate that for the illuminant [15].

This problem can be solved if we can add back color in the sheen area separately, and since we have broken out the sheen ς separately, we can in fact do that. Thus if we can estimate the interpolated basic luminance $\hat{e}$, and also the shade contribution σ to luminance, which we assume has the same color, we can generate RGB color for these contributions to luminance by simply multiplying luminance by the chromaticity 3-vector χ. Note that we already know the luminance for matte and shade.

For the sheen contribution, so far we have the scalar luminance ς. We can also obtain the specular chromaticity χ_{spec} by identifying it with the chromaticity of the maximum-luminance input value over all pixels — here we are using the neutral-interface model of dielectrics, which states that specular color equals light color [15]. Then we generate an x, y-dependent difference of chromaticity, $\Delta\chi$, equal to the chromaticity for a pixel from Eq. (4.9) subtracted from χ_{spec}. At pixels with specular contribution, we expect the matte chromaticity to be replaced by that for a highlight, so we add the sheen ς times $\Delta\chi$ to the interpolated color. The following equation summarizes this method at each pixel position:

$$\rho'_k(l) = (\hat{E}(l) + \hat{\varsigma}(l) - \hat{\sigma}(l)) \times \chi_k + \hat{\varsigma}(l) \times (\chi_{spec} - \chi_k) \tag{5.10}$$

where $k = 1, 2, 3$ corresponds to three color channels and l is the new, interpolated, lighting direction.

As Figure 5.3 depicts, this method works quite well. The PSNR value between the input (Figure 5.3(a)) and regenerated RGB image (Figure 5.3(c)) is 42.37dB for this synthetic example.

5.4.2 Three-Shade and -Sheen Method

Although the chromaticity based method works well enough, it may produce some artefacts where robust regression generates near-zero slopes. Note that this will happen in locations where a majority of the pixel values are very dark, and thus LMS regression returns slopes that are almost zero and hence the matte contribution is very dark. In such a case the interpolated color would be basically that of the sheen, and is therefore greyish.

However we determined that modeling shade and sheen for each color channel and then applying RBF to each of them can solve the problem. The reason is that now we are

actually regressing color and not only luminance and therefore we can achieve a better fit. The following equations demonstrate this method, where $k = 1, 2, 3$ corresponds to the three color channels, ρ_k is the value of the kth color channel and $\hat{E}$ and χ are approximated luminance for matte contribution and chromaticity respectively. Note that in this method, as in the previous method, we model sheen in the highlight area and shade everywhere except highlight area:

$$\varsigma'_k(w^+) = \rho_k(w^+) - \hat{E}(w^+) \times \chi_k, \quad \varsigma'_k(\neg w^+) \equiv 0 \tag{5.11}$$

$$\sigma'_k(\neg w^+) = \hat{E}(\neg w^+) \times \chi_k - \rho_k(\neg w^+), \qquad \sigma'_k(w^+) \equiv 0 \tag{5.12}$$

We then model each of shade and sheen using a different set of RBF coefficients, and finally color is the combination of matte contribution, shade and sheen for each color channel as shown in Eq. 5.13:

$$\hat{\rho_k}(l) = \hat{E}(l) \times \chi_k + \hat{\varsigma'_k}(l) - \hat{\sigma'_k}(l) \tag{5.13}$$

Note that the matte contribution for each color channel is chromaticity χ times approximated luminance $\hat{E}$.

In other words we performed Lum-PTM to get luminance in the matte area ($\hat{E}$) as well as chromaticity (χ). We then applied RBF to each of the three ς'_k and σ'_k to model sheen and shade for each color channel and finally we used Eq. 5.13 to get full color output.

Experimental results show an improvement in qualitative results (compare Figure 5.3(c) and (d)) as well as the quantitative results. The PSNR is 48.67 between the input (Figure 5.3(a)) and regenerated RGB image (Figure 5.3(d)), a large improvement.

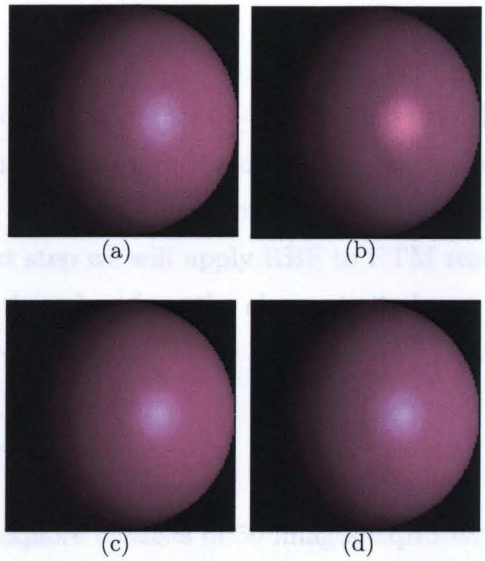


Figure 5.3: (a): Actual colored sphere. (b): Regenerated colored sphere using chromaticity times luminance. (c): Regenerated colored sphere using specular chromaticity. (d): Regenerating colored sphere, regressing three different channels.

Chapter 6

Experimental Results

In this chapter we will present experimental results for seven different datasets, captured using a hemispherical dome as shown in Figure 2.2, in two main steps. First we will apply *Modified Robust PTM* to the data and recover the matte component, chromaticity and surface normal. In the next step we will apply RBF to PTM results to interpolate shadows and specularities. We will then show how the *chromaticity based* and *three-shade and -sheen* methods will perform on real data.

6.1 Robust-PTM

The first dataset that we explore consists of 50 images captured under 50 different lighting directions and contains both shadows and highlights. Figure 6.1 shows some of these images. We solve the modified Lum-PTM equation as per Eq. 3.5, using the LMS regression technique to identify matte, shadow and highlight contributions automatically. Having this information, we can then estimate chromaticity, and the matte contribution for any existing and new lighting directions. Suppose the illumination direction we are interested in is the one for the input image shown in Figure 6.1(d) with $l = (0.48, -0.61, 0.64)^T$.

Figure 6.2 presents the result of performing Robust-PTM on the input data. Figure 6.2(b) is actual luminance of the input image. Figure 6.2(a) is the shadow and specularity map solved by Robust-PTM. This figure shows weights corresponding to the input image where w^0 is shown as white, w^+ as green, and w^- as red. The approximated luminance for the matte contribution is shown in Figure 6.2(c). This is the polynomial PTM-based image. Figure 6.2(d) presents the recovered chromaticity which is intrinsic color without

Figure 6.1: Some images from Barbara dataset

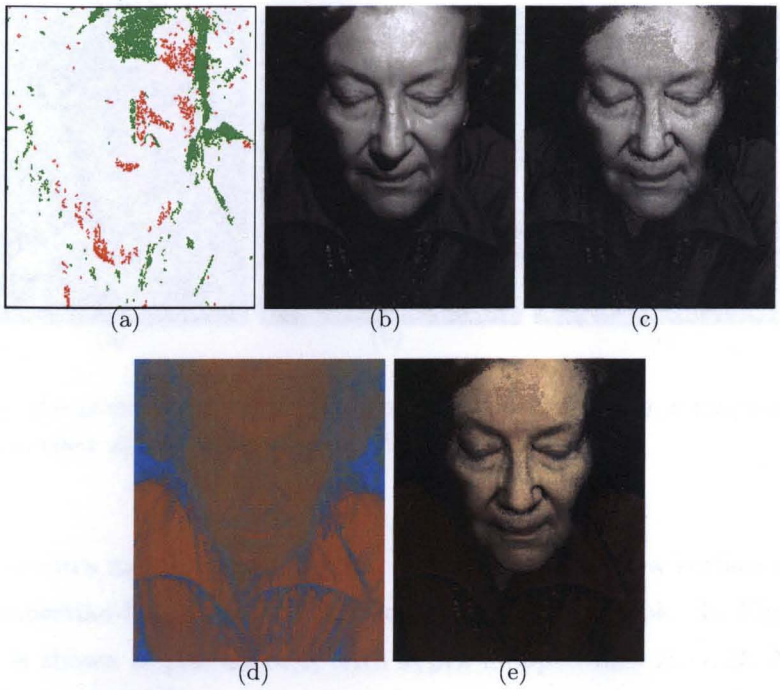


Figure 6.2: (a): Shadow and specularity map for $u = 0.48, v = -0.61, w = 0.64$ obtained by Robust-PTM. (b): Actual luminance for $u = 0.48, v = -0.61, w = 0.64$. (c): Approximated luminance for matte component using Robust-PTM. (d): Recovered chromaticity. (e): Approximated colored version for matte contribution.

magnitude, as in Eq. 4.9. Finally the colored version of the matte is depicted in Figure 6.2(e) which is obtained by simply multiplying chromaticity and approximated luminance. This is what PTM matte relighting generates. Note that using robust slopes can generate black output, from slopes that are near zero, since dark values may form a majority at some pixels — the area around the nose in this case — as shown in Figure 6.2(c). Least-squares based PTM might return brighter matte values for these cases. Also this image shows that the base luminance value does include some specular content which is due to the non-linear regression of PTM and the fact that PTM can model the reflectance of any general surface and not only Lambertian surface.

At this point we can generate surface normals and albedos either by the mean of the modified Robust-PTM coefficients we already have, Eq. 4.2, or by performing Robust-PST using inlier illumination directions and luminances, Eq. 4.6. However we showed earlier that

(a) (b) (c)

Figure 6.3: (a): Recovered surface normal: pseudocolor with x, y, z mapped into R, G, B. (b): Recovered surface albedo. (c): Intrinsic image.

Robust-PST generates more accurate results. Figure 6.3 shows how surface normal, surface albedo and Lambertian-based matte intrinsic image, Eq. 4.8, look. In Figure 6.2(a), the surface normal is shown in pseudocolor, with x, y, z mapped into R, G, B. Note that since Figure 6.2(e) includes shading, its appearance is more realistic comparing to Figure 6.3(c).

6.2 RBF

Now we are ready to interpolate shadow and specularity using different sets of Radial Basis Functions as explained in Chapter 5. Figure 6.4 shows the result of the *chromaticity based method* discussed in section 5.4.1.

Figure 6.4(d) shows the reconstructed luminance of Figure 6.1(d). Comparing the reconstructed and original luminance shown in Figure 6.2(b) demonstrates that the chromaticity based method works very well for luminance images. However the colored image version contains some gray pixels around the nose and in the forehead, which is due to the near-zero slopes generated by robust regression — the interpolated color in these areas is basically that of the sheen and therefore is greyish. This artefact is removed using *three-shade and -sheen model*, described in section 5.4.2, as shown in Figure 6.5(d).

We showed that our method can regenerate the input data accurately and with high PSNR. PSNR values between the input and the thusly regenerated RGB images range from 27.54 to 50.43 with median value 35.61, justifying the suitability of this approach. Figure 6.6 shows how the method will work for new lighting directions. We chose the average between

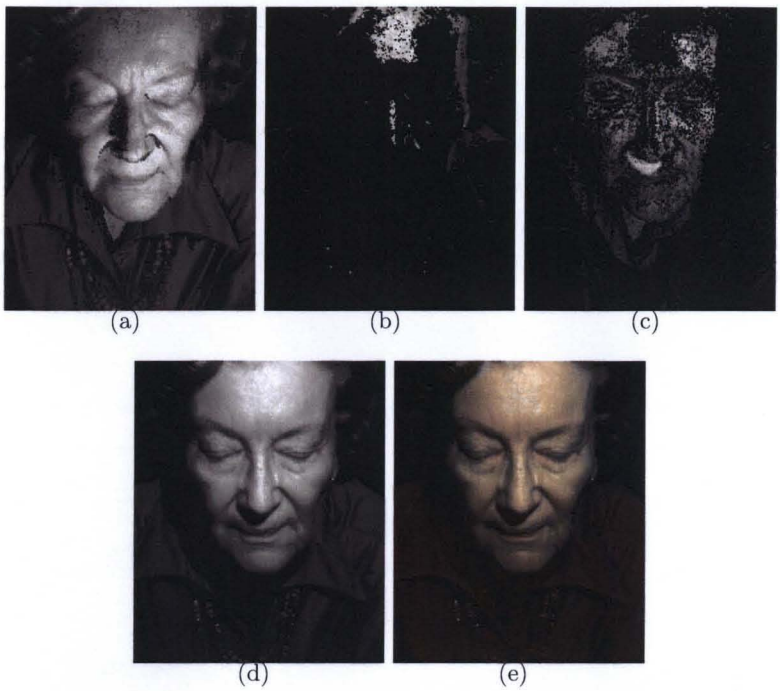


Figure 6.4: Chromaticity-based method.(a): Approximated matte contribution. (b): Modeled sheen contribution. (c): Modeled shade contribution. (d): Reconstructed luminance image of the input data. (e): Reconstructed colored image of the input data.

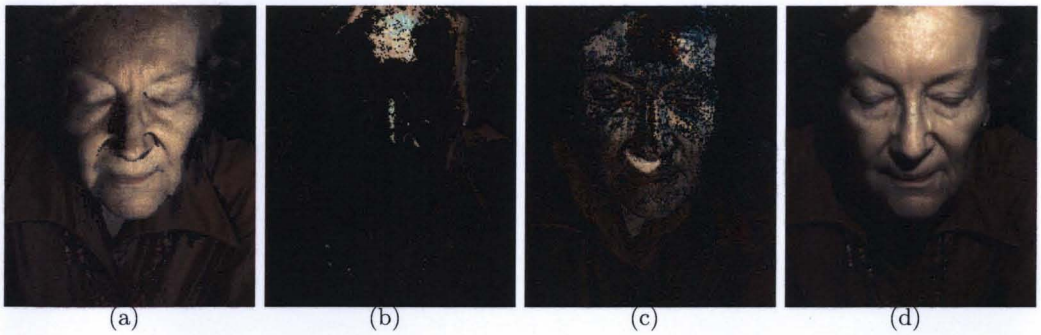


Figure 6.5: Three-shade and -sheen method.(a): Approximated colored matte contribution. (b): Modeled colored sheen contribution. (c): Modeled colored shade contribution. (d): Reconstructed colored image of the input data.

(a) (0.22,0.35,0.91) (b) (0.19,-0.57,0.80) (c) (0.19,-0.57,0.80) (d) (-0.0,-1.0,-0.0)

Figure 6.6: (a,d): Input images. (b): Interpolated result of the chromaticity-based method for light between (a) and (b). (c): Interpolated result of three-shade and -sheen method for light between (a) and (b).

two available lighting directions as the direction of interest.

6.3 Further Results

In this section we will present more interpolated results using the *three-shade and -sheen model*, displayed in Figure 6.7. The left and right images shown are two of the inputs under two known lighting directions, say a and b. The middle images are the interpolated results for the averaged lighting direction $(a + b)/2$.

Overall, results for interpolating specularities and shadows are seen to be excellent, producing a new kind of results in this field. However in situations where shadows are completely black, as in Figure 6.7(n) — a synthetic example —, the method can produce streaks resulting from hard shadow regions. Nevertheless even for cases where pixels are saturated, as in Figure 6.7(g,i), the sheen model simply sees these as extra, bright information and successfully models them in interpolants. Overall the new method is indeed seen to be a promising new approach.

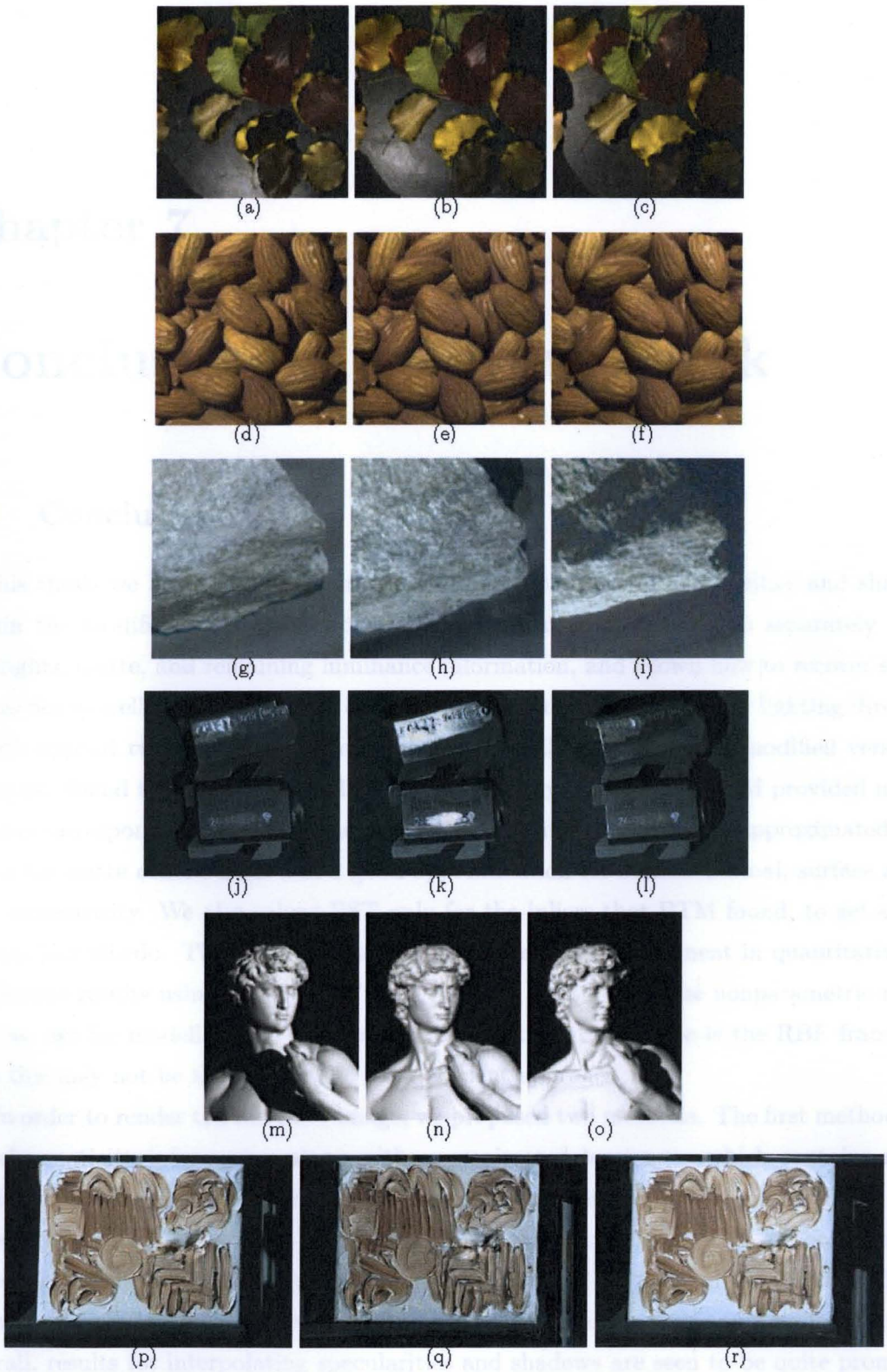


Figure 6.7: Left, right: two of inputs. Center: Interpolant for mean of left and right lighting directions.

Chapter 7

Conclusion and Future Work

7.1 Conclusion

In this thesis we have presented a method to interpolate both specularities and shadows, within the modified PTM framework, by applying robust methods to separately model highlights, matte, and remaining luminance information, and shown how to recover surface properties as well as to produce an accurate RGB rendition under a new lighting direction.

We applied two robust regression techniques, LMS and LTS, to a modified version of PTM and found that LMS has smaller time complexity. LMS-based PTM provided us with weights corresponding to matte, shadow and specularity, as well as the approximated luminance for matte contribution and a good approximation for surface normal, surface albedo and chromaticity. We also solved PST only for the inliers that PTM found, to get surface normal and albedo. The experimental results showed an improvement in quantitative and qualitative results using Robust-PST to get surface properties. The nonparametric regression we use for modelling the sheen and the departure from matte is the RBF framework (but this may not be the best or the most efficient approach).

In order to render the full color image, we proposed two methods. The first method used the chromaticity information along with approximated luminance which contains matte, shade and sheen contribution. Although this method generated quite good results, it sometimes produces artefacts when PTM slopes are near zero. The second method proposed for rendering the full color image solved this problem. This method regressed shade and sheen for each color channel separately and not only luminance shade and luminance sheen. Overall, results for interpolating specularities and shadows are seen to be quite promising,

especially by the mean of *three-shade and -sheen model*, i.e., the second method described in Chapter 5.

7.2 Future Work

The RBF framework used for interpolating shadow and specularity may not be the best available approach, and in future one could consider both the smoothing of RBF as well as other methods. For RBF, the Gaussian base functions may not necessarily be the best choice, and several other common functions could be used, such as a thin-plate model.

In future, we will examine mechanisms for reducing the space and time complexity of the RBF modeling used. In particular, one straightforward strategy would be to examine the possibility of using a Principal Components Analysis for a linear dimensionality-reduction scheme.

We intend to apply the developed method to artworks, with a view to determining their 3D structure and surface properties so that they can be measured before and after they are moved, e.g. by lending to other institutions. There is much interest in determining whether any damage has occurred in such circumstances. As well appearance changes for famous artworks under re-lighting are also of interest.

Bibliography

[1] V. Argyriou, S. Barsky, and M. Petrou. Generalisation of photometric stereo technique to q-illuminants. In *Proceedings of 19 th British Machine Vision Conference*, 2008.

[2] V. Argyriou and M. Petrou. Recursive photometric stereo when multiple shadows and highlights are present. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6, 2008.

[3] H. G. Barrow and J. M. Tenenbaum. Recovering intrinsic scene characteristics from images. In *Computer Vision Systems*, pages 3–26, 1978.

[4] S. Barsky and M. Petrou. The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows. *IEEE Transaction on Pattern Analalysis and Machine Intelligence*, 25(10):1239–1252, 2003.

[5] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[6] M. Chandraker, S. Agarwal, and D. Kriegman. Shadowcuts: Photometric stereo with shadows. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[7] E.N. Coleman and R. Jain. Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry. pages 180–199, 1992.

[8] J. Dong and M. Chantler. Capture and synthesis of 3d surface texture. *International Journal of Computer Vision*, 62(1-2):177–194, 2005.

[9] J. Dong, G. Sun, and G. Chen. Conversions between three methods for representing 3d surface textures under arbitrary illumination directions. *Image Vision Computing*, 26(12):1561–1573, 2008.

[10] M. S. Drew, N. Hajari, Y. Hel-Or, and T. Malzbender. Specularity and shadow interpolation via robust polynomial texture maps. In *Proceedings of British Machine Vision 2009*, 2009.

[11] F. Y. Edgeworth. On observations relating to several quantities, 1887. Hermathena.

[12] Y. Hammer, S. Bengtson, T. Malzbender, and D. Gelb. Imaging fossils using reflectance transformation and interactive manipulation of virtual light sources. In *Manipulation of Virtual Light Sources, Palaeontologia Electronica*, 2002.

[13] C. Hernández, G. Vogiatzis, and R. Cipolla. Shadows in three-source photometric stereo. In *Proceedings of the 10th European Conference on Computer Vision*, pages 290–303, 2008.

[14] C. Julia, A.D. Sappa, F. Lumbreras, J. Serrat, and A.M. Lopez. Photometric stereo through an adapted alternation approach. In *Proceedings of IEEE Conference on Image Processing*, pages 1500–1503, 2008.

[15] H.-C. Lee, E. J. Breneman, and C. P. Schulte. Modeling light reflection for computer color vision. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(4):402–409, 1990.

[16] G. R. Liu. *Mesh Free Methods: Moving Beyond the Finite Element Method.* CRC Press, 2002.

[17] T. Malzbender, D. Gelb, and H. Wolters. Polynomial texture maps. In *In Computer Graphics, SIGGRAPH 2001 Proceedings*, pages 519–528, 2001.

[18] D. Miyazaki, K. Hara, and K. Ikeuchi. Photometric stereo beyond glass: Active separation of transparent layer and five-light photometric stereo with m-estimator using laplace distribution for a virtual museum. In *International Workshop on Photometric Analysis for Computer Vision*, pages 325–329, 2007.

[19] B. S. Morse, T. S. Yoo, P. Rheingans, D. T. Chen, and K. R. Subramanian. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. In SMI'01: Proceedings of the International Conference on Shape Modeling & Applications, page 89, 2001.

[20] S. Nayar, K. Ikeuchi, and T. Kanade. Determining shape and reflectance of hybrid surfaces by photometric sampling. *IEEE Trans. on Robotics and Automation*, 6(1):418–431, August 1990.

[21] J. Padfield, D. Saunders, and T. Malzbender. Polynomial texture mapping: A new tool for examining the surface of paintings. In *ICOM Committee for Conservation*, volume 1, pages 504–510, 2005.

[22] B. Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, 1975.

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

[24] P. J. Rousseeuw and K. Driessen. Computing lts regression for large data sets. *Data Mining Knowledge Discovery*, 12(1):29–45, 2006.

[25] P. J. Rousseeuw and A. M. Leroy. *Robust regression and outlier detection.* John Wiley & Sons, Inc., 1987.

[26] H. Rushmeier, G. Taubin, and A. Gueziec. Applying shape from lighting variation to bump map capturing. In *Proceedings of Eurographics Rendering Workshop 1997*, pages 35–44, 1997.

[27] F. Solomon and K. Ikeuchi. Extracting the shape and roughness of specular lobe objects using four light photometric stereo. *IEEE Transaction Pattern Analysis and Machine Intelligence*, 18(4):449–454, 1996.

[28] 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 Vision Comput.*, 25(7):1050–1057, 2007.

[29] F. Verbiest and L. VanGool. Photometric stereo with coherent outlier handling and confidence estimation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[30] A. Wenger, A. Gardner, C. Tchou, J. Unger, T. Hawkins, and P. Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Transaction on Graphics*, 24(3):756–764, 2005.

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

[32] A. Yuille and D. Snow. Shape and albedo from multiple images using integrability. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 158, 1997.