

**PHOTOMETRIC STEREO VIA LOCALITY
SENSITIVE HIGH-DIMENSION HASHING**

by

LIN ZHONG

B.Sc., Zhejiang University, 2002

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

in

THE FACULTY OF GRADUATE STUDIES
(COMPUTER SCIENCE)

THE UNIVERSITY OF BRITISH COLUMBIA

April 2005

© Lin Zhong, 2005

Abstract

Photometric stereo is a shape-from-shading method for recovering three-dimensional surface orientation information from two-dimensional images with differing illumination but the same viewing geometry. According to the former photometric stereo methods, the inconvenience of calibration and the cost of searching for the gradient of the best match between reference images' brightness and target images' brightness remain as the major problems in this area. Lately, the increasing interest in geometry reconstruction by using photometric stereo has led to a new method for improving the original photometric stereo method. This approach is based on the assumption that two points with the same surface property should reflect the same light and show the same brightness. This new method largely simplifies the traditional calibration experiment by getting the reference object's and target object's information at the same time. Moreover, this new convenient method provides greater opportunity for photometric stereo to be applied to practical real-time robot vision.

In this thesis, we extend the new photometric stereo method of Hertzmann & Seitz that uses multiple images of an object together with a calibration object. For each point in the registered collection of images, we have a large number of brightness values. Photometric stereo finds a matching collection of brightness values from the calibration object and overdetermines the surface normal. With a large number of images in high dimensions, finding similar brightnesses becomes costly. To speed up the search, we apply locality sensitive high dimensional hashing (LSH) to compute the irregular target object's surface orientation. The experimental results of a simplified photometric stereo experiment show consistent results in surface orientation. LSH can be implemented very efficiently; and it offers the possibility of practical photometric stereo computation with a large number of images. Finally, we present the idea that LSH should be applied in a variety of computer vision areas.

Contents

Abstract	ii
Contents	iii
List of Figures	v
Acknowledgements	vii
1 Introduction	1
1.1 Photometric Stereo	2
2 Background	6
2.1 Shape from Shading and Reflectance Map	7
2.1.1 Diffuse Reflection	7
2.1.2 Specular Reflection	8
2.1.3 Phong Model	9
2.2 Photometric Stereo	9
2.2.1 Surface Curvature	11
2.3 Non-Lambertian Photometric Stereo Modeling	12
3 Implementation and Experiments	14
3.1 Experiment Setting	15
3.2 Photometric Stereo from High-Dimension Images	16

3.2.1	Least Squares Fitting of Ellipse	20
3.3	Match From Reference Images for Target Images	22
3.4	High-Dimension Locality Sensitive Hashing	23
3.4.1	Notation	23
3.4.2	The Algorithm	25
3.5	Experimental Results	31
4	Conclusion	51
	Bibliography	53

List of Figures

3.1	Video images with both the reference and target objects	15
3.2	The LSH hash tables structure	27
3.3	Initialization of high-dimensional hash table	29
3.4	Approximate Nearest Neighbor query answering algorithm	29
3.5	Three images of ball under different illustrations	32
3.6	The boundary contour of the reference ball	32
3.7	Three images of doll face under different illustrations	33
3.8	The gradient estimation of the ceramic doll face	33
3.9	Gradient estimation of the doll face from different angles	34
3.10	Eight images of a billiard ball under a hand-held moving light source	35
3.11	The boundary contour of the reference billiard ball	35
3.12	Eight input images of the target elephant	36
3.13	The gradient estimation of the target red elephant	37
3.14	Gradient Aspect and Needlemap estimation of elephant's side	38
3.15	Gradient slope map2	39
3.16	Aspect and needlemap of front side of elephant estimation	40
3.17	Gradient slope map3	41
3.18	Back side of elephant's aspect and needlemap estimation	42
3.19	The gradient estimation for a painted bottle	44
3.20	The 3D reconstructed model of the green bottle	45
3.21	Convergence performance with varying image numbers	46

3.22	Convergence performance with varying the number of hash tables . .	47
3.23	The cosine of the angle between	49
3.24	Comparison between bottle reconstructed and laser scanned	50

Acknowledgements

I would like to thank my supervisor Dr. James J. Little and Dr. Robert Woodham for all their guidance, patience and leadership. Without their effort, this thesis would not have been possible. Thanks to Dr. David Lowe for giving me suggestions and being my thesis's second reader. I am also grateful to Dr. Steven M. Steitz for providing me with image data. I thank my family and my friends, who have encouraged and supported me in my work.

LIN ZHONG

THE UNIVERSITY OF BRITISH COLUMBIA
April 2005

Chapter 1

Introduction

Vision is our most powerful but also most complicated sense. A person sees seamless images through the vision apparatus of the eyes and the brain, which is the amazing function we call *vision*. Human vision receives seamless images, including the details of shape, colour, position and motion, without any noticeable breaks while they are being updated. In other words, people perceive a coherent three-dimensional world with its invariant properties. These properties, in some way, are sensed by human directly. Basically, the changes of brightness and colour are the most important information for human eyes. Human vision can also immediately distinguish the difference among objects based upon previous experience. However, there also exist some limitations for human vision, such as *change blindness*, *misconception* and *limited spectral sensitivity*. *Change blindness* means that human vision adapts with difficulty large changes in a scene and misunderstand real objects. It often happens as the result of eye movements, image flips, movie cuts and other disturbances. *Misconception* means that human vision is influenced by the shape of an object. *Limited spectral sensitivity* refers to the human eye's limited wavelength range, which ranges only from 400nm to 700nm.

As for computer vision, received information consists of mathematically characterized two-dimensional images. The explicit challenge of computer vision is how

to acquire a three-dimensional description from two-dimensional images. Moreover, how to robustly produce the result is another considerations. Compared to human vision, human judgment is difficult to quantify for computer vision. For example, what information about scenes can be extracted from an image using only basic assumptions about physics and optics? What kinds of computations are required to be performed? How are real-world models and knowledge represented and used? These are the key questions of computer vision. For photometric stereo, how to solve the above questions would help photometric stereo get the 3D information from the 2D images correctly and efficiently.

The information we can get directly from 2D images are the brightness and its distribution. Brightness is determined by the light source's radiance, the object's geometry, the object's irradiance factor and also the camera's performance. Physical-based computer vision pays attention to the relationship between geometrical structures and radiometry. Recovering the three-dimensional object's shape from a two-dimensional image is an important problem — shape from shading — in computer vision. In terms of the computation of shape from shading, the method of using multiple 2D images, obtained from the same viewpoint but under different illumination conditions, has been presented as an effective way to obtain additional local radiometric constraints. The basic principle of this method is to use multiple images to overdetermine the solution locally; further, this method strengthens robustness by permitting local validation of the radiometric models.

1.1 Photometric Stereo

The photometric stereo method is a shape-from-shading technique to recover the orientation of surface patches from a number of images taken under different lighting conditions [14], which has been implemented in different experimental environments and has also produced consistently satisfactory results. The traditional photometric stereo method is based on physical-based modelling, which follows the principles

of optics. It works well for a regular object's computation; however, when it encounters complicated irregular objects, it fails to build up their physical models. The traditional photometric stereo technique is not enough in such cases. Visual scientists often use a system of units (photometric units) that scale the physical power (radiometric units) by matching the spectral sensitivity of the eye. Optics determines that the surface normal vector relates the object's geometry to its image's irradiance. In an orthographic projection, the viewing direction and hence the phase angle is constant for all the surface elements. Thus, for a fixed light source and geometry, the ratio of scene radiance to scene irradiance depends on the surface normal vector. But the equation cannot be inverted locally because image brightness provide only one measurement, whereas surface normal vector has 2 degrees of freedom (DOF). Under this circumstance, Woodham [27, 28] developed the photometric stereo method by using more than two local brightness measurement in order to overdetermine the 2 DOF of the surface orientation.

In the image irradiance equation, the formulation assumes that each surface element receives illumination only directly from the light source(s). This is correct when scenes consist of a single convex object. Actually, the surface radiance is from the light source, but it is impacted by other factors, such as shadows, inter-reflections and other non-local radiance effects. In our work, we ignore those factors and assume the object's surface is made of the same material. Surface shading provides information of the surface geometry. However, if illumination, camera geometry, or reflectance varies the appearance of the object would be different. As a result, without knowing the way to build the model — the relationship of the contribution of each factor — it is very difficult to compute the shape. To address this problem, we introduce the theory that two points with the same surface orientation reflect the same light toward the viewer. This is the basic idea of shape by example, in which we search the same appearance points and assume their surface normals are identical.

The motivation of this thesis is to overcome the limitation of high computational demands in example-based photometric stereo when large numbers of images are used. Under the assumption that two points with the same surface property should have the same appearance in an image, we adapted the method of [11] by placing the reference object and target object in the same image, which means that calibration of the camera or lighting environment is unnecessary in this case. The reason for using this method is to deal with cases when the target object is hard to move. In that situation, instead of using the former calibration method [28], putting the reference object beside the target object would allow for flexibility for the photometric stereo. Following this setup, we acquired multiple images for both objects. Each pixel on the object has d brightness values from d images. Then the problem becomes finding the best match between the brightness vector of the reference object and the brightness vector of the target object; this means searching in a high-dimensional space. Some techniques have already been applied for these purposes. The first method is the lookup table [28]; the total size of the lookup table is (2^{bd}) , which depends on the dimension d and the number of bits b into which brightness is quantized. This is prohibitively large when d — the number of the images — is large. Also the table is sparse, and time-consuming weighted interpolation is also needed to fill the empty entries in the table. Hertzmann & Seitz [11] used the ANN (approximate nearest neighbour) method to solve the problem of matching. It solves the problem well, but when the dimension exceeds 10 – 20 [10], it is also highly time consuming. However it depends on the degree of accuracy required. The prohibitive computation means the photometric stereo technique is not available for real-time applications. The practical benefits of real-time implementation are the ability to integrate photometric stereo and motion, and to replace the harmful laser range sensor for medical shape detection. Recently, a new method, locality sensitive hashing [10], has been introduced in database applications, usually in the context of high-dimensional similarity searching. It keeps the speed of hashing, and adds in

the property of clustering similar points together.

We implemented a robust system analyzing high dimensional images. Our approach simplifies the traditional shape from shading experiment by avoiding calibration as in [28]; moreover, it can rapidly and accurately estimate the surface orientation of irregular objects from a large database of example images via LSH. We applied the algorithm to several examples: Lambertian and non-Lambertian cases, multiple reference objects, different surface materials. We also compared the results with the exhaustive search method and the laser scanned images to show the accuracy of the method. Moreover, in this thesis, we analyzed computational complexity to illustrate the efficiency of this algorithm.

In this thesis, we described the method in the following order. Chapter 2 introduced related work in computer vision. Chapter 3 demonstrated the experiment's details, including the environmental setting, and explained the speedup technique used in this thesis. The final chapter provided the results of our method and conclusions.

Chapter 2

Background

Given a fixed viewpoint and a certain illumination, objects with various surface materials, following optic laws, will each have a distinctive appearance. All of these constraints determine the colour and intensity of the image of an object. The local reflectance character depends not only on the irregular geometry of the object but also on the reflectance surface material of the objects. We also need to consider the inter-reflections and casting shadows. Moreover, the wavelength of the light must also be taken into consideration for the reflectance model building. How to model all those factors to build up a general reflectance model exactly is very complicated and difficult. However, research on reflectance models is still ongoing. Book [26] collects quite a few introductions to physical modeling.

After thinking about these physical constraints on reflectance, Horn [12] first made a great observation that image irradiance can be expressed as a function that only depends on surface orientation, which is suitable for many practical image applications. Horn's research introduces the idea of the reflectance map and points out a new formulation for the problem of shape from shading. The reflectance map makes explicit the relationship between surface orientation and image brightness. It is a representational tool used in developing methods for recovering surface shape from images.

2.1 Shape from Shading and Reflectance Map

Shape from shading is a method for determining the shape of a surface from its image. Following standard geometry, the surface is represented as $z = f(x, y)$ in a left-handed Euclidean coordinate system, when the viewer looks in the positive Z direction. The image projection is orthographic and in the XY -plane. Thus, the surface gradient (p, q) is denoted as

$$p = \frac{\partial f(x, y)}{\partial x}, q = \frac{\partial f(x, y)}{\partial y}$$

the surface normal vector of that point (x, y) is $[p, q, -1]$. We could also simply use $[p, q]$ to represent the surface orientation. The image irradiance equation is

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

where $E(x, y)$ is the image irradiance and $R(p, q)$ is the reflectance map [14, 16]. The reflectance map determines the image brightness as a function of the surface orientation. Then the shape-from-shading problem is defined as computing for a smooth surface $z = f(x, y)$. For a single image, shape-from-shading problems are typically solved by exploitation of *a priori* constraints on the reflectance map $R(p, q)$ [5, 15]. In the following we will discuss the different reflectance map models.

2.1.1 Diffuse Reflection

When the surface is microscopically rough, the light rays will reflect and diffuse in many different directions. Reflection from rough surfaces such as clothing, paper, and asphalt roadways leads to a type of reflection known as diffuse reflection. Lambertian reflection is the ideal case of diffuse reflection. For the ideal Lambertian material surface, which is illuminated by a distant light source, appears equally bright from all viewing directions and reflects all incident light, absorbing none.

In Lambertian reflection, the image irradiance is proportional to the $\cos(\theta_i)$, where the θ_i is the incident light angle. Consider a source of radiance E illuminating

a Lambertian surface. The scene radiance is

$$L = \frac{1}{\pi} E \cos \theta_i, \text{ for } \theta_i \geq 0,$$

Taking dot products of the corresponding unit vectors, we obtain

$$\cos \theta_i = \frac{1 + p_s p + q_s q}{\sqrt{1 + p^2 + q^2} \sqrt{1 + p_s^2 + q_s^2}}$$

This gives us a good idea of how brightness depends on surface orientation. This result is called the reflectance map, denoted by $R(p, q)$ [14]. The reflectance map depends on the properties of the object's surface material and the distribution of light sources. The radiance cannot be negative, so we should, strictly speaking, impose the restriction $0 \leq \theta_i \leq \pi/2$. The radiance will be zero for values of θ_i outside this range.

$$R(p, q) = \frac{1 + p_s p + q_s q}{\sqrt{1 + p^2 + q^2} \sqrt{1 + p_s^2 + q_s^2}}$$

For other diffuse reflection cases, the radiance will also depend on the *Albedo* — the material's bidirectional reflectance factor. It varies spatially, independent of the gradient.

2.1.2 Specular Reflection

Reflection off smooth surfaces such as mirrors or a calm body of water leads to a kind of reflection known as specular reflection. Each individual light ray of the bundle follows the law of reflection. If the bundle of light rays is incident upon a smooth surface, then the light rays reflect and remain concentrated in a bundle upon leaving the surface. Specular surface reflects all incident light in a direction that lies in the same plane as the incident ray and the surface normal. Specular reflection is also a function of the light incidence angle θ . If $\theta = 0^\circ$, there is almost no specular reflectance. When $\theta = 80^\circ$, specular reflectance is high. A full specular reflectance function is the Bi-directional Reflectance Distribution Function (BRDF). For many materials the specular coefficient is approximately constant. Helmholtz Reciprocity

is one of BRDF's important properties:

$$f_r(\theta_i, \theta_o) = f_r(\theta_o, \theta_i)$$

where θ_i is the incident angle and the θ_o is the output angle. The equation illustrates that the reflectance of the material is not one-way street; its incoming to outgoing pathway is the same as its outgoing to income pathway.

2.1.3 Phong Model

The Phong reflection is another type of reflectance; it is a combination of the diffuse and specular illumination. This is an empirical model based not on physics but on physical observation. Its image irradiance is proportional to [13]

$$R(p, q) = \frac{\cos \theta_i [(1 - \alpha) + \alpha \cos^n(s/2)]}{\cos(\phi/2)}$$

where θ_i is the incident angle, ϕ is the phase angle (the angle between the vector pointing to the light source and the vector pointing to the viewer), and s is the off-specular angle (the angle between the vector pointing to the viewer and the vector that defines, relative to the light-source direction and the surface normal, the direction of perfect specular reflection). α is a fraction, $0 \leq \alpha \leq 1$. The factor α adjusts how much of the incident light is reflected specularly, and n is a number that models how compact the specular patch is comparing with the perfect specular reflection, i.e., n would be very large, like $n > 200$, if the surface is specular. When n is small, the surface is duller. Parameters α and n vary with respect to the material's properties.

2.2 Photometric Stereo

The photometric stereo method has been developed to recover the orientation of surface patches from a number of images taken under different lighting conditions

[24, 27, 17, 28]. Woodham [27, 28] presents the three image irradiance equations:

$$E_1(x, y) = R_1(p, q),$$

$$E_2(x, y) = R_2(p, q),$$

$$E_3(x, y) = R_3(p, q),$$

The three image irradiance equations overdetermine the solution of p , q , at each point of (x, y) , because the solution uses the three brightness measurements of E_1 , E_2 and E_3 to configure the two parameters p and q . The images can only describe the object's 2-D information, and the three image irradiance equations present the way to connect 3-D information with 2D images. By measuring the $[E_1, E_2, E_3]$ under the three light source situation, we acquire the 3D p and q by computing the gradient of the object. The implementation method is simple. It keeps the three light source and camera still, and takes images of the calibration object and the target object at the same position one by one. The calibration object of known shape is made of the same material as that of the target object's surface. It uses the triple of measured brightness values $[E_1, E_2, E_3]$ to search for the corresponding gradient (p, q) in a lookup table. The method is straightforward but still has some limitations. Assume the image has $2^8=256$ gray values, then the full size of the lookup table is $2^8 \times 2^8 \times 2^8 = 2^{24}$ entries. Besides that, we consider that the per entry is two *double type* size (for the gradient p and q) to store the gradient value. Then the full size is more than 1G. Moreover, the huge table is very sparse, which results in time-consuming interpolation to fill the table. However, the interpolation is no guarantee that all the table entries are filled. To solve the large table search, Woodham's work [28] accelerates the search by using 2^6 gray value; the other 2^2 value is separated as the three tables choose an index. Thus, by using the lookup table with 2^{18} entries, the retrieving achieves a near-real-time throughput at 15 Hz. If the surface material's albedo also varies spatially, independent of the gradient,

then the image irradiance equation will be [7, 17, 22, 23]

$$E_1(x, y) = \rho(x, y)R_1(p, q),$$

$$E_2(x, y) = \rho(x, y)R_2(p, q),$$

$$E_3(x, y) = \rho(x, y)R_3(p, q),$$

$$E_4(x, y) = \rho(x, y)R_4(p, q)$$

where the $\rho(x, y)$ is the albedo, a function of (x, y) . Therefore, the former image irradiance equation is a special case, where the albedo is the constant (and normalized to 1). For this situation, there are 3DOF (degrees of freedom) in the equation, p , q , $\rho(x, y)$. If the albedo parameter ρ is constant, then we can simplify the problem as the former; if the albedo varies along with the position of the surface, then we need one more light source of image irradiance equation to over-determine the 3DOF. Thus, In addition to a fixed scale factor, the reflectance map gives the dependence of scene radiance on surface orientation. It is often convenient to plot the surface $R(p, q)$ as a function of the gradient (p, q) . The pq -plane is called gradient space, and every point in it corresponds to a particular surface orientation. The above expression also assumes that any additive offsets, e.g., due to atmospheric scattering, have been removed.

2.2.1 Surface Curvature

We use the representation for surface curvature based on the gradient (p, q) [28]. There are 3DOF to the curvature of a point of a smooth surface. One representation is the 2×2 matrix of the second partial derivatives of the surface $z = f(x, y)$.

$$p_x = \frac{\partial^2 f(x, y)}{\partial x^2}, p_y = \frac{\partial^2 f(x, y)}{\partial x \partial y}, q_x = \frac{\partial^2 f(x, y)}{\partial y \partial x}, q_y = \frac{\partial^2 f(x, y)}{\partial y^2}.$$

The Hessian matrix of $z = f(x, y)$ is

$$H = \begin{vmatrix} p_x & p_y \\ q_x & q_y \end{vmatrix}$$

Then the surface curvature matrix is

$$C = (1 + p^2 + q^2)^{-\frac{3}{2}} \begin{bmatrix} q + 1 & -pq \\ -pq & p^2 + 1 \end{bmatrix} H$$

The principal curvatures of the matrix C are k_1 and k_2 , which are also the two eigenvalues of the matrix C . ω_1 and ω_2 are eigenvectors of the C , which are also the direction of k_1 and k_2 . Thus k_1 , k_2 , ω_1 , and ω_2 together are the four independent parameters that describe the surface curvature of that the object. Also, Besl and Jain [4] introduced the idea that the sections of a smooth surface could be segmented into one of eight basic types based on the sign and the zeros of Gaussian and mean curvature. The Gaussian curvature K , also called the total curvature, is the product, $K = k_1 k_2$, of the principal curvatures. The mean curvature H is the average, $H = (k_1 + k_2)/2$, of the principal curvatures. The surface curvature provides another description for the 3D shape information.

2.3 Non-Lambertian Photometric Stereo Modeling

Much research is devoted to the recovery of non-Lambertian reflectance parameters. Nayar [22] applied photometric stereo using a "hybrid reflectance model." Tagare and deGigueiredo [25] developed the theory of photometric stereo for the class of *m-lobe* reflectance maps. Kay & Caelli [20] continued their work by investigating the problem from a practical point of view. They applied nonlinear regression to a large number of input images. Those methods not only recover the surface orientation but also compute reflectance parameters. The problem is that their methods requires quite a lot of images, and the algorithms are fairly complicated. Many non-Lambertian surfaces exhibit near-Lambertian behaviour outside their regions of specularity. By segmenting these surface areas, it is a very attractive option to apply a linear algorithm, which was developed for Lambertian surfaces, to surfaces with non-Lambertian reflectance and treat highlights as deviations from Lambertian

Law. This technique was proposed by Coleman and Jain [7]. The above work deals mostly with gray-scale images.

Some recent work uses colour images to get more redundant images by separating an RGB colour image into 3 independent images. These methods are called Shape-from-Color. The surface should be illuminated by several light sources that are spectrally distinct and whose directions do not lie in the same plane, which means they are not linearly dependent on each other. This method was applied to the task of face recognition. Christensen [6] introduced the method of colour photometric stereo for surfaces with an arbitrary reflectance. The method is a generalization of [28] and also uses look-up tables. The disadvantage of this method is that the surface should be either uniformly coloured or its colours should form distinct separable clusters in the colour space, which significantly restricts the choice of acceptable surfaces. Another disadvantage is the need for a preliminary calibration. To overcome these limitations, most subsequent work on photometric stereo turned to using analytic models rather than the empirical models of surface reflectance [2]. Some of these methods yield good results, while real-world materials do not fit these traditional models. Therefore, researchers developed the approach of combining the developed empirical models and the analytic models to simulate the real-world's complicated surface material [21, 30, 1, 11, 29]. They introduced the method of using more than one calibration object, each of them has a weight, while at the same time segmenting the real target object with different reflectance properties. The disadvantage of this method is that it requires quite a lot of images and the matching algorithms are pretty inefficient. In this thesis, we show that these major difficulties are overcome with the new technique—high-dimension locality sensitive hashing, with which the photometric stereo method becomes a practical shape recovery application.

Chapter 3

Implementation and Experiments

We use more than three images to overdetermine the mapping relationship between the geometry and the brightness and so overcome the limitations of the original photometric stereo method. Woodham [28] and Horn [14] used only three images (3DOF) to solve the problem of Photometric stereo — computing the 2DOFs of the gradient p and q . This methodology works well when the reference object and the target object are easy to move to the same spot to simulate the same illumination condition. Also it analyzes the reference object and target object separately, which is not convenient enough for real-time processing. On the other hand, Hertzmann & Seitz's [11] work, uses multiple images to reconstruct the geometry of the target object by simply putting the calibration object and the target object in the same scene. This method is much more flexible than Woodham's [28] work. However, their geometry is not really accurate as the method's major objective is to determine the geometry's reconstruction, and also the speed is 5 minutes, which is too long. Based on the definition of photometric stereo, and inspired by the novel idea of texture reconstruction, we would like to adapt the methods by putting the calibration object and the target objects in the scene together, using the locality high-dimension

hashing technique to accelerate the search for the mapping relationship between the gradient and the corresponding brightness of the objects.

3.1 Experiment Setting

The reference sphere and the target object have been juxtaposed with a certain distance on the same scene. Their surfaces have been painted with the same red tempera paints to get the Lambertian effect. We set a Canon Optura Pi NTSC digital video camcorder stationary to facing the objects and moved the light source, a Philip 150W bulb, by hand, 4 meters from the objects. Then we recorded a series of video sequence, in which the reference object and target object were shot with same exposure under different light illuminations. We separated the 640×480 video into 640×480 image frames. Due to the hand-held light source's movement and the digital camera's focus is automatically adjusted, it is necessary to eliminate those blurred images. An example of the frame is showed in Figure 3.1. After that, we separated the calibration object and the target object from the image and analyzed them. Figure 3.10 and Figure 3.12 present the reference object and the target object under the same light source.

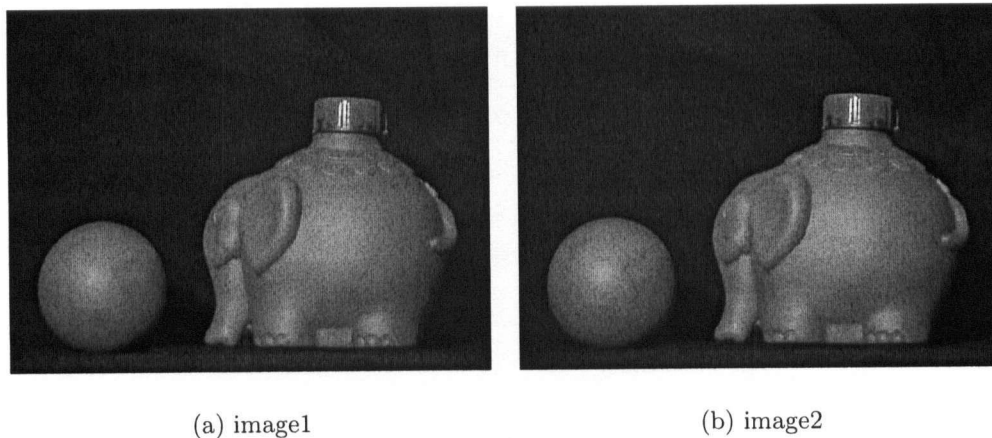


Figure 3.1: Video images with both the reference and target objects

3.2 Photometric Stereo from High-Dimension Images

We captured many 2D images under different lighting, so that for a particular pixel position for all these images, we got an array of the correspondent brightness values. These brightness values could be treated as a very big number by concatenating them; however, this method would take out the independence of the data to each other. Considering this, we would handle the multiple brightness values as a high dimensional brightness vector.

Determining the shape from the shading information of multiple images is the basic idea of photometric stereo. The radiometry of the object in the image depends on illumination, camera geometry, and the object's reflectance factor of the object. A change in one of them will affect the appearance of the object in the image. Unless we know all the factors accurately, it is very hard to compute the shape. In this case, we propose a calibration method by addressing the orientation consistency cue [11]:

Cue: Two points with the same material and the same surface orientation reflect identical brightness when they are under the same illumination situation.

This cue holds under the following assumption:

Assumption: Both points have the same BRDF property, the light sources are directional (i.e., distant), the camera is orthographic, and there are no shadows, interreflections and transparency effects, or other non-local effects that do not depend purely on the BRDF. Also, the surface orientation, incident illumination, and viewer direction are the same for both points.

The above cue is clear about presenting the distribution of the object's radiometry of the object. If a point has a highlight on an object, the other point with the same surface geometry should have the same highlight too. Therefore, we can apply the cue since we know one point's brightness and its surface orientation, and since another point has the same brightness, we can say that the second point has the same surface orientation as the first point. In this way, we can extend this

knowledge to compute the other points' surface orientations.

Now, we will present our implementation algorithm for computing the surface orientation of an irregular shape object by using its known shape object as calibration. Also, we introduce an effective technique for accelerating the searching process in real-time.

After isolating the reference object and the target object from the captured video frames, and following the basic theory, we present the following notation representation. We use $I_1^r, \dots, I_d^r$ to denote the reference images and $I_1^t, \dots, I_d^t$ to present the target images, where d is the dimension. Further, we make the assumption that the corresponding reference I_i^r and targets I_i^t are exposed under the same illumination condition. For each pixel position $p(x, y)$ in all reference images, let V_p^r be the brightness vector, which is the set of brightnesses observed at that pixel over n images:

$$V_p^r = [I_{1,p}^r, \dots, I_{d,p}^r]^T; \quad (3.1)$$

Similarly, for every pixel $q(x, y)$ in all target object images, V_q^t is the brightness vector for the target object.

$$V_q^t = [I_{1,q}^t, \dots, I_{d,q}^t]^T; \quad (3.2)$$

For our case, we tried the 4, 8, 16, 32, 64 observation vectors to determine one gradient vector $[p, q]$.

$$V_p^r \Longleftrightarrow [p, q]_p; \quad (3.3)$$

The idea comes from photometric stereo by multiple images [24]. It is very difficult to straightforwardly set up the relationship between the brightness appearance and the gradient object. Thus, the three image irradiance equations are presented as:

$$\begin{aligned} E_1(x, y) &= R_1(p, q), \\ E_2(x, y) &= R_2(p, q), \\ E_3(x, y) &= R_3(p, q), \end{aligned} \quad (3.4)$$

which use 3DOF brightness to overdetermine the 2DOF gradient. If the surface material's bidirectional reflectance albedo factor also varies spatially, independent of the gradient, then the image irradiance equations become

$$\begin{aligned} E_1(x, y) &= \rho(x, y)R_1(p, q), \\ E_2(x, y) &= \rho(x, y)R_2(p, q), \\ E_3(x, y) &= \rho(x, y)R_3(p, q), \\ E_4(x, y) &= \rho(x, y)R_4(p, q), \end{aligned} \tag{3.5}$$

Then there are 3DOF, including the *albedo* and the gradient, which need to be determined. Therefore, we may use 4DOF brightness to overdetermine this situation by using 4-images irradiation equations. Moreover, although we make the above assumptions, in the our experiments, we could not get the illumination to be as accurate as the former calibration method [28]. This's the reason why we would rather use more than 4-images irradiance equations to overdetermine the gradient and the *albedo*.

Above all, given a pixel q on the target object, the surface normal at q is determined simply by retrieving a point p on the reference object with the best matching of the brightness vector, which minimizes $\|V_p^r - V_q^t\|$. A complete correspondence determines the surface normal for each pixel on the target object.

$$V_q^t \implies [p, q]_q \tag{3.6}$$

It is very novel to realize that this formulation treats the photometric stereo methodology as 2-images stereo matching problem—figure out the pixel by pixel similarity between the image of the target object and the image of the reference object. Woodham[27] addresses the correspondence problem in binocular stereo should be fundamentally exclusive when solving the photometric stereo, however, under the assumption made in this paper, the camcorder captured both the reference and target objects at the same time. The light illumination is distant to the objects, so that the mapping relationship between the reference object and target object is

acceptable. Compared to the model-fitting methods [2, 14, 22, 27, 11] done before, we present an efficient hashing mapping technique for our model. If the reflectance map's parametric form varies, the hashing method would be a good alternative choice.

First, let us process the images of the reference object. As we mentioned in the experimental setting, the images with both reference object and target object are captured by a stationary camcorder. At the same time, we move the light source by hand 3-4 meters away from the objects to simulate distant illumination. We select d frames from the captured video series. The method needs to make sure that the illumination situations are not linearly dependent on each other, and also that the images need to be very clear, for when the illumination changes, the focus of the lens of the digital camcorder will change as well, so that some of the images are not out of focus, which are not as suitable as good samples to serve as the references. The captured images are in RGB mode. Instead of processing all the three-color channels for further reference, it is reasonable to analyze only the most sensitive color channel. For our case, the reference object and the target object are both painted with red, flat tempera paint, so red is the most sensitive channel here. After that, we need to preprocess the d images. First, we separate the image with the reference objects and the target objects into two images. Second, we need to isolate the reference object from the background. Our method is to sum the d reference images, and compute the threshold of the summed image according to its histogram. The threshold should be the dividing value to separate the black background and the object on the histogram. Based on that, we can then figure out the equation of the reference sphere object for further computing the geometry of the reference sphere. Here we use the least square fit-ellipse algorithm to get the 2D ellipse equation from the reference sphere's boundary data. Also, the threshold should be applied to the target images to separate them from its background. The purpose of computing the 2D equation is to get the 3D ellipse equation. Then we

can get each pixel of the ellipse's gradient. The robust algorithm of getting the fitting ellipse equation will be introduced in the following section.

3.2.1 Least Squares Fitting of Ellipse

Here we adapted a new algorithm for fitting ellipses [8]. It is a direct ellipse specific fitting algorithm, which can also handle bad data very well. The ellipse equation can be presented as

$$F(\vec{a}, \vec{x}) = \vec{a} \cdot \vec{x} = ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (3.7)$$

where $\vec{a} = [a, b, c, d, e, f]$ and $\vec{x} = [x^2, xy, y^2, x, y, 1]$. The aim here is to determine the parameter $\vec{a}$. In general, the problem of parameter estimation is often cast as an optimization problem. One of the traditional methods for optimization uses the least squares fitting. The other two general techniques are Kalman filtering and robust clustering. For the least squares fitting method, the focus is on computing the set of parameters that minimize the distance between the real-data points and the formulated curves and surfaces. Given finite input data set points $D = (x, y)_i$, $i \in N$, which in this case is the coordination set of the reference object's boundary points, the problem is to fit the ellipse curve equation $F(\vec{a}, \vec{x})$ to D and minimize the distance measure:

$$\frac{1}{N} \sum_{i=1}^N dist(F(\vec{a}, D)) \rightarrow minimum \quad (3.8)$$

It is difficult to solve the optimization problem in general, and we do not guarantee there is a solution for it. In this case, we adapted the method by arbitrarily scaling the parameters to get the right equation of the ellipse equality constraint $4ac - b^2 = 1$ [9] to solve the ellipse fitting problem. Then we can cast the problem into a quadratic

constraint, which also expresses it in the matrix form $\vec{a}^T C \vec{a} = 1$:

$$\vec{a}^T \begin{bmatrix} 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \vec{a} = 1 \quad (3.9)$$

Then the problem reduces to minimize $E = \|D\vec{a}\|^2$, subject to the constraint $\vec{a}^T C \vec{a} = 1$ where the matrix D is defined as $D = [\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n]^T$. By applied the Lagrange multiplier λ and differentiation, the algorithm gets the following system of simultaneous equations

$$2D^T D \vec{a} - 2\lambda C \vec{a} = 0 \quad (3.10)$$

$$\vec{a}^T C \vec{a} = 1 \quad (3.11)$$

It is equivalent to the following:

$$S \vec{a} = \lambda C \vec{a} \quad (3.12)$$

$$\vec{a}^T C \vec{a} = 1 \quad (3.13)$$

where S is the scatter matrix $D^T D$. Then the problem is solved by considering the generalized eigenvectors $S \vec{a} = \lambda C \vec{a}$. If (λ_i, u_i) solves this, where λ_i is the eigenvalue and u_i is the eigenvector, then so does $(\lambda, \mu u_i)$ for any μ . According to $\vec{a}^T C \vec{a} = 1$, we can figure out the parameter μ , and we can find the value of $\mu_i^2 u_i^T C u_i = 1$ giving

$$\mu_i = \sqrt{\frac{1}{u_i^T C u_i}} = \sqrt{\frac{1}{u_i^T S u_i}} \quad (3.14)$$

Finally, setting $\vec{a}_i = \mu_i u_i$. Follow this algorithm, it yields six eigenvalue-eigenvector pairs (λ_i, u_i) . Each of them gives a local minimum if the term under the square root of the solution of μu is positive. S is positive, so the denominator $u_i^T S u_i$ is also

positive for all u_i . Therefore, the square root exists if $\lambda_i > 0$, so any solutions for figuring $\vec{a}$ must have positive eigenvalues.

The problem of minimizing $E = \|D\vec{a}\|$ subject to $4ac - b^2 = 1$ generates exactly one solution for $\vec{a}$, which has been proved in [8]. Now, we can transfer the ellipse equation $F(\vec{a}, \vec{x})$ into

$$(x - cx)^2 + \lambda^2(y - cy)^2 = r^2 \quad (3.15)$$

where (cx, cy) is the center of the ellipse. The object surface is give in the format of $z = f(x, y)$ in a left-handed Euclidean coordinate system, in which the viewer looks in the direction of the positive Z -axes. The equation of the sphere can be written down as

$$(x - cx)^2 + \lambda^2(y - cy)^2 + z^2 = r^2 \quad (3.16)$$

The gradient of a given point on the reference sphere can be expressed as

$$p = -\frac{x}{z}, q = -\frac{y}{z} \quad (3.17)$$

Then the surface normal of that point can be presented as $[p, q, -1]$.

3.3 Match From Reference Images for Target Images

Above all, the least-squares ellipse-fitting algorithm helps to compute the gradient for each point on the reference object. Now we need to set up the mapping relationship between the gradient and the brightness. Given that the brightness data is more than 3 images, the next step is to think about how to store the high-dimensional data. Also, we should think about what kind of storage is convenient for the next step, that is, searching for the gradient of the irregular object. There are three factors that should be taken into consideration: the table's size, the searching speed, and the hit accuracy. In general, the task of finding the reference object brightness vector closest to the target object's brightness vector is a nearest-neighbor search problem. Some techniques have already been applied for this purpose. The first

method is the lookup table [28]; it stores the gradient by using the brightness as the index; the total size of the lookup table depends on the dimension of the brightness. If the dimension of the brightness is 3, then the size of the table would be $(2^d)^3$. Also the table is sparse, and so time-consuming weighted interpolations are also required to fill the empty entries in the table. Although the lookup table’s initialization is slow, this method is suitable for the search when the dimension is less than 4. However, when the dimensions increase, this method can not do the job well. Hertzmann & Seitz’s work, they used the ANN (approximate nearest neighbour) method to solve the problem. Basically, it uses the $k - d$ trees [3] to solve the problem. To some degree, it can solve the problem well, but when the dimension exceeds 10 – 20 [10], it is also highly time consuming. Recently, a new method, locality sensitive hashing [19], has been introduced in database applications, usually in the context of high-dimensional similarity searching. It preserves the fast character of hashing, but also adds in the property of similar points that are clustered together. The following section will describe in detail how to apply locality-sensitive hashing in the photometric stereo.

3.4 High-Dimension Locality Sensitive Hashing

3.4.1 Notation

To determine the geometry of the target sphere, we need to compute the target object’s surface normal. In other words, we need to figure out the object’s gradient. We already have the mapping relationship between the calibration object’s brightness and its gradient. For the target object, the job is to figure out its gradient distribution by using the mapping relationship according to its high dimensional brightness vector. The main goal is to find the target object’s best brightness vector to match the reference object’s brightness vector, and use the reference object’s gradient as the target object’s gradient. Then the problem can be posed as a **Nearest**

Neighbor Search (NNS) problem. The NNS problem is also defined as ϵ -NNS:

Given a set P of points in a normed space l_p^d , preprocess P so as to efficiently return a point $p \in P$ for any given query point q , such that $d(p, q) \leq (1 + \epsilon)d(q, P)$, where $d(q, P)$ is the distance of q to the its closest point in P .

For the purpose of devising a main memory algorithm for solving the ϵ -NNS) problem, Indyk & Motwani [19] introduced the Locality Sensitive Hashing (LSH) technique for high-dimensional data. In our case, we adapted the original LSH algorithm [10] to search for matches in the high dimensional brightness vector spaces.

The basic idea of LSH is to hash the high dimensional points so as to ensure that the probability of collision is much higher for points that are close to each other than for those that are far apart. A hash function is a function that converts input from a large domain into output in a smaller range. In our case, the input data are the image size brightness vectors $p = [x_1, \dots, x_d]$; however, there are 2^{bd} possible values of p while we may have on the order of 10^5 pixels in the image of the calibration object. Therefore, our input data is from a large domain, but the cardinality of the input data is small compared to its domain. Based on that, using hashing to solve the matching problem is reasonable. In the following, we will present the algorithm in detail.

Let l_r^d denote the Euclidean space R^d under the l_r norm, where the point vector p is $[x_1, \dots, x_d]$. The length of the vector is $(|x_1|^r + \dots + |x_d|^r)^{1/r}$. $x_{d'}$ is the brightness of a certain pixel in image d' , where $1 \leq d' \leq d$. Furthermore, $d_r(p, q) = \|p - q\|$ denotes the distance between the points p and q in l_r^d . Also H^d presents the d -dimensional Hamming metric space; it could be understood as the space of binary vectors of length d under the standard Hamming matrices.

3.4.2 The Algorithm

First of all, we illustrate Indyk & Motwani's [19] algorithm in terms of inputs represented as binary vectors using the standard Hamming metric. Let C be the largest coordinate in the whole points set P . Then we can embed P into the Hamming cube $H^{d'}$ with $d' = Cd$, by transforming each point $p = (x_1, \dots, x_d)$ into a binary vector.

$$v(p) = \text{Unary}_C(x_1) \dots \text{Unary}_C(x_d) \quad (3.18)$$

where $\text{Unary}_C(x)$ is the unary representation of x , i.e., is a sequence of x ones followed by $C - x$ zeroes. For example, if x is 67, and C is 255, then

$$\text{Unary}_C(x) = \overbrace{1 \dots 1}^{67} \overbrace{0 \dots 0}^{255-67=188} \quad (3.19)$$

For any pair of points p, q with coordinates in the set $1 \dots C$,

$$d_1(p, q) = d_H(v(p), v(q)). \quad (3.20)$$

Where d_1 is the distance under norm l_1 and d_H is the Hamming distance function. That embedding equation preserves the Hamming distances between the two high-dimensional points. The Hamming distance between points $[78, 45, 30]$ and points $[82, 56, 89]$ with $C = 255$ is $4 + 11 + 59 = 74$. By doing this, we discover the connection that allows us to express the distance between a pair of high-dimensional points. The result also showed that the closest points have the minimum difference for all the dimensions of the points. Therefore, in the sequence we can concentrate on solving ϵ -NNS in the Hamming space $H^{d'}$. The unary representation in Hamming space provides us with a straightforward framework for the algorithm. However, it is very expensive to convert the data into unary representation when C and dimension d is large. For this reason, we need to work out a modified algorithm and in time make it run independent of C .

Then, we implemented an improved LSH algorithm without converting the high dimensional data to the Hamming space. The details of the algorithm are

as follows. l is the number of hash tables, and there are l subsets $I_1 \dots I_i \dots I_l$. The dimension of the data is denoted as d . Let $p|_I$ denote the projection of vector $v(p) = [x_1, \dots, x_d]$ on the coordinate set I , and let $x_{d'}$ be the brightness of point p in image d' , where $k \leq d$. We compute $p|_{I_i}$ by selecting the coordinate positions and applying the hashing function $g_i(p)$ to a certain bucket of the hash table T_i , $i \in l$. Denote $g_i(p) = p|_{I_i}$. Through that, we project the high-dimensional data $v(p)$ to a certain hash number $g_i(p)$. By the preprocessing, we store each $p \in P$ in the bucket $g_i(p)$, for $i = 1, \dots, l$, and get l compact hash tables for the next step's nearest neighbour searching. By computing the projection per I_i , we get the whole projection $p|_I$. As the total number of buckets per hash table may be large, we compress the buckets by assigning them to a standard fixed chaining number of bucket hashing. Thus, there are two levels of the hashing technique for a certain pixel point p :

- Apply the LSH function to map the high-dimensional brightness vector p to the bucket $g_i(p)$.
- Use the standard hash function to map the contents of these buckets into a hash table of size M .

The maximum bucket size of the latter hash table is denoted as B . In the algorithm the hash table is a chained hash table. When the number of points matching a certain hash number exceeds the size B of the bucket, a new bucket is dynamically allocated and linked to and from the old bucket. However, to process the large amount of data, dynamically allocating the space is not an advisable choice here, since the images' size are large, which means the table is large. In our implementation we use the simpler fixed size buckets method for each hash table item rather than chaining. The approach is as follows: if a bucket in a given hashing index is full, a new matching point cannot be added to it, since it will be added to some other index with high probability. In our application, we put the point into the next bucket

available for hashing indexing. By using the fixed bucket size policy, we reduce the link structure's overhead. The whole structure is clearer and also convenient for the next step's search. The structure of the hash table is presented in 3.2.

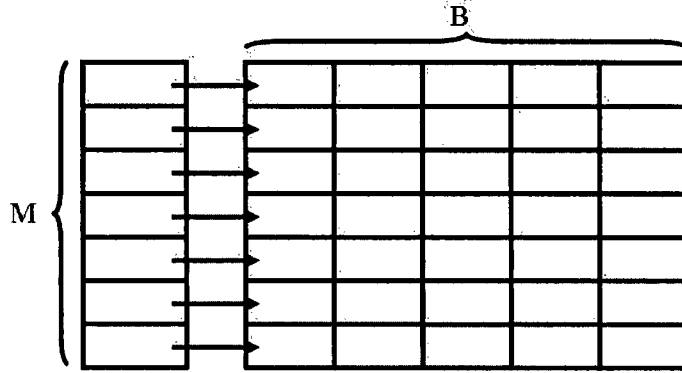


Figure 3.2: The LSH hash tables structure

Each bucket not only stores the high-dimension brightness vector but also keeps its correspondent gradient vector $[p, q]$. The brightness vector in our case is stored as *unsigned char* set and the 2D gradient vector is saved as two *float* number. After that, we must discuss how to determine the suitable parameters l , the number of the hash tables and M , the size of the hash tables. The number n of points from the reference object's surface, the size M of the hash table, and the maximum bucket size B are related in the following equation:

$$M = \alpha \frac{n}{B} \quad (3.21)$$

where α is the memory utilization parameter. Basically, $\alpha \geq 1$. It is the ratio of the memory allocated for the index to the size of the data set, and should be taken into consideration when the dimension is very large and the number of buckets per hash table index exceeds the memory. In our application, we set the whole table size to contain all the data set in the hash table as:

$$M = \lfloor \frac{n}{B} \rfloor \quad (3.22)$$

It is also very important to choose the l – the number of hash tables. In our experiments, we compared the results from using different choices of l . The more hash tables, the more accurate the search would be. However, beyond a certain number, the result does not improve significantly. Then there is no reason to waste space and time to do the search in those redundant hash tables. The main goal is to find the threshold for the choice of l , and to get the best matching results. Basically, from the definition of [10], let $B(p, r)$ notate the set of point elements from P within the distance r from p . Also following is the definition of (r_1, r_2, p_1, p_2) - sensitive as:

- if $p \in B(q, r_1)$ then $P_{rH}[h(a) = h(p)] \geq p_1$
- if $p \notin B(q, r_2)$ then $P_{rH}[h(a) = h(p)] \leq p_2$

where H is the hashing function family, $p_1 > p_2$ and $r_1 < r_2$. Then the l is calculated as

$$l = n^\rho \quad (3.23)$$

and

$$\rho = \frac{\log 1/p_1}{\log 1/p_2}. \quad (3.24)$$

For more detail see [18].

After setting the parameters for the hash table's size and its total number, the hash tables are constructed to store the reference object's brightness vectors, including their corresponding gradient vector. Fig 3.3 shows the initialization procedure for filling the hash tables with the reference object's data.

To process a query q , the brightness vector from the irregular target object, we search all indices hash tables $g_1(q), \dots, g_l(q)$ until we encounter at least $c \cdot l$ ($c \geq 1$) points matching the current brightness vector. Clearly, the number of accesses is always upper bounded by the number of indices, which is equal to l . Let $p_1, \dots, p_t$ be the points encountered in the process of searching in the hash table T_i , by the hashing number index, and we get a fixed number of buckets linked by the index. The job

```

Algorithm Initialization for the hash table
Prehash Points set  $P$  (the high-dimensional data vectors) and  $l$  (number
of hash tables),
for ( $i = 1, \dots, l$ )
    Initialize hash table  $T_i$  by generating a random hash function  $g_i(\cdot)$ 
for  $i = 1, \dots, l$ 
    for ( $j = 1, \dots, n$ )
        Insert the point  $p_j$  to the bucket  $g_i(p_j)$  of the hash table  $T_i$ 
Output Hash tables  $T_i, i = 1, \dots, l$ 

```

Figure 3.3: Initialization of high-dimensional hash table by inserting the points of brightness vectors and their gradient vector.

is to choose the best fit – the closest to q from those buckets. Then approximate $K - NNS$ is applied to minimize the distance between the query point q and the hit bucket's point. The optimal value of K – the number of the buckets per hash index – is chosen to maximize the probability that a point p “close” to q will fall into the same bucket as q , and also to minimize the probability that a point p' far away from q will fall into the same bucket. In general, we may return one best match point from each hash table, and put those best fits from different hash tables together. Then the problem remains as a $K - NNS$ problem for these points, and pick up the “closest” point p to q , and use p 's gradient vector as query point q 's gradient. The query strategy is presented in Fig 3.4.

```

Algorithm Approximate Nearest Neighbor Query
hashquery (A set of  $n$  query points  $q$  from the target object)
for ( $i = 1, \dots, n$ )
for (hash tables  $T_j, j = 1, \dots, l$ )
    generated by the hash table initialization algorithm
Output A set  $S$  of  $l$  (or fewer) approximate nearest neighbors according
to  $q$ 
Return the nearest neighbours for each  $q_i$  found in set  $S$ 

```

Figure 3.4: Approximate Nearest Neighbor query answering algorithm

The LSH function we used takes advantage of a linear combination of high-

dimensional data. By doing this, we also get the distance between any two points. Those brightness vectors are presented as $[x_1, \dots, x_d]$, and we make the LSH mapping strategy the hashing function in our implementation as

$$h(x_1, \dots, x_d) = (a_1 \cdot x_1 + \dots + a_d \cdot x_d) \bmod M \quad (3.25)$$

The permutation hashing method is good for implementation; how to choose the parameters and make this method random enough is very important at this stage. The hashing function should also satisfy the requirement that there be a low probability of collision. Here, we also denote the hash function as $h(\vec{x}) = (\vec{a}\vec{x}^T \bmod M)$.

- $M = 2^C - m, m$ is a prime;
- a_i is a random number from interval $[0, M - 1]$.

where C is a number to make the $M \leq N - N$ is the pixel number from reference image, and a_i is sufficiently random to give low probability of collision. For our experiment, we choose C from 8 to 15. By doing that, the l hash tables have different size of length and different bucket size for each hash index. The hashing should be doing well; otherwise the result will not be satisfactory. By tried some hashing number M and $\vec{a}$, the results of our experiment are reasonably accurate. The results of our experiments are discussed in the next section.

In general, through all the above steps, we first get the reference object's one-to-one mapping relationship between the brightness vector and the gradient vector. Then we store the mapping into the high-dimensional hash tables to preserve the locality similarities between the near neighbour points. For the best fit for the irregular target object's gradient vector, we figure out the best fit from each hash table, then pick up the "closest" point among those best fits and calculate the query point's approximate gradient. As a result, the irregular target object's surface orientation has been figured out.

3.5 Experimental Results

In this section we report the results of our experiments with the locality-sensitive hashing method. The surface orientation's estimations are evaluated in three formats as the figure shows: (i) the slope angle of the gradient $\tan^{-1} \sqrt{p^2 + q^2}$; (ii) the aspect of the gradient $\tan^{-1}(p/q)$; (iii) the needlemap for the surface orientation $[p, q, -1]$.

The first experiment's source image is provided by [28]. The reference sphere (260×310 pixels) (Figure 3.5) and the target doll face (260×380 pixels) (Figure 3.7) were made with ceramic material, which approximates the perfect Lambertian situation. Figure 3.6 shows the reference sphere's contour estimation by the least-square fitting algorithm and its gradient mapping. Both the reference object and the target object images were taken in a calibration manner; the light illumination for the reference object and the target object is identical. Because of the quality of the lighting, using only two hash tables already estimates the doll's surface orientation very well. Figure 3.8 shows the gradient estimation for the doll face by varying the number of hash tables. Figure 3.9 shows the doll face's orientation estimation from a different viewing point.

The second experiment uses the example shown in Figure 3.1: one billiard ball (208×185 pixels) (Figure 3.10) and one plastic elephant toy (371×271 pixels) (Figure 3.12). After painting them with the same red tempera paint, we put them together, and shot a video with the Sony Camcorder. Figure 3.11 shows the reference ball's gradient estimation. We took the video sequence for the elephant toy from three viewing angles. Figures 3.13(t), 3.15(i), and 3.17(i) shows the three examples' gradient estimation with varying the input image number and the number of the hash tables. Figures 3.14, 3.16, and 3.18 illustrated the gradient aspect and needlemap evaluation for the three examples.

The third experiment's source data is derived from [11]. The reference object is also a billiard ball (328×322 pixels) 3.19(e), and the target object is a painted bot-

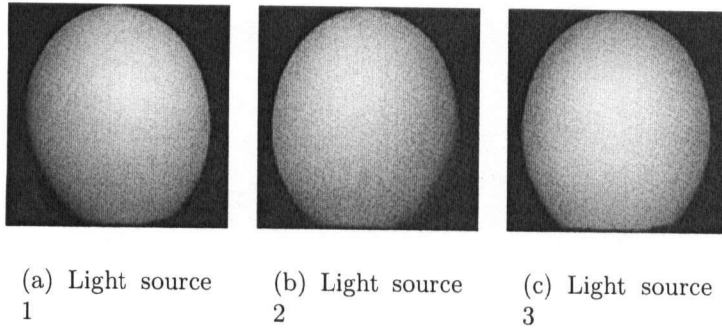


Figure 3.5: Three images of ball under different illumination: (a) light source 1; (b) light source 2; (c) light source 3.

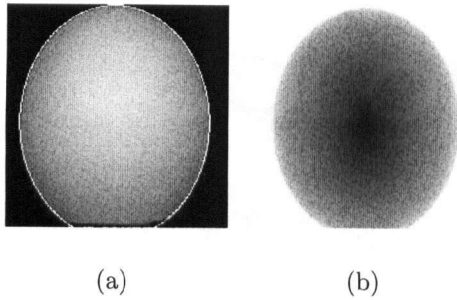


Figure 3.6: The boundary contour of the reference ball and the gradient of the reference ball.

tle (398×1176) 3.19(a). In this case, the object is an example of a non-Lambertian surface with distinct specular highlights. Both the calibration object and the test object appear in the same image; there are 8 colour images. Figure 3.19 shows its gradient estimation. Also, we reconstructed the green bottle's depth map in figure 3.20.

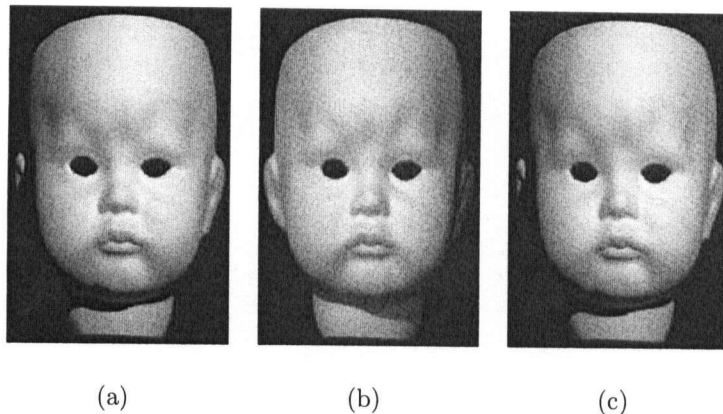


Figure 3.7: Three images of doll face under different illustrations: (a) light source 1; (b) light source 2; (c) light source 3.

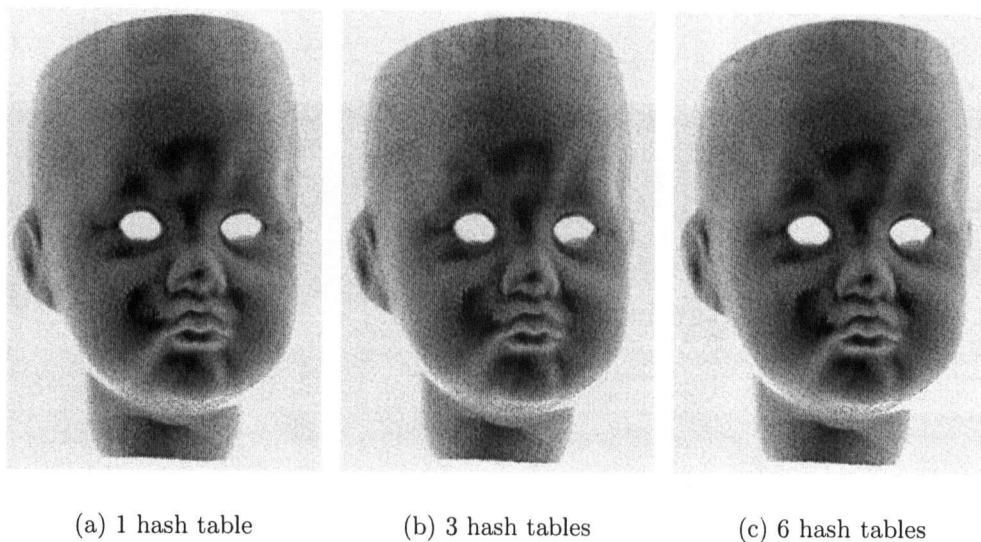


Figure 3.8: The gradient estimation of the ceramic doll face by varying the number of hash tables. Since the input images are acquired from the original photometric stereo method, and they are quite accurate. Therefore, only three hash tables already give a very good estimation of the gradient. The estimation used 3 hash tables is much smoother than the one only used 1 hash table. We could see the difference from the forehead part.

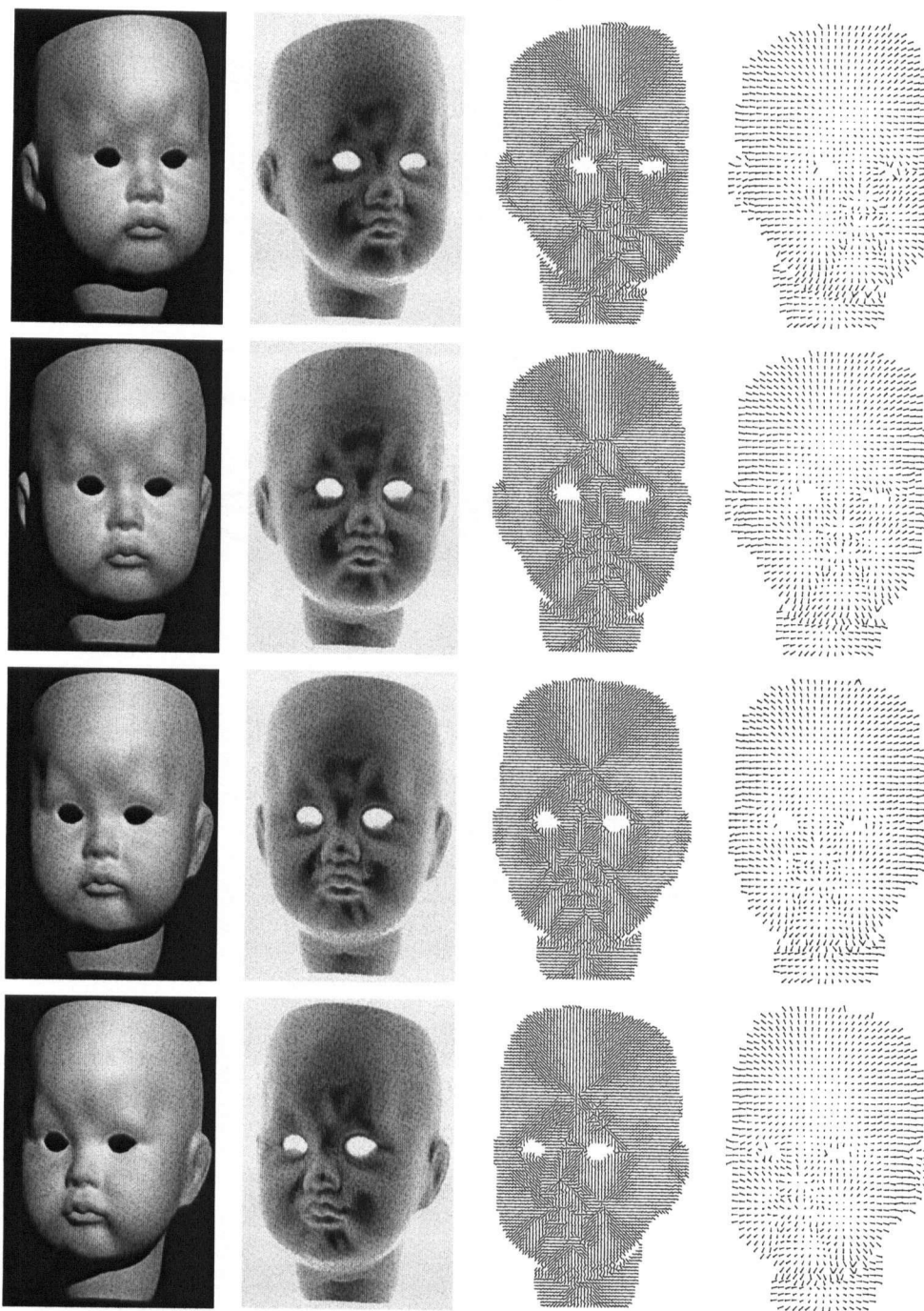
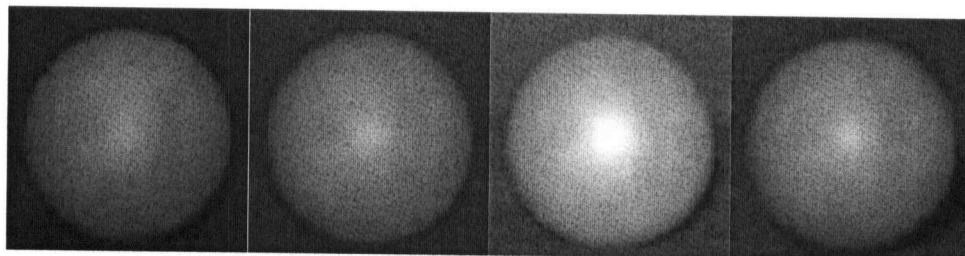
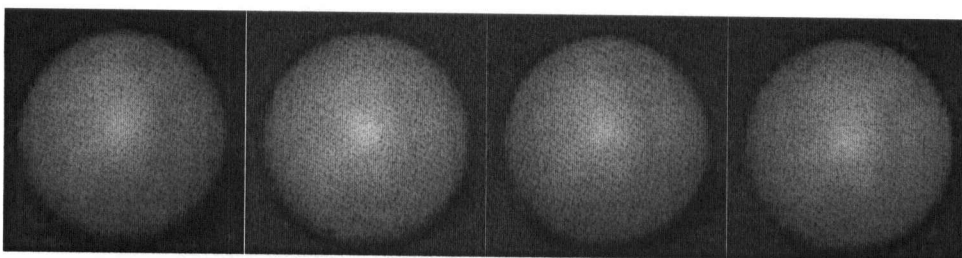


Figure 3.9: Gradient estimation of doll face from different angles, including the gradient slope angle and gradient aspect estimation



(a) Light source 1 (b) Light source 2 (c) Light source 3 (d) Light source 4



(e) Light source 5 (f) Light source 6 (g) Light source 7 (h) Light source 8

Figure 3.10: Eight images of a billiard ball under a hand-held moving light source

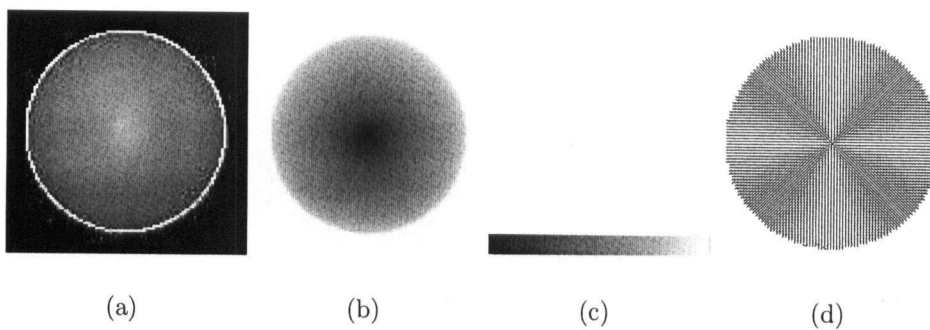
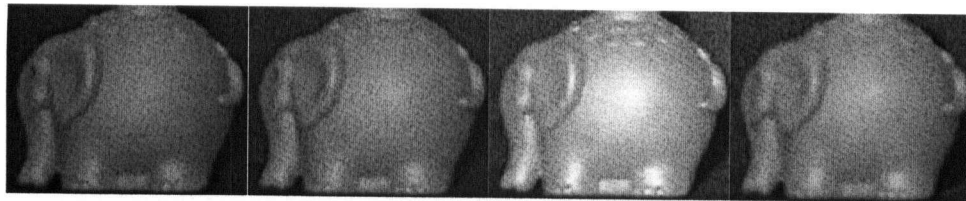


Figure 3.11: The boundary contour of the reference billiard ball and the gradient of the reference billiard ball

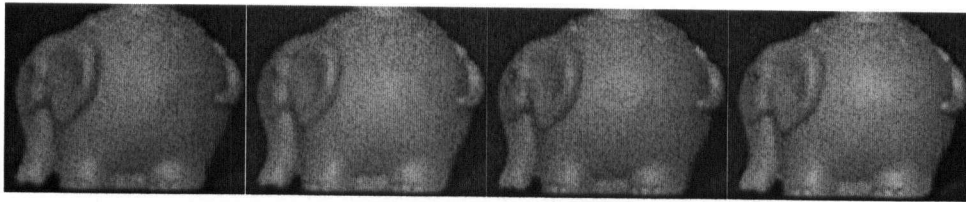


(a) Light 1

(b) Light 2

(c) Light 3

(d) Light 4



(e) Light 5

(f) Light 6

(g) Light 7

(h) Light 8

Figure 3.12: Eight input images of the target elephant under the same light source as the billiard ball

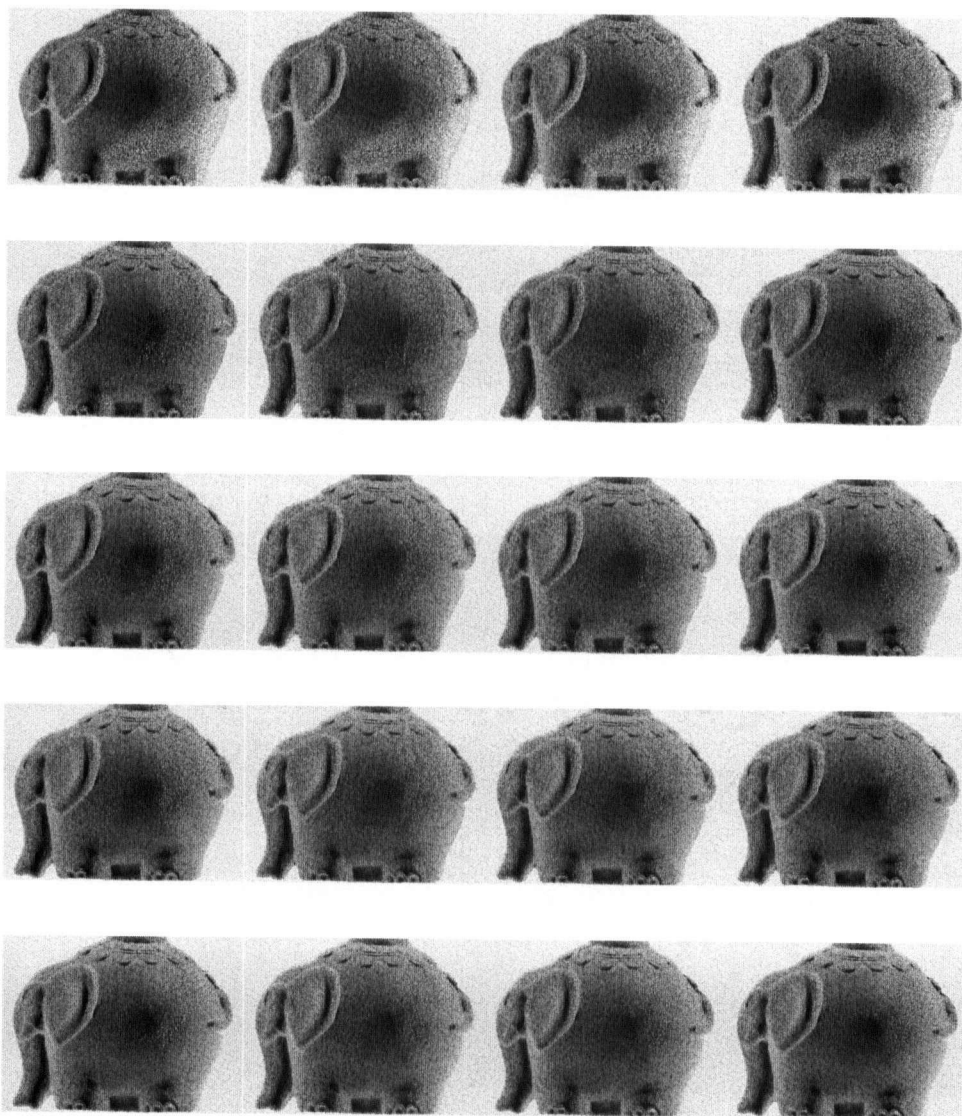
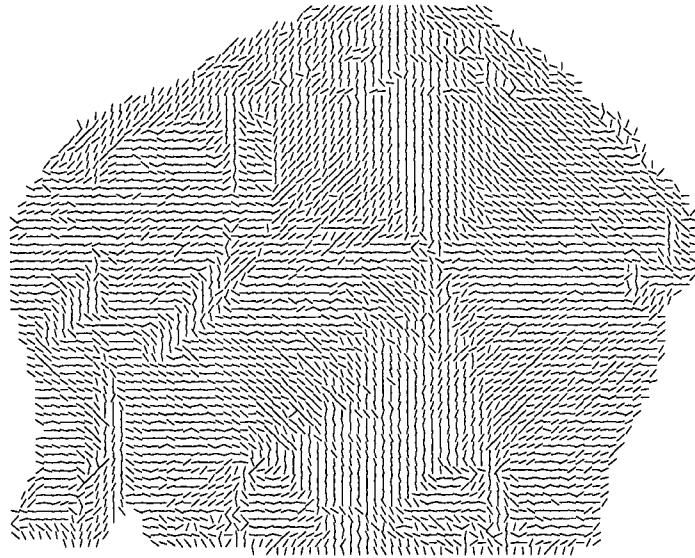
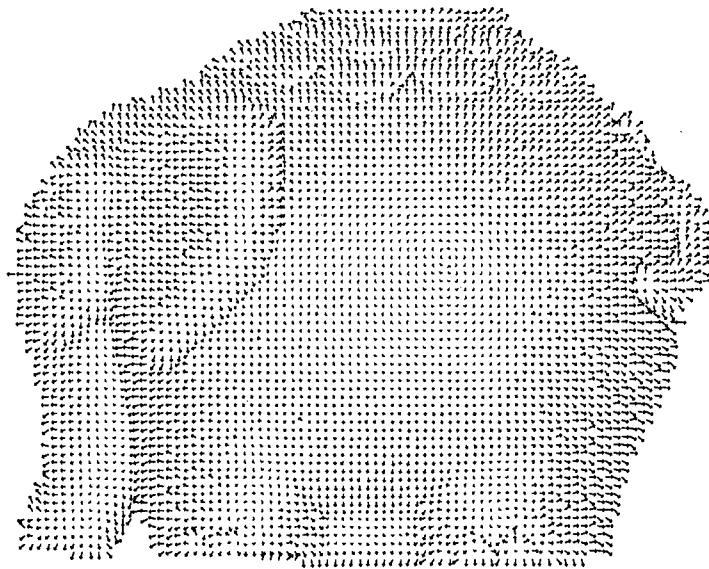


Figure 3.13: The gradient estimation of the target red elephant according to different hash numbers and based on different number sample images. The images show the slope angle of the gradient. Row 1 to Row 5 uses 4, 8, 16, 32, 64 images separately. Column 1 to Column 4 uses 1, 2, 4, 6 hash tables separately.

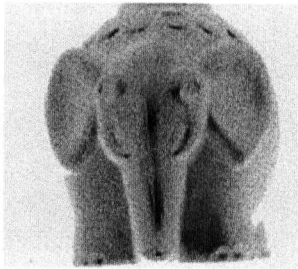


(a) Gradient aspect of elephant's side

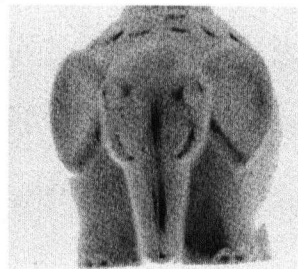


(b) Needlemap of elephant's side

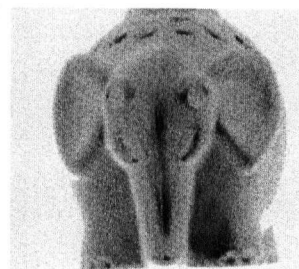
Figure 3.14: Gradient Aspect and Needlemap estimation of elephant's side



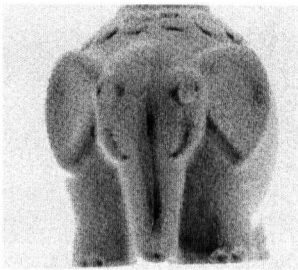
(a) 2 hash tables and 16 images



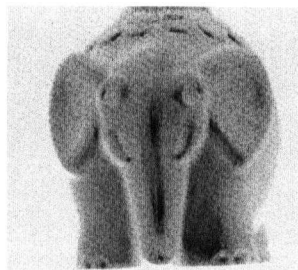
(b) 4 hash tables and 16 images



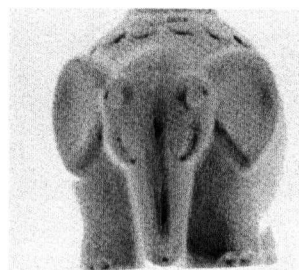
(c) 6 hash tables and 16 images



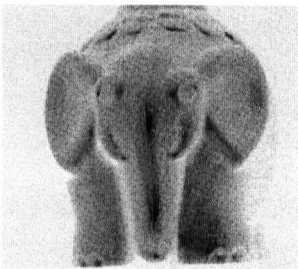
(d) 2 hash tables and 32 images



(e) 4 hash tables and 32 images



(f) 6 hash tables and 32 images



(g) 2 hash tables and 64 images

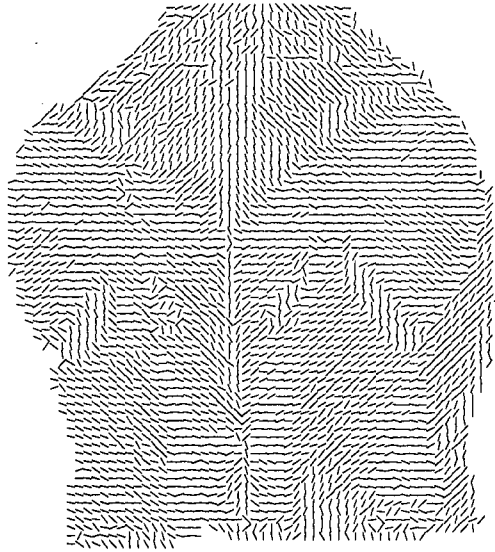


(h) 4 hash tables and 64 images

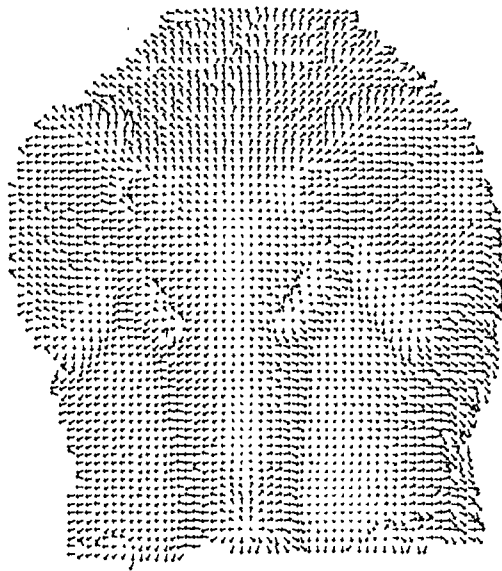


(i) 6 hash tables and 64 images

Figure 3.15: Gradient slope map2: the gradient estimation of the target red elephant according to different hash number and based on different numbers sample images

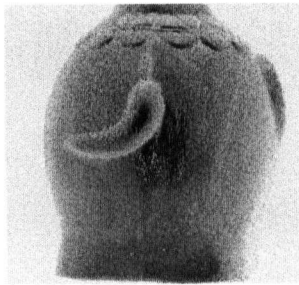


(a) Aspect of front side of elephant

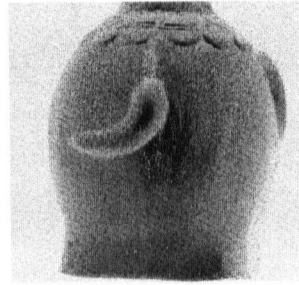


(b) Needlemap of front side of elephant

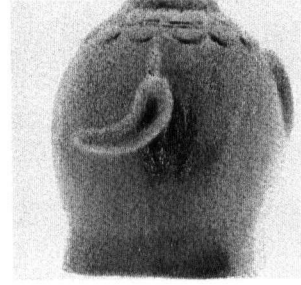
Figure 3.16: Aspect and needlemap of front side of elephant estimation



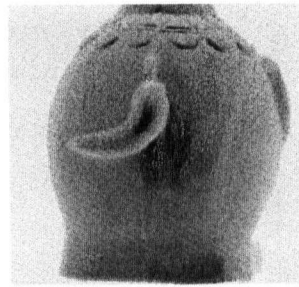
(a) 2 hash tables and 16 images



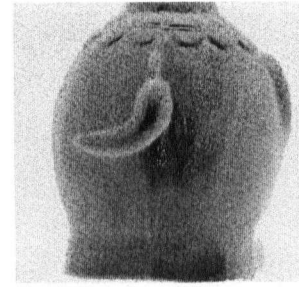
(b) 4 hash tables and 16 images



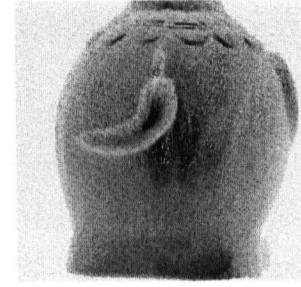
(c) 6 hash tables and 16 images



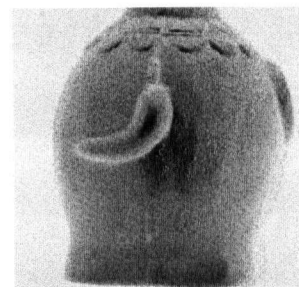
(d) 2 hash tables and 32 images



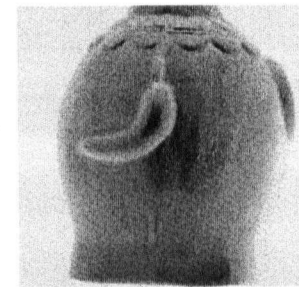
(e) 4 hash tables and 32 images



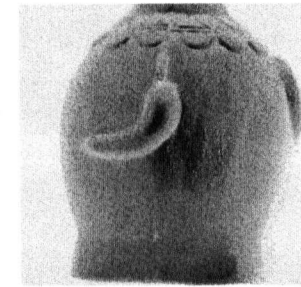
(f) 6 hash tables and 32 images



(g) 2 hash tables and 64 images

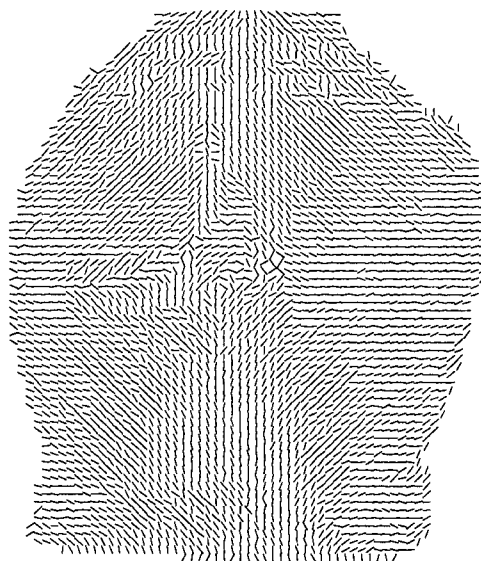


(h) 4 hash tables and 64 images

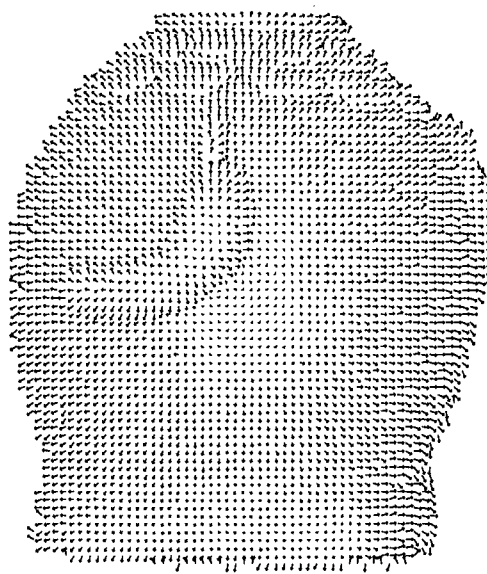


(i) 6 hash tables and 64 images

Figure 3.17: Gradient slope map3: The gradient estimation of the target red elephant according to different hash numbers and based on different number sample images



(a) Aspect of back side of elephant



(b) Needlemap of back side of elephant

Figure 3.18: Back side of elephant's aspect and needlemap estimation

We also analyzed the results by varying the number of images d and the number of hashing l . Figure 3.21 — the plastic elephant experiment — shows that the more input images are used, the more accurate the results. The resolution of the 640×480 camcorder is not high, so we need more images to overdetermine the gradient. The results in Figure 3.22 present the change in the average angular difference when we increase the number of hash tables. The figure shows that the larger l is, the more accurate the gradient estimation. There is a threshold for the choice of l . In Figure 3.22, the threshold is 6. It means that using 6 hash tables already generate a good result for the gradient estimation. Adding more hash tables does not improve accuracy.

For comparison purposes we computed the gradient for the bottle example by brute-force lookup; i.e., we built a table with all the approximately 10,000 pixels in the calibration object. Each entry has a key of 24×8 bits. Computing the gradient by direct lookup and finding the best-matching key by linear search in the table takes approximately 4 hours on a 2.4GHz Intel machine with 2GB of memory. In contrast our LSH version took 5 minutes for the bottle example and 90 seconds for the elephant example with 80×8 bit keys.

We also computed the point-by-point angular difference between the bottle's normal computed by LSH and the normals computed by brute-force lookup. Figure 3.23 shows an image of the angle's cosine between the normals.

In addition, we also compared our results to the laser-scanned results. Figure 3.24 presents the difference between them. We selected several slices of the same position, both from the reconstructed bottle and the laser-scanned model. The reconstructed bottle shows that our result is very close to the laser-scanned model.

Another significant performance improvement of our method is speed. The table size of the lookup-table of Woodham's method is 2^{bd} , where d is the dimension of the images and b is the number of quantization levels. The size of the LSH hash table is $l \times N_r$, where l is the number of the hash tables and N_r is the reference

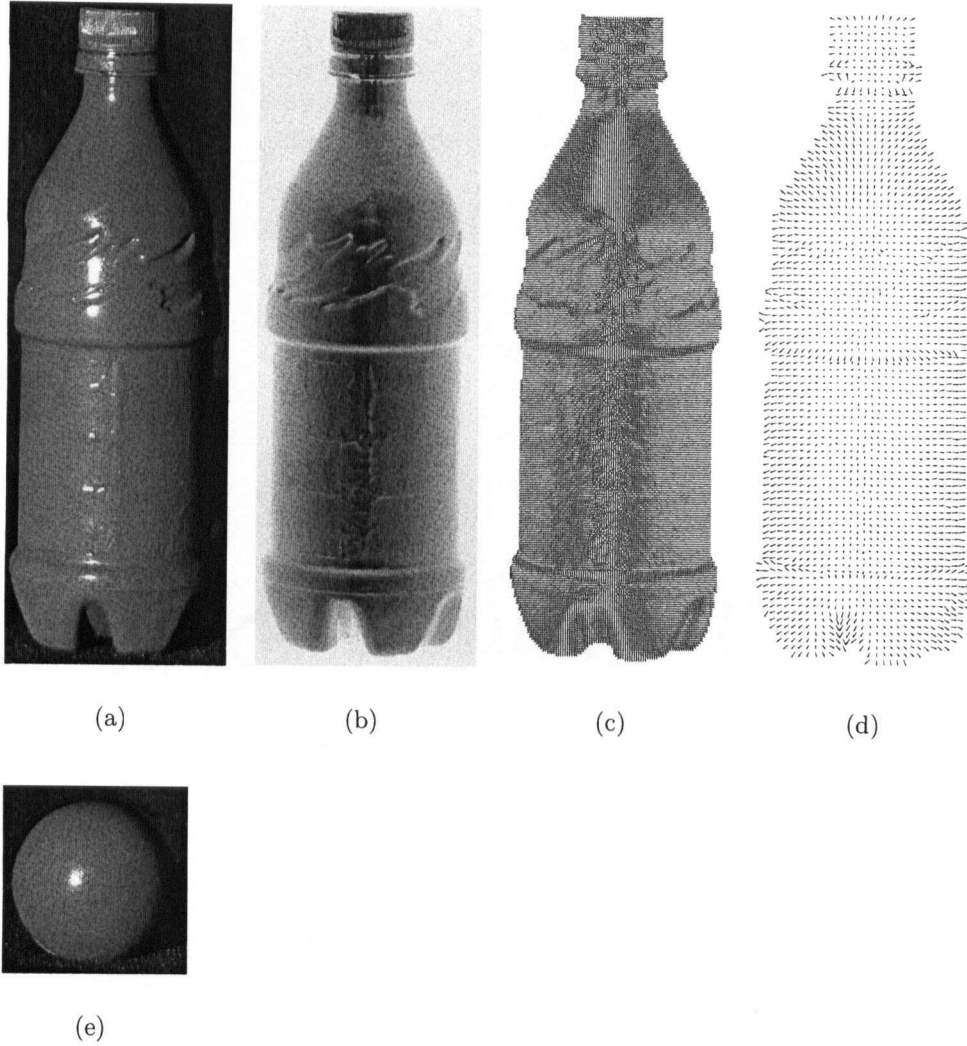
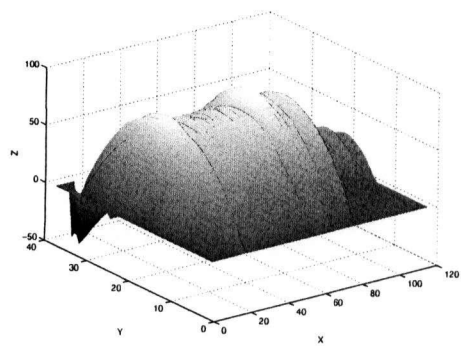
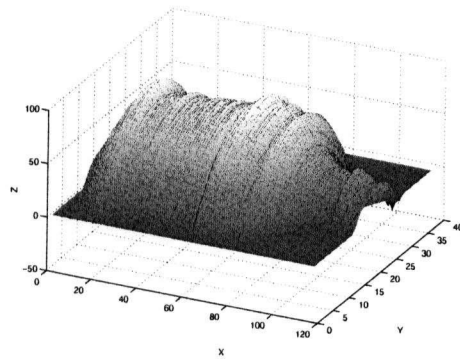


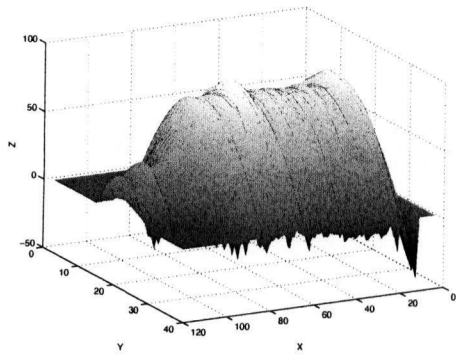
Figure 3.19: The gradient estimation for a painted bottle. (a) shows one input image. (b) shows one reference image (c) shows the slope angle of the gradient. (d) shows the aspect of the gradient. (e) shows the needlemap of the surface orientation.



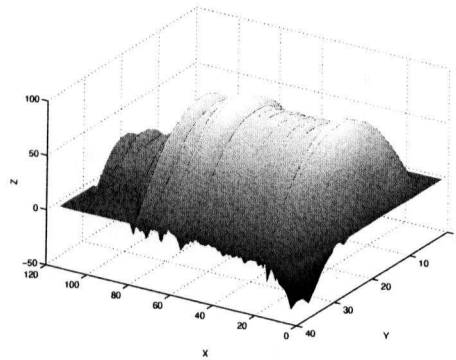
(a)



(b)



(c)



(d)

Figure 3.20: The 3D reconstructed model of the green bottle

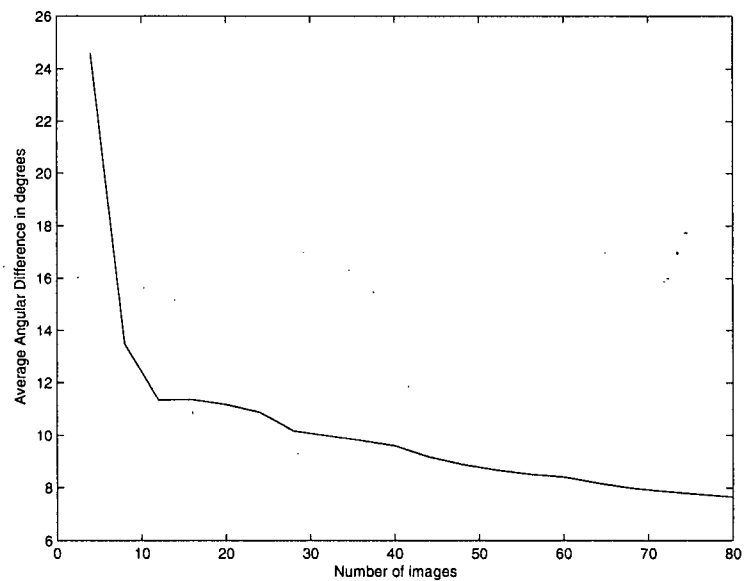


Figure 3.21: Convergence performance with varying image numbers. The vertical axis displays the average angle between the normal computed with the LSH method using the first n images and 6 hash tables, and the normal computed by brute force with 80 images, for the plastic elephant example.

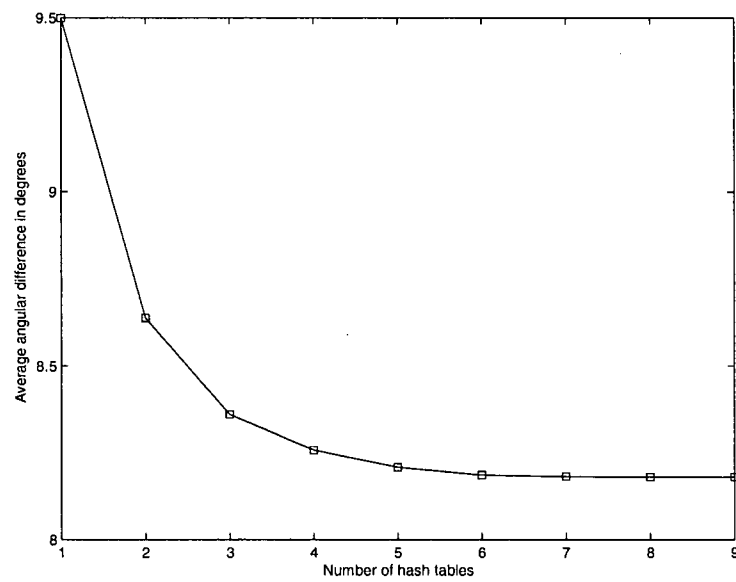


Figure 3.22: Convergence performance with varying the number of hash tables. The vertical axis displays the average angle between the normal computed with the n hash tables and 64 input images, and the normal computed from brute force search from 80 input images for the plastic elephant example.

objects' image size exclusive of the background size. The search is $O(N)$ problem. Here we assume the pixel numbers in reference image (N_r) and the target image (N_t) are similar size N . The table size for brute force search method is $d \times N_r$, where d is the input image numbers. The computational complexity is $O(N^2)$. For ANN matching method, the table size is $d \times N_r$, and the computational complexity is $O(N)O(1/\epsilon)^d$ [19], which is exponential in d . In our implementation, the table size is $l \times N_r$ the total number of operations is $(d + l \times B) \times N$, where l is the number of hash tables and B is the bucket size of the hash items. Thus the computational complexity is $O(N)$; i.e., the LSH hashing is constant per pixel. Evaluating the hashing function is simple, and the tables can be limited in size, so the results can be computed rapidly. There is a cost involved in setting up the hash table, but we envision inspection applications where that cost can be absorbed at the beginning, and only inexpensive hashing operations are done per object.

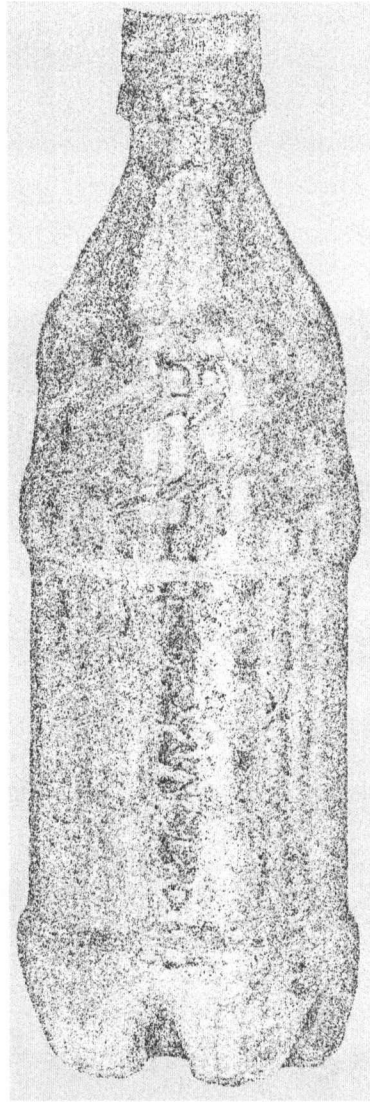


Figure 3.23: The cosine of the angle between the gradient computed by LSH and the gradient computed by brute force lookup. The image shows values between 1.0 in white and 0.985 (approximately the cosine of 10 degrees) in black.

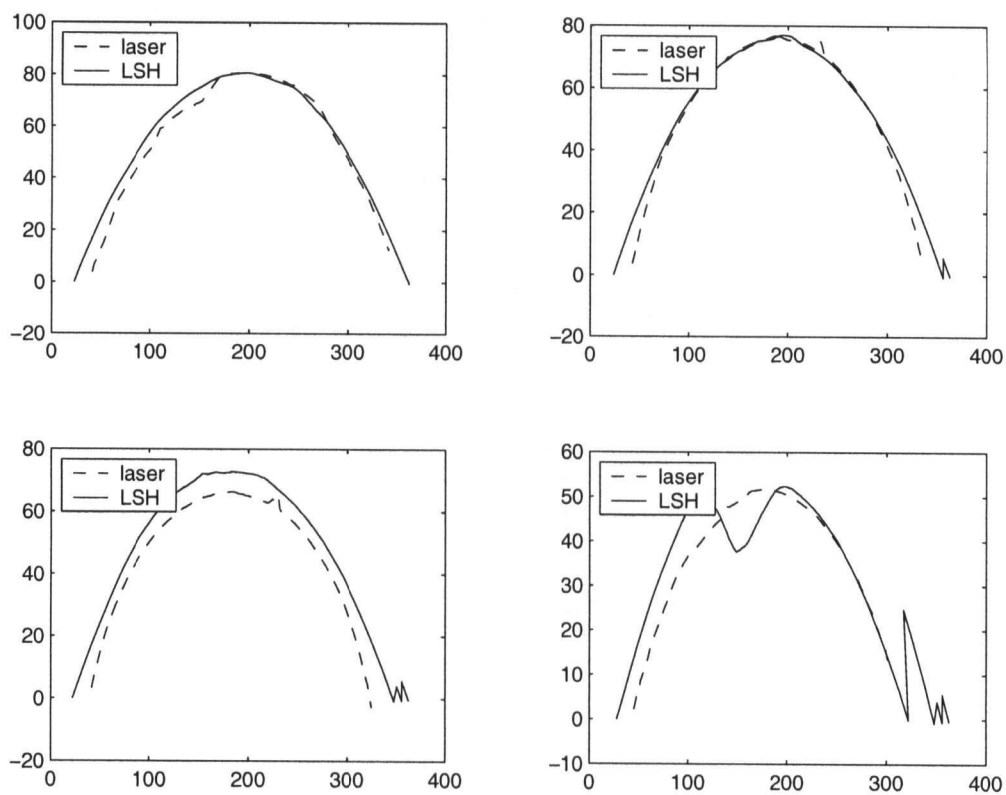
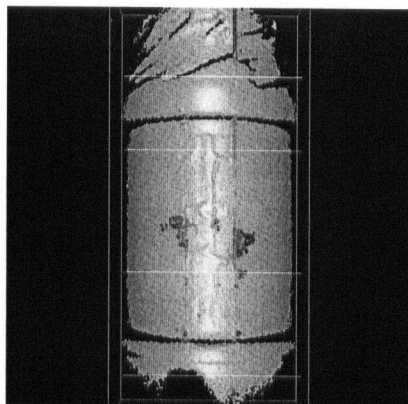


Figure 3.24: Comparison between bottle reconstructed and laser scanned by different slices marked in white lines (note spike in the laser scan). Median error over the entire scan is 0.5 mm

Chapter 4

Conclusion

In this thesis, we present a new method that uses locality sensitive hashing to rapidly compute the high-dimensional data's best match for photometric stereo. The advantage of putting the reference object and target object in the same scene, following Hertzmann & Seitz, is that there is no need to move the reference object and target object one by one to get the same illumination, which largely simplifies the experimental process. Through this method, we can only move the easy handle reference object around the hard to move target object and get its surface orientation. According to the robustness of the algorithm, we found that low-resolution video sequence input image frames (the toy elephant example) also generated a satisfactory gradient estimation. The multiple results show that this method works very accurately for both Lambertian and non-Lambertian cases, and the computation time is much reduced.

The simplified experiment and the robust searching algorithm broaden the practical applications for photometric stereo. The idea is simple but very useful. The practical benefit of the fast photometric stereo implementation is that it could be integrated with motion detection. It provides an efficient reconstruction for an object's geometry, which can be an appropriate alternative for laser range sensors. The data acquisition process—taking photos by cameras—is not harmful to the

patients, and also is cheaper than the laser range sensors and also easy to control. The fast speed LSH photometric stereo method would allow the viewer to observe the surface model immediately, and also enable the viewer to control the model for where he wants to look more in detail at the same time. By doing so, it would save a lot of time for both the patients and the doctor to make other appointments for further examinations by slow photometric stereo method.

Moreover, the efficient photometric stereo via LSH could also be embedded into a robot to get better path exploration by detecting the surrounding obstacles' shape. Also out of the efficient matching method, we could use more than one reference object to build up the search hash tables, which would not affect the performance of the searching speed. Based on that, we can further do photometric stereo on colourful objects.

There are some interesting unanswered questions regarding for future work: how can we verify the accuracy of results? For the Laser Scanned result, we find that there are some holes on the model; and for brute force result, we find that the search is based on the accuracy of reference images. How can we bound the number of hash tables and the dimensions of the input images? For different objects and different lighting, do we always need a large number of images? How can we use photometric stereo via LSH to handle varied and colourful objects? Also when moving the camcorder, the auto focus may blur the images, therefore how do we select the subset of images that leads to a fast and accurate solution? Moreover, how do we speed up the algorithm to real time, and how do we derive depth data effectively from orientation data?

Bibliography

- [1] S. Barsky and M. Petrou. The 4-source photometric stereo technique for three-dimensional surfaces in the presence highlights and shadows. *IEEE Trans.PAMI*, 25(10):1239–1252, October 2003.
- [2] R. Basri and D. Jacobs. Photometric stereo with general unknown lighting. *Computer Vision and Pattern Recognition*, 2:374–381, 2001.
- [3] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communication of the ACM*, 18(9), September 1975.
- [4] P. J. Besl and R. C. Jain. Three-dimensional object recognition. *Computer Vision Graphics Image Process*, 33:33–80, 1986.
- [5] P. Brou. Using the gaussian image to find the orientation of objects. *Int. J. Robotics Res.*, 3:89–125, 1984.
- [6] P. H. Christensen and L. G. Shapiro. Three-dimensinal shape from color photometric stereo. *Int'l J. computer Vision*, 13(2):213–227, 1994.
- [7] E. Coleman and R. Jain. Obtaining 3-dimensional shape of textured and specular surface using four-source photometry. *Comp. Graphics and Image Proc.*, 18:308–328, 1982.
- [8] A. Fitzgibbon, M. Pilu, and R. B. Fisher. Direct least square fitting of ellipses. *IEEE Transactions on PAMI*, 21(5):38–55, May 1999.
- [9] A. W. Fitzgibbon and R. B. Fisher. A buyer's guide to conic fitting. In *British Machine Vision Conf.*, Birmingham, England, 1995.
- [10] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Similarity search in high dimensions via hashing. In *VLDB '99: Proceedings of the 25th International Conference on Very Large Data Bases*, pages 518–529. Morgan Kaufmann Publishers Inc., 1999.

- [11] A. Hertzmann and S. M. Seitz. Shape and materials by example: A photometric stereo approach. *Conference on Computer Vision and Pattern Recognition*, 1(9):533–540, June 2003.
- [12] B. K. P. Horn. Understanding image intensities. *Artif. Intell.*, 8:201–231, 1977.
- [13] B. K. P. Horn. Hill-shading and the reflectance map. In *IEEE*, volume 69, pages 14–47, 1981.
- [14] B. K. P. Horn. *Robot Vision*. The MIT Press, Cambridge, Massachusetts, 1986.
- [15] B. K. P. Horn and K. Ikeuchi. The mechanical manipulation of randomly oriented parts. *Sci. Am.*, 251(8), 1984.
- [16] B. K. P. Horn and eds. M. J. Brooks. *Shape from Shading*. MIT Press, Cambridge, Mass., 1989.
- [17] K. Ikeuchi. Determining surface orientations of specular surfaces by using the photometric stereo method. *IEEE Trans. PAMI*, 3:661–669, 1981.
- [18] P. Indyk. Nearest neighbors in high-dimensional spaces. *Handbook of Discrete and Computational Geometry (2nd edition)*, 2003.
- [19] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbor - towards removing the curse of dimensionality. In *STOC '98: Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM Press, 1998.
- [20] G. Kay and T. Caelli. Estimating the parameters of an illumination model using photometric stereo. *Graphical Models and Image Processing*, 56(5):365–388, September 1995.
- [21] S. Magda, T. Zickler, D. Kriegman, and P. Belhumeur. Beyond lambert: Reconstructing surfaces with arbitrary brdfs. In *ICCV 2001*, pages 391–398, 2001.
- [22] S. Nayar, K. Ikeuchi, and T. Kanade. Determining shape and reflectance of hybrid surfaces by photometric sampling. *IEEE Trans. on Robotics and Automation*, 6(4):418–431, 1990.
- [23] M. A. Penna and S. S. Chen. Shape-from-shading using multiple light sources. *Int. J. Intell. Syst.*, 1, 1986.
- [24] William. M. Silver. *Determining shape and reflectance using multiple images*. MIT, Cambridge, MA, 1980.

- [25] Hemant D. Tagare and Rui J. P. deFigueiredo. A theory of photometric stereo for a class of diffuse non-lambertian surfaces. *IEEE Trans. PAMI*, 13(2):133–152, 1991.
- [26] L. B. Wolff, S. A. Shafer, and G. E. Healey. *Physics-based vision: Principles and Practice, Shape Recovery*. Jones and Bartlett, Boston, MA, 1992.
- [27] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1), 1980.
- [28] Robert J. Woodham. Gradient and curvature from the photometric-stereo method, including local confidence estimation. *Optical Society of America*, 1(3):3050–3068, 1994.
- [29] L. Zhang, B. Curless, A. Hertzmann, and S. M. Seitz. Shape and motion under varying illumination: Unifying structure from motion, photometric stereo, and multi-view stereo. In *The 9th IEEE International Conference on Computer Vision*, pages 618–625, Oct. 2003.
- [30] T. Zickler and P. Belhumeur, and D. J. Kriegman. Helmholtz stereopsis: Exploiting reciprocity for surface reconstruction. In *ECCV 2001*, volume 3, pages 869–884, 2002.