Reconstructing an Object's Shape from its Appearance Manifold under Moving Light

by

Yu Qiong

A dissertation submitted to

The Graduate School of
Information Science and Technology

of

The University of Tokyo

in partial fulfillment of the requirements
for the degree of Master of Science

Advisor:

Associate Professor Yoichi Sato

February 2007

Reconstructing an Object's Shape from
its Appearance Manifold under Moving Light

by

Yu Qiong

Abstract

Reconstructing an Object's Shape from
its Appearance Manifold under Moving Light

by

Yu Qiong

Master of Science in Information and Communications Engineering

The University of Tokyo, Institute of Industrial Science

Associate Professor Yoichi Sato, Advisor

This thesis presents a technique for recovering the shape of an object from its appearance manifold composed of a set of images that can be taken from a fixed viewpoint camera under a moving light source. We assume the distant illumination, a convex object shape and the variations of the object's appearance under a moving light are caused by the difference in the surface normals.

For one position of an object, its different appearances under the different illuminations can be seen as a vector in the high-dimensional space. These input images of an object give us an appearance manifold that the embedding structure of the surface normals is hidden inside. So from the high-dimensional input appearance manifold, we can use a dimensionality reduction technique called 'Isomap' to recover the embedding three-dimensional surface normals of the object.

True surface normals of the boundary points can be computed directly from one image by using the sobel filters, then these boundary points are used as the reference points to transform the Isomap result three-dimensional vectors into the true distribution of the surface normals.

The proposed method is available for a wide range of reflectance materials such as plastic, ceramic, steel and some so on. Furthermore, this technique is easy to implement and do not need complex equipment. The only requirement for our method is to take the different images of an object under different lighting directions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The appearance of an object is determined by several factors such as illumination, viewing position, the surface shape and the reflectance property of the object. Changing any one of these factors should lead to the object having a different appearance. Based on the relation among these factors, in the computer vision field, some important inverse problems have been addressed through reflectance analysis [KN06], illumination estimation [HWG05], and the shape construction researches [Sil80, NIK90, HS05, ZTCS99].

In the real world, for most of objects' appearance, these factors are related nonlinearly if the object has complex reflectance properties. Based on the reflectance light range and intensity, there are diffuse reflectance and the specular reflectance models. The most simple reflectance model is a uniform diffuse reflectance model called lambertian model that can be easily computed. And for most of the objects have a combination of these two types reflectance properties, so the estimation of the unknown factor tends to be difficult unless some knowledge about the scene is given a priori. Most of the previous inverse approaches thus estimate some of these factors from images of a scene assuming one or two of the factors are given. How to construct the shape of the complex reflectance property object without too much assumptions makes the topic of shape reconstruction still be a hot research in the computer vision field.

The previous studies have demonstrated that the shape of an object can be recovered from a single image or multiple images of the object. Always these studies based on the images need assume that some knowledge about the scene is given: the illumination is known or the surface materials are known. Furthermore, the constrain of some works can only success in constructing the shape of a lambertian object. For example most of the shape-from-shading approaches estimate an object's shape from a single image assuming distant illumination and uniform lambertian reflectance [ZTCS99].

The classical photometric stereo approach presented in [Hor86, Woo81] recovers the shape of a lambertian object from multiple images of the object taken under known light sources. Photometric stereo has been intensively studied as a fundamental computer vision problem. For instance, uncalibrated photometric stereo approaches estimate the shape of a lambertian object up to a linear ambiguity under unknown lighting [BKY99, BJ03]. Some researches applied other analytic reflectance models to photometric stereo to deal with non-lambertian surfaces [NIK90] and the previous approaches have shown promising results for objects with various surface

materials. Furthermore, some approaches use the combination of the photometric stereo and the geometric method to get a good shape reconstruction result [VH06].

However, as noted in [HS05], real world materials sometimes have complex appearances that prevent us from extracting their shapes by using the analytic reflectance models. To cope with this problem, the use of a calibration object was proposed in the early works on photometric stereo [Sil80, KH86]. In stead of computing a reflectance map based on some analytic reflectance model, the images of a calibrated object with a known shape, such as a sphere(including all of kinds of surface normals), with the same surface materials as a target object was captured under various lighting conditions and used as a empirical reflectance map. A more sophisticated example-based photometric stereo approach [HS05]. can handle the objects with non-uniform surface materials and do not need a particular calibration object for each target object.

Now the question we ask next is, given only images of an object captured under various lighting conditions, is it possible to achieve the shape of the object without any calibration object? Is there any information left in the input images that we can use as reference? Recently, Koppal and Narasimhan presented a novel approach for clustering surface normals of a scene of unknown geometry and surface materials under unknown illuminations [KN06]. Their approach shows how effective it is to analyze the temporal variation in the appearance of a scene for clustering surface normals for representing its meaningful geometric structure.

1.1 The Proposed Method

Our proposed method can directly recover an object's shape from its appearance changes under a freely moving unknown light. In the procedure of shape reconstruction, we need not to make any analytic reflectance model or to cluster the surface normals into the groups for estimating the surface normals. Assuming distant illumination and a convex object shape, the temporal variations in the appearance of the object surface under a moving light source reflect the difference in the surface normals. Through analyzing the difference between the different positions on the object's surface in the different appearances, we can make it possible to discover the shape of the object from this high-dimensional input appearance manifold.

Our proposed method is easy to implement, do not need complex equipment and the requirement in the whole process is the input consists of different images of an object taken from a fixed view position under different lighting directions. Also no calibration is required for lights and camera. In addition, the order of the input images does not affect the estimation results and this lets us be able to gather the input images from a freely moving light source around the object.

Chapter 2

Related Research

2.1 Classical Approach about Geometric and Photometric Stereo Method

Nowadays, in the field of computer vision, the shape reconstruction of the object can be resolved by two main methods: geometry method and photometric method. Geometry method uses the images taken in the different view points under the same light source and photometric method use the images taken from the fixed view point but under different light sources. [Sil80] gives the compare of these two methods and reports: geometry method does well with the rough surfaces with discontinuities of surface orientation and does well with the textured surfaces with varying surface reflectance; photometric method works best on smooth surfaces with few discontinuities and the surfaces with the uniform surface properties. For either method has its own goodness and shortcomings, some researches combine these two methods' goodness to do the shape reconstruction such as [VH06].

2.2 Shape Reconstruction under the Freely Moving Light Sources

We have described if we can know the light source positions and the reflectance model, the determination of the surface normal for the surface point of the object can be achieved only from the images. But in the real world situation, the illumination is unknown or sometimes is set to be freely moving around the test object, the shape reconstruction becomes difficult and not stable [BKY99, H.H94, AD97]. Recently some researches are challenging to solve the problem of shape reconstruction under the unknown illuminations [A.G03, MD05, OR02]. The generalized bas-relief (GBR) ambiguity problem [BKY99] is that when a lambertian surface is observed in fixed view-position images under varying distant illumination, there is an equivalence class of surfaces given by the generalized bas-relief (GBR) ambiguity that could have produced these images. [MD05] gives a solution by using the interreflections to resolve the GBR ambiguity as figure 2.1 showing.

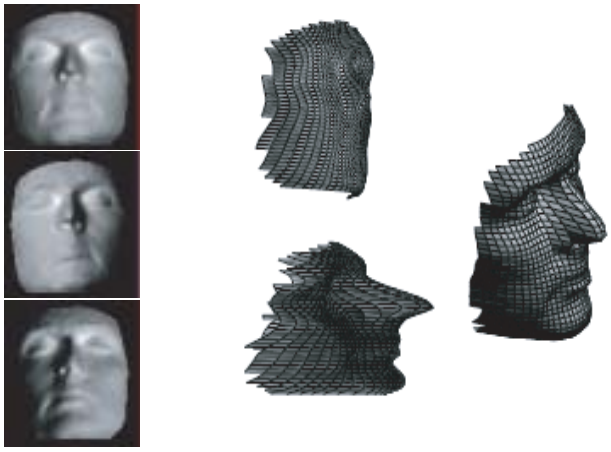


Figure 2.1: GBR surfaces and the GBR ambiguity is resolved by considering the interreflections.

Figure 2.2: The bottle reconstruction result

2.3 Example-based Photometric Model and Analytic Reflectance Model

The example-based photometric stereo method uses the images of a reference object and a target object taken from a fixed view point camera under the same illumination and through finding the points with the same intensities to get the surface normal of the position of the target object from the reference object such as a sphere. But the previous example-based photometric research has some problems such as the scene must have a single BRDF; the reference object must be made of the same material as the target object and need for a calibration object.

Recently, [HS05] presents a method of overcome these problems by some simple observations and using a small number of reference objects(typically two). The orientation-consistency cues that two points with the same surface orientation reflect the same light toward the viewer. From the figure 2.2, the orientation of each point on the bottle in the highlight can be determined by finding the corresponding point on the sphere. This method is available to a wide range of material objects. Another result about a velvet surface shape is reconstructed as the figure 2.3 showing: the right velvet's surface shape is reconstructed from the left reference object with the same material to the target object.

Be different from the classic example-based method, this approach can use the reference

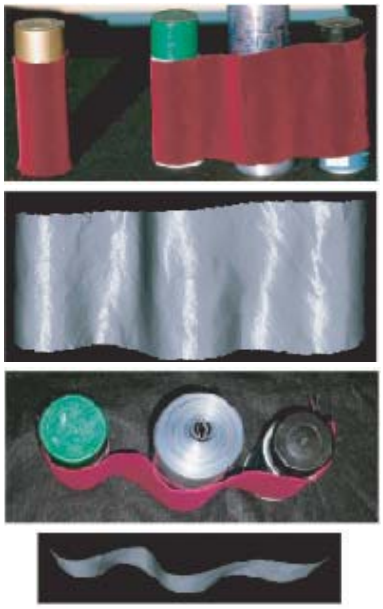


Figure 2.3: The velvet reconstruction result

objects with the different material from the target object to help do the shape reconstruction such as figure 2.4 showing. The left images show that the shape reconstruction result of a target cat by using two different materials spheres as the reference objects.

This research also achieves to do the segmentation for the surface materials as the figure 2.4 right images show the results of the groups of pixels with similar materials but different surface normals. The different colors mean the different materials.

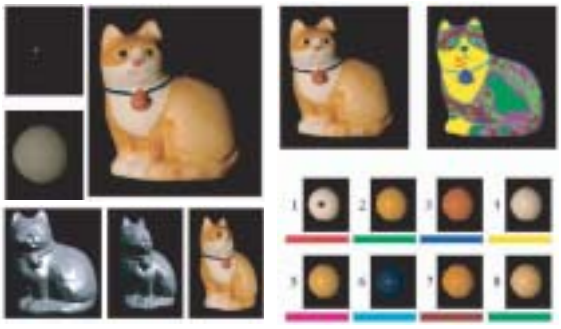


Figure 2.4: Left: Cat shape reconstruction result by using two different materials reference spheres; Right: the analysis on the cat surface materials

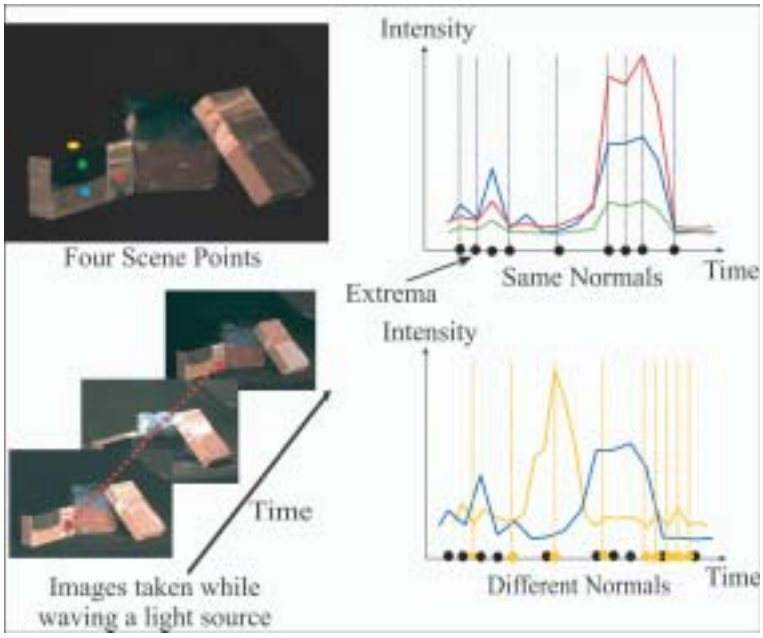


Figure 2.5: Appearance profile and extrema

2.4 Scene Analysis based on the Appearance Clustering

[KN06] performs the iso-normals clusters by analyzing the appearance of each pixel in a scene in different illumination even if the geometry, material and the illumination are all unknown.

In this approach, an appearance profile is defined as a vector composing the intensities measured at a pixel in a static scene under a continuously moving distant point light source, as illustrated in 2.5. For the same normals points, they often have the extrema (peaks and valleys) at the same time instances; for the different surface normals points, the extrema locations are different. This observation makes extrema locations be seen as the good features for this approach algorithm.

Figure 2.6 shows this approach result of the link between extrema and geometry: the top image shows the same surface normals points with different materials share most of extrema and the bottom image shows the different surface normals surface points share no extrema.

The more important effect of the clustering a scene into the regions of same surface normals can help to estimate the other scene properties(shape, material properties and illumination), for example separating diffuse and specular components of the scene, extracting the scene structure and some so on.

Figure 2.7 right images show the result of the objects (in the left images) with the same color means the same surface normals but with the different materials. Figure 2.8 shows another clustering result on doing the separation diffuse and specular components in these books.

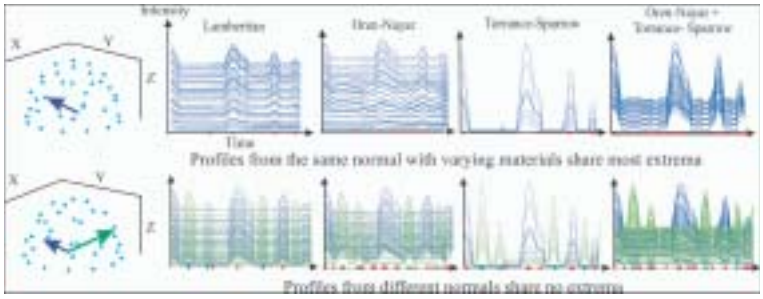


Figure 2.6: Simulated Results to show the link between Extrema and Geometry.

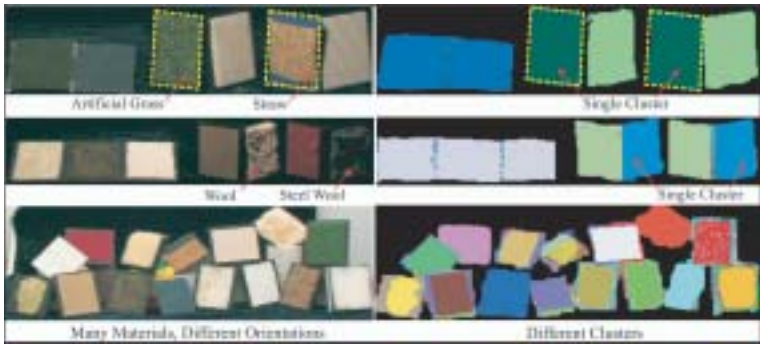


Figure 2.7: Result of the objects with different materials but the same surface normals

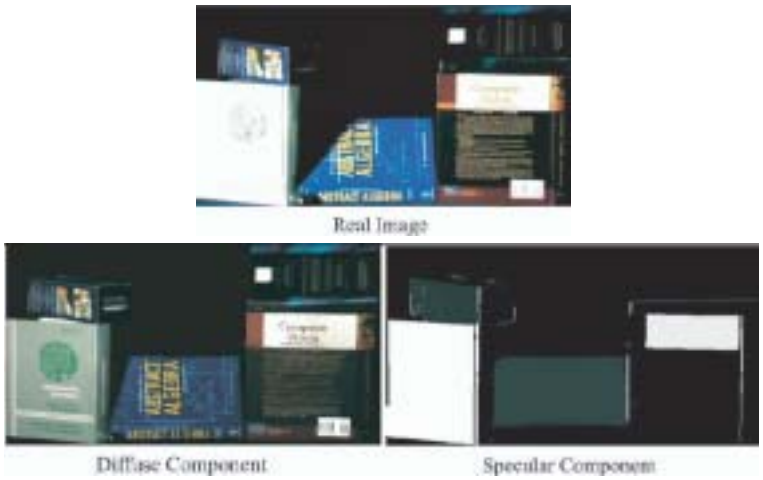


Figure 2.8: The result of separation diffuse and specular components.

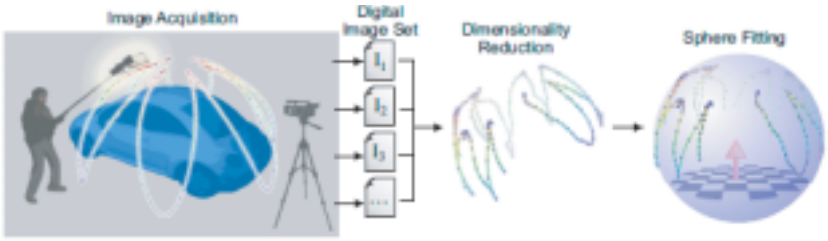


Figure 2.9: System overview: images requirement, dimensionality reduction, sphere fitting.

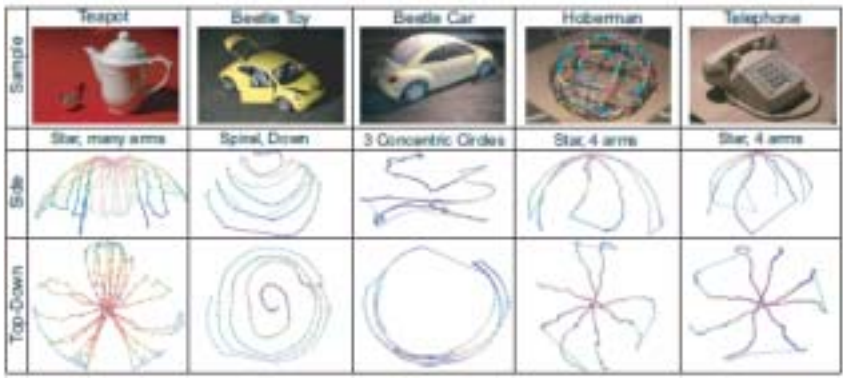


Figure 2.10: Recovery results for a series of objects and the light waving patterns

2.5 Illumination Estimation from the Images

We know that the shape reconstruction of the object can be recovered from the appearance of the object. On the other hand, the illumination estimation can also be achieved from the object's different images. [HWG05] estimates the illumination from a set of images taken from a fixed view-position camera under a hand waving light source. This work uses the dimensionality reduction method 'Isomap' to extract the intrinsic structures describing the illumination distributions from the input images. This algorithm has the merit of simple, fast and reasonably robust and its main steps are shown in the figure 2.9. : gathering the images from a uncalibrated fixed view position camera under a hand held waving light source, doing the dimensionality reduction to get the intrinsic structures in three-dimensional space, fitting the isomap results onto the upper half of a sphere surface to get the illumination vectors. The result light waving patterns are shown in the figure 2.10: the top images are the sample images with arbitrary light position and the bottom images are the result 3D light-waving patterns from the top and side views.

From the introduction of some related researches, we can get a conclusion the shape of reconstruction is still a hot problem in computer vision field and not only the shape reconstruction of the object but also the illumination estimation can be achieved sucessfully only from the appearances of the object in the different unknown light sources.

Chapter 3

Overview of Proposed Method

In this chapter, the overview of our research is explained by analyzing the input appearance manifold, the main steps of this work and the embedding structures extracted by the dimensionlity reduction method. The main task of our research is to estimate the surface normals of the object from the input appearance manifold. The system overview is explained as figure 3.1: gathering an input appearance manifold by taking the different images in the different illuminations; doing the dimensionality reduction; transformation to the surface normals and constructing the 3D shape by the relaxation method.

The assumptions of our approach are required as the following:

- Distant illumination: light sources are sufficiently distant from the object, and thus all light sources project parallel rays onto the object surface.
- Orthographic projection: images of the object are captured from a distant viewing point under orthographic projection.
- Convex shape: the object is assumed to have a convex shape. There are no interreflections or shadows cast on the object surface.

3.1 Input Appearance Manifold

Consider a set of images of an object captured under n different illuminations seen from a fixed view point. Let I_i^p be the intensity of a each pixel or each corresponding surface point p seen under the ith illumination; then the observation vector $\vec{o_p}$ of this pixel p, also known as the *appearance profile*, so this point's corresponding *appearance profile* is:

$$\vec{o_p} = [I_1^p, \cdots, I_n^p]^T . \tag{3.1}$$

From the input images of an object with m surface points ($p = 1, \cdots, m$), we obtain m observation vectors. Note that observation vectors can be also thought of as data points in an n-dimensional vector space. To examine the variations in the appearance of surface points

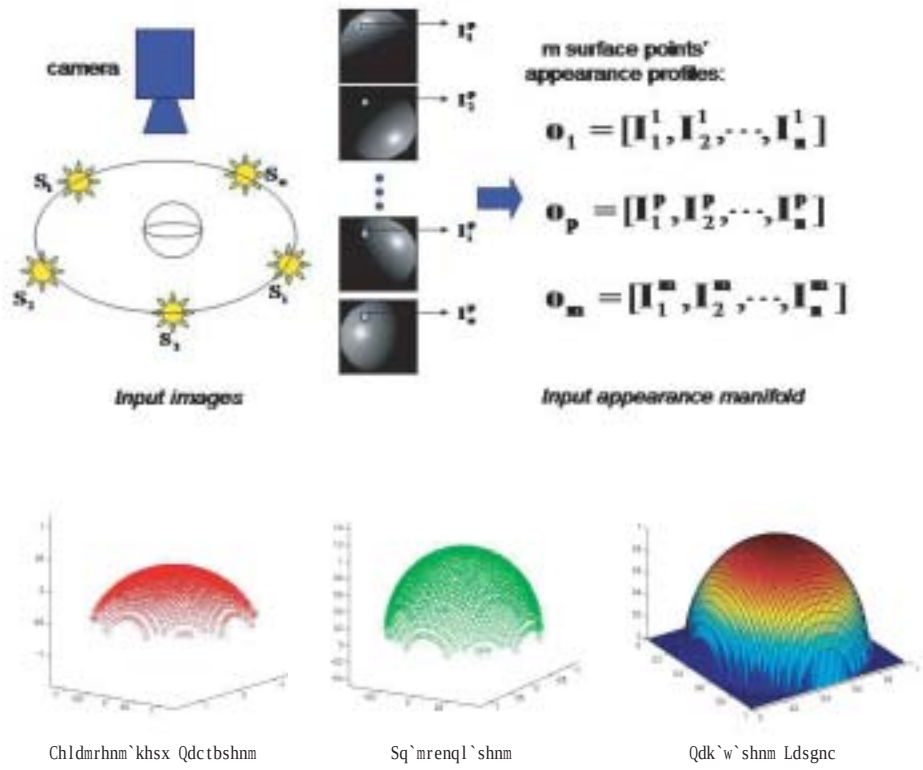


Figure 3.1: System overview including the main steps: Input Appearance Manifold; Dimensionality Reduction; Transformation; Relaxation Method.

under a moving light source, the observation vectors are normalized as that equation: $o_p = \frac{o_p}{|o_p|}$ and used as inputs for the shape recovery. From the assumption of our approach, we know that both illumination and viewing directions are consistent for the pixels in one appearance, so it seems reasonable to think that the variations of the observation vectors reflect the distribution of the object's surface normals and data points o_p lie on a manifold whose intrinsic structure reveals the distribution of the object's surface normals. We refer to this manifold as an *appearance manifold* and use an effective embedding method to find a three-dimensional embedding of this appearance manifold.

3.2 Main Steps of the Proposed-Method

Based on the idea about the differences between the observation vectors reflect the surface normal differences among all surface points, we can get the embedding three-dimensional structure showing the surface normals by using the dimensionality reduction techniques from these

input appearance manifold. The concrete procedure of our proposed method is accomplished by these main steps as follows:

1. **Discover low-dimensional representation**.

   The nonlinear embedding method, a dimensionality reduction, Isomap, is employed for discovering three-dimensional structure underlying the appearance manifold

2. **Transform outputs into a surface normal distribution**.

   Since the converged solution from Isomap does not necessarily correspond to the true distribution of object's surface normals. We use the occluding boundary as reference points to transform the output from Isomap

3. **Estimate shape from surface normals**.

   The height field of the object surface is recovered from the transformed data points by using the relaxation method.

3.3 The Embedding Surface Normals

In the case of an appearance manifold, however, the set of n-dimensional input data generally has nonlinear structures. In order to reveal the intrinsic structures underlying an appearance manifold, we employ a nonlinear embedding method, called isometric feature mapping (Isomap), proposed by Tenenbaum et al. in [TSL00].

Isomap has been used to find perceptually meaningful low dimensional manifolds of natural images, such as the images of a face with different poses and lighting directions. Isomap learns a manifold of input data as a graph by connecting k-nearest neighbors among all data points. A low-dimensional embedding of this manifold is estimated such that the geodesic distances between all pairs of points are preserved even after dimensionality reduction.

In the case of an appearance manifold, differences between all pairs of n-dimensional data points o_p are due to differences in their surface normals. Therefore, if a three-dimensional embedding of this appearance manifold is discovered by Isomap, it should reveal the distribution of surface normals of the object. This is the key idea of our algorithm for shape recovery through the appearance manifold.

In the Chapter 5, we will investigate the validity of the condition of the appearance manifold by using analytic reflection models.

Chapter 4

Shape from Appearance Manifold

In this chapter, we introduce the detailed algorithm steps in the process of the shape reconstruction. As the only input information, the images of the object are required to be taken from a fixed viewpoint camera under the different illuminations. We assume there are a set of images captured under n different illuminations and there are m surface points in one image. So the input appearance manifold is obtained from these images that can be seen as m n-dimensional vectors for m points. As we have defined in the chapter 3, for one surface point p, its corresponding *appearance profile* is remembered as a vector o_p: $o_p = (I_1^p, \cdots, I_n^p)$. The input appearance manifold o is the only input for our algorithm.

4.1 Dimensionality Reduction from the Input Appearance Manifold

4.1.1 Dimensionality Reduction Techniques

The purpose of dimensionality reduction is to find the meaningful low-dimensional structures hidden in their high-dimensional observations and now it is well used in some research fields, such as biology, medical and some so on.

1. MDS and PCA

   For the merit of extracting intrinsic structure from the complex and large data, for some researches on computer vision and compute graphics, the principle component analysis (PCA) and multidimensional scaling(MDS) are the classical and well-used dimensionality reduction techniques. PCA and MDS dimensionality reduction methods are easy to implement and guaranteed to discover the intrinsic embedding structure of data lying on or near a linear high-dimensional space. The PCA does the dimensionality reduction for extracting a low-dimensional embedding that best saving their variance as measured in the high-dimensional original data. The MDS is best preserving the Euclidean distance between the pair of these high-dimensional data points.

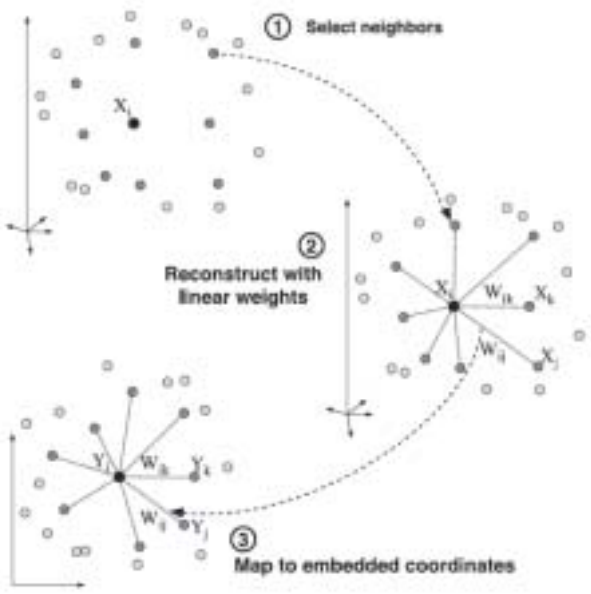


Figure 4.1: LLE three steps algorithm.

2. LLE(Locally Linear Embedding)

   Unlike classical dimensionality reduction methods PCA and MDS, the LLE [TK00] can extract the nonlinear degrees of freedom that hidden in the complex input data. As the figure 4.1 is showing, the LLE algorithm can be divided into these three steps:

   (1) Choosing the neighbors for each data point $\vec{X}_i$.

   (2) Compute the weights W_{ij} that best linearly reconstruct $\vec{X}_i$ from its neighbors and the reconstruction errors are measured by this cost function:

$$\epsilon(W) = \sum_i |\vec{X}_i - \Sigma_j W_{ij} \vec{X}_j|^2 \tag{4.1}$$

   where the $W_{ij} = 0$ if $\vec{X}_i$ and $\vec{X}_j$ are not neighbors; the $\sum_j W_{ij} = 1$ and this cost function can be minimized by solving a least-squares problem [TK00].

   (3) Compute the low-dimensional vectors $\vec{Y}_j$ best constructed by W_{ij} through minimizing the equation 4.2:

$$\Phi(Y) = \sum_i |\vec{Y}_i - \Sigma_j W_{ij} \vec{Y}_j|^2 \tag{4.2}$$

   The detailed algorithm is given in the [TK00] and the LLE method successes in extracting the intrinsic structure from the images of the person's faces as the figure 4.2. The bottom images correspond to the solid top-right path points, illustrating one particular mode of variability in pose and expression.

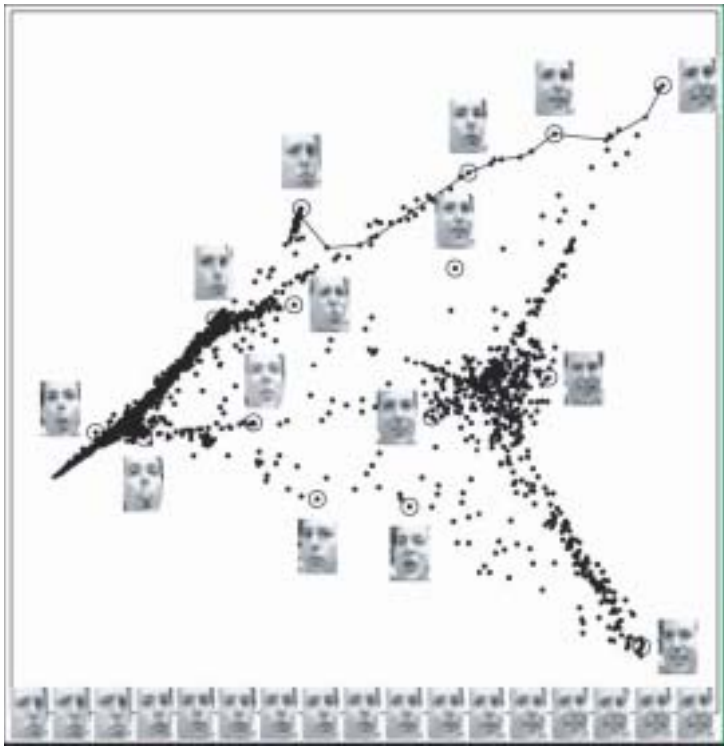


Figure 4.2: LLE result for the face images: the embedding space described by the first two coordinates (pose and expression).

3. Isomap

   Another dimensionality reduction approach called Isomap can also extract the nonlinear degrees of freedom from the complex input data in the high-dimensional space. In [TSL00], the author introduces the detailed isomap algorithm and some successful examples of using isomap to extract the intrinsic structures from some complex images. The isomap method builds on the classical MDS but uses a geodesic manifold distance matrix as the input matrix, so it can best preserve the intrinsic geometry of the data. For some research that using the PCA or MDS can not extract the true structures, the isomap algorithm can success in detecting the true intrinsic structures. Figure 4.3 shows that using the isomap algorithm to some face images gets a good result: the input consists a sequence of 4096-dimensional vectors, representing 698 face images (64pixels by 64 pixels) and isomap learns the intrinsic embedding in three -dimensional space: the x axis shows the left to right pose, the y axis shows the up-down pose and the slider position shows the light direction. Another example is shown as figure 4.4: the result embedding two-dimensional structure extracted from the human handwritings images by using the isomap algorithm.

   The detailed isomap algorithm used in our method will be introduced in the next section.

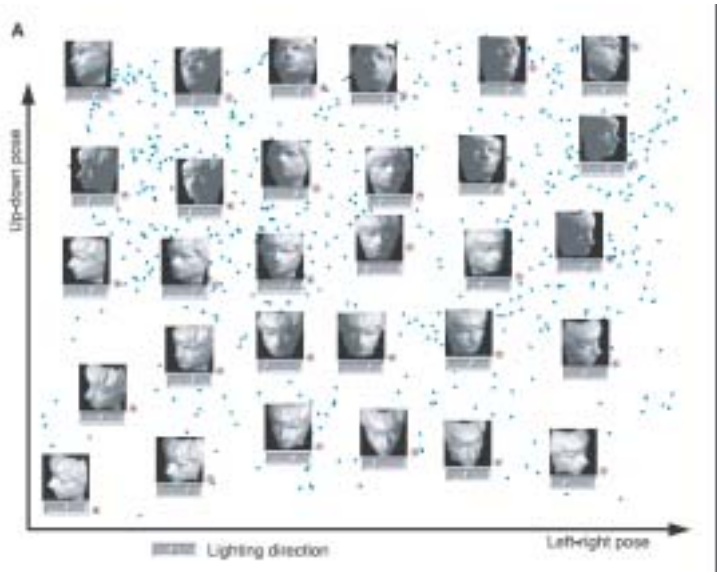


Figure 4.3: Isomap results for the dimensionality reduction to the face images.

Figure 4.4: Isomap results for the dimensionality reduction to the handwriting images.

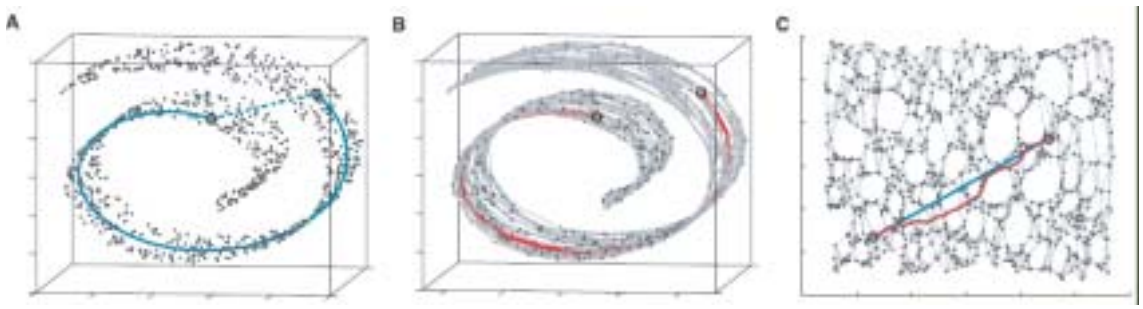


Figure 4.5: Isomap three steps algorithm

4.1.2 Isomap for Embedding the Nonlinear Surface Normals

In our work, the dimensionality reduction technique, Isomap, is used to extract the three-dimensional structures that reflect the surface normals of the object from the input appearance manifold. Isomap takes a input matrix called distance matrix D that is computed between all pairs of pixel' corresponding appearance profiles $o_p = (I_1^p, \cdots, I_n^p), p = 1, \cdots, m$. Here m is the surface points number and n is the illumination number(equals the images number).

So the whole isomap algorithm can be divided into three steps such as the figure 4.5 showing: compute the Euclidean distance matrix; modify the geodesic distance instead of the Euclidean distance and extract the embedding structures. The detailed steps are explained by combining the input appearance manifold for our method as the following:

1. **Input distance matrix**

   We define this input distance matrix $D(i,j) = d(i,j)$ by computing the Euclidean distance between all pairs of pixels corresponding appearance profiles. Before we compute the distance $d(i,j)$, we need to normalize these input appearance profiles by $o_i = \frac{o_i}{|o_i|}$. For example, the Euclidean distance $d(i,j)$ between the pixel i and pixel j, is computed by this equation: $d(i,j) = |o_i - o_j|$. Here the matrix $D(i,j)$ is symmetric: $d(i,i) = 0; d(i,j) > 0, i \neq j$.

2. **Geodesic distance matrix**

   We compute the geodesic distance between each point and its nearest k neighbor points where the parameter k is determined before. For each pixel's corresponding vector o_i, find its nearest k neighbors based on the input distance matrix $D(i,j)$, and modify the distance matrix $D(i,j)$ by using this equation:

$$D(i,j) = \begin{cases} D(i,j) & if I_j is the neighbor of I_i \\ \infty & otherwise \end{cases}$$

   Then compute the shortest distance between each pair of neighbors vectors by the Dijkstra's algorithm and now the matrix D only includes the shortest path for each pairs of neighbor vectors.

3. **Apply MDS to the matrix**

   Use Multidimensional Scaling(MDS) to get the embedding three-dimensional vectors that describe the relative surface normal structures for the surface points on the object. This MDS output is a $3 \times m$ matrix(m is the pixel number). The detailed algorithm of MDS is introduced in the next section.

4.1.3 Multidimensional Scaling(MDS) Algorithm

MDS is a famous and popular data analysis function well used in some research fields. It takes a dissimilarity matrix $D(i, j)$ as the input matrix that describes the distances between all element pairs (i, j) and the results are in another co-ordinate system with the original distance relationships best preserved. If the new co-ordinate d is lower than the original co-ordinate n, the dimensionality reduction has been done. In our algorithm, the classical MDS algorithm is used in the last step of the isomap algorithm and [KJ79] gives that the classical MDS algorithm works as the following steps:

1. **Construct the distance matrix $D = d_{rs}$**

   The distance matrix's element d_{rs} is computed from the Euclidean Distance between the pair of high-dimensional vector r and vector s:

   $$d_{rs} = |V_r - V_s| \tag{4.3}$$

2. **Construct the matrix A, B based on the distance matrix $D = d_{rs}$.**

   The matrix $A = a_{rs}$ is computed from the distance matrix $D = d_{rs}$ by the equation:

   $$a_{rs} = -\frac{1}{2} d_{rs}^2 \tag{4.4}$$

   The matrix $B = b_{rs}$ is computed based on the matrix A by the equation:

   $$b_{rs} = a_{rs} - \frac{1}{n}\sum_{s=1}^{n} a_{rs} - \frac{1}{n}\sum_{r=1}^{n} a_{rs} + \frac{1}{n^2}\sum_{r,s=1}^{n} a_{rs} \tag{4.5}$$

3. **Construct the required coordinates in another d-dimensional space.**

   (1) Find the d largest eigenvalues $\lambda_1, \lambda_2, ...\lambda_d$ of the matrix B. (in our work, d is required to be three),

   (2) Find these eigenvalues corresponding eigenvectors $X = (X_{(1)}, X_{(2)}, ..., X_{(d)})$ and these eigenvectors should be normalized by $X'_{(i)}X_{(i)} = \lambda_i (i = 1, 2, ..., d)$. (Here $X_{(1)}$ is a vector of $(m \times 1)$

   (3) In our method, the required coordinates for the whole m pixels $I_r, (r = 1, 2...m)$ in the d-dimensional space are described as:

   $$I_1 = (X_{11}, X_{12}, ...X_{1d}), ..., I_m = (X_{m1}, X_{m2}, ...X_{md})$$

   .

In our work, the parameter d is set to be three for the surface normals are three-dimensional vectors and m is the surface points number. So the result is a matrix $I = (I_1, ..., I_i, ..., I_m)$ where the I_i is the three-dimensional vector describing the surface normal for pixel i.

4.1.4 Attentions in the Procedure of the Dimensionality Reduction

In this process of dimensionality reduction, there are some detailed places that should be paid attention to as follows:

1. **Determination of the Neighbors Number k**

   As the only one parameter of the isomap algorithm, the choice of neighbor points number k is very important for the whole isomap algorithm. If the neighbors number k is not enough, the whole surface points can not be connected with each other, leading to that the number of output three-dimensional structures is not enough.

2. **Normalization for the Appearance Profile**

   The appearance profile for each surface pixel should be normalized to set to the same length with the other pixels' surface pixels before computing the distance matrix D. This step is very important in the process of the shape reconstruction for the objects with the texture surface. In order to make the length of every pixel's appearance profile be similar, we set the light sources be a good distribution on a sphere such as the figure 4.6 shows the illumination distribution. In the real experiment, that can be obtained through moving the light source around the object at a suitable speed.

3. **Isomap Outputs are not Stable.**

   As Figure 4.7 is showing that these three-dimensional isomap result points are in the different positions of the co-ordinate, we know that the isomap outputs are random, not stable. So the mirror result may appear in the axis x, axis y, axis$z$; In our algorithm, we use the symmetric points of these result points instead to do the next transformation. For example, we use the right figure 4.7 three-dimensional points with the z' value $z' = -z$ instead of the left isomap results z.

The isomap result vectors in the three-dimensional space can not describe the correct surface normals directly because they just show the relative relations between these surface points corresponding surface normals. So in order to get the correct surface normals of these surface points, we must do some modification for these three-dimensional isomap results. The modification process will be introduced in the next section.

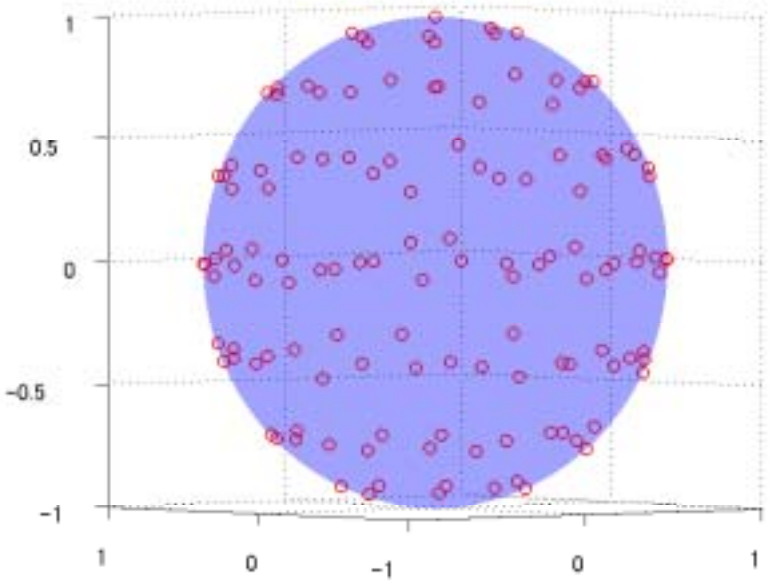


Figure 4.6: Good illumination distribution.

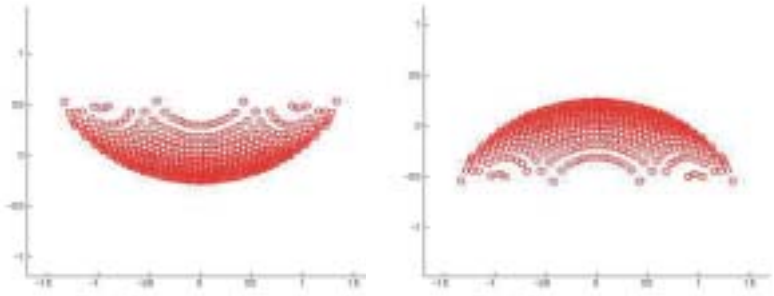


Figure 4.7: Isomap mirror pair outputs.

4.2 Transformation Solution from Isomap into Surface Normal Distribution

There is an important issue when the output from isomap is used for recovering the object shape. Although isomap computes a globally optimal solution, the converged solution does not necessarily correspond to the true distribution of the object's surface normals; what isomap computes is the relative relationships of its surface normals. In our work, we use the occluding boundary of the object as reference to transform the output from isomap to the distribution of surface normals of the object.

The occluding boundary is the curve on the object surface that is projected as a silhouette in the input image. Occluding contours of an object can be found by applying a gradient-based edge detector to the image of the object. The surface normal on the occluding boundary lies in a plane parallel to the image plane.

Let $C_b = (c_x^b, c_y^b, c_z^b)$ be surface normals of those boundary points b. Assuming that C_b are on a plane through the origin $(0, 0, 0)$ with its surface normal $(0, 0, 1)$, c_x^b and c_y^b of those points can be computed as their gradient directions in the 2D image coordinate system defined in the input image. This transforms the viewing direction $\vec{w_o}$ into the direction $(0, 0, 1)$ as well from the definition of the occluding boundary.

Let $E_b = (e_x^b, e_y^b, e_z^b)$ be the output three-dimensional coordinates of the corresponding boundary points from Isomap. A transformation $M : R^3 \rightarrow R^3$ is estimated such that the transformed points $M(E_b)$ minimize

$$\sum_{\text{all } b} \|M(E_b) - C_b\|^2, \tag{4.6}$$

where the correspondences between E_b and C_b are given based on the initial pixel location of o_b, which is maintained through the dimensionality reduction process.

Then we achieve the transformation relation matrix T and R from a set of mapping steps between C_b and E_b.

Finally, we modify the isomap three-dimensional results I by the equation: $R\dot{I} + T$ to get the results. In the process of transformation, the rotation and transformation matrix in two-dimensional and three-dimensional space are used, some optimization problems are solved by using the function in the MATLAB. The more specific steps of the transformation process are explained in the following sections:

4.2.1 Achieving the Occluding Boundary Points' True Surface Normals C_b

We can separate the occluding boundary points of the object from the other surface points and use the sobel filter to a image to get these boundary positions' corresponding three-dimensional surface normals described as C_b (the axis z value of surface normal is zero). The detailed steps are as follows:

-1	0	1	-1	-2	-1
-2	0	2	0	0	0
-1	0	1	1	2	1

Table 4.1: The sobel filter: x-filter and y-filter

Step1: Available for the occluding boundary points

1. Separate the background pixels and the surface pixels:

We know that if one pixel is the background pixel, its corresponding different appearance does not change in the different illumination (different images). So we can achieve the surface pixels of the object.

2. Find the boundary points by comparing one pixel and its up, down, left, right around four pixels from one image.

Step2: Apply the sobel filters to one image

1. Sobel filter

We know that the surface normals of some boundary points can be computed directly from one image by doing the sobel filter to the images. Here we use the x-sobel filter and y-sobel filter as table 4.1:

2. The true surface normals of the boundary points C_b

The images pixel value results after doing the x-sobel filter are remembered as X and the results after y-sobel filter are remembered as Y, so the surface normals of the boundary points can be written as $C_b = [-X_b; -Y_b; 0]$ and they are lying in a plane parallel to the image plane. These normal surfaces are correct and used to help to do the fitting for the isomap results.

4.2.2 The Mapping Relation between C_b and E_b

In this section, we introduce the detailed process of finding the relation M between the true surface normals C_b and the isomap three-dimensional results E_b. We have known that the C_b achieved from the sobel filter are on the plane $z = 0$. So the first step is to make all of the isomap result boundary points' surface normals E_b on one plane and rotate this plane to be parallel with $z = 0$; Then in this plane, do the fitting between these two-dimensional points on the plane by using the rotation and transformation method. The specific algorithm is introduced in the following steps:

STEP1: Projection to a Plane

Firstly, we determine a plane on which all E_b lie. In fact, the isomap result surface normals of these boundary points E_b are always not in the same plane. An approximate plane P :

$ax+by+cz=d$ can be determined by minimizing the sum of the distance between every point in E_b and plane P such as the equation: $D_{sum}=\sum_{i=1}^{f} d_i$, where d_i is the distance between the point $i(x_i,y_i,z_i)$ and the plane P computed by:

$$d_i=\frac{|ax_i+by_i+cz_i-d|}{\sqrt{a_i^2+b_i^2+c_i^2}} \tag{4.7}$$

where the parameter f is the number of the boundary points. So we can find the suitable plane P and project the boundary points E_b to plane P to get the E_b'. This optimization problem can be solved by the function in the MatLab. Finally we can get the corresponding three-dimensional E_b' in the plane P.

STEP2: 3D Rotation for the Plane

In order to do the mapping between the points E_b' and C_b in the same plane, for the boundary points C_b are in the plane $z=0$, the plane P should be rotated to be parallel with plane: $z=0$. We define that the normal of the plane P is $n(a,b,c)$; vector n's projection to the plane $z=0$ is remembered as n_{xy} and projection to the plane $y=0$ is remembered as n_{xz}; α is the circle between the n_{xy} and axis x; β is the circle between the n_{xz} and axis z.

Firstly rotate these points E_b' α around axis z, then rotate β around axis y, the result points will in a new plane that is parallel to the plane $z=0$. The results points $E_b'(x,y,z)$ are $E_b'=R_y*R_z*E_b'$ with the same value of z values. The rotation matrix R_z around axis z and the rotation matrix R_y around axis y are as follows:

$$R_z=\begin{pmatrix} cos\alpha & sin\alpha & 0 & 0\\ -sin\alpha & cos\alpha & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1\end{pmatrix} R_y=\begin{pmatrix} cos\beta & 0 & sin\beta & 0\\ 0 & 1 & 0 & 0\\ sin\beta & 0 & cos\beta & 0\\ 0 & 0 & 0 & 1\end{pmatrix}$$

STEP3: Mapping between the Boundary Points on the Plane

Because $E_b'(x_i,y_i,z)$ and $C_b(x_i,y_i,0)$ are in the parallel planes, this step is to do the fitting between two set of two-dimensional points: $e_b(x_i,y_i)=E_b'(x_i,y_i)$ and $c_b(x_i,y_i)=C_b(x_i,y_i),(i=1...f)$ where f is the number of the boundary points. Firstly we normalize these two set points, then do the transformation $T(T_x,T_y)$ and rotation R_{xy} for the points e_b to make the sum of the difference between each pair points $e_b(x_i,y_i)$ and $c_b(x_i,y_i)$ be the minimum.

Do the transformation and rotation as the following:

$$\begin{aligned} e_{r_x} &= e_{r_x}+T_x;\\ e_{r_y} &= e_{r_y}+T_y;\\ e_r &= R_{xy}*e_r; \end{aligned}$$

The sum difference between each pair of point $e_b(x_i, y_i)$ and point $c_b(x_i, y_i)$ is computed by:

$$D_s = \sum_{i=1}^{n} \sqrt{(e_b(x_i) - c_b(x_i))^2 + (e_b(y_i) - c_b(y_i))^2} \tag{4.8}$$

In my method we set the initiate value of T is $T_x = 0, T_y = 0$, and find the best circle θ when rotate the points e_b to make the difference be minimum. Then we start to rotate and transform the points e_b to make D_s be minimum again and again. Here we use the Matlab optimization function to solve this problem. Finally we can find the $T(T_x, T_y)$ and the rotation matrix R_{xy}.

4.2.3 Transformation for the Whole Isomap Results

We have found the transformation and rotation relations that have been explained before: (rotation matrix R_z, R_y in three dimensional space; R_{xy} in the plane and the transformation $T(T_x, T_y)$). We use the following steps to modify the whole isomap results:

1. Rotate them around axis z by matrix R_z, around axis y by matrix R_y in three dimensional space.
2. Do the rotation and transformation by R_{xy}, T in two dimensional space.
3. Do the normalization for each three-dimensional result surface normal.
4. Replace these three dimensional results to their corresponding pixels places in the original image and get the object's surface normals in the images.

As the figure 4.8 showing, the top two images are the three-dimensional results of a sphere directly from the isomap algorithm; the bottom two images are the three dimensional surface normal vectors after doing the transformation algorithm by using the occluding boundary points as the reference.

4.3 3D Shape Reconstruction by Relaxation Method

The surface normals of the surface points can be achieved from the previous section algorithm. The relaxation method is used to construct the three-dimensional shape for the object from the surface normals.

A needle diagram is given by the known surface orientation for every picture cell, so we want to represent the surface shape by using the depth map that is giving the height above some reference plane.

We define the parameter p, q are the partial derivatives of $z(x, y)$ with respect to x and y:

$$p = -\frac{dz}{dx}, q = -\frac{dz}{dy} \tag{4.9}$$

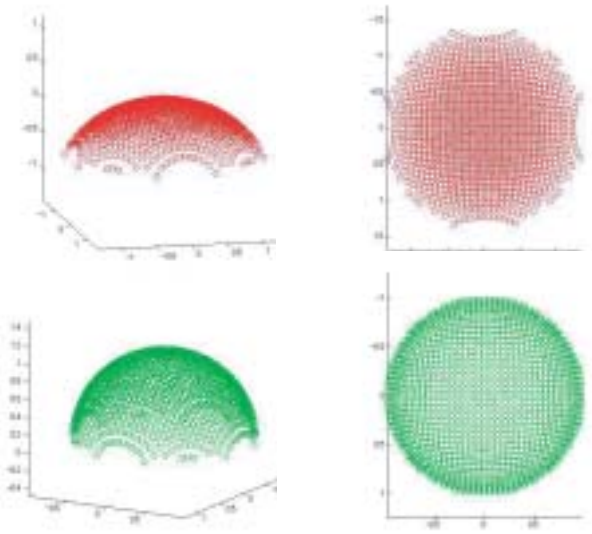


Figure 4.8: Top(red):Isomap three-dimensional results; Bottom(green): Surface normal results after do the transformation.

We want to make the observed orientation p, q be the same as the z_x, z_y, so the total error is remembered as:

$$(p(x,y) + \frac{\partial z}{\partial x}(x,y))^2 + (q(x,y) + \frac{\partial z}{\partial y}(x,y))^2 \tag{4.10}$$

In order to minimize this equation, the calculus variance method can be used to minimize this equation within a boundary. The detailed algorithm is written in the [Hor86].

$$\int\int_A (p(x,y) + \frac{\partial z}{\partial x}(x,y))^2 + (q(x,y) + \frac{\partial z}{\partial y}(x,y))^2 dxdy \tag{4.11}$$

An iterative method can be used to solve this equation and one pixel height can be obtained from the around four pixels heights as:

$$\begin{aligned} z^{n+1}(x,y) &= \frac{1}{4}\{z^n(x+1,y) + z^n(x-1,y) + z^n(x,y+1) + z^n(x,y-1)\} \\ &+ \frac{1}{4}\{\frac{\partial p}{\partial x}(x,y) + \frac{\partial q}{\partial y}(x,y)\} \end{aligned}$$

The detailed theory and algorithm can be seen in the [Hor86], so the result depth map can be obtained and we can get the 3D shape of the object from the surface normals of the object.

Chapter 5

Applicable Surface Materials

Since illumination and camera position are assumed to be sufficiently distant from an object, illumination and viewing directions are consistent over the object surface; then the reflection equation for a point p when it is illuminated by the ith illumination with unit radiance is

$$I_i^p = f_p(\vec{\omega_i}, \vec{\omega_o})(\vec{n_p} \cdot \vec{\omega_i}), \tag{5.1}$$

where $\vec{n}_p$ is the surface normal of p, $\vec{\omega_i}$ and $\vec{\omega_o}$ are incident and reflection directions that are consistent over all surface points of the object, and $f_p(\vec{\omega_i}, \vec{\omega_o})$ represents a bidirectional reflectance distribution function (BRDF) of the point p that represents how much of the incident light $\vec{\omega_i}$ is reflected on the object surface toward $\vec{\omega_o}$.

5.1 Lambertian Surfaces

Let us start with the case where objects have uniform Lambertian reflectance. The BRDF for a Lambertian surface is known to be a constant. From (5.1), the equation for a Lambertian surface is given as

$$I_i^p = \mathrm{kd}(\vec{n_p} \cdot \vec{\omega_i}), \tag{5.2}$$

where kd is a constant albedo over the object surface.

Suppose that q and w are two points on an object surface; then their intensities are computed from (5.2) as: $I_i^q = \mathrm{kd}(\vec{n_q} \cdot \vec{\omega_i})$ and $I_i^w = \mathrm{kd}(\vec{n_w} \cdot \vec{\omega_i})$. Comparing these equations, it can be clearly seen that under the same incident direction $\vec{w_i}$, their intensity difference is only caused by the difference between their surface normals $\vec{n_q}$ and $\vec{n_w}$.

Since this is true for all n illumination directions ($i = 1, \cdots, n$), it can be concluded that the differences in observation vectors $\vec{o_p} = [I_1^{\mathrm{p}}, \cdots, I_n^{\mathrm{p}}]^T$ among m surface points result from their surface normal differences, and thus points o_p construct an appearance manifold.

5.2 Textured Lambertian Surfaces

Suppose that two surface points q and w have the same surface normal $\vec{n}$, but different diffuse parameters $\mathrm{kd}_q \neq \mathrm{kd}_w$. From (5.2), their intensities under the ith illumination are $I_i^q = \mathrm{kd}_q(\vec{n} \cdot \vec{\omega_i})$ and $I_i^w = \mathrm{kd}_w(\vec{n} \cdot \vec{\omega_i})$, and their observation vectors become

$$\begin{aligned} \vec{o_q} &= \mathrm{kd}_q \left[(\vec{n} \cdot \vec{\omega_1}), \cdots, (\vec{n} \cdot \vec{\omega_n})\right]^T, \\ \vec{o_w} &= \mathrm{kd}_w \left[(\vec{n} \cdot \vec{\omega_1}), \cdots, (\vec{n} \cdot \vec{\omega_n})\right]^T. \end{aligned} \tag{5.3}$$

We can see that kd_q and kd_w are just scalar values of the same vector $[(\vec{n} \cdot \vec{\omega_1}), \cdots, (\vec{n} \cdot \vec{\omega_n})]^T$ in these equations.

As a result, by normalizing each observation vector $\vec{o_p}$ by its length $\|\vec{o_p}\|$, the effect of diffuse parameters kd_p or kd_w can be canceled. Accordingly, the variations in these vectors are due to their surface normal differences, and this leads to an appearance manifold for the textured Lambertian surface.

5.3 Specular Surfaces

The proposed approach can be applied to non-Lambertian surfaces as well. Let us take several reflection models as examples. Supposing uniform reflectance properties over the object surface, the intensity of a surface point is computed as

Blinn-Phong model:

$$I_i^p = \mathrm{ks}\frac{n+2}{2p}\cos^n \angle(\vec{h_i}, \vec{n_p}) \tag{5.4}$$

Torrance-Sparrow model [TS67]:

$$I_i^p = \mathrm{ks}\frac{1}{(\vec{v} \cdot \vec{n_p})}\exp(\frac{\angle(\vec{h_i}, \vec{n_p})^2}{2\sigma^2}). \tag{5.5}$$

Ward isotropic reflection model [War92]:

$$I_i^p = \mathrm{ks}\frac{1}{\sqrt{(\vec{n} \cdot \vec{\omega_i})(\vec{n} \cdot \vec{\omega_o})}}\frac{\exp(-\tan^2 \angle(\vec{h_i}, \vec{n_p})/\sigma^2)}{4\pi\sigma^2}, \tag{5.6}$$

where ks is a constant for the specular reflection component, and σ is the standard deviation of a surface slope. $\vec{h_i}$ is the bisector of the light source direction $\vec{\omega_i}$ and the viewing direction $\vec{v}$, and the function $\angle(\vec{h}, \vec{n})$ computes the angle between two vectors.

Comparing intensities I_i^p for all surface points ($p = 1, \cdots, m$) illuminated by the ith illumination from the direction $\vec{w_i}$, we can see that only the surface normals $\vec{n_p}$ differ in their reflection equations in the case where reflectance properties ks, sigma or n are uniform over the object surface.

This is true for all illumination directions ($i = 1, \cdots, n$). One can say that the temporal variations in their appearance also reflect the differences in their surface normals, and the appearance changes of the specular surface should thus construct an appearance manifold.

5.4 Surfaces with Both Components

If the object surfaces consist of both uniform specular and diffuse reflectance components, the intensities of their surface points are computed as the addition of its diffuse component and specular component from the Dicromatic reflection model. Based on our analyses of Lambertian surfaces (Section 5.1) and specular surfaces (Section 5.3), it can be saied the differences among the observation vectors are still due to the differences in the surface normals $\vec{n_p}$ among p. Our algorithm is therefore able to extract the surface normals of objects with both diffuse reflectance and specular reflectance properties from their appearance manifold.

The most complex reflectance model is a textured specular and diffsue reflectance model. This type refelactance property object is still a challange for our approach. In the future work part we will discuss it.

Chapter 6

Experimental Results

This chapter is divided into two sections: the first section (6.1) shows some CG objects results and the error computation for our approach; the second section (6.2) shows the results of some real objects with different surface materials by using our proposed method.

The equipment that used in our approach is:

1. Machine
   - Inter(R) Core(TM)2 CPU 6700 2.66GHz
   - 3.50 GB RAM
2. Software
   - MATLAB © R2006b
3. Camera
   - SONY DXC-9000 3CCD COLOR VIDEO CAMERA ©

6.1 Synthetic Data

6.1.1 Illumination Distribution Setting

We set the light source moving on a sphere surface just as the figure 6.1 showing: the light positions are described as the small red points on a sphere. For the determination of the light positions, we define two circles θ and ϕ: θ is the circle between the light position vector and axis z; ϕ is the circle between the illumination's projection vector to plane $z = 0$ and axis x. So the concrete positions of the light sources are determined as:

$$L_p = (\sin\theta\cos\phi, -\sin\theta\sin\phi, \cos\theta) \tag{6.1}$$

We can set the light source positions for making the CG images through setting the different circles θ,ϕ values. Based on some reflectance models introduced in Chapter 5, we make the different reflectance property CG images to do the test for our approach.

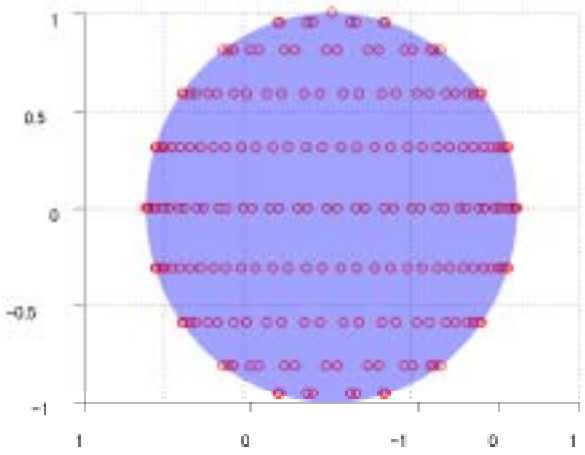


Figure 6.1: Setting the light source positions on a sphere.

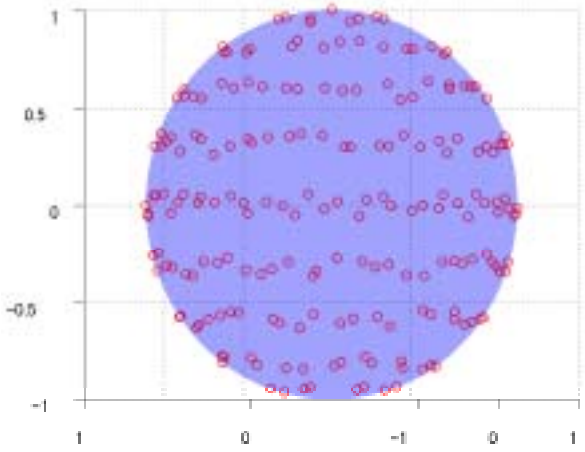


Figure 6.2: Setting the light source positions on a sphere randomly.

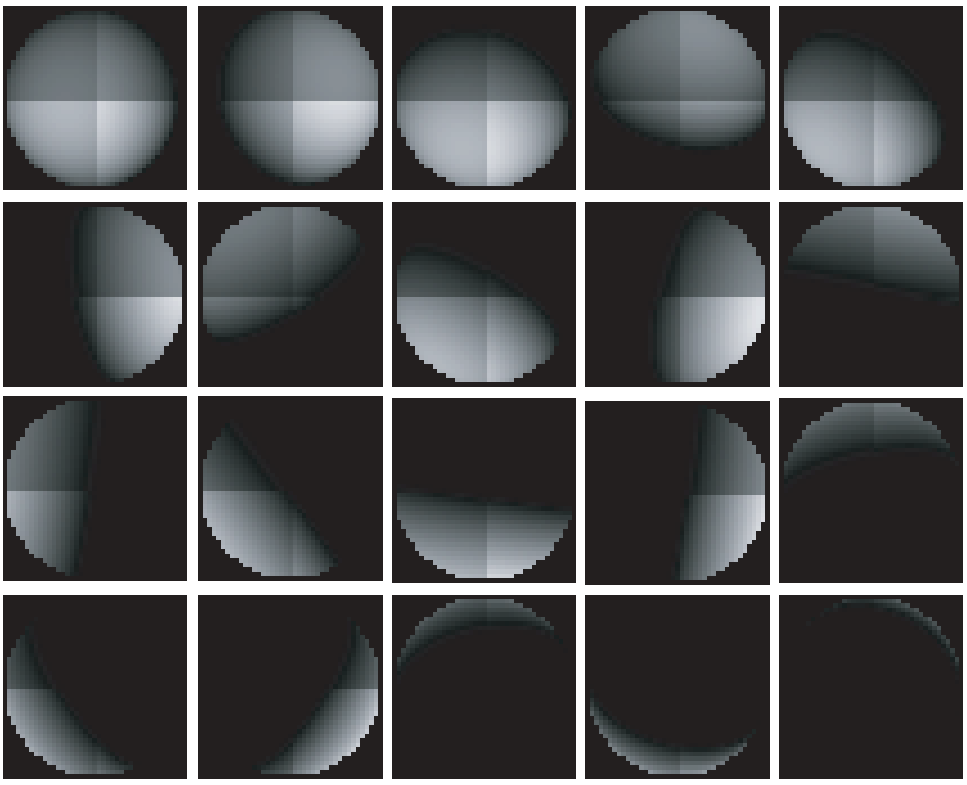


Figure 6.3: Some of the input images of a texture sphere with the lambertian reflectance property

In our approach, the light sources do not have the need to be set at the known positions or in an accurate order. In the test for the CG objects, we use the illumination such as the figure 6.2 showing the illumination distribution: the positions are allowed to move around randomly in some range. In the procedure of the real objects test, we can use a hand-held light to achieve the suitable illumination distribution for the light source is not required so much for our approach. Further, the images taken in the different illuminations do not need to be arranged at a fixed sequence.

6.1.2 The Texture and Lambertion Surface

1. Sphere

   - Input Appearance Manifold

     We use a CG sphere's different appearances in different illumination such as figure 6.2 showing the distribution of the illumination.

     This sphere is a texture lambertion sphere only with the diffuse reflectance property parameter kd is set to be the different values $0.4, 0.5, 0.6, 0.9$ in the four different positions.

     Figure 6.3 shows one part of the input images that made by using the equation of

the lambertion texture surface reflectance property:

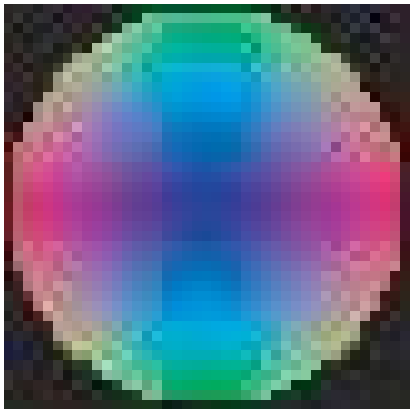
$$I = kd * (N \cdot L)$$

where L is the vector of illumination; N is this pixel's corresponding surface normal; kd is the diffuse reflectance parameter. We use 200 input images and in each image there are about 1245 surface pixels. So these input images give us 1245 200-dimensional vectors to extract the intrinsic three-dimensional surface normals that hidden inside. In the process of dimensionality reduction, the parameter k that presents the number of neighbors needs to be adjusted in order to make the output vectors number be equal to the number of the surface points. Here we set $k = 30$ and use our approach to get the result surface normals of this sphere.

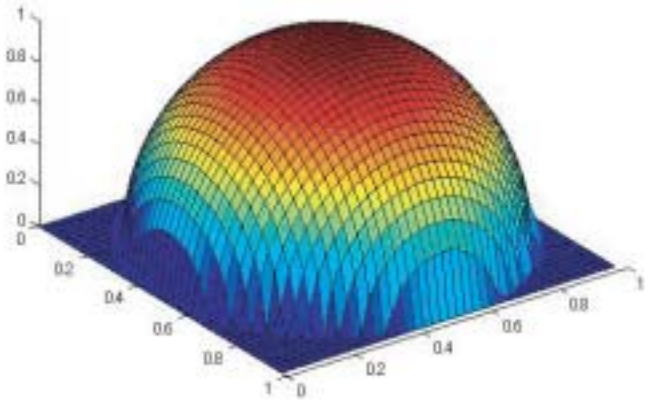
- Results of the Sphere

  In order to compare the result and the original images difference, the true color map and the true 3D shape of a sphere are also shown in figure 6.4. The top image shows the true color map of a sphere's surface normals (red shows x value of the surface normal; green shows the y value and blue shows the z value). The bottom image is the true 3D shape of a sphere.

  Figure 6.5's top image is the result color map and the bottom image is the result 3D shape of this sphere computed from the surface normals by the relaxation method.

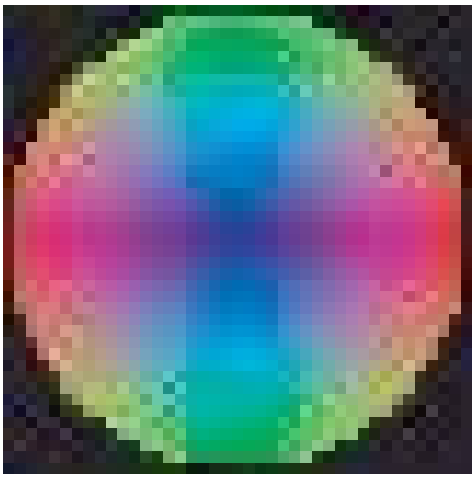


true sphere color map

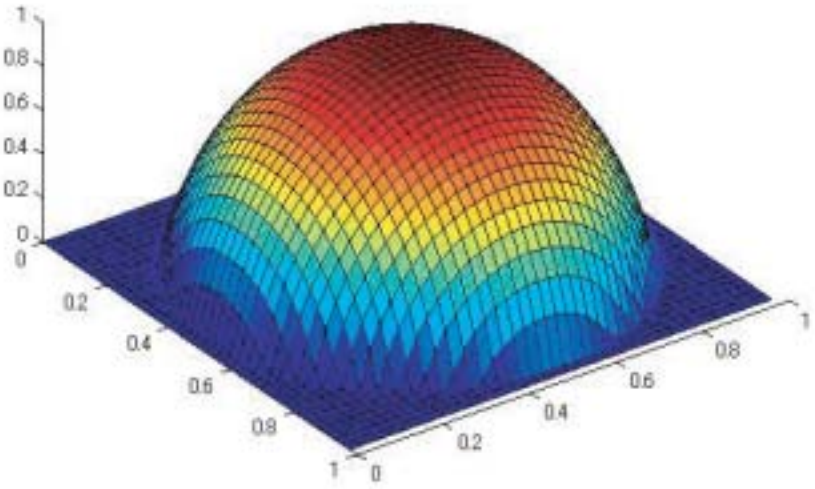


true sphere 3D shape

Figure 6.4: True color map and 3D shape of a texture lambertian sphere.



result color map



result 3D shape

Figure 6.5: Result color map and 3D shape of this sphere by using our proposed-method.

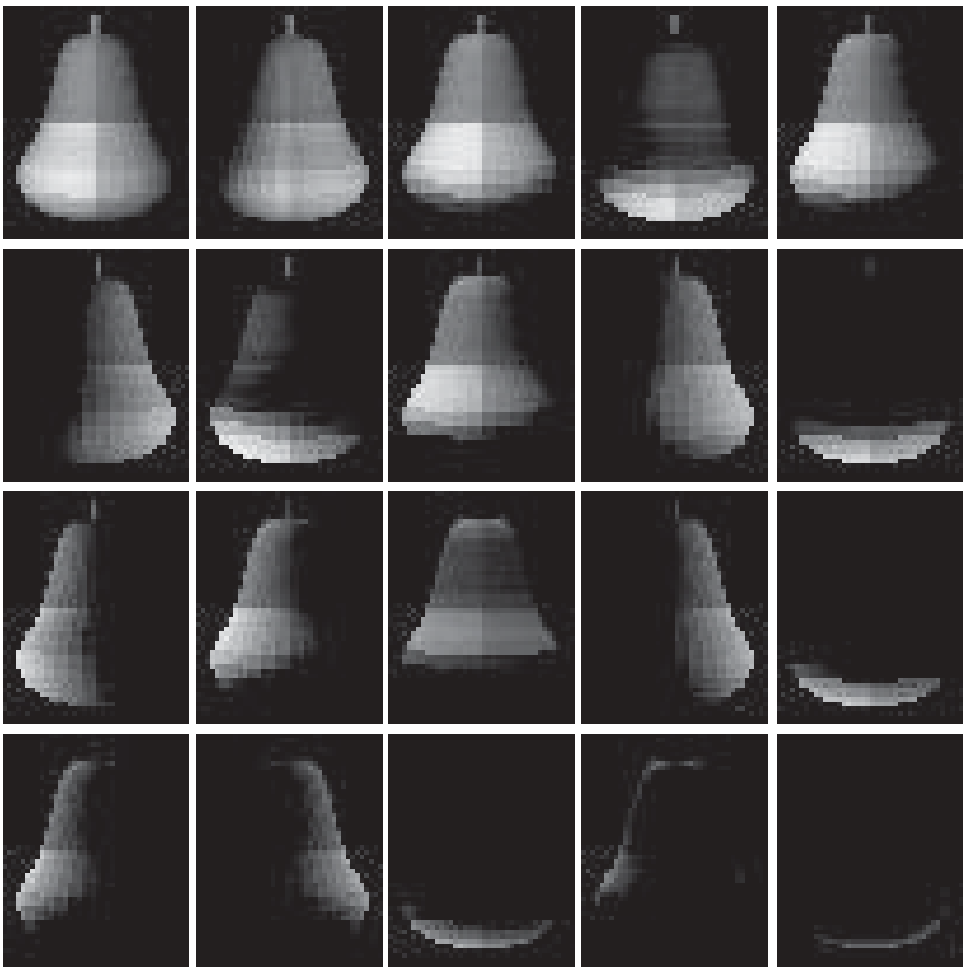


Figure 6.6: Some of the input images of a texture pear with the lambertian reflectance property.

2. Pear

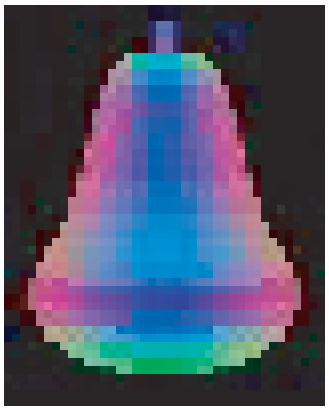
- Input Appearance Manifold

  (1). Input images number

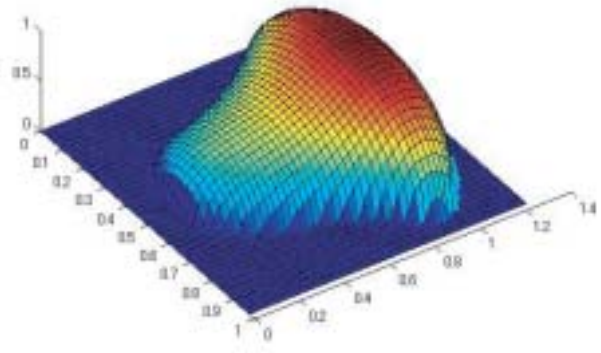
  This is another experiment used the CG texture diffuse pear as the test object. Some input images made in the different illuminations are shown as figure 6.6. We also use about 200 input images and the size of one image is 40×50 pixels including about 942 surface pixels. So the input appearance manifold is seen as 942 200-dimensional vectors for 942 surface pixels.

  (2). Reflectance property parameters

  This CG pear only has the different diffuse reflectance properties in the different surface positions. Here we set the diffuse reflectance parameter kd is $0.4, 0.5, 0.6, 0.9$ at the up-left, up-right, down-left and down-right four blocks;



true pear color map



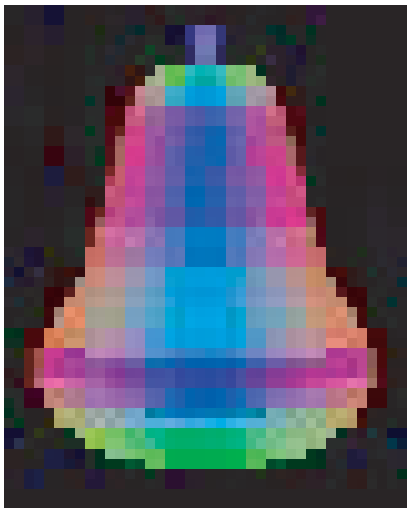
true pear 3D shape

Figure 6.7: True color map and 3D shape of the pear.

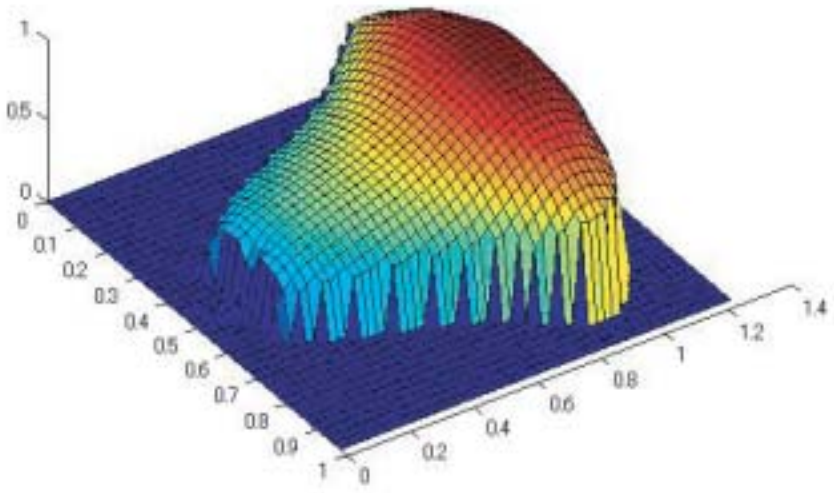
- Results of the CG Pear

  Figure 6.7 shows the true color map and the true 3D shape of this pear.

  Figure 6.8's top image shows the result color map which describes the result surface normals obtained from our proposed-method and the bottom image shows the 3D shape of this CG pear by using the relaxation method.



result color map



result 3D shape

Figure 6.8: Result color map and 3D shape of the texture lambertian pear by using our method.

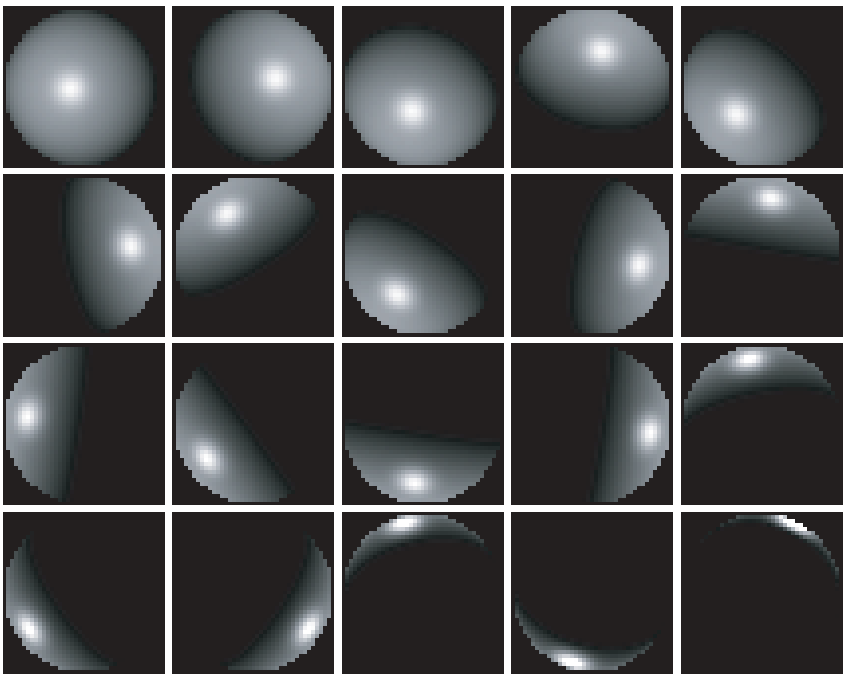


Figure 6.9: Some of the input images of a no texture sphere with specular and diffuse reflectance properties.

6.1.3 The Diffuse and Specular Objects

Here we also use some CG objects with the diffuse and specular two types of the reflectance properties as the experiment objects. The objects are the same with the last part: a sphere and a pear.

1. A Diffuse and Specular Sphere

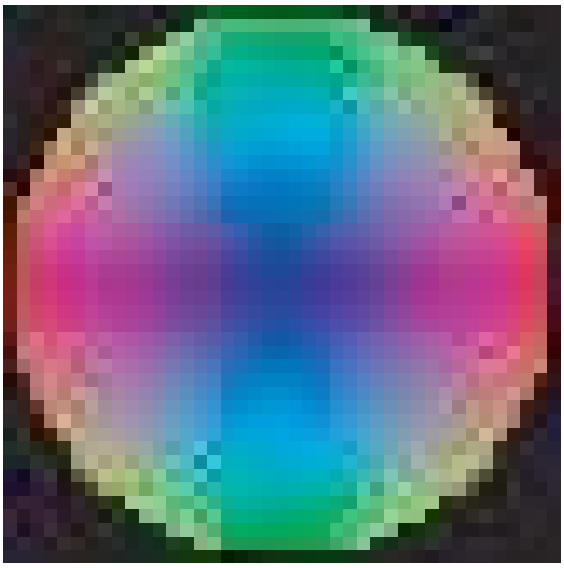
   - Input Appearance Manifold

     This sphere is a no texture sphere but with the diffuse and specular two different reflectance properties: the diffuse reflectance parameter $kd = 0.6$ and the specular reflectance parameters $ks = 0.4; \sigma = 0.15$.

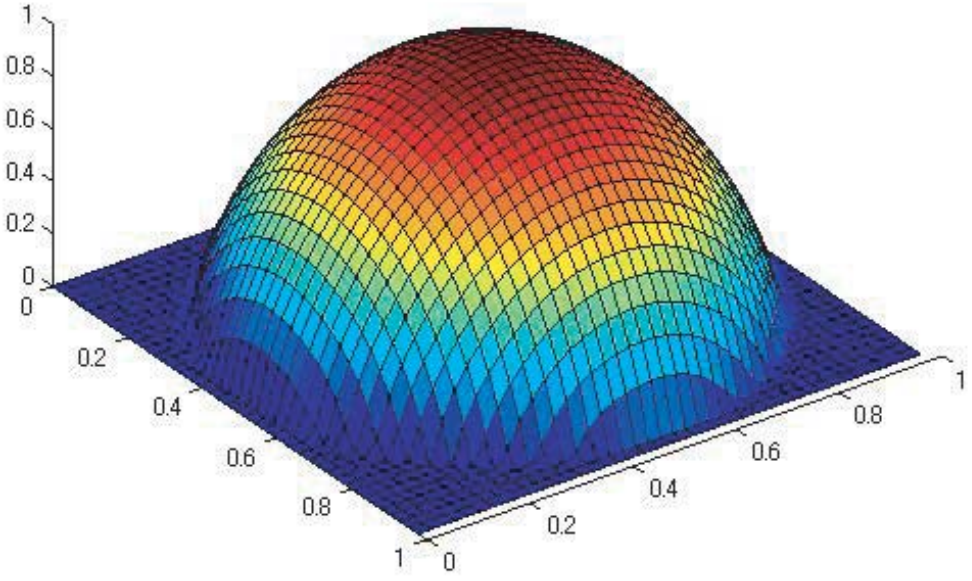
     Figure 6.9 shows one part of the input images computed by using the equation of the specular and diffuse reflectance model that has been introduced in the Chapter 5. We also use 200 input images and in each image there are 1245 surface pixels. So these input images give us 1245 200-dimensional vectors to extract the intrinsic three-dimensional surface normals that hidden inside. The parameter k is set to be 30 in the process of the dimensionality reduction.

   - Results of the Sphere

     Figure 6.10's top image shows the result color map of this sphere's surface normals and the bottom image shows the result 3D shape of this spcular and diffuse sphere.



result color map



result 3D shape

Figure 6.10: The result color map and The result 3D shape of a sphere by using our method.

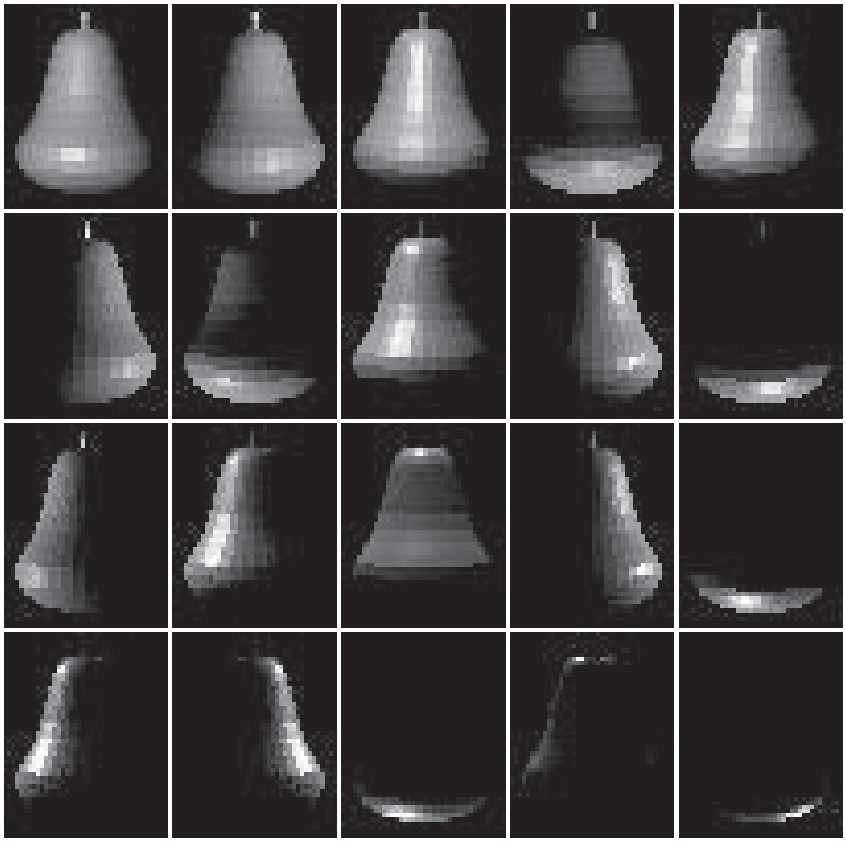


Figure 6.11: Some of the input images of a no texture pear with specular and diffuse reflectance properties.

2. A Diffuse and Specular Pear

   - Input Appearance Manifold

     (1). Input images number

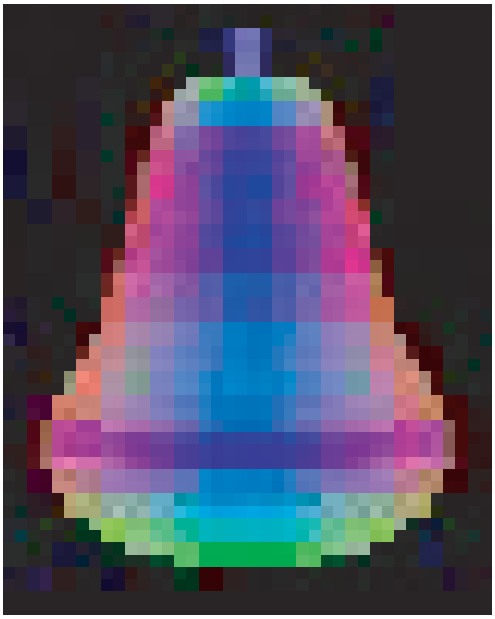
     We use the CG diffuse and specular pear without the texture as the object. Some input images are shown as figure 6.11. We also use about 200 input images and the size of one image is 40×50 pixels including about 942 surface pixels. So the input appearance manifold can be seen as 942 vectors in the 200-dimensional space.

     (2). Reflectance property parameters

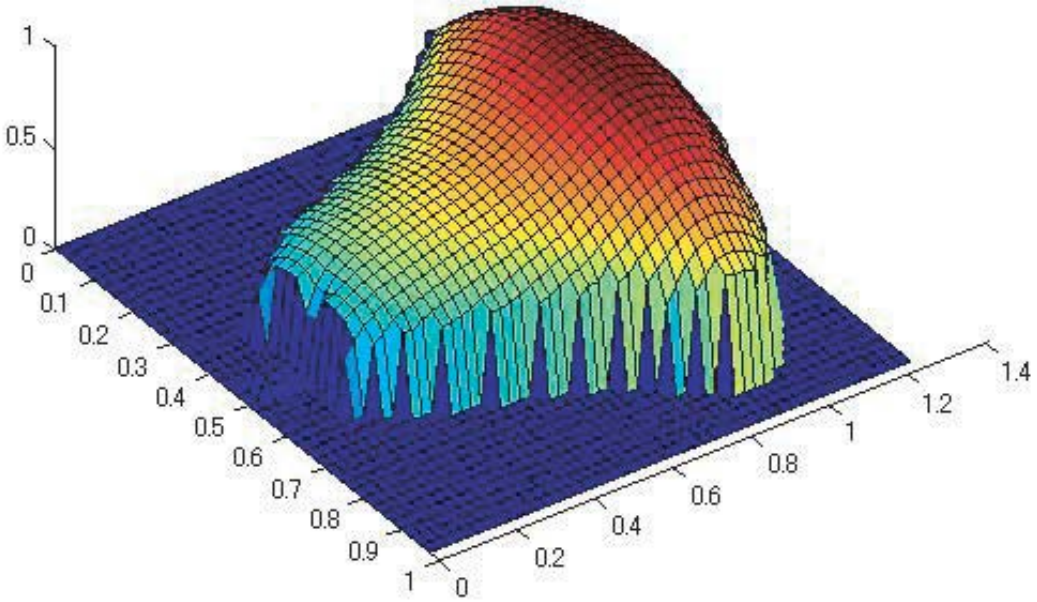
     The diffuse and specular reflectance property parameters are set as the same as the sphere: the diffuse reflectance parameter $kd = 0.6$ and the specular reflectance parameters $ks = 0.4; \sigma = 0.15$.

   - Results of the CG Pear

     Figure 6.12's top image shows the result surface normals color map of a diffuse and specular pear. The bottom image shows the result 3D shape of this diffuse and specular pear.



result color map



result 3D shape

Figure 6.12: The result color map and 3D shape of a diffuse and specular pear by using our method.

shape	diffuse	specular	texture	$\bar{E}_h$	S_h^2	$\bar{E}_n$	S_n^2
sphere	Y	N	Y	0.1253	0.0564	0.1474	0.0009
sphere	Y	Y	N	0.1403	0.0727	0.0606	0.0004
pear	Y	N	Y	0.0837	0.0147	0.1390	0.0053
pear	Y	Y	N	0.0885	0.0143	0.1295	0.0027

Table 6.1: Errors of the CG sphere and the CG pear

6.1.4 The Error Computation

1. Error Definition

   For the CG objects, the true surface normals and the true height of the object are known, so we can make a quantitative computation for the errors caused by our proposed-method. In this part, the mean error $\bar{E}$ and the variance error S^2 are used to evaluate our CG images results.

   The mean error $\bar{E}$ is computed by:

   $$\bar{E} = \sqrt{(E_1^2 + ... + E_n^2)/n} \tag{6.2}$$

   The variance error S^2 is computed by:

   $$S^2 = \frac{1}{n}\{(E_1 - \bar{E_n})^2 + ... + (E_1 - \bar{E_n})^2\} \tag{6.3}$$

   where n is the number of the surface pixels; for the error computation of the surface normals, E_i is the absolute different degrees $E_i = \arccos(N_{Ti} \cdot N_{Ri})$ between our result surface normal vector N_{Ri} and the true surface normal vector N_{Ti} for pixel i;

   And for the height error computation, E_i is the absolute value $E_i = |H_{Ri} - H_{Ti}|$ between result height H_{Ri} and the true height H_{Ti} for pixel i.

2. The Errors Caused by the CG Images

   Table 6.1 shows the mean and variance errors of surface normals and the height of our results (sphere and pear).

   In this table the mean error and the variance error of surface normal is remembered as $\bar{E}_n$, S_n^2; The mean error and the variance error of the height is remembered as $\bar{E}_h$, S_h^2 These objects are mentioned before: a texture diffuse sphere, a no-texture specular and diffuse sphere, a texture diffuse pear and a no-texture specular and diffuse pear.

   From the results of these four objects, the mean errors of surface normal $\bar{E}_n$ are about 7° and the variance errors are small.

6.2 Real Objects

This section introduces the results of some real objects with different kinds of materials such as plastic, ceramic and steel.

6.2.1 Real Experimental System

Unlike most of image-based relighting work requires some complex equipment such like a calibrated camera, a known position light probe, a dynamic range photography and some so on, our approach only needs to take a set of different images for an object under a freely moving light source that can be obtained by using a normal handheld light and an uncalibrated position-fixed digital camera.

The process is setting the camera to take the images at regular intervals as we are waving a light source around the object from the approximate fixed distance. Here we need to pay attention to avoid making the cast shadow on the objects. This process is easy to implement and do not need long time (It takes us about $10 - 20$ minutes to get the enough number of the input images). Because these images are not required in order, we can use them as the input images after check whether these are in good condition and the input images number can be added or decreased as the whole algorithm needs.

6.2.2 Results of Different Materials Objects

In this section, some results of using our proposed-method to the real objects are introduced as the following parts:

Plastic Orange

1.The object:

Figure 6.13 shows the original image of this plastic orange which looks like a real normal orange in our life but is made of plastic. This type plastic makes us think this orange has the specular and diffuse two types of the reflectance properties. The color of this orange is nearly a uniform orange color.

2.Input images of this plastic orange

We take about 64 images for this orange in the 64 different illuminations as the input images for our algorithm. The size of this orange is 45×47 pixels and the surface points are 1664. Figure 6.14 shows one part of these input images. And these input images can be seen as 1664 64-dimensional input appearance manifold.

3.The result colormap and result 3D shape.

Figure 6.15 and figure 6.16 show the result color map and the three dimensional shape of this real plastic orange.

Figure 6.13: A plastic orange.

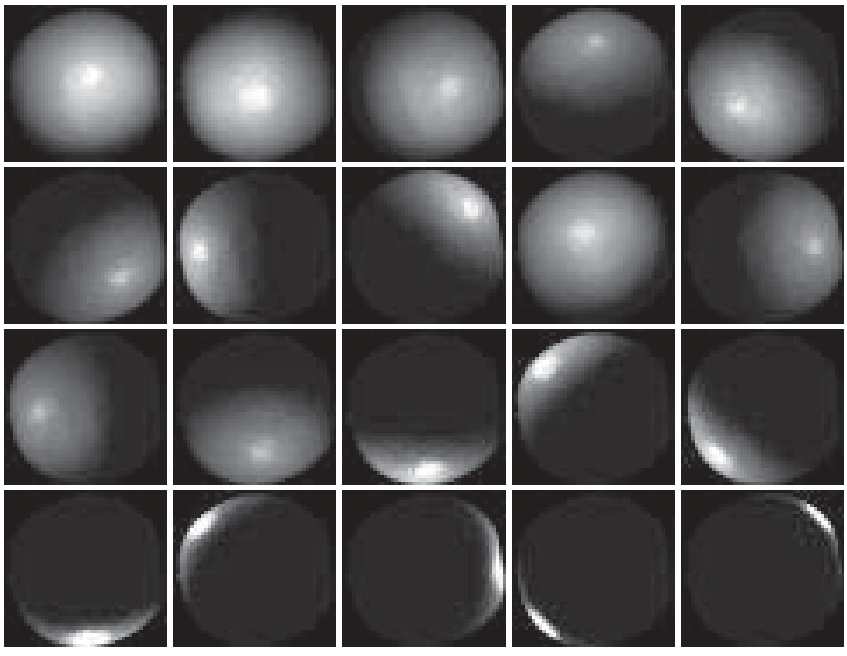
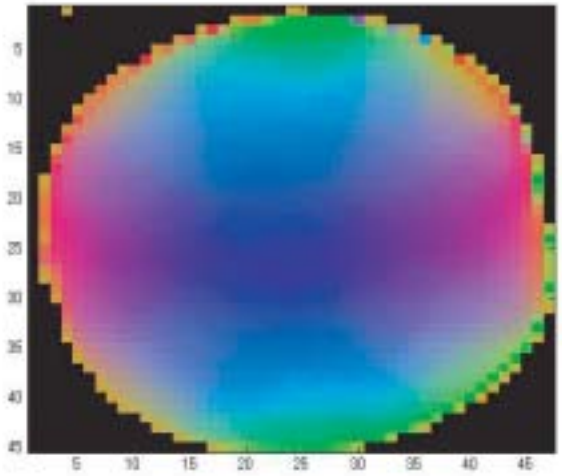
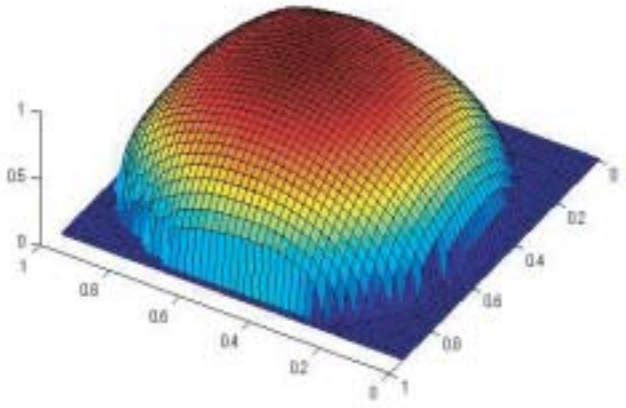


Figure 6.14: Some input images of a plastic orange.



qdrtks nq`mfd bnknq l`o

Figure 6.15: Result color map of a plastic orange



qdrtks nq`mfd 2C rg`od

Figure 6.16: Result 3D shape of a plastic orange.

Figure 6.17: A ceramic cat.

Ceramic Cat

1.The object:

Figure 6.17 shows the original image of this ceramic cat with some colorful texture. This type material has the texture and the diffuse reflectance properties.

2.Input images of a ceramic cat

We take about 192 images for this little cat in the 192 different illuminations as the input images. The image's size is 53×46 pixels and there are 1718 surface points in one image. Figure 6.18 shows one part of these input images that can be seen as 1718 192-dimensional input appearance manifold.

3.The result colormap and result 3D shape.

Figure 6.19 shows the result color map of this cat by using our approach. Figure 6.20 shows some images of the three-dimensional shape of this ceramic cat from the different view positions.

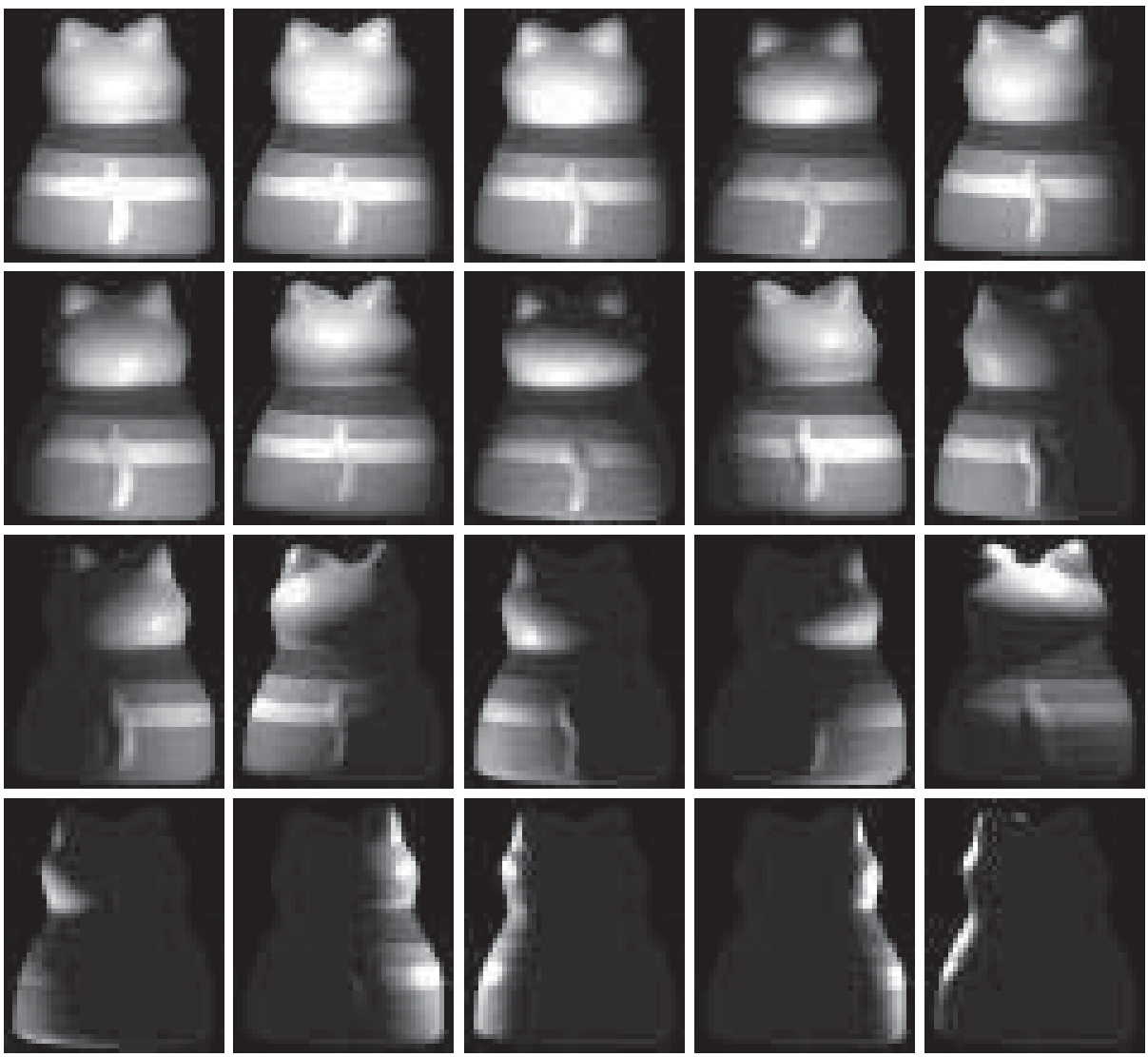
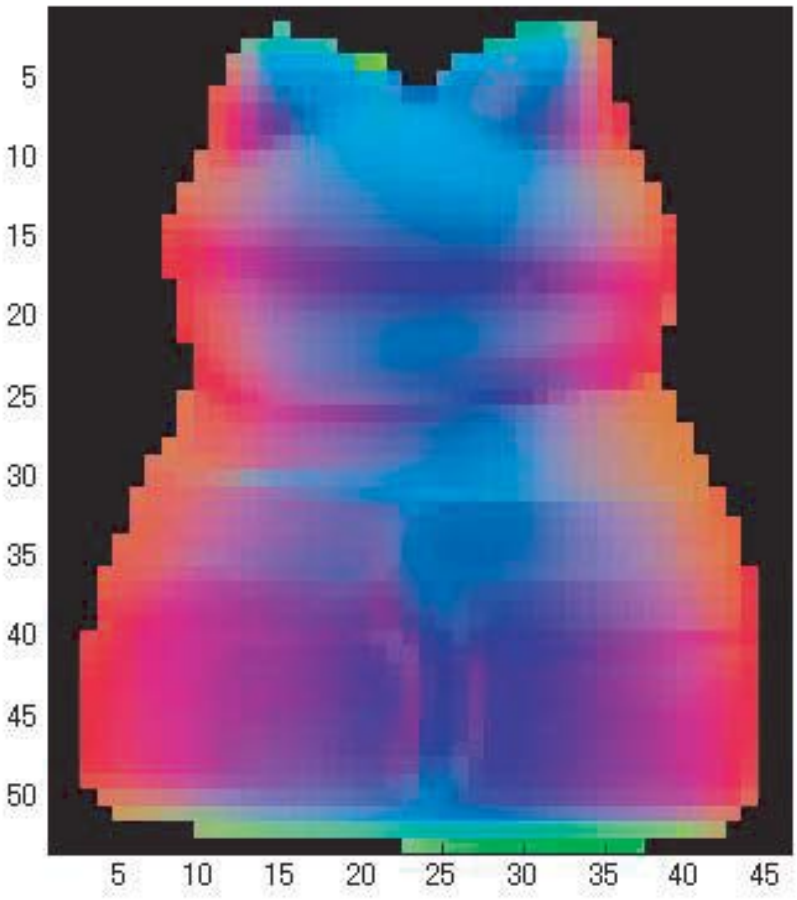
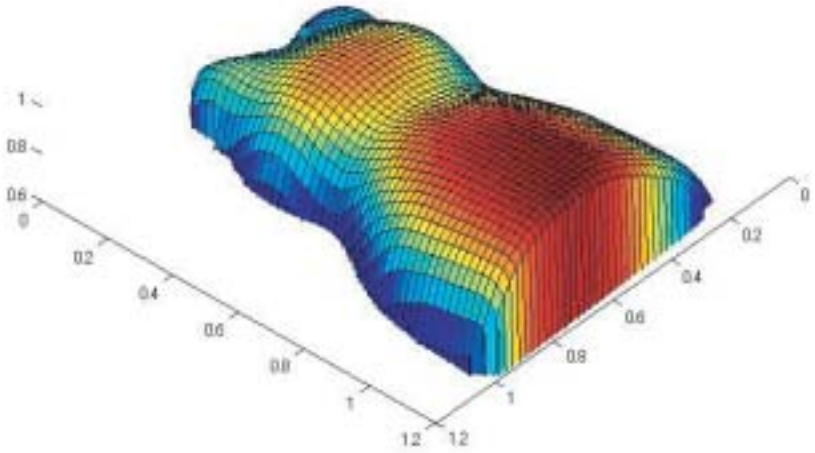


Figure 6.18: Some input images of a ceramic cat.

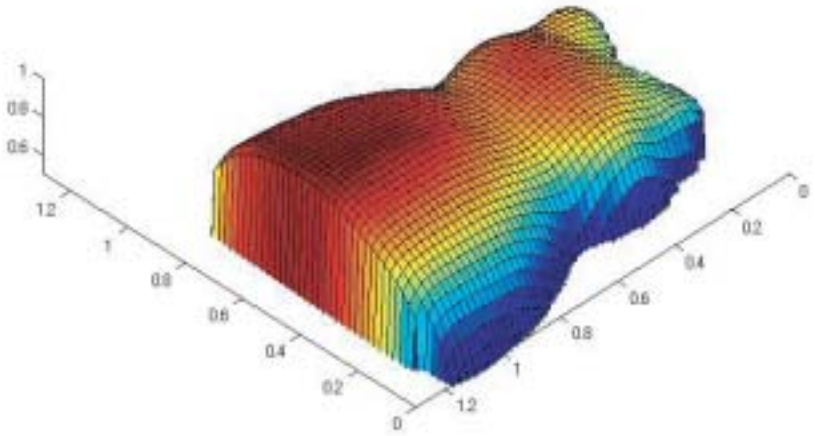


qdrtks bnknq l`o

Figure 6.19: Result color map of a ceramic cat.



qdrtks 2C rg`od'0(



qdrtks 2C rg`od'1(

Figure 6.20: Result 3D shape of a ceramic cat.

Figure 6.21: A steel bird

Steel Bird

1.The object:

Figure 6.21 shows the image of a bird made of steel and this bird has the specular reflectance property.

2.Input images of a steel bird

We take about 168 images for this steel bird in the 168 different illuminations as the input images. The size of the image is 58×65 pixels and there are 2213 surface points in one image. Figure 6.22 shows some input images and the whole images can be seen as 2213 168-dimensional input appearance manifold.

3.The result color map and result 3D shape.

Figure 6.23 shows the result color map of this bird by using our approach. Figure 6.24 shows the result of this steel bird's three-dimensional shape and the tail shape cleared recovered by our proposed method.

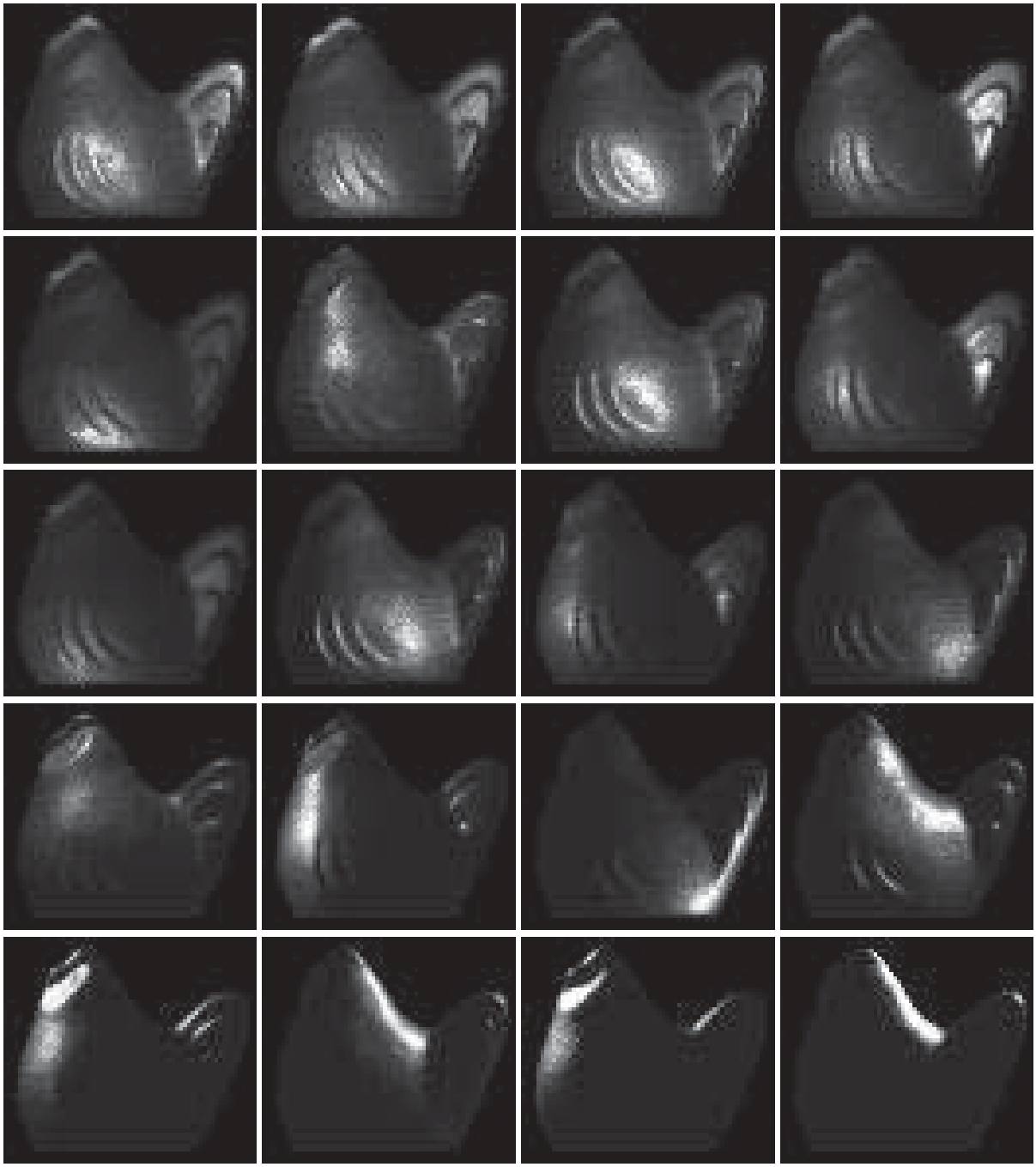


Figure 6.22: Some input images of a steel bird.

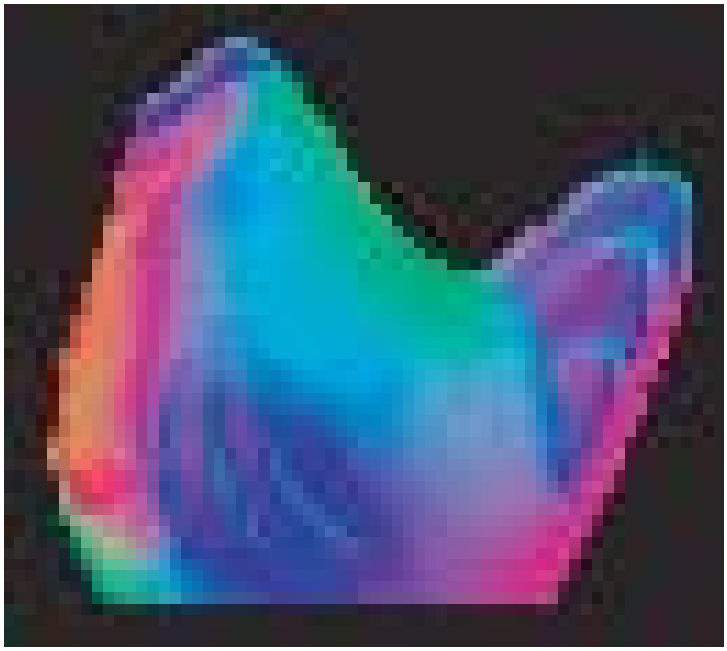


Figure 6.23: Result color map of a steel bird

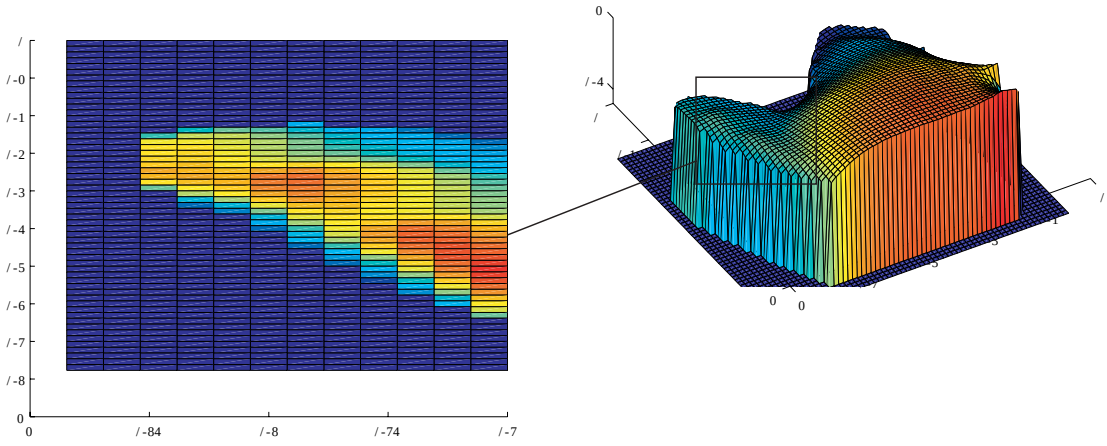


Figure 6.24: Result 3D shape of a steel bird and its tail.

6.2.3 Some Experiment Attentions

In the whole experiment, we should pay attention to the following points:

1. The Illumination Number

   For some object with specular reflectance property or with some texture, we can take more images under different illuminations.

2. The Surface Pixels Number The determination of the input image size makes an important effect on the result of shape reconstruction because adding the pixels number can make the result three-dimensional shape more correctly but make the speed of computation for the isomap algorithm slowly. So we should choose the suitable size for the input images and consider the computer machine's performance at the same time.

3. Some Difficult Objects

   The transparent and the black objects can not be used as the experiment objects for our method. For some special object such as with a strong specular reflectance property and some special materials such like velvet and cloth may be suitable for our method and we will challenge this type object in our future work.

Chapter 7

Conclusion

7.1 Summary

In the introduction (Chapter 1), we have explained some difficult problems on the research on the shape reconstruction for the object through discussing the determination of the object's appearance. Our proposed method on the shape reconstruction is also abstractly mentioned in this chapter. Then a wide overview of research about the shape reconstruction and some previous researches with the relation to our method (Chapter 2) were presented through showing their key ideas and results.

Based on the introduction on the whole overview (Chapter 3) of our approach such as the input appearance manifold, the embedding structures extracted by doing dimensionality reduction method and the main steps of our proposed method, the algorithm of the shape reconstruction (Chapter 4) was explained as these following steps: doing dimensionality reduction to the pixel appearance manifold by using the isomap algorithm, transformation the isomap outputs to the surface normals of the objects based on the reference occluding boundary points and estimation the 3D shape of the object from the surface normals by using the relaxation methods .

Furthermore, in the (Chapter 5), we discussed the applicable materials for our proposed method based on the different materials reflectance models. From the discussion, we gave a clear explanation about the basic reason which type object is suitable for our method and based on this theory, in the final chapter (Chapter 6), we showed the good results of the CG and real objects with different materials and most of them have the combination reflectance properties: the specular and the diffuse reflectance properties.

7.2 Discussion and Future work

Throughout the course of this work, several insights were gained from the discussing the whole procedure of the shape reconstruction and the applicable reflectance models for this approach.

They ranged from the objects' reflectance property to the dimensionality reduction method used in our approach and they are given below:

- *Is it suitable for these type objects? velvet, cloth...*

  From the theory on the applicable materials discussed in the Chapter 5, the object with texture, diffuse and specular reflectance properties is still a challenge object for our algorithm. But if we can deal with the input images such like separating the images into the diffuse and specular two parts, using the intrinsic images (the illumination images)[Wei01] instead of the original images, our proposed method may be suitable for this type object with the complex reflectance properties. Furthermore, some objects made of the velvet or the cloth will be used as the experiment objects in the future work.

- *Can MDS be in stead of the Isomap?*

  In the isomap algorithm, the procedure of computing the geodesic distance between the neighbor points costs most of the algorithm time. So instead that using the Isomap to compute the geodesic distance, we can try to use the MDS algorithm or the LLE algorithm to do the dimensionality reduction for the input appearance manifold. On the other hand, the distance matrix may be obtained from computing the circle between the input vectors instead of computing the Euclidean distance. This is also a valuable challenge for our future work.

- *The balance between the number of surface points and the system's memory and speed*

  We have known that the input for the whole algorithm is all of the surface points' corresponding appearance manifolds and one surface point is correspond to one high-dimensional vector. The larger size of the input image meaning the more surface points will add the number of the result output surface normals and make the result shape more correct. On the other hand, the run time of the whole algorithm will be added more. Depending on the performance of machine and the wanted speed of the whole algorithm, we can make a good choice of the suitable input image's size and keep a good balance between them.

- *The illumination number and distribution*

  In the chapter 6, we have explained the illumination distribution on a sphere used in our approach. But in the real test, the illumination setting is obtained from waving a hand-held light source around the object remaining the distance between the light source and the object surface. The determination of the illumination number (about 100 to 200 images) depends on the test object shape and material. For example, the object with stronger specular reflectance property needs more illumination number. The more complex shape needs more light sources number. But for whatever object, the need for the light source is to set a nearly sparse illumination distribution around the object. The number of illumination and the more concrete relation will be discussed in the future.

Chapter 8

Acknowledgements

I would like to thank Yoichi Sato, my professor and research adviser, whose strictness to us foreign students in the past two years of our academics, coupled with warm care outside of our work, enabled me to have a good time while making a lot of progress in my studies;

Imari Sato, a best friend and a good teacher, who provided a lot of help and support in both studies and life, from whom I have learned a lot about what it takes to be a good researcher, and from whom I found the confidence and strength to continue on whenever I encountered problems in my study;

and Takahiro Okabe, from whom I have received a lot of good advice and learned a lot of research methods, and whose wide mathematical knowledge proved to be a lot of help in my research.

To all the members of Sato Laboratory, these past two years with you will be most cherished.

A lot of thanks, and best wishes!

Bibliography

[AD97] A.Yuille and D.Snow. Shape and albedo from multiple images using integrability. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 158–164, 1997.

[A.G03] A.Georghiades. Incorporating the torrance and sparrow model of reflectance in uncalibrated photometric stereo. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 816–823, 2003.

[BJ03] Ronen Basri and David W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, 2003.

[BKY99] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille. The bas-relief ambiguity. *International Journal of Computer Vision*, 35(1):33–44, 1999.

[H.H94] H.Hayakawa. Photometric stereo under a light source with arbitrary motion. In *J. Opt. Soc. Am. A 11*, pages 3079–3089, 1994.

[Hor86] B. Horn. Robot vision. *MIT Press, Cambridge, MA*, 1986.

[HS05] A. Hertzmann and S. M. Seitz. Example-based photometric stereo: Shape reconstruction with general, varying brdfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1254–1264, 2005.

[HWG05] Jack Tumblin Holger Winnemoller, Ankit Mohan and Bruce Gooch. Light Waving: Estimating Light Positions From Photographs Alone. volume 24, pages 433–438, 2005.

[KH86] Berthold Klaus and Paul Horn. *Robot Vision*. MIT Press, 1986.

[KJ79] J.T.Kent K.V.Mardia and J.M.Bibby. *Multivariate Analysis*. Academic Press, 1979.

[KN06] S. Koppal and S. G. Narasimhan. Clustering appearance for scene analysis. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1323–1330, 2006.

[MD05] F.Kahl M.Chandraker and D.Kriegman. Reflections on the generalized bas-relief ambiguity. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages I–788–795, 2005.

[NIK90] S. Nayar, Katsushi Ikeuchi, and Takeo Kanade. Determining shape and reflectance of hybrid surfaces by photometric sampling. *IEEE Transactions on Robotics and Automation*, 6(4):418–431, August 1990.

[OR02] O.Drbohlav and R.Sara. Specularities reduce ambiguity of uncalibrated photometric stereo. pages 46–62, 2002.

[Sil80] W. M. Silver. Determining shape and reflectance using multiple images. *Master's thesis, MIT, Cambridge,MA*, 1980.

[TK00] Sam T.Roweis and Lawrence K.Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290:2323–2326, 2000.

[TS67] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surface. *J. Optical Society of America*, 57:1105–1114, 1967.

[TSL00] J. B. Tenenbaum, V. Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.

[VH06] George Vogiatzis and Carlos Hernandez. Reconstruction in the round using photometric normals and silhouettes. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1847–1854, 2006.

[War92] G. J. Ward. Measuring and modeling anisotropic reflection. *Proc. SIGGRAPH '92*, pages 265–272, 1992.

[Wei01] Yair Weiss. Deriving intrinsic images from image sequences. In *Eighth International Conference on Computer Vision (ICCV) 2001*, 2001.

[Woo81] Robert J. Woodham. Analysing images of curved surfaces. *Artif. Intell.*, 17(1-3):117–140, 1981.

[ZTCS99] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, 21(8):690–706, 1999.