Photometric Stereo Applied to Metallic Surface Inspection

Ohenhen Godwin

Institute for Automation
Departement Product Engineering
University of Leoben
Leoben, Austria

December 14, 2007

Diploma thesis submitted to the University of Leoben
in partial fulfillment of the requirements for the degree
of Diplom-Ingenieur

Certification Page

I hereby declare, that I composed this thesis and that all work contained therein is my own, except where stated.

Leoben, December 14, 2007 Godwin Ohenhen

Acknowlegement

"*One who shows appreciation for a gift is in a position to get another*"— a popular Edo saying. I am indeed very grateful to a number of people who helped me, directly or indirectly, during the course of this thesis.

My profound gratitude goes to my supervisor Prof. Paul O'Leary for his innumerable and sustained assistance, and from whom I have learnt so much both academically and otherwise. Without his able supervision this work might not have seen the light of day.

A lot of thanks go to Matthew Harker for his help with the Matlab® codes and very useful contributions during the performance of the experiments. I want to thank Ewald Fauster also for making the intricacies of writing this text in LATEX easy. To Taweepol Suesut I say thank you for all the suggestions you gave and the help you rendered during and after the construction of the experimental set-up.

My thanks go to the secretary of the Institute for Automation, Mrs Doris Widek for her help, kind words and great concern. I thank also all members of staff of this institute for their various support and contributions.

It is with a great sense of gratitude that I say thank you to the management of vatron gmbh in Linz for their interest and financial support for this work. I wish to say I big thank you to Dr. Johann Reisinger for his invaluable help.

I am very grateful for the perseverance and encouragement I got from my family. I wish to say a big thank you to my brother, Fred and his family, for their continued support. I cannot find enough words to express my gratitude to Miss Birgit Auer. I will always be grateful to you for all that you have done for me, which is simply so much. To my children, Ebuwa and Ivie, I love you dearly and you continue to be the source of my inspirations.

I acknowledge all my friends for being there for me every time I needed them. My thanks go to Adewumi, Toka, Christian, Katja, Eric, Andrianna, and all the others. Thanks to you all for the times we spent together and for your faith in me.

Abstract

This thesis presents a new solution to photometric stereo and its application to the automatic inspection of metallic surfaces. Four images are acquired with one camera and four light sources with different positions. This enables the reduction or elimination of undesirable effects associated with specular reflection. Three cases are considered: without specular reflection, the use of pseudo-inverse in obtaining a least squares approximation for the surface normal vectors; in the case where one image pixel is subject to specular reflection then the three remaining pixels are used, enabling complete reconstruction; when more than one pixel is affected then the normal vector is assumed to be vertical.

Surface reconstruction from the surface normal vectors is performed using a new technique based on global discrete polynomial moments. This is a new general solution to surface reconstruction from gradient fields. The equation needing to be solved is a partitioned Lyapunov equation - commonly encountered in control engineering. This reconstruction method is numerically more efficient than past solutions and delivers better reconstruction performance.

This solution enables the reconstruction of the surface geometry independent of the surface albedo. This is important for surface inspection.

Kurzfassung

Diese Arbeit präsentiert einen neuen Lösungsansatz im Bereich Photometric Stereo (PS) sowie dessen Anwendung in Bezug auf die Möglichkeiten der automatischen Oberflächeninspektion von Metallen. Vier Bilder werden durch eine einzige Kamera und vier unterschiedlich positionierte Lichtquellen aufgenommen. Dadurch sollen unerwünschte Nebeneffekte der spekularen Reflexion reduziert bzw. eliminiert werden. Drei Fälle werden betrachtet: 1) ohne spekulare Reflexion durch die Verwendung der Pseudoinversen mit der eine ,,Methode der kleinsten Quadrate" die Oberflächen–Normalvektoren zu erhalten, 2) der Fall in dem ein Bildpixel Gegenstand spekularer Reflexion wird und die drei verbleibenden Pixel verwendet werden, um eine vollständige Rekonstruktion zu erhalten und 3) wenn mehr als ein Pixel betroffen unter der Annahme, dass sich der Normalvektor vertikal verhält.

Die Oberflächenrekonstruktion mittels Oberflächen–Normalvektoren wird durch den Einsatz einer neuen Methode gezeigt, welche auf global-diskreten polynominalen Momenten basiert. Es handelt sich hierbei um eine neue, generelle Lösung zur Oberflächenrekonstruktion ausgehend von Gradientenfeldern, die aufzulösende Gleichung ist eine geteilte Lyapunov Gleichung, wie sie gewöhnlicherweise im Bereich der Regelungstechnik Anwendung findet. Die in diesem Ansatz gewählte Rekonstruktionsmethode ist numerisch effizienter als bisherige Lösungsvorschläge und zeigt bessere Rekonstruktionsergebnisse.

Eine solche Vorgehensweise erlaubt die Rekonstruktion der Oberflächengeometrie unabhängig von der jeweiligen Oberflächenalbedo, ein für die Oberflächeninspektion nicht zu hoch einschätzbarer Faktor.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Quality control has become a vital part of industrial production process. In the steel industry, for example, different types of inspection techniques and methods are often employed to inspect intermediary and finished products during and after production. Surface inspection methods are many and varied, what is finally adopted is based on a number of factors. Some of this methods form a group of inspection method known as non-destructive testing technique (NDT). A summary of NDT methods can be found in [37].

The use of computer vision in the task of inspecting material surfaces automatically has been widely studied and documented as shown in [2, 9] and [36]. However, a lot still needs to be done as suggested by [36] to improve the quality and suitability of this branch of science for inspection . There are obvious advantages as well as disadvantages of this method over others, these are summarised below as:

Advantages

1. **Flexibility:**

   A potentially greater class of objects can be inspected without any contact to the objects. For example touching a highly polished surface with a probe may not be acceptable.

2. **Speed:**

   A large number of measurements can be made in a given time without any contact to the object.

3. **Reliability:**

   Any inferences made about surface in particular should be more reliable due to the large number of measurements made. On the other hand, inspection done by coordinate measuring machines typically uses four or five readings to test a plane or cylindrical hole [36]. Three points define a plane, in which case there is only one reading serving to check the "goodness of the fit". A feature

sampled over an invariant Cartesian grid of say 1mm×1mm would lead to a large number of readings. Least square approximation technique can then be used to get the best fit to the surface.

4. **Automatic Registration:**

A vision system can determine the position and orientation of the object before inspection takes place, eliminating the need to place the object in a known position or to register it manually. Whilst coordinate measuring machines (CMMs) could achieve this time required to gather enough data to be able to accurately recognize an arbitrary pose would be considerably larger.

5. **Increased Productivity:**

Computer vision systems enable small batch jobs to be inspected efficiently. This cannot be done economically with some other methods. If the system is a real-time one information about any defects detected could be passed back to the manufacturing stage so that remedies could be effected.

6. **Tireless**:

The system once programmed can perform the same task repeatedly and to the same accuracy without been tired.

Disadvantages

1. **Lack of Access:**

A visual inspection system can only check visible features. Consequently any feature which cannot be seen from any position of the camera(s) cannot be inspected by this method. Such features are defects imbedded within the materials and are referred as subsurface defects. Also, it may be difficult to reliably inspect features which can not be completely imaged in its totality by the camera from a single viewpoint.

2. **Resolution:**

The resolution of the imaging device limits the size of flaws it can detect. Thus flaws smaller than its resolution go undetected.

1.1 What is Unique in this Thesis

The idea of photometric stereo is not new and its application to surface inspection has been used before [2] and [10]. What is new, however is the use of the number of light source in solving the problem of specularity without need to go into more laborious computations such as thresholding and other probabilistic measures in eliminating specularity in image pixels as suggested in many works such as [10] and [11] just to mention but two.

It is one thing to perform photometric stereo and another to reconstruct the inspected surface. A new method of surface reconstruction using discrete polynomial moments was developed and tested in the course of this work. A number of methods are available for surface reconstruction using moments and these are summarized in [34] together with their advantages and limitations. In their work Bang-Hwan and Rae-Hong [38] used the Legendre polynomials derived from gradients obtained from multi-image photometric stereo for surface reconstruction. However, they only succeeded in reconstruction small patches of the image at a time. This was as a result of the error encountered in generating their polynomial basis. This error limited the degree and size of the polynomial that could be used. However, we believe that the new method proposed and used in this thesis is better than any previous work because the new method can reconstruct any surface of any size.

1.2 Organisation of the Thesis

This thesis is organized as follows:

Chapter two: This chapter is divided into two parts: in the first part a brief review of the mathematical foundation of projective geometry as it pertains to image formation and transformation is given. The second part introduces the various reflection models and explains the model chosen for this work. This chapter enables us understand the processes involved in the theme of this work.

Chapter three: The method of photometric stereo in its original form is presented in this chapter. Problems associated with the use of this technique for metallic surface inspection are also presented. The second part of this chapter is a discussion on two types of moments and the polynomial from which they are generated. Specifically, non-orthogonal and orthogonal moments are presented.

Chapter four: This chapter describes the experiment carried out in acquiring the images using four white light source. The new surface reconstruction algorithm is also presented and their application to the acquired images is shown. The results are also shown.

Chapter five: The results of the experiment is discussed and conclusions are drawn.

Chapter six: It is our belief that there is still room for continuation as far as this work is concerned. Areas where this could be done are highlighted in this chapter.

Appendix MATLAB® source code.

Chapter 2

Mathematical And Geometrical Backgrounds

2.1 Introduction

The word geometry is derived from the Greek word *geometria* which means "measurement of the earth" [1]. In the realm of image processing four different types of geometries are encountered. These are: Euclidean; similarity; affine; and projective geometries.

Euclidean geometry is familiar to us because it describes our three-dimensional world so well. In Euclidean geometry, the sides of objects have lengths, intersecting lines determine angles between them, and two lines are said to be parallel if they lie in the same plane and never meet. Moreover, these properties do not change when the Euclidean transformations (translation and rotation) are applied. When the imaging process of a camera is considered it is seen that Euclidean transformation is insufficient in describing the process because lengths and angles are no longer preserved and parallel lines may intersect.

The relationships between the various types of geometries are shown in Table 2.1 below.

2.2 Projective Geometry

Projective geometry applies to a variety of disciplines. This section gives a brief introduction to projective geometry and its application to computer vision. Projective geometry in two dimension is the geometry of the projective transformations of a plane. These transformations model geometric distortions which arise when a plane is imaged by a perspective camera. The most important aspect of projective geometry is the introduction of homogeneous coordinates which represents projective transformation as matrix multiplication [4]. This compact form allows many of the significant aspects of projective transformations and projective geometry to be demonstrated using simple matrix algebra

	Euclidean	Similarity	Affine	Projective
Transformations				
rotation	X	X	X	X
translation	X	X	X	X
uniform scalling		X	X	X
non-uniform scalling			X	X
shear			X	X
perspective projection				X
composition of projection				X
Invariants				
length	X			
angle	X	X		
ratio of length	X	X		
parallelism	X	X	X	
incidence	X	X	X	X
cross-ratio	X	X	X	X

Table 2.1: The four different geometries, the transformations allowed in each, and the measures that remain invariant under those transformations [1].

computations. In Euclidean coordinates, many of these derivations become difficult, if not impossible.

Another major contribution of projective geometry to computer vision according to [4] is the formulation of invariants under projective transformation. Under perspective imaging certain geometric properties are preserved, such as collinearity (a straight line is imaged as a straight line), whilst others are not, for example parallel lines are not imaged as parallel lines in general. Projective geometry models this imaging and also provides a mathematical representation appropriate for computations. A wide variety of these invariants are available for sets of points and lines as well as higher order algebraic curves.

2.2.1 Homogeneous Coordinates

As it is well known a point in the plane $\mathbb{R}^2$ may be represented as (*x,y*)in the Euclidean plane $\mathbb{R}^2$ [18]. To represent the same point in the projective plane, we simply add a third coordinate of 1 at the end:(*x,y,1*)[1]. Overall scaling is unimportant, so the point (*x,y,1*) is the same as $(\alpha x, \alpha y, \alpha)$ for any non-zero (α). In other words,

$$(X, Y, W) = (\alpha X, \alpha Y, \alpha W) \tag{2.1}$$

for any $(\alpha \neq 0)$. (Thus the point (0,0,0) is disallowed). Because scaling is unimportant, the coordinates (X,Y,W) are called the *homogeneous coordinates* of the point. In vector notation (X,Y,W) is written [X,Y,W]^T

[1]In general, a point in an n dimensional Euclidean space is represented as a point in an $(n+1)$ dimensional projective space

In other to represent a line in the projective plane, we begin with the standard Euclidean formula of a line

$$(ax + by + c) = 0, \tag{2.2}$$

and use the fact that the equation is unaffected by scaling to arrive at the following

$$\begin{aligned} (aX + bY + cW) &= 0, & (2.3)\\ \mathbf{u}^T\mathbf{p} = \mathbf{p}^T\mathbf{u} &= 0, & (2.4) \end{aligned}$$

where $\mathbf{u} = [a, b, c]^T$ is the line and $\mathbf{p} = [X, Y, W]^T$ is a point on the line. We see that points and lines have the same interpretation in the projective space.

To transform a point in the projective plane back into the Euclidean coordinates, we simply divide by the third coordinate: $(x,y) = (X/W, Y/W)$. We immediately see that the projective plane contains more points than the Euclidean plane, that is; points whose third coordinate is zero. These points are called *ideal points* or *points at infinity*. There is a separate ideal point associated with each direction in the plane: for example, the points (1,0,0) and (0,1,0)are associated with horizontal and vertical directions, respectively. All the ideal points lie on a line, called the *ideal line* or *line at infinity*, which is represented as (0,0,1).

2.2.2 Image Formation

Consider the figure shown in Fig.2.1. Let p denotes a scene point with coordinates *(X, Y,Z)* and p' denotes its image with coordinates (x, y, z). Since the three points p,o,p' are collinear, then $\vec{op'} = \lambda\vec{op}$, for some number λ. This gives the perspective projection equations, to which we are familiar:

$$x = -f\frac{X}{Z} \tag{2.5}$$

$$y = -f\frac{Y}{Z} \tag{2.6}$$

where the point (X, Y, Z) in the world is projected to the point (x,y) on the image plane. Equations 2.5 and 2.6 are inherently nonlinear. Converting to homogeneous coordinates, however, makes them linear, that is

$$\mathsf{p\prime} = \mathsf{T}_p\mathsf{p} \tag{2.7}$$

where $\mathsf{p\prime} = [x, y, w]^T$ and $\mathsf{p} = [X, Y, W]^T$ and the perspective projection matrix T_p is given by:

$$\mathsf{T}_p = \begin{bmatrix} -f & 0 & 0 & 0 \\ 0 & -f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.8}$$

The *collineation* of $\mathbb{R}^2$ is defined as the mapping from the plane to itself such that the collinearity of any set of points is preserved.

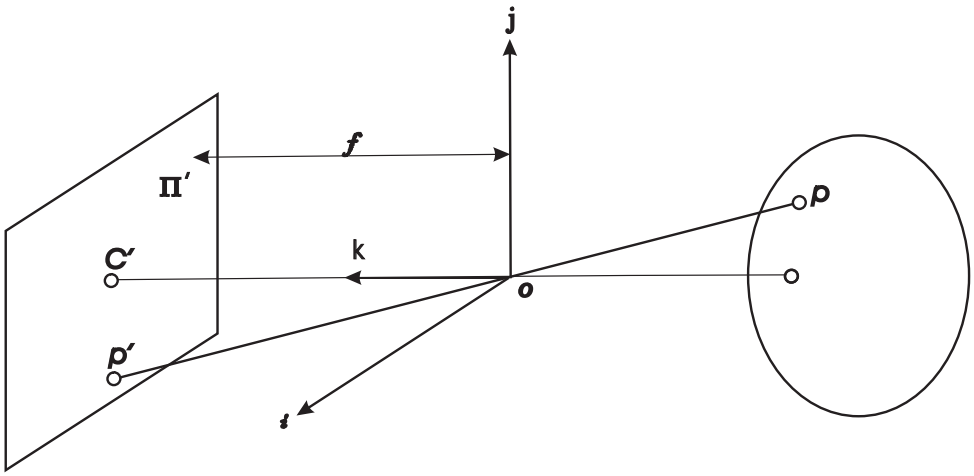


Figure 2.1: The perspective projection of a point onto a plane.

In applying projective geometry to the imaging process, it is customary [18] to model the world as a three dimensional projective space, equal to $\mathbb{R}^3$ along with points at infinity. This is similar to modelling of a two-dimensional projective transformation of the image in a plane $\mathbb{R}^2$. Central projection is simply a map from $\mathbb{R}^3$ to $\mathbb{R}^2$. Consider points in $\mathbb{R}^3$ wrtitten in homogeneous coodinates $(X, Y, Z, W)^T$. Let the centre of projection be $(0, 0, 0, 1)^T$, then it is seen that for all points $(X, Y, Z, W)^T$; when X, Y and Z are fixed but varying W this forms a single ray and passing through the point, the centre of projection, and hence all map to the same point. Thus, the final coordiante of (X, Y, Z, W) is irrelevant to where the point is imaged. The image point is the point in $\mathbb{R}^2$ with coordinate $(x, y, z)^T$. The mapping may be represented by a mapping of 3D homogeneous coordinates, represented by 3×4 matrix T with the block structure $\mathsf{T} = [\mathbf{I}_{3\times3}|\mathbf{0}_3]$, where $\mathbf{I}_{3\times3}$ is the 3×3 identity matrix and $\mathbf{0}_3$ is a zero 3-vector. Generally image projection is represented by an arbitrary 3×4 matrix of rank 3 acting on the homogeneous coordinates of the point in $\mathbb{R}^3$ mapping to the image point in $\mathbb{R}^2$. The matrix T is called the *camera matrix*.

Following from the above, the action of a projective camera on a point in space may be expressed in terms of a linear mapping of homogeneous coordinates as:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \mathsf{T}_{3\times4} \begin{pmatrix} X \\ Y \\ Z \\ W \end{pmatrix} \tag{2.9}$$

Furthermore, if all the points lie on a plane (this plane can be choosen as the plane $Z = 0$) this reduces the linear mapping to

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \mathsf{H}_{3\times3} \begin{pmatrix} X \\ Y \\ W \end{pmatrix} \tag{2.10}$$

which is a projective transformation.

Referring to Fig.2.1, each point $\mathbf{p}$ is transformed into a point $\mathbf{p}'$:

$$\mathbf{p} = \mathsf{T}\mathbf{p}' \tag{2.11}$$

The entire image formation process includes perspective projection,along with matrices for external and internal calibration:

$$\begin{aligned}\mathbf{P} = \mathsf{T}_i\mathsf{T}_p\mathsf{T}_e &= \begin{bmatrix} k_u & k_v & u_0 \\ 0 & k_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} -f & 0 & 0 \\ 0 & -f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \\ &= \begin{bmatrix} \alpha_u & -\alpha_u cot\theta & u_0 \\ 0 & a_v/sin\theta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} R & \mathbf{t} \end{bmatrix} \\ &= \mathsf{AD} \end{aligned} \tag{2.12}$$

where $\mathsf{T}_i, \mathsf{T}_e$ are internal and external calibration matrices respectively and α_u and α_v are the scale factors of the image plane (in unit of the focal plane f), θ is the skew ($\theta = \frac{\pi}{2}$ for most real cameras). The point (u_0 and v_0) is the principal point, R is the 3×3 rotation matrix and $\boldsymbol{t}$ is the 3×1 translation vector. The matrix A contains the internal parameters and the perspective projection, while D contains the external parameters.

It is sometimes convenient to decompose the 3×4 projection matrix $\mathbf{P}$ into a 3×3 matrix P and a 3×1 matrix p

$$\mathbf{P} = [\mathsf{P} \quad \mathsf{p}] \tag{2.13}$$

so that

$$\mathsf{P} = \mathsf{AR} \quad \text{and} \quad \mathsf{p} = \mathsf{A}\boldsymbol{t} \tag{2.14}$$

2.2.3 Orthographic Projection

The figure shown in Fig 4.1 depicts the central projection of points in 3D onto an image plane. The essential geometric properties of this projection can be modelled by the mapping of 3D projective space onto a projective plane , which conveniently can be represented by a linear homogeneous transformation.

Just as discussed in Section 2.2.2 for 2D transformation, a general transformation in 3D onto a plane is defined by a 4×4 matrix multiplication:

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & t_{44} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}. \tag{2.15}$$

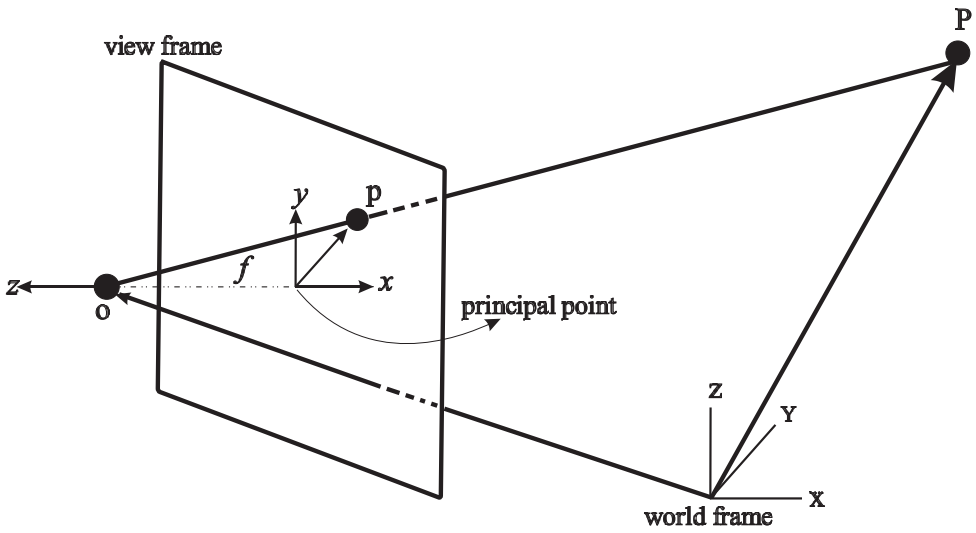


Figure 2.2: Perspective camera geometry

A projection onto space of one lower dimension can be achieved by eliminating one of the coordinates of the transformed projective space. If, for example, the plane defined by $x_4 = 0$ is chosen, that means all points on the plane can be represented by the homogeneous coordinate vector, $(x_1, x_2, x_3)^T$. The image projection is then given by :

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix}. \tag{2.16}$$

or x = TX.

The Euclidean projection of a point **P** in the world coordinate frame to a point **p** in the camera frame is given by :

$$\mathbf{p}_{cam} = \mathsf{R}(\mathbf{P}_{world} - \mathbf{O}) \tag{2.17}$$

where the matrix R given by

$$\mathsf{R} = \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{R}_2 \\ \mathbf{R}_3 \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{2.18}$$

is the rotation matrix from the world coordinate frame to the camera coordinate frame. **O** is the translation vector from the the world origin to the camera origin. The origin of the camera is taken to be the centre of projection. The transformation is carried out by applying translation **O** followed by rotation R.

These transformation can be applied by a single homogeneous 4×4 transformation matrix:

$$\mathrm{T}_E = \begin{bmatrix} \mathbf{R}_1 & -(\mathbf{R}_1.\mathbf{O}) \\ \mathbf{R}_2 & -(\mathbf{R}_2.\mathbf{O}) \\ \mathbf{R}_3 & -(\mathbf{R}_3.\mathbf{O}) \\ 0\,0\,0 & 1 \end{bmatrix} \tag{2.19}$$

Next, the transformed point is projected into the image plane by the matrix:

$$\mathrm{T}_{proj} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/f & 0 \end{bmatrix} \tag{2.20}$$

The composite transformation matrix $\mathsf{T} = \mathsf{T}_{proj}\mathsf{T}_E$

$$\mathrm{T} = \begin{bmatrix} \mathbf{R}_1 & -(\mathbf{R}_1.\mathbf{O}) \\ \mathbf{R}_2 & -(\mathbf{R}_2.\mathbf{O}) \\ \mathbf{R}_3/f & -(\mathbf{R}_3.\mathbf{O})/f \end{bmatrix} \tag{2.21}$$

As opposed to perspective transformation orthographic projection results from the limit where the rays from the centre of projection are parallel. This limit can be represented by letting the focal length approach infinity while keeping the scale factor at unity. The form of the perspective transformation matrix becomes,

$$\mathrm{T}_{orth} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & -(\mathbf{R}_1.\mathbf{O}) \\ r_{11} & r_{12} & r_{13} & -(\mathbf{R}_2.\mathbf{O}) \\ r_{11} & r_{12} & r_{13} & -(\mathbf{R}_3.\mathbf{O}) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.22}$$

The main difference between orthographic projection and weak prospective is that distances along directions parallel to the image plane are preserved under orthography. For this reason, orthographic projections are used extensively to define 3D dimensions of object [4]

2.2.4 Mappings between planes

The general projective transformation matrix can be used to specify the mapping between two planes in space. Here the points in space are assumed to lie on a plane. Without loss of generality, it can be assumed that the first plane corresponds to the X,Y plane of the world coordinate system and the second plane is the image plane. That is

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & t_{44} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 0 \\ 1 \end{bmatrix}. \tag{2.23}$$

which can be rewritten as

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} & t_{14} \\ t_{21} & t_{22} & t_{24} \\ t_{31} & t_{32} & t_{34} \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix}. \tag{2.24}$$

This shows that the general projective mapping between planes in space is specified by a

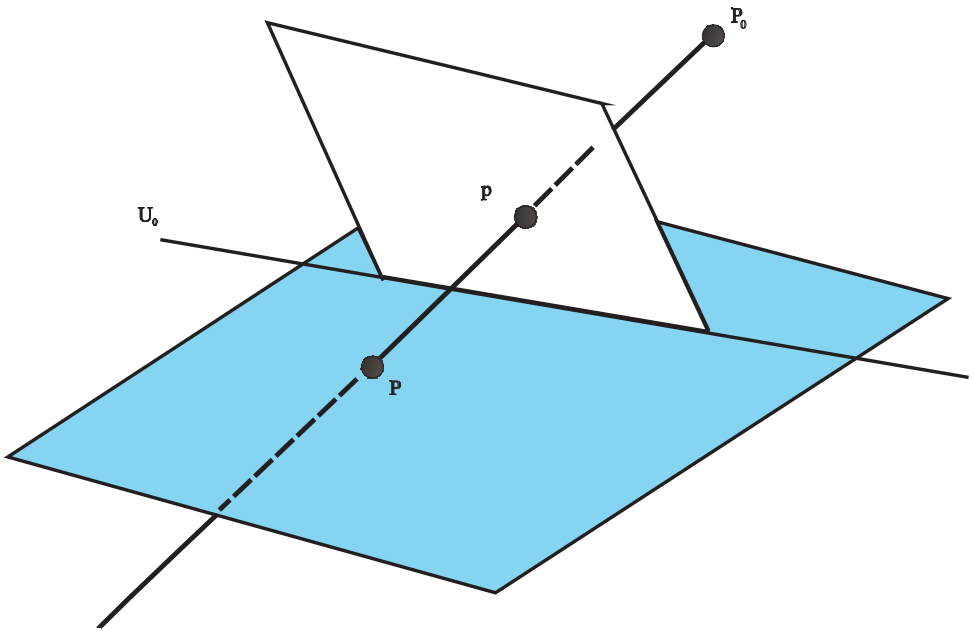


Figure 2.3: The perspective mapping between two planes.
Note that the line $\boldsymbol{u}_0$ is fixed under the perspectivity.

3×3 homogeneous transformation. The elements of the 3×3 matrix corresponds to the first, second and fourth columns of the original matrix T.

The perspective mapping between two planes is a central projection from a single point in space where corresponding points in the planes are collinear with the centre of projection. In the case of perspective mapping it is observed that the first two columns of the 3×3 matrix must be orthogonal and have the same norm in a coordinate frame where $f = 1$. These perspective transformation could be called *perspectivities*[4]. The geometry of perspective transformation is shown in Fig.2.3. The line of intersection between the two planes is fixed under perpectivity. Perspective mapping of the plane do not form a group since the composition of the two perspectivities is not in general a perspective transformation, this means that the special form of the perspective matrix is not in general preserved by the product of two such matrices.

2.3 Illumination Geometry

In this section the definition of illumination angles relating the light source to the viewed object is introduced. These angles and how they are positioned in space are shown in Fig 2.6 below. It is assumed that the test surface mounted in the (x, y) plane is perpendicular to the camera axis (the z-axis). Orthogonal camera model, briefly discussed in Section 2.2.3, is also assumed. Also assumed is that the test surface is illuminated by a point

light source located at infinity. This means that the incident vector field is uniform in magnitude and direction over the test area.

For the purpose of this thesis the illumination angles are defined as follows:

i. τ is the tilt angle, that is the angle that the projection of the illuminant vector incident onto the test surface plane makes with an axis in that plane.

ii. σ the slant angle is the angle that the illuminant vector makes with a normal to the test surface.

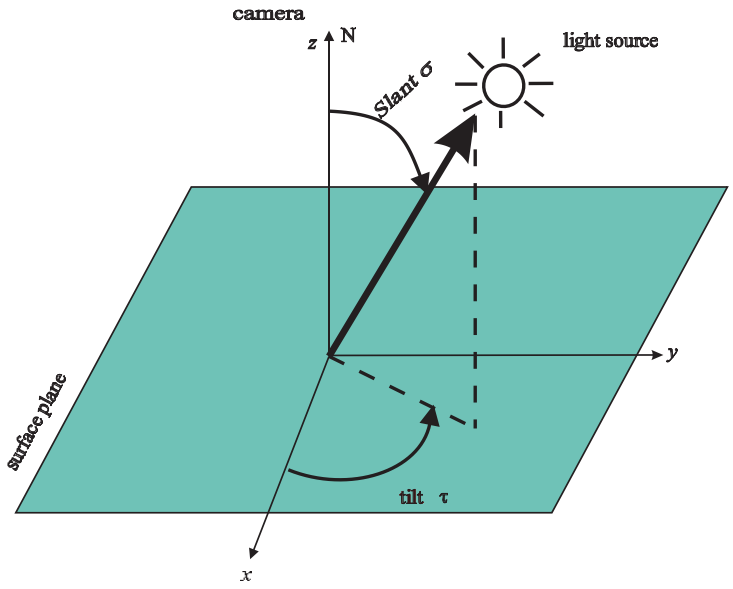


Figure 2.4: Illumination geometry showing the tilt angle τ and the slant angle σ.

2.3.1 Diffuse and Specular Reflections

When light strikes a smooth surface such as a mirror or a still body of water the resulting reflection is highly directional, it is known as *specular reflection* (Fig.2.5a) and it has a spectral distribution similar to that of the illuminate. For a normal object this reflections is only a part of the total reflection.

Another type of reflection, which is obtained from a rough surface is called *diffused reflection*. In this type of reflection a collimated beam emerges in all directions (Fig.2.5b). Diffused reflection may in turn be divided into two parts: directionally diffused and uniformly diffused. In the former case the incident light is scattered in all directions, while in the latter case the light is scattered uniformly over the surface after undergoing multiple internal reflections. The spectral distribution is modified by the colour of the object.

In practice the reflection process may well be a combination of both diffuse and specular components. An example of this is a *spread reflection* (2.5c), which has a dominant directional component that is partially diffused by surface irregularities.

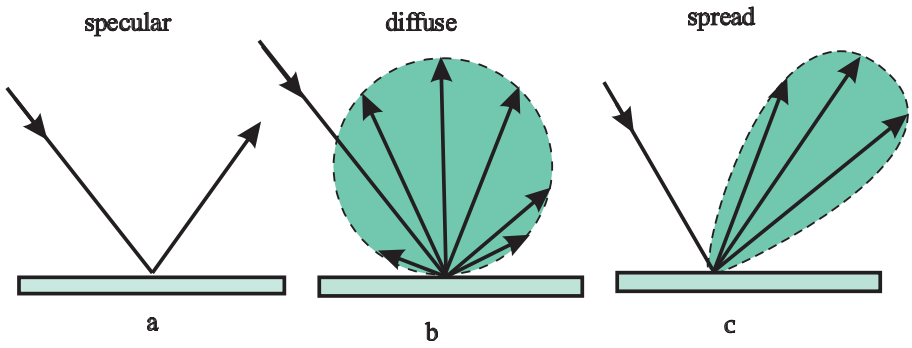


Figure 2.5: Various types of reflection. (a) specular, (b) diffuse, (c) spread

2.4 Mathematical Representation of Reflection Models of Image Formation Process from a Surface

Given a light source, a surface, and an observer, a reflectance model describes the intensity and spectral composition of the reflected light reaching the observer. It, therefore, describes the manner in which incident light interacts with an object surface. Three things happen when a ray of light interacts with a solid material: it may be reflected; absorbed; or transmitted. The manner in which light is reflected from an object surface is dependent upon various optical properties, together with the surface microstructure of the object material, and the wavelength, angle of incidence, and polarisation of the incident light [2].

Various optical modelling techniques used for explaining the formation of 2D images from 3D objects are now presented.

2.4.1 Surface Roughness Model

The manner in which light is reflected by a surface is dependent on the shape characteristics of the surface. To analyse the reflection of incident light, a mathematical model of the surface will be used to describe what happens when light is reflected. To this effect two models are reviewed. These are: the height distribution model; and the slope distribution model [3].

2.4.1.1 Height distribution model

This model expresses the height coordinate of the surface as random function from which the shape of the surface is determined by the probability distribution of the height coordinates. For this purpose two types of surface are differentiated. A surface is said to be *isotropic* if it exhibit the same surface texture in all directions. Conversely a surface whose texture changes as it is rotated about its normal but with the direction of illumination unchanged is said to be anisotropic. Consider the figure shown in Fig 2.6. For a surface which is isotropic the height of a point on the surface is represented by a Gaussian random

function:

$$p(s) = \frac{1}{\sigma_s\sqrt{2\pi}}\mathrm{e}^{s^2/2\sigma_s^2} \tag{2.25}$$

where σ_s is *rms roughness* of the surface [19].

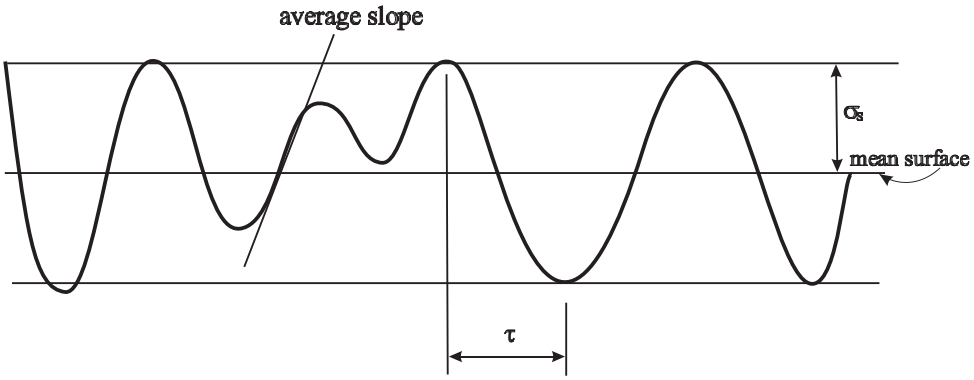


Figure 2.6: surface height distribution model.

For an *anisotropic* surface the surface roughness is represented as the measure of the standard deviation of the surface heights denoted by σ_s (i.e. the root-mean-square roughness) and the average roughness denoted by R_{cla} (Centre Line Average CLA). Both functions are represented mathematically as shown below.

$$\sigma_s = \sqrt{\frac{1}{n}\sum_{n=1}^{n}\left[s(x) - \overline{s(x)}\right]^2} \tag{2.26}$$

$$R_{cla} = \frac{1}{n}\sum_{n=1}^{n}|s(x)| \tag{2.27}$$

s(x) represents the height of a surface at a point x along the profile and $\overline{s(x)}$ is the expectation of s(x) and n is the number of pixels. Hence, they provide measures of the localised surface deviation about a nominal path [3].

2.4.1.2 Slope distribution model

[3] postulated that the scattering of light rays by a surface is dependent on the local slope of the surface and not the local height of the surface. The slope model, he concluded, is more suitable for the investigation of the problem of surface reflection. For this purpose, it is useful to think of a surface as a collection of planar micro-facets. This is shown below in Fig.2.7. For a surface, which is mathematically smooth and whose facets ε are small compared to the area L of the surface patch (that is $L >> \varepsilon$), we may use two slope parameters, p_{rms} and q_{rms}, as a measure of roughness. They correspond to the standard deviation of the surface partial derivatives p and q

$$p_{rms} = \sqrt{\frac{1}{n}\sum_{n=1}^{n}\left(\frac{\partial s(x,y)}{\partial x} - \overline{\frac{\partial s(x,y)}{\partial x}}\right)^2} \tag{2.28}$$

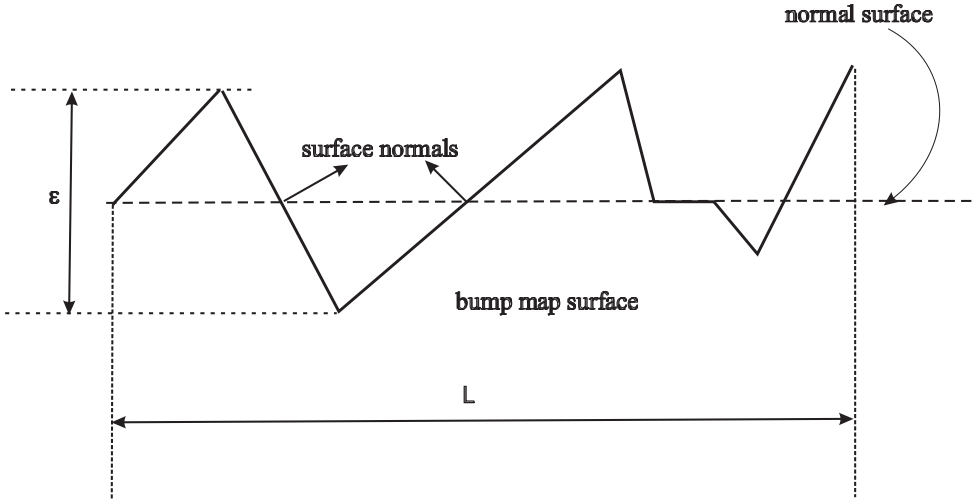


Figure 2.7: A surface as a collection of planar micro-facets.

$$q_{rms} = \sqrt{\frac{1}{n}\sum_{n=1}^{n}\left(\frac{\partial s(x,y)}{\partial y} - \overline{\frac{\partial s(x,y)}{\partial y}}\right)^2} \tag{2.29}$$

where $p = \frac{\partial s(x,y)}{\partial x}$ and $q = \frac{\partial s(x,y)}{\partial y}$ are partial derivatives measured along the x and y axes respectively. Therefore, p_{rms} and q_{rms} can be used to describe surfaces with both isotropic and directional roughness.

2.5 Reflection Modelling

The figure below Fig 2.9 illustrates the geometry of light reflection at a surface. The **B**idirectional **R**eflectance **D**istribution **F**unction **(BRDF)** is the basis of all reflection models. It is the general model that relates the energy arriving at a surface from the direction of the illuminate, to the reflected intensity in the direction of the viewer. It is defined as the ratio of the total reflected intensity in the direction(θ_r, ϕ_r) to the energy incident per unit time and per unit area onto the surface from the direction(θ_i, ϕ_i) [20].

The BRDF is material and wavelength (λ) dependent, and with the variables shown in Fig.2.9 the function is represented as:

$$F_{bdrf}(\lambda, \theta_i, \phi_i, \theta_v, \phi_v) = \frac{dI_r(\theta_r, \phi_r; \theta_i, \phi_i)}{I_i(\theta_i, \phi_i)\cos\theta_i d\omega_i} \tag{2.30}$$

where λ is the wavelength of the incident light, (θ_i and ϕ_i) denotes its direction, and (θ_v and ϕ_v) describes the direction to the viewer.

Fig 2.8 shows the various components of the reflection model and the angle dω_i, which is known as the incident solid angle. In the following sub-sections two reflection models are considered: the Phong reflection model; and the Lambertian reflection model.

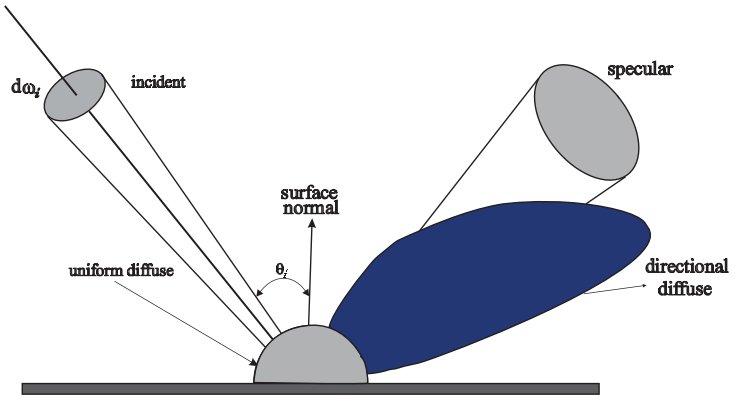


Figure 2.8: Components of light reflection model.

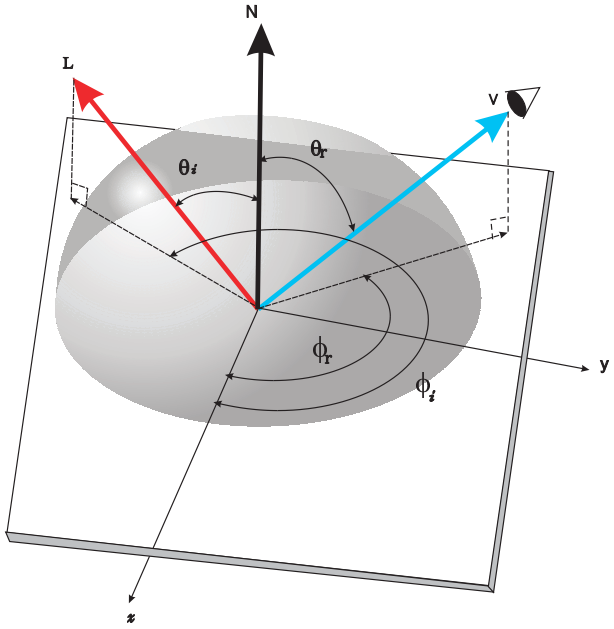


Figure 2.9: The geometry of light reflection [3].

2.5.1 The Phong Reflection Model

The most commonly used type of reflection model in computer vision is known as Phong's reflection model. It was developed by Bui Tuong Phong [5]. It is a linear combination of three parts: ambient; specular; and diffuse, see Fig 2.8. The ambient, or constant component, is a surrogate in the absence of a simple mechanism to model global diffuse inter-reflection. The specular component is a lobe, which spreads out around the specular direction and is modelled by using a cosine function raised to a power. This model is given in terms of unit vectors associated with the geometry of the point under consideration.

In the figure shown below Fig 2.10, L is the light source, N is the surface normal and R is the reflected light. By Nell's law the angle θ is constant. Therefore, the Phong's model

states that

$$I(n.\Phi) = I_a k_a + (\sum_{lights} I_i k_d(\boldsymbol{L.N}) + I_i k_s(\boldsymbol{R.V})^n) \tag{2.31}$$

where,

I represents the intensity of the surface location under consideration;
I_a is the constant intensity of the ambient light;
I_i is the intensity of the input light;
k_d is the coefficient of diffuse reflection for the material;
k_s is the coefficient of specular reflection;
n is a shininess constant for this material; which decides how "evenly" light is reflected from a shiny spot [21];
N is the local surface normal;
R is the direction a perfectly reflected ray of light (represented as a vector) would take from the point where the ray strikes the surface;
V is the direction towards the viewer (such as a virtual camera);
ϕ is the angle between the mirror vector **R** and the viewing vector **V**; and **(L.N)** and **(R.V)** and are dot products of the respective vectors.

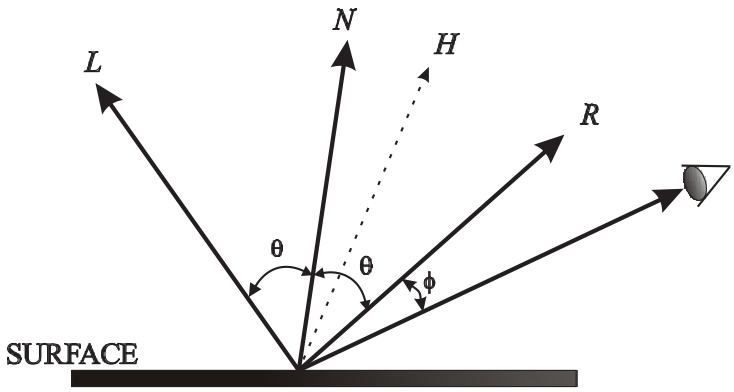


Figure 2.10: [Phong's Vectors]Vectors used in the Phong's reflection model.

In Phong's model, the light sources are considered as point sources situated at infinity. Therefore, the angle θ between the incident light and the normal to a planar surface is constant over the surface. The observer is assumed to be positioned at infinity and hence the angle ϕ is constant over a planar surface also. The diffuse and specular terms are modeled as local components only. Shadows are not considered. The colour of the specular term is assumed to be that of the light source.

In the this model the diffuse term does not use the direction towards the viewer $\boldsymbol{V}$, as the diffuse term is equal in all directions from the point, including the direction of the viewer. The specular term, however, is large only when the reflection vector $\boldsymbol{R}$ is nearly aligned with viewpoint vector $\boldsymbol{V}$, as measured by the n power of the cosine of the angle between them, which is the *dot product* of the normalized direction vectors $\boldsymbol{R}$ and

$\boldsymbol{V}$. When n is large, representing an almost mirror-like reflection, the specular reflection will be very small because the high power of the cosine will go rapidly to zero with any viewpoint not aligned with the reflection.

Estimating the specular component involves the computation of the reflected vector $\boldsymbol{R}$ This is computationally expensive [3] and therefore, it is replaced with the computation of $\boldsymbol{H}$**,** a vector half-way between $\mathbf{L}$, and $\mathbf{V}$,. This is called the Blinn's method. Therefore, $\boldsymbol{H}$**,** is given by

$$\boldsymbol{H} = \frac{\boldsymbol{L} + \boldsymbol{V}}{\mathbf{2}} \tag{2.32}$$

Then the specular term in Phong's model becomes

$$I_{specular} = I_i k_s (\boldsymbol{N}.\boldsymbol{H})^n \tag{2.33}$$

2.5.2 The Lambertian Illumination Model

Let us consider a diffuse surface, which is an optically rough surface, reflecting a portion of the incident light with radiant intensity uniformly distributed in all directions. A Lambertian surface will look equally bright from any illumination direction [3]. This means that the reflected intensity is independent of the viewing direction.

However, the intensity does depend on the light source's orientation relative to the surface. This can be represented mathematically as the dot product of the surface derivative vector with the illuminant vector. Thus the Lambert's Law is represented as

$$I(x,y) = \rho\lambda(\mathbf{N}.\mathbf{L}) = \rho\lambda\frac{-p\cos\tau\sin\sigma + \cos\sigma}{\sqrt{p^2+q^2+1}} \tag{2.34}$$

where *i(x,y)* is the image intensity;

$$N = \left(\frac{-p}{\sqrt{p^2+q^2+1}}, \frac{-q}{\sqrt{p^2+q^2+1}}, \frac{1}{\sqrt{p^2+q^2+1}}\right) \tag{2.35}$$

where N is the unit vector normal to the surface *s(x,y)* at the point *x,y*;

$p = \frac{\partial s(x,y)}{\partial x}$ and $p = \frac{\partial s(x,y)}{\partial y}$ are surface partial derivatives measured along the x and y axes respectively; $L =(\ \cos\tau.\sin\sigma, \sin\sigma.\sin\tau, \cos\sigma\ )$ is the unit vector towards the light source; σ and τ are the illuminant vector angles as shown in Fig.2.4.
ρ is surface albedo, a material dependent coefficient;
λ is the strength of the light source,

However, a number of assumptions are needed before this model can be used. These are highlighted below as:

i. The surface is *ideally diffused*, which means the entire incident light is equally re-distributed in all directions, and its reflectance function is uniform.

ii. The viewer is far away from the surface relative to the size of test surface, so that orthographic projection in the image system can be assumed.

iii. Light sources are supposed to be at infinity from the surface, such that the light source energy does not depend on the position of the surface. This means that we assume that illumination is constant over the whole surface.

iv. For a perfect Lambertian model both *self* and *cast shadows* are ignored as well as *inter-reflections*.

v. Only incident angle in the range from 0 to 90 degrees are considered. Greater angles (giving rise to negative $\boldsymbol{N.L}$) are blocked by the surface and the reflected energy is 0. The light is incident on the back of the surface, which means that it is blocked by the object.

It has to be said that the Lambertian model cannot describe specular reflections, which occur at places where the direction of direct reflection equals the viewing direction. The problem of specularity and how it was dealt with is discussed in the next chapter.

Chapter 3

Implementation

3.1 Photometric Stereo

The idea of photometric stereo was first introduced by Robert J. Woodham [6] and since then it has seen an increased industrial usage for material surface inspection. For example it was used by Smith [2] to inspect ceramic tiles where he isolated surface albedo and surface flaws.

The fraction of light reflected by an object surface in a given direction depends upon the optical properties of the surface material, the surface microstructure, the spatial and spectral distribution and state of polarisation of the incident illumination. In chapter two reflectance models were developed to characterise image irradiance with respect to the illumination environment, viewing angles and material properties. The technique of photometric stereo uses these reflection models to estimate the surface properties from the transformation of image intensities arising from illumination change.

In photometric stereo several images of a static scene from the same viewpoint is taken, while alternating the illumination direction. This means, that a particular pixel in each of the consecutively acquired images corresponds to the same object point. For any particular surface location or pixel there exists three degrees of freedom (or unknowns), which are the surface reflectance factor (albedo) and two degrees of freedom, which specify the orientation of the surface. Fig.3.1 shows the schematic representation of this technique.

3.2 The Gradient Space

For a lot of surfaces, the fraction of the incident illumination reflected in a given direction depends only on the surface orientation [6]. Consider the arrangement shown in Fig.3.2. The reflectance characteristics of the surface can, therefore be represented as a function $\phi(i, e, g)$ of the three angles *i,e* and *g*. These angle are known as incident, emergent and phase angles respectively. The incident angle is the angle between the incident ray and surface normal, The emergent angle is the angle between the emergent ray and the surface normal, while the phase angle is the angle between the incident and emergent rays. They

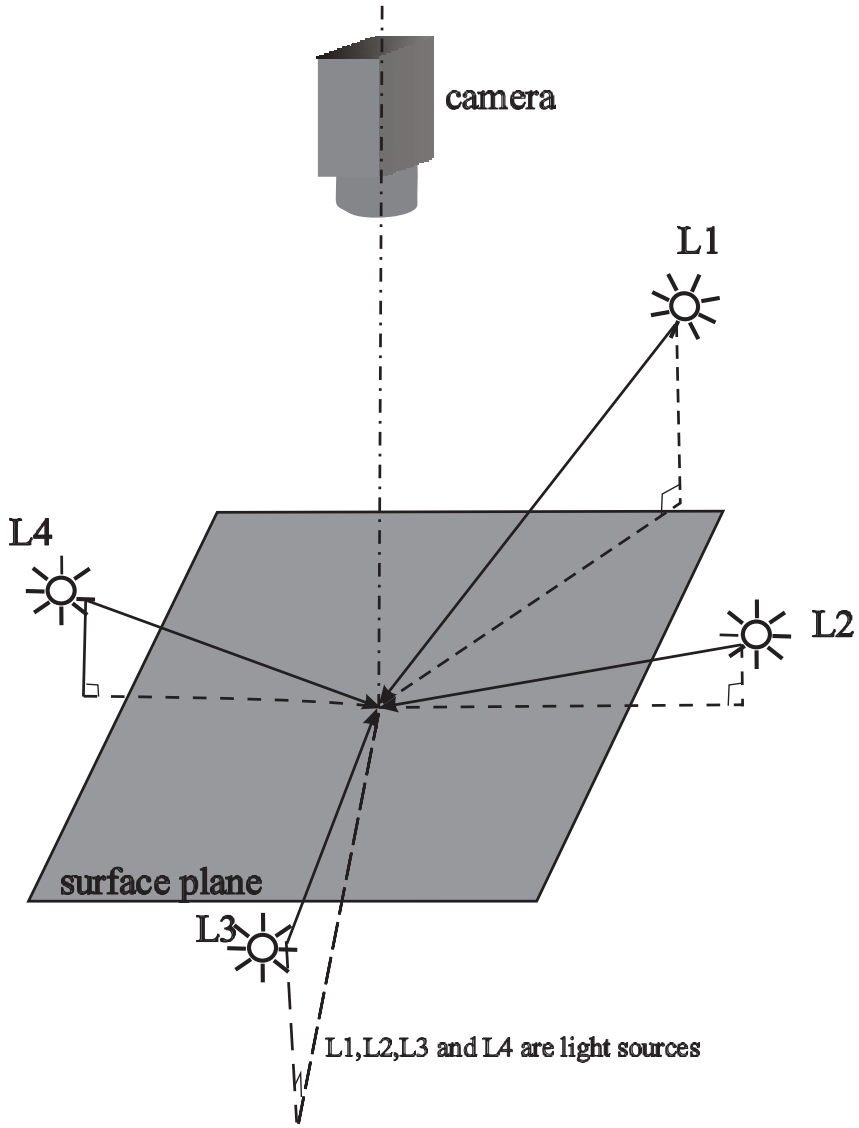


Figure 3.1: A schematic diagram of a 4-light source photometric stereo.

are defined relative to a local surface normal. The function $\phi(i, e, g)$ determines the ratio of surface radiance to irradiance measured per unit surface area, per unit solid angle, in the direction of the viewer. This function is related to the BDRF defined in chapter two.

Fig 3.3a shows how perspective transformation of a surface is done by an image forming device. However, if the size of the object viewed is small compared to the viewing distance, then the perspective projection can be approximated as an orthogonal projection. This is shown in Fig 3.3b.

Consider an imaging device which performs orthographic projection. To standardise the imaging geometry, it would be convenient to choose a coordinate system such that the viewing direction is aligned with the negative z-axis. Let us also assume an appropriate

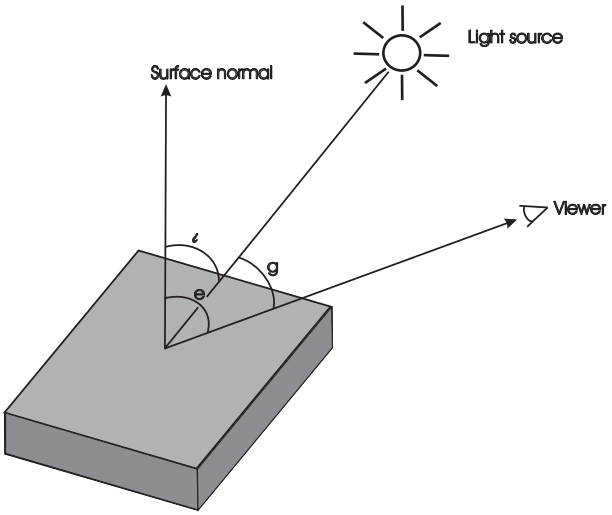


Figure 3.2: Incident, emergent and phase angles.

scaling of the image plane such that object point (x, y, z) maps onto the image point (u, v) where $u = x$ and $v = y$. One advantage for doing this is that image coordinates (x, y) and objects coordinate (x, y) can be referred to interchangeably.

If the equation of a plane is given as $\mathbf{ax + by + cz + d = 0}$, then a surface normal in this plane is (**a,b,c**). We can extend this to a curved surface by consideration of the tangent plane at a point on the surface patch. If the equation of a curved surface is given by

$$z = f(x, y) \tag{3.1}$$

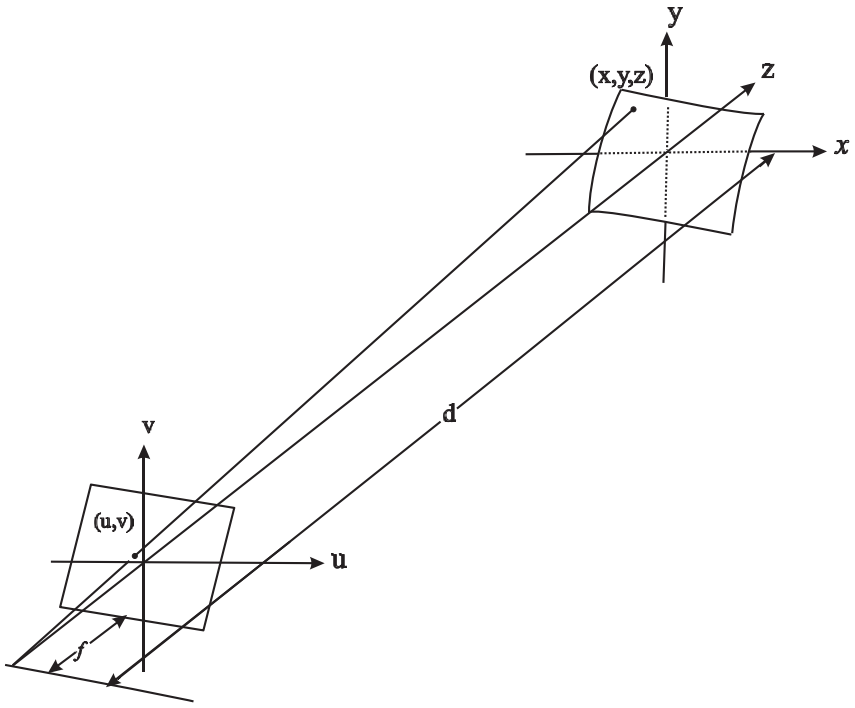
then the surface normal is given by the vector

$$\left[\frac{\partial f(x,y)}{\partial x}, \frac{\partial f(x,y)}{\partial y}, -1\right] \tag{3.2}$$

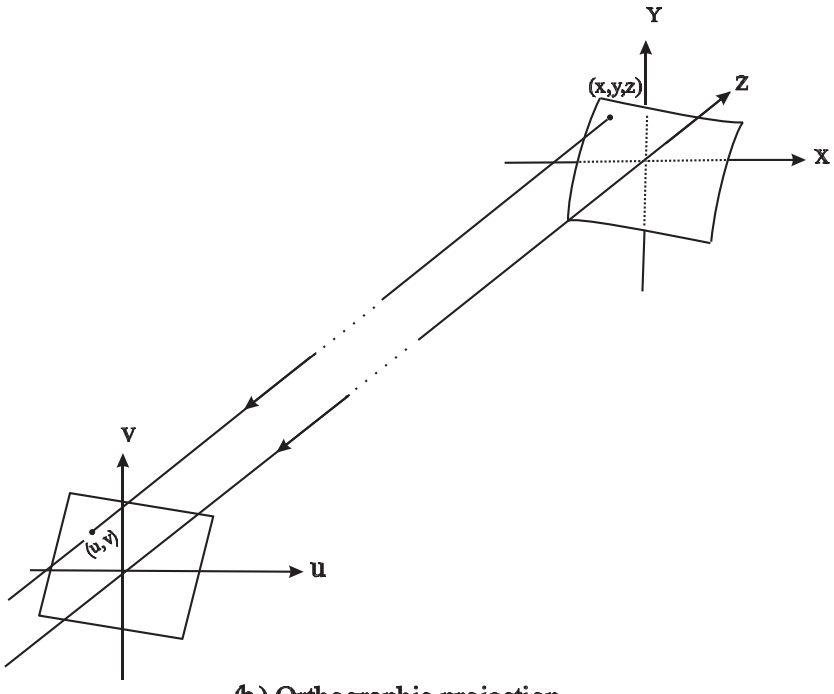
If parameters p and p are defined as:

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

then the surface normal can be written as $[p, q, -1]$. The quantity (p, q) is called the *gradient* of $f(x, y)$ and *gradient space* is the two dimensional space of all such points (p, q). Gradient space is a convenient viewer-centred representation of surface orientation. Parallel planes map into a common point in the gradient space. Planes perpendicular to the viewing direction map onto the origin of gradient space. Moving away from the origin in gradient space, the distance from the origin equals the tangent of the emergent angle, e, between the surface normal and the viewing point.



(a) Perspective projection



(b) Orthographic projection

Figure 3.3: Characterizing image projections. (a) illustrates perspective projection. (b) For objects that are small relative to the viewing distance, the image projection can be modelled as orthographic projection

3.3 The reflectance map and Image Irradiance Equation

The surface normal vector relates geometry to image irradiance because it determines the angles i and e appearing in the surface reflectance function $\phi(i, e, g)$. In this work orthographic projection is assumed, this means that the viewing direction and hence the phase angle is constant for all the surface elements.An ideal imaging device produces image irradiances proportional to the scene radiances. Thus for a fixed light source and geometry, the ratio of scene radiance to irradiance depends only on the gradient co-ordinates p and q (i.e. on the surface normal vector). If we suppose each surface element receives the same irradiance, then the scene radiance, and hence image intensity, depends only on the surface normal defined by p and q.

The *reflectance map* is represented as a function $\mathbf{R}(p, q)$. A reflectance map captures the surface reflectance of an object material for a particular light source, object surface and viewing geometry [6]. It determines the proportion of light reflected as a function of p and q. The viewed image intensity is directly proportional to the surface radiance [7]. Setting the proportionality constant to one, the image intensity and reflectance map are equivalent. This is expressed mathematically as:

$$\mathrm{I}(x, y) \quad = \quad \mathbf{R}(p, q) \tag{3.3}$$

Recall that for two vectors x and y, $\mathbf{x}.\mathbf{y} = \|x\|\|y\| \cos\theta$, where θ is the angle between x and y. Therefore expressions for $\cos(i)$, $\cos(e)$ and $\cos(g)$ can be derive using normalised dot products of the surface normal vector, $\mathbf{n} = [p, q, -1]$, the vector $\mathbf{l} = [p_s, q_s, -1]$, which points in the direction of the light source, and the vector $\mathbf{V} = [0, 0, -1]$, which points in the direction of the viewer. Thus,

$$\cos(i) \quad = \quad \frac{1 + pp_s + qq_s}{\sqrt{1 + p^2 + q^2}\sqrt{1 + p_s^2 + q_s^2}} \tag{3.4}$$

$$\cos(e) \quad = \quad \frac{1}{\sqrt{1 + p^2 + q^2}} \tag{3.5}$$

$$\cos(g) \quad = \quad \frac{1}{\sqrt{1 + p_s^2 + q_s^2}} \tag{3.6}$$

Equations 3.4 - 3.6 are used to transform a surface reflectance function $\phi(i, e, g)$ into a reflectance map $\mathbf{R}$(p,q). In the simplest case of a Lambertian surface

$$\mathrm{I}(x, y) \quad = \quad \mathrm{I}_a\rho\cos(i) \quad = \quad \mathrm{I}_a\rho\left(\frac{1 + pp_s + qq_s}{\sqrt{1 + p^2 + q^2}\sqrt{1 + p_s^2 + q_s^2}}\right) \tag{3.7}$$

where $\mathrm{I}_a\rho$ is also known as the reflectance factor or the *albedo*. This reflectance function corresponds to the phenomenological model of a perfectly diffuse (lambertian) surface which appears equally bright from all viewing directions.

3.4 Photometric Stereo and The Reflectance Map

The idea of photometric stereo was discussed in section 3.1. Suppose two images, represented as $\mathrm{I}_1(x, y)$ and $\mathrm{I}_2(x, y)$, are obtained by varying the direction of incident illumination. Since there has been no change in the imaging geometry, each picture element, *pixel*, (x, y) in the two images corresponds to the same object point and hence to the same same gradient (p, q). The effect of varying the direction of incident illumination is to change the reflectance map $\mathbf{R}(p, q)$ that characterises the imaging situation.

Let the reflectance maps corresponding to $\mathrm{I}_1(x, y)$ and $\mathrm{I}_2(x, y)$ be $\mathbf{R}_1(p, q)$ and $\mathbf{R}_2(p, q)$ respectively. Based on Equation 3.3 these two views are characterised by two independent equations:

$$\mathrm{I}_1(x, y) = \mathbf{R}_1(p, q) \tag{3.8}$$
$$\mathrm{I}_2(x, y) = \mathbf{R}_2(p, q) \tag{3.9}$$

Thus we can determine the surface normal parameters from two images. This means that two reflectance maps $\mathbf{R}_1(p, q)$ and $\mathbf{R}_2(p, q)$ are required. Defining the two light source vectors as $[p_1, q_1, -1]$ and $[p_2, q_2, -1]$ and assuming that Equations 3.8 and 3.9 are linear and independent there will be a unique solution for p and q. [8] showed this to be as follow:

$$p = \frac{\left(\mathrm{I}_1^2\mathrm{r}_1 - 1\right) q_2 - \left(\mathrm{I}_2^2\mathrm{r}_2 - 1\right) q_1}{p_1 q_2 - q_1 p_2} \tag{3.10}$$
$$q = \frac{\left(\mathrm{I}_2^2\mathrm{r}_2 - 1\right) - \left(\mathrm{I}_1^2\mathrm{r}_1 - 1\right)}{p_1 q_2 - q_1 p_2} \tag{3.11}$$

where provided $p_1/q_1 \neq p_2/q_2$; $\mathrm{r}_1 = \sqrt{1 + p_1^2 + q_1^2}$ and $\mathrm{r}_2 = \sqrt{1 + p_2^2 + q_2^2}$ This gives a unique solution for surface orientation at all points in the image.

However if Equations 3.10 and 3.16 are non-linear, which translates to the fact that either there are no solutions or there are more possible solutions. In the case of a Lambertian reflectance function, we have to introduce another image to remove such ambiguities, ie

$$\mathrm{I}_3(x, y) = \mathbf{R}_3(p, q) \tag{3.12}$$

It is now shown that for such a surface three images are sufficient to determine the gradients and the reflectance factor.

From Fig 3.2 and using Equation 3.7, the following equation can be deduced

$$\mathrm{I}(x, y) = \mathrm{I}_a \rho \cos(i) \tag{3.13}$$

but

$$\cos(i) = \frac{\mathbf{s.n}}{\|\mathbf{s}\|\|\mathbf{n}\|} \tag{3.14}$$

where $\mathbf{s}$ is the illumination source vector and $\mathbf{n}$ is the local surface normal. $\|s\|$ denotes the magnitude of vector $\mathbf{s}$. If $\mathrm{I}_a\rho$ is replaced with R then Equation 3.7 can be rewritten as

$$\mathrm{I}(x,y) \;=\; R\frac{\mathbf{s.n}}{\|\mathbf{s}\|\|\mathbf{n}\|} \tag{3.15}$$

Unit length of the vectors $\mathbf{s}$ and $\mathbf{n}$ are assumed, therefore the imaging equation simplify to:

$$\mathrm{I}(x,y) \;=\; R\mathbf{sn} \tag{3.16}$$

where R is known as the surface reflectance factor or the surface albedo. Let

$$\mathrm{I} \;=\; [\mathrm{I}_1,\ \mathrm{I}_2,\ \mathrm{I}_3]^T \tag{3.17}$$

be a column vector of the intensity values recorded at a point(x,y) where ^{T} represents the transposition of the vectors. And let

$$\mathbf{n} \;=\; [\mathrm{n}_1,\ \mathrm{n}_2,\ \mathrm{n}_3]^T \tag{3.18}$$

be a column vector corresponding to a unit surface normal ar the point (x,y), also let

$$\begin{aligned}
\mathsf{s}_1 &= [\ \mathrm{s}_{11},\ \mathrm{s}_{12},\ \mathrm{s}_{13}\ ]^T \\
\mathsf{s}_2 &= [\ \mathrm{s}_{21},\ \mathrm{s}_{22},\ \mathrm{s}_{23}\ ]^T \\
\mathsf{s}_3 &= [\ \mathrm{s}_{31},\ \mathrm{s}_{32},\ \mathrm{s}_{33}\ ]^T
\end{aligned}$$

be unit column vectors defining the three directions of incident illumination.
Therefore

$$\mathsf{S} = \begin{bmatrix} \mathrm{s}_{11} & \mathrm{s}_{21} & \mathrm{s}_{31} \\ \mathrm{s}_{12} & \mathrm{s}_{22} & \mathrm{s}_{32} \\ \mathrm{s}_{13} & \mathrm{s}_{23} & \mathrm{s}_{33} \end{bmatrix}$$

If we represent $\mathsf{n} = [\mathrm{n}_1, \mathrm{n}_2, \mathrm{n}_3]^T$ to be the unit vector corresponding to a unit surface normal at (x,y) to solve for the reflectance factor R, Equation 3.16 is transformed to

$$R\mathsf{n} = \mathsf{S}^{-1}\mathrm{I} \tag{3.19}$$

Due to the unit length of n, R is computed as

$$R = \|\mathsf{S}^{-1}\mathrm{I}\| \tag{3.20}$$

Given R, the unit surface normal vector n can be computed as

$$\mathbf{n} = \frac{1}{R}\mathsf{S}^{-1}\mathrm{I} \tag{3.21}$$

For each pixel, the reflectance factor R and the three components of the unit vector n are computed. The inverse of the light source matrix S only exists, if the positions of the lighting do not lie on a straight line [9].

Equation 3.2 can be written as

$$\mathbf{n} = [\mathrm{n}_1\ \mathrm{n}_2\ \mathrm{n}_3]^T\ =\ [\ p\quad q\ -1\ ]^T \tag{3.22}$$

Therefore,

$$p\ =\ \frac{\mathrm{n}_1}{\mathrm{n}_3} \tag{3.23}$$

$$q\ =\ \frac{\mathrm{n}_2}{\mathrm{n}_3} \tag{3.24}$$

3.5 Dealing with Specularity in Photometric Stereo

The use of a three-light source photometric stereo leads to poor results. Two causes were identified by [3]. These are:

1. The surface is not entirely Lambertian and can contain some specular component;
2. Some points might be in shadow for one or more of the images.

Specularity detection is important for obvious reasons. A lot of work has been done to tackle this problem. A brief review of two of these works is now presented.

In their work, Coleamn and Jain [10] proposed a method to detect specularity component from a four light-source stereo technique. As already pointed out only three light sources are sufficient to recover surface shape for Lambertian surface. But, if a point on the surface has orientation such that its specular spikes is in the same direction as one of the three light sources, the computed normal will be higher than the surface normal. A fourth light source is added to detect the existence of specularity by computing four surface normal vectors, one normal for each combination of three images.

They illustrated the difference between the calculated normals with and without specular contribution. At any given point (x, y) on the surface, if none of the intensities has a specular component, the resulting four surface normals will appear very close to each other. In this case the surface normal is the average of the computed four normals.

If specularity exists in one of the images, the intensity value from this image will elevate the resulting surface normal causing a deviation among the resulting four surface normals. The existence of specularity will cause a high deviation in both direction and magnitude of the vectors.

A thresholding procedure was then used to eliminate the specular effects. First, the relative deviation in the surface reflectance factor is computed from the four images intensity. Before computing the surface normal at point (x,y), the relative deviation is checked against a threshold value which is chosen to indicate a specular contribution. If relative deviation is greater than the largest amount of the reflectance deviation allowed, the surface normal is chosen from the combination of the three intensity values which has the smallest reflectance factor. In the other hand, if there is no specular contribution the surface normal is computed as the average of all four normals

They concluded that this method can apply only when specular regions are not overlapped and specular contribution does not appear in two or more images. Adjusting the angle of incidence would prevent the overlap of specular regions. Finally, they recommended that the phase angles between all source vectors and the view vector must not be so large as to prevent all four sources from contributing measurable intensity values from the specular reflection.

In their work Gendy and Shalaby [11] used also four-source photometric stereo. However, their method of computing the surface normals and reflectance factors avoided the inverse matrix method. Instead they used the method suggested by McGunnigle [12]. In this method the light sources have the same slant (σ) and tilt (τ) angles but their positions are increased by 90^0. For the definition of σ and τ see Fig.2.4. They chose positions with $\tau = 0^0, 90^0, 180^0$, and 270^0. With these angles they substituted into Equation 3.7 to yield:

$$i_0(\mathrm{x,y}) = i_o\rho(\mathrm{x,y})\frac{-\mathrm{p(x,y)}\sin(\sigma)+\cos(\sigma)}{\sqrt{\mathrm{p(x,y)}^2+\mathrm{q(x,y)}^2+1}} \tag{3.25}$$

$$i_{90}(\mathrm{x,y}) = i_o\rho(\mathrm{x,y})\frac{-\mathrm{q(x,y)}\sin(\sigma)+\cos(\sigma)}{\sqrt{\mathrm{p(x,y)}^2+\mathrm{q(x,y)}^2+1}} \tag{3.26}$$

$$i_{180}(\mathrm{x,y}) = i_o\rho(\mathrm{x,y})\frac{\mathrm{p(x,y)}\sin(\sigma)+\cos(\sigma)}{\sqrt{\mathrm{p(x,y)}^2+\mathrm{q(x,y)}^2+1}} \tag{3.27}$$

$$i_{270}(\mathrm{x,y}) = i_o\rho(\mathrm{x,y})\frac{\mathrm{q(x,y)}\sin(\sigma)+\cos(\sigma)}{\sqrt{\mathrm{p(x,y)}^2+\mathrm{q(x,y)}^2+1}} \tag{3.28}$$

where i_o is it the incident intensity, i_0, i_{90}, i_{180}, and i_{270} are image intensities at $\tau = 0^0, 90^0, 180^0$, and 280^0 respectively.

Using Equations $3.26 - 3.28$ and knowing that only three images are needed to fully determine p, q and albedo estimated intensities are computed for the four images. If there is no specularity contribution, the difference between any of the original and estimated image intensity is small. Otherwise the specular component will increase the original image intensity. They detected specular contributions by computing the differences between the original and estimated values and checking it against a determined threshold, which they concluded is about 20% of the average image intensity. When specularity is detected they excluded that pixel from their computation of $q(x, y)$, since their computation of $p(x, y)$ is independent of specular component

In the first part of this chapter it has been shown how images of a surface during photometric stereo can be acquired and how the image gradients can be estimated while

attempting to remove specularity in the acquired images. Using photometric stereo for quality surface inspection requires that a three-dimensional reconstruction of the surface be made to reveal the defects on the surface.

The reconstruction of images of objects can be done using moment sets which describe the images. This is as a results of a useful property of the Radon transform [39]. For clarification purpose a definition of this transform is given.

Consider a coordinate system shown in Fig 3.4 and taken from [40]. The function $g(s, \theta)$ is a projection of $f(x, y)$ on the axis s of θ direction. The function $g(s, \theta)$ is obtained by the integration along the line whose normal vector is in θ direction. The value $g(0, \theta)$ is so defined that it is obtained by the integration along the line passing the origin of (x, y)-coordinate. The Radon transform of a square integrable function $f(x, y)$, which is an image in this case, to the projection $g(s, \theta)$ can be expressed, therefore, as

$$g(s, \theta) \; = \; \iint_{-\infty}^{\infty} f(x, y)\delta(s\cos\theta \; + \; y\sin\theta \; - \; s)\mathrm{d}y\mathrm{d}x \tag{3.29}$$

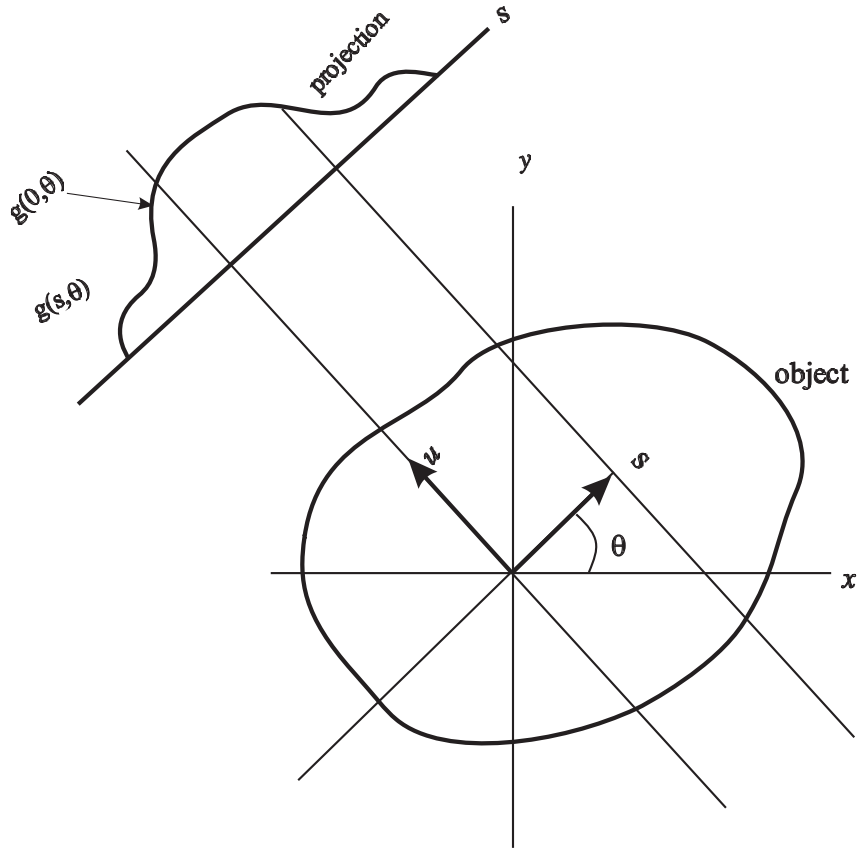


Figure 3.4: The Radon transform of a function$f(x, y)$ to a projection $g(s, \theta)$.

where $\delta(.)$ denotes the Dirac delta function. Equation 3.29 can be rewritten as

$$g(s,\theta) \;=\; \iint_{-\infty}^{\infty} f(x,y)\delta(s-\omega.[x,y]^T)\mathrm{d}y\mathrm{d}x \tag{3.30}$$

where $\omega = [\cos\theta, \sin\theta]$ [39]. Because of this property of the Radon transform an integrable function $F(s)$ sampled over the interval [-1,1] is expressed as

$$\int_{-1}^{1} g(s,\theta)F(s)ds \;=\; \iint_{-\infty}^{\infty} f(x,y)F(s-\omega.[x,y]^T)\mathrm{d}y\mathrm{d}x \tag{3.31}$$

It would be shown that moments of an image can be obtained in two different ways: either from polynomials or by the use of moments generating functions. The next section, Section 3.6, shows how moments can be generated from polynomials while Section 3.7 shows how moments can be obtained from a moment generating function. It is also shown in this chapter how particular moments can be used to describe particular geometric properties of an image. This chapter ends with an algorithm which was utilised for this thesis.

3.6 A Brief Introduction to Statistical Moments

Statistical moments are applicable to many different aspects of image processing. This field was pioneered by Hu [15] in the early 1960's. It ranges from invariant pattern recognition and image encoding to pose estimation. When applied to images they describe the image content (or distribution) with respect to its axes. They are designed to capture both global and detailed geometric information about the image. In this thesis they were used to characterise grey level images in order to be able to extract the flaws from the surface.

An images can be represented as a discrete function $f(x,y)$ with the sampled points defined on the nodes of a square lattice. These values can be embedded in a continuous function $g(x,y)$ defined on the square surrounding every pixel. Because of the Weierstrass theorem a uniform approximation of the image using polynomials can be made [41]. This theorem is stated below as

Theorem 3.1 *If f is a continuous complex valued function on [a, b], then there exists a sequence of polynomials $P_n(x)$ such that*

$$\lim_{n\longrightarrow\infty} P_n(x) \;=\; f(x) \tag{3.32}$$

uniformly on [a,b], where [a,b]$\in \mathbb{R}$. If f is real valued, the P_n's may be taken real.

The proof of this theorem can be found in [41]. Representing images as polynomials has advantages. These advantages were given by Eden *et al* in [13] and are cited here.

1. The correspondence between array values and polynomial coefficients allows the discrete set of values to be described by an analytical continuous function. This

function has a very simple expression.Operations such as interpolation, projection, and affine transformation take on a consistent interpretation and the numeric computations entailed by an operation may be simplified. A local continuous form of an image is used to extract edge or local texture properties. This may be so because the basis function may have properties which are passed unto the moments and thus producing descriptions which are invariant under rotation, scale, translation, and orientation [14].

2. Polynomials fit the geometric forms of images harmoniously. Slowly varying surfaces in images are well represented by polynomials

In continuous form an image can be considered as a two-dimensional Cartesian density distribution $f(x,y)f(x,y)$, two dimensional continuous function. Therefore, a moment of order $(p+q)$ evaluated over the entire image plane ξ can be represented generally as:

$$M_{pq} = \iint \psi_{pq}(x,y)f(x,y)\mathrm{d}x\mathrm{d}y\ ; \qquad p,q = 0,1,2,...,\infty \tag{3.33}$$

ψ_{pq} is known as the weighted kernel or the *basis function*. This produces a weighted description of $f(x,y)$ over the entire plane ξ.

Two theorems help to establish the fact that the moment M_{pq} does in fact exists and is unique over the function $f(x,y)$.

Theorem 3.2 ***The uniqueness theorem:***
Assuming that the intensity function $f(x,y)$ is piece-wise continuous and bounded in the region ζ , the moment sequence M_{pq} is uniquely determined by the intensity function $f(x,y)$ and conversely.

Theorem 3.3 ***Existence theorem:***
Assuming that the intensity function $f(x,y)$ is piece-wise continuous and bounded in the region ζ , the moment M_{pq} of all orders exist and are finite.

The proofs of these theorems can be found in [16] and [17] respectively.

If the moment has to be applied to digital images Equation 3.33 needs to be expressed in discrete form. Recall that that the probability density distribution of a continuous variable is different from that of the probability of a discrete distribution.

Let us assume that the whole image plane ξ is divided into square pixels of dimension 1×1, with constant intensity I over each square. Let P_{xy} also be a discrete pixel value, then

$$P_{xy} = I(x,y)\Delta A \tag{3.34}$$

where ΔA is the sampled or pixel area equal to one. Thus, analysing M_{pq} defined in Equation 3.33 over a complete discrete image intensity plane, we have:

$$M_{pq} = \sum_x \sum_y \psi_{pq}(x, y) P_{xy} \; ; \qquad p, q = 0, 1, 2, ..., \infty \tag{3.35}$$

where ψ_{pq} remains the basis function. The choice of basis function depends on the application and the desired invariant properties. The choice may introduce constraints including limiting the x and y range.

In sections 3.8 and 3.9 two types of moments are reviewed. These are orthogonal moments and non-orthogonal moments.

3.7 The Moment Generating Function

To describe the distribution of a random variable $f(x)$ the *characteristic function* given below can be used:

$$X(\omega) = \int_{-\infty}^{\infty} f(x) \mathrm{e}^{j\omega x} \mathrm{d}x = E[\mathrm{e}^{j\omega x}] \tag{3.36}$$

This function shown here is for the signal density $f(x)$, where $j = \sqrt{-1}$ and ω is the spatial frequency. This is essentially the Fourier transform of the signal and has a maximum at the origin $\omega = 0$, as $f(x) \geq 0$.

If $f(x)$ a one dimensional continuous function is the density of a positive, real valued variable x, such that $x \in \mathbb{R}$, then a continuous exponential distribution can be defined. If $j\omega$ in Equation 3.36 is replaced with s this produces a real valued integral of the form:

$$M^x(s) = \int_{-\infty}^{\infty} f(x) \mathrm{e}^{xs} \mathrm{d}x = E[\mathrm{e}^{xs}] \tag{3.37}$$

where $E[.]$ is the expectation and $M^x(s)$ exists as a real number. $M^x(s)$ is called the *moment generating function*, shown here for a one-dimensional distribution. Expressing the exponential in terms of an expanded Taylor series produces:

$$\mathrm{e}^{xs} = \sum_{n=0}^{\infty} \frac{x^n s^n}{n!} = 1 + xs + \frac{1}{2!} x^2 s^2 + \cdots + R_n(x) \tag{3.38}$$

where $R_n(x)$ is the error term. We see that this series will only converge and represent $x(s)$ completely if $R_n(x) = 0$. This means that if the distribution is finite in length, all values outside this length must be zero (or in terms of an image, all values outside the sampled image plane must be zero). If this is so, and Equation 3.38 is substituted into

Equation 3.37, this produces

$$\begin{aligned} M^x(s) &= \int_{-\infty}^{\infty} f(x)\mathrm{e}^{xs}\mathrm{d}x \\ &= \int_{-\infty}^{\infty} (1 + xs + \frac{1}{2!}x^2s^2\cdots)f(x)\mathrm{d}x \\ &= 1 + sm_1 + \frac{1}{2!}s^2m_2 + \cdots, \end{aligned} \tag{3.39}$$

where m_n is the n^{th} moment about the origin. Differentiating 3.37 n times with respect to s produces:

$$M_n^x(s) = E[x^n\mathrm{e}^{xs}] \tag{3.40}$$

If $M^x(s)$ is differentiable at zero, then the n^{th} order moments about the origin are given by

$$M_n^x(0) = E[x^n] = m_n \tag{3.41}$$

The first three moments of this distribution, therefore, are

$$\begin{aligned} M_0^x(s) &= E[\mathrm{e}^{xs}] \quad ; \quad M_0^x(0) = 1 \\ M_1^x(s) &= E[x\mathrm{e}^{xs}] \quad ; \quad M_1^x(0) = x \\ M_2^x(s) &= E[x^2\mathrm{e}^{xs}] \quad ; \quad M_2^x(0) = x^2 \end{aligned} \tag{3.42}$$

Finally, it is possible to evaluate the moment of a distribution by two methods. Either by using the direct integration method (i.e by integrating Equation 3.36, alternatively it is evaluated using the moment generation function just shown.

3.8 Orthogonal Polynomials and Moments

The common form of a polynomial is given as:

$$\mathrm{P}(x) = a_0 + a_1x + a_2x^2 + \cdots + a^nx^n. \tag{3.43}$$

However a more general definition of a polynomial of degree n as given by [23] is

$$\mathrm{P}(x) = \sum_{i=0}^{n} a_i\phi_i(x) \tag{3.44}$$

The quantity n is known as the degree of the polynomial. Orthogonal polynomials are defined as

Defintion 3.1 *Let $\lambda(t)$ be a nondecreasing function on the real line $\mathbb{R}$ having final limits as $t \to -\infty$ and $t \to +\infty$, and assume that the induced positive measure $d\lambda$ has finite moments of all orders* [24]*,*

$$\mu_r = \mu_r(d\lambda) := \int_{\mathbb{R}} t^r \, d\lambda(t), \quad r = 1, 2, 3, \ldots, \qquad \textit{with} \quad \mu_0 > 0 \tag{3.45}$$

Let $\mathbb{P}$ be the space of real polynomials and $\mathbb{P}_d \subset \mathbb{P}$ the space polynomials of degree $\leq d$. For any pair u, v in $\mathbb{P}$, one may define an inner product *as*

$$(u, v) = \int_{\mathbb{R}} u(t)v(t) \, d\lambda(t) \tag{3.46}$$

If $(u, v) = 0$, then u is said to be orthogonal *to v. If $u = v$, then*

$$\|u\| = \sqrt{(u, v)} = \left(\int_{\mathbb{R}} u^2(t) \, d\lambda(t) \right)^{1/2} \tag{3.47}$$

is called the norm *of u.*

Moments produced using orthogonal basis sets are called orthogonal moments, examples of which are the Legendre moments and the Zernike moments.

3.8.1 Legendre Moments

The kernel of Legendre moments are products of the Legendre polynomials defined along rectangular image coordinate axes inside a unit circle. The Legendre moments of order $(p + q)$ are defined as [17]

$$\mathrm{L}_{pq} = \frac{(2p + 1)(2q + 1)}{4} \int_{-1}^{1} \int_{-1}^{1} \mathrm{P}_p(x)\mathrm{P}_q(y) f(x, y) \mathrm{d}x\mathrm{d}y \tag{3.48}$$

where the functions $\mathrm{P}_n(x)$ denote the Legendre polynomial of order n [see 3.8.1.1] below.

The pth order term is given by the Rodrigues formula [25]

$$\mathrm{P}_p(x) = \frac{1}{2^P p!} \frac{\mathrm{d}^p}{\mathrm{d}x^p} (x^2 - 1)^p, \qquad x \in [-1, 1] \, . \tag{3.49}$$

In order to evaluate the Legendre moments, the image coordinate space has to be scaled so that their respective magnitudes are less than 1. If the image dimension along each image coordinate axis is N pixels, and i, j denote the pixel coordinate indices along the axes, $0 \leq i, j \leq N$, then the discrete Legendre moments can be written as

$$\mathrm{L}_{pq} = \frac{(2p + 1)(2q + 1)}{(N - 1)^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \mathrm{P}_p(x_i)\mathrm{P}_q(y_j) f(i, j) \tag{3.50}$$

where x_i, y_j denote the normalised pixel coordinates in the range [-1,1], given by

$$x_i = (2i/N) - 1; \qquad y_j = (2j/N) - 1 \tag{3.51}$$

3.8.1.1 Legendre Polynomials

The Legendre polynomial $P_n(x)$ of order n is defined as

$$\mathrm{P}_n(x) = \sum_{k=0}^{n} (-1)^{(n-k)/2} \frac{1}{2^n} \frac{(n+k)! x^k}{\left(\frac{n-k}{2}\right)! \left(\frac{n+k}{2}\right)! k!} \quad , \qquad \mid x \mid \leq 1, \text{ and } (n-k) \text{ is even.} \tag{3.52}$$

The above series expansion of Legendre polynomials can be obtained using Equation 3.49. The Legendre polynomials form a complete orthogonal set inside the circle, hence

$$\int_{-1}^{1} \mathrm{P}_m(x)\mathrm{P}_n(x)\mathrm{d}x = \frac{2}{(2m+1)}\delta_{mn} \tag{3.53}$$

where δ_{mn} denotes the *Kronecker delta*.

The equation below provides a recursive relation in Legendre polynomials:

$$\mathrm{P}_n(x) = \frac{(2n-1)xP_{n-1}(x) - (n-1)P_{n-2}(x)}{n} \tag{3.54}$$

where
$\mathrm{P}_0(x) = 1; \qquad \mathrm{P}_1(x) = x; \qquad \mid x \mid \leq 1; \qquad \text{and,} \quad n > 1.$
The integral formula for Legendre polynomials is given by

$$\int \mathrm{P}_n(x)\mathrm{d}x = \frac{x\mathrm{P}_n(x) - \mathrm{P}_{n-1}(x)}{n+1} \tag{3.55}$$

The polynomial expressions for $\mathrm{P}_n(x)$ up to the fourth order are given below:

$$\begin{aligned} \mathrm{P}_0(x) &= 1 \\ \mathrm{P}_1(x) &= x \\ \mathrm{P}_2(x) &= (3x^2 - 1)/2 \\ \mathrm{P}_3(x) &= (5x^3 - 3)/2 \\ \mathrm{P}_4(x) &= (35x^4 - 30x^2 + 3)/8 \end{aligned} \tag{3.56}$$

3.9 Non-orthogonal Moments

Non-orthogonal moments are moments which do not satisfy the condition for orthogonality given in Equation 3.46. Examples are *geometric moments*, *Cartesian moments* and *regular moments*. These sets of moments are the simplest amongst moment functions, the kernel functions are defined as a product of the pixel coordinates [17]. Only a brief review of how they can be obtain is given here. There exist a number of works on how these moments can be used for surface reconstruction. Examples can be found in [14, 25, 27].

Geometric moments can be defined with the basis set $\{x^p y^q\}$. The $(p+q)^{th}$ order two-dimensional geometric moments are denoted by m_{pq}, and can be expressed as

$$m_{pq} = \iint_{\zeta} x^p y^q f(x,y) \mathrm{d}x\mathrm{d}y \qquad p,q = 1,,2,3,\ldots \tag{3.57}$$

where ζ is the region of the pixel space in which the image intensity function $f(x,y)$ is defined. Equation 3.57 has the form of the projection of the equation of the function $f(x,y)$ onto the monomial $x^p y^q$. The basis set $\{x^p y^q\}$, while complete, is not orthogonal. It is assumed that $f(x,y)$ is a piecewise continuous, bounded function and that it can have non-zero values only in the finite region of the $x-y$ plane.

Analysing a two-dimensional irradiance distribution $f(x,y)$

$$M^{xy}(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\,(ux+vy) f(x,y)\mathrm{d}x\mathrm{d}y \tag{3.58}$$

and expanding the exponential using Taylor series produces

$$M^{xy}(u,v) \;=\; \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} \frac{u^p}{p!}\frac{v^q}{q!} m_{pq} \tag{3.59}$$

where m_{pq} are the moments of the two dimensional distribution.

The discrete version of the Cartesian moment defined in Equation 3.57 for an image consisting of pixels P_{xy} is obtained by replacing the integrals in Equation 3.57 with summations, as

$$m_{pq} \;=\; \sum_{x=1}^{M}\sum_{y=1}^{N} x^p y^q P_{xy} \tag{3.60}$$

where m_{pq} is the two-dimensional Cartesian moment, $M \mathrm{and} N$ are the image dimensions and the monomial product $x^p y^q$ is the basis function.

3.10 Shape Representation Using Moments

Describing shapes from given moments falls under what is known as the *moment problem*, see [26] and [27]. It has also been called the *inverse moment*, or *Hausdorf's moment problem* see [25]. The inverse moment problem, simply put, states that if given a sequence of numbers $\{\mu_{n=0}^{\infty}\}$, under what conditions is it possible to determine a function $a(t)$ of bounded variation in the interval [0 1] such that

$$\mu_n \;=\; \int_0^1 t^n \mathrm{d}a(t) \qquad for \quad n \;=\; 1,2,3,\cdots \qquad [22] \tag{3.61}$$

Applying this to image reconstruction, it simply states that: "if only a finite set of moments of an image is given, how well can we reconstruct the image?" [25].

Geometric moments of different order represent different spatial characteristics of the image intensity distribution. Thus, a set of moments can form a global shape descriptor of the image. The representation given below is first shown for geometric moments and then followed by the Legendre moments.

3.10.1 Shape representation using geometric moments

By definition the moment of order zero (m_{00}) represents the total intensity of the image. For a binary image this term gives the geometrical area of the image region. The first two order moments m_{10} and m_{01} provide the intensity moments about the $y-$axis and $x-$axis of the image respectively, and are used to find the Centre of Mass (COM) of the image. Accordingly, the centre coordinates, or *intensity centroid*, $(\overline{x}\ ,\ \overline{y})$ is given by

$$\bar{x} \ = \ m_{10}/m_{00}; \qquad \bar{y} \ = \ m_{01}/m_{00} \tag{3.62}$$

It is sometimes convenient to evaluate the moments with the origin of the reference system shifted to the intensity centroid of the image [17]. This transformation makes the moment computation independent of the position of the image reference system. The moments computed with respect to intensity centroid are called *central moments*, are are defined as

$$\mu_{pq} \ = \ \iint_{\zeta} (x-\bar{x})^p (y-\bar{y})^q f(x,y) \mathrm{d}x\mathrm{d}y \qquad p,q \ = \ 1,2,3,\cdots \tag{3.63}$$

From the definition of central moments we have that

$$\mu_{00} \ = \ m_{00}; \qquad \mu_{10} \ = \ \mu_{01} \ = \ 0 \tag{3.64}$$

The method commonly used for image reconstruction using geometric moments is called *moments matching* and it is covered in many literatures [14],[17],[25]and[27]. For more shape description moments and moment sets which are invariant with respect to image plane transformation see [17] and [15].

3.10.2 Shape representation using Legendre moments

The Legendre moments of order $(p+q)$ in Equation 3.48 can be approximated by discrete summations and in terms of geometric moments can be expressed as follows:

$$\mathrm{L}_{pq} = \frac{(2p+1)(2q+1)}{4} \sum_{i=0}^{p} \sum_{j=0}^{q} a_{pi} a_{qi} m_{ij} \tag{3.65}$$

where a_{pi} denotes the coefficient of x^i in the series expansion of $P_p(x)$ as given in Equation 3.48. Assigning particular values for p, q in Equation 3.65, we get the following relations

in Legendre moments up to the second order.

$$\begin{aligned} \mathrm{L}_{00} &= m_{00}, \\ \mathrm{L}_{10} &= (3/4)m_{10}, \\ \mathrm{L}_{01} &= (3/4)m_{01,} \\ \mathrm{L}_{20} &= (5/4)[(3/2)m_{20} - (1/2)m_{00}], \\ \mathrm{L}_{02} &= (5/4)[(3/2)m_{02} - (1/2)m_{00}], \end{aligned} \tag{3.66}$$

The orthogonality property of the Legendre polynomial helps in expressing the image intensity function $\mathrm{f}(x, y)$ in terms of its Legendre moments using the Fourier expansion theorem on orthogonal functions. This is expressed mathematically as:

$$f(x, y) \;=\; \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} \mathrm{L}_{ij}\mathrm{P}_i(x)\mathrm{P}_j(y), \qquad |x|,|y| \leq 1 \tag{3.67}$$

Equation 3.67 is called the inverse Legendre moments transform. For a finite number n of Legendre moments, an approximated version $f'(x, y)$ of the intensity function can be reconstructed as follows:

$$f'(x, y) \;=\; \sum_{i=0}^{n}\sum_{j=0}^{i} \mathrm{L}_{i-jj}\mathrm{P}_{i-j}(x)\mathrm{P}_j(y), \qquad |x|, |y| \leq 1 \tag{3.68}$$

The analysis done so far is restricted to using the Legendre moments as continuous orthogonal moments and applied to a continuous set of points. However these moments can be utilised on a finite set of points to generate shape descriptor moments. Legendre moments suitable for such computations are known as discrete Legendre moments.

Consider an image model represented as $n \times n$ pixel array, without loss of generality, on the set $\Omega = \{(i, j), \quad 0 \leq i, j \leq n\}$. This means that we have the digital image $\{f(x, y); \;\; (x, y) \in \Omega\}$.

The orthogonal system on Ω is defined as $\{P_p(x)P_q(y), \;\; 0 \leq p, q \leq n\}$ for $(x, y) \in \Omega$, where the orthogonality of $\{P_p(x), \; 0 \leq p \leq n\}$ is defined as follows

$$\sum_{x=0}^{n} P_p(x)P_q(x) \;=\; C_p(n)\delta_{pq} \tag{3.69}$$

Here $C_p(n) \;= \sum_{x=0}^{n} P_p^2(x)$ is the *normalising constant* [27].

From the above, we can represent the digital image $\{f(x, y), \; 0 \leq x, y \leq n\}$ by the following expansion after first employing $(T + 1)^2$ discrete moments

$$f_T(x, y) \;=\; \sum_{p=0}^{T}\sum_{q=0}^{T} \tau_{pq}\lambda_{pq}P_p(x)P_q(y), \qquad (x, y) \in \Omega \tag{3.70}$$

where $\tau_{pq} = (C_p(n)C_q(n))^{-1}$ is the normalising constant and

$$\lambda_{pq} = \sum_{i=0}^{n}\sum_{j=0}^{n} f(i,j)P_p(i)P_q(j) \tag{3.71}$$

defines the (p,q) order discrete moment with respect to the basis $\{P_p(x)P_q(y),\ 0 \leq p,q \leq n\}$.

An example of $\{P_p(x); 1 \leq p \leq n\}$ is a discrete analog of Legendre orthogonal polynomials given by [27] as

$$P_p(x) = \sum_{s=0}^{p} \frac{(-1)^s \binom{p}{s}\binom{p+s}{s}}{n(n-1)\cdots(n-s+1)} x^{[s]} \tag{3.72}$$

for $0 \leq p \leq n$ and $0 \leq x \leq n$, where $x^{[s]} = x(x-1)\cdots(x-s+1)$ with $x^{[0]} = 1$. In particular, $P_0(x) = 1$, $P_1(x) = 1 - 2x/n$, $P_2(x) = 1 - 6x/n + 6x(x-1)/(n(n-1))$.

Different methods have been used to generate discrete Legendre polynomial as basis for obtaining moments used for image reconstruction. These method are numerically involved and the computation takes time. Some of these methods are given in [29] and [30]. In the next section a new surface approximation algorithm is presented, which is numerically more efficient than previous solutions.

3.11 New surface Approximation method

This method is based on using polynomials to describe the surface and estimating the polynomials coefficients. In computing for the coefficients the reconstruction is reduced to least square approximation. The obtained coefficients are back-substituted into the polynomial equations to give the required surface.

3.11.1 Representing Images as Polynomials

Images are in discrete forms but can be approximate as continuous functions, see [13]. It has also be shown that these function can be represented as a polynomial. It is now shown how this polynomial can be derived. To do this the interpolation theory in one dimension is stated below

Theorem 3.4 . *If given a sequence of n distinct numbers x_k (called* **nodes***), and for each x_k a second number y_k, there exist a function f so that*

$$f(x_k) = y_k\ ; \quad k = 1,2,3,\ldots,n \ \text{ and } y_k \neq y_j \tag{3.73}$$

where the pair (x_k, y_k) is called a data point and the function f is called an **interpolant** *for the data points.*

Equation 3.73 can be written as

$$g(y_k) \;=\; \sum_{j=1}^{n} a_j y_k^{j-1} \tag{3.74}$$

where $k \;=\; 1, 2, \ldots, n$ and a_j are coefficients. If the function f in Equation 3.73 is replaced by a polynomial $p(x)$ then the equation can be written as

$$P(x) \;=\; \sum_{j=1}^{n} a_j x^{j-1} \tag{3.75}$$

which is unique and satisfy the n equations. Note that this equation is similar to that given in Equation 3.44 for the equation of a polynomial.

If g in equation 3.74 is defined as

$$g \;=\; [g(y_1) \ldots g(y(n)]^T \tag{3.76}$$

as the vector of the function values at the points of interest and

$$a \;=\; [a_1 \ldots a_n]^T \tag{3.77}$$

the vector of the polynomial coefficient values, therefore the system of equations represented in Equation 3.75 can be rewritten as

$$g \;=\; \mathsf{P}_{\mathrm{V}}(y_1, \ldots, y_n).a \tag{3.78}$$

where Pv is the $N \times N$ matrix defined by

$$\mathsf{P}_{\mathrm{V}} \;\triangleq\; \begin{bmatrix} 1 & y_1^1 & y_1^2 & \cdots & y_1^n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & y_n^1 & y_n^2 & \cdots & y_n^n \end{bmatrix} \tag{3.79}$$

where the matrix in Equation 3.79 is known as the *Vandermonde matrix*. This derivation can now be generalized.

Any polynomial basis $\mathsf{P}(x)$ which are sums of monomials can be defined by post-multiplying the Vandermonde matrix P_V by an upper triangular matrix A [32], i.e.

$$\mathsf{P}_A \;\triangleq\; \mathsf{P}_{\mathrm{V}}\mathsf{A} \tag{3.80}$$

whereby the Vandermonde matrix of degree d_x for n_x points in x is defined as,

$$\mathsf{P}_{\mathrm{V}} \;\triangleq\; \begin{bmatrix} 1 & x_1^1 & \cdots & x_1^{d_x} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_{n_x}^1 & \cdots & x_{n_x}^{d_x} \end{bmatrix} \tag{3.81}$$

Note: It has not been implied that this polynomial basis is orthogonal.

3.11.2 The New Method

Surfaces being inspected have coordinates in x and y directions with elevation in the z direction, the surface height. The axes are well aligned at experimentation time with the camera axis, therefore a suitable model for the smooth surface is provided by a two-dimensional tensor polynomial regression [31].

Let us consider a given point $p(i,j)$ which is on an invariant cartesian grid $\mathsf{G}(i,j)$ and whose size is denoted as $n_x \times n_y$ with its corresponding elevation $z(i,j)$. A bivariate polynomial for $z(i,j)$ of degree d_x in x and d_y in y can be formulated as a sum of monomials, compare Equation 3.35, that is

$$z(i,j) \;=\; \sum_{m=0}^{d_x}\sum_{n=0}^{d_y} \mathsf{C}(m,n) x_i^m y_j^n, \tag{3.82}$$

where C is the matrix of coefficients. Equation 3.82 can also be written as a tensor product of a vector in x and y; hence the name *bivariate tensor polynomial* [31],

$$z(i,j) \;=\; \begin{bmatrix} y_i^0 & \cdots & y_i^{d_y} \end{bmatrix} \mathsf{C} \begin{bmatrix} x_j^0 & \cdots & x_j^{d_x} \end{bmatrix}^T \quad . \tag{3.83}$$

The complete polynomial surface can now be written as

$$\mathsf{Z} \;=\; \mathsf{Y_V C X}_V^T \tag{3.84}$$

Equation 3.84 is the starting point for the polynomial regression. The computation is stated as follows: If given Z, X_V and Y_V we sought the values of C, which minimizes the cost function $\varepsilon = \mid \mathsf{Z} - \mathsf{X}_v \mathsf{C} \mathsf{Y}_V^T \mid^2$. Therefore, the reconstruction method has been reduced to a least square approximation method.

Let us denote the partial derivatives of the surface in x and y directions as

$$\mathrm{I}_x \;=\; \mathsf{Y}_V \mathsf{C} \frac{\mathrm{d}}{\mathrm{d}x} \mathsf{X}_V^T \tag{3.85}$$

$$\mathrm{I}_y \;=\; \frac{\mathrm{d}}{\mathrm{d}y} \mathsf{Y}_V \mathsf{C X}_V^T \tag{3.86}$$

The subscript would be dropped from here onwards. From Equations 3.85 and 3.86 the cost function is given as

$$\varepsilon \;=\; \|\mathrm{I}_x \;-\; \mathsf{YC}\frac{\mathrm{d}}{\mathrm{d}x}\mathsf{X}^T\|_F^2 \;+\; \|\mathrm{I}_y \;-\; \frac{\mathrm{d}}{\mathrm{d}y}\mathsf{YCX}^T\|_F^2 \tag{3.87}$$

where $\|.\|_F$ is the *Fronenius norm.*

Since $\|\mathsf{A}\|_F^2 \;=\; \mathrm{trace}(\mathsf{AA}^T)$, Equation 3.87 becomes

$$\varepsilon \;=\; \mathrm{trace}\{(\mathrm{I}_x - \mathsf{YC}\frac{\mathrm{d}}{\mathrm{d}x}\mathsf{X}^T)(\mathrm{I}_x - \mathsf{YC}\frac{\mathrm{d}}{\mathrm{d}x}\mathsf{X}^T)^T\} + \mathrm{trace}\{(\mathrm{I}_y - \frac{\mathrm{d}}{\mathrm{d}y}\mathsf{YCX}^T)(\mathrm{I}_y - \frac{\mathrm{d}}{\mathrm{d}y}\mathsf{YCX}^T)^T\} \tag{3.88}$$

To solve Equation 3.88, it is expanded and differentiate with respect to C because this would minimize the cost function and the result set equal to zero, i.e.

$$\mathsf{Y}^T\mathsf{Y}\mathsf{C}\frac{\mathrm{d}}{\mathrm{d}x}\mathsf{X}^T\frac{\mathrm{d}}{\mathrm{d}x}\mathsf{X} \;-\; \mathsf{Y}^T\mathrm{I}_x\frac{\mathrm{d}}{\mathrm{d}x}\mathsf{X} \;+\; \frac{\mathrm{d}}{\mathrm{d}y}\mathsf{Y}^T\frac{\mathrm{d}}{\mathrm{d}y}\mathsf{Y}\mathsf{C}\mathsf{X}^T\mathsf{X} \;-\; \frac{\mathrm{d}}{\mathrm{d}y}\mathsf{Y}^T\mathrm{I}_y\mathsf{X} \;=\; \underset{\sim}{\mathbf{O}} \tag{3.89}$$

If the basis function are carefully chosen to be orthonormal[1], then Equation 3.89 simplifies to :

$$\frac{\mathrm{d}}{\mathrm{d}y}\mathsf{Y}^T\frac{\mathrm{d}}{\mathrm{d}y}\mathsf{Y}\mathsf{C} \;+\; \mathsf{C}\frac{\mathrm{d}}{\mathrm{d}x}\mathsf{X}^T\frac{\mathrm{d}}{\mathrm{d}x}\mathsf{X} \;-\; \mathsf{Y}^T\mathrm{I}_x\frac{\mathrm{d}}{\mathrm{d}x}\mathsf{X} \;-\; \frac{\mathrm{d}}{\mathrm{d}y}\mathsf{Y}^T\mathrm{I}_y\mathsf{X} \;=\underset{\sim}{\mathbf{O}} \tag{3.90}$$

Equation 3.90 is a *Lyapunov* equation of the form;

$$\mathsf{A}_1\mathsf{C} \;+\; \mathsf{C}\mathsf{A}_2 \;+\; \mathsf{A}_3 \;=\; 0 \;. \tag{3.91}$$

from which C is readily obtained.
The general Lyapunov equation is of the form

$$A^TP \;+\; PA \;+\; Q \;=\; 0 \tag{3.92}$$

where $A, T, P, Q \in \mathbb{R}^{n\times n}$ with P and Q being symmetric. This equation is commonly encountered in countrol engineering and the solution can be found in [42].

Care has to be taken in solving for C in Equation 3.91. The first column of the derivatives $\frac{\mathrm{d}}{\mathrm{d}x}\mathsf{X}$ and $\frac{\mathrm{d}}{\mathrm{d}y}\mathsf{Y}$ are zero (because they are derivatives of constants). The matrices can be partitioned as:

$$\frac{\mathrm{d}}{\mathrm{d}y}\mathsf{Y}^T\frac{\mathrm{d}}{\mathrm{d}y}\mathsf{Y} \;=\; \begin{bmatrix} \mathbf{0} & \vec{\mathbf{0}}^T \\ \vec{\mathbf{0}} & \vec{\mathrm{A}}_1 \end{bmatrix} \tag{3.93}$$

The value of C is back-substituted into equation 3.84 to reconstruct the surface.

This algorithm was used to reconstruct surfaces of objects whose images have been captured using a CCD camera and the gradients obtained using photometric stereo. The results of these computation are shown in the next chapter.

[1] An orthogonal matrix A has the property $\mathsf{A}^T\mathsf{A} \;=\; \mathsf{W}$, where W is a diagonal matrix containing the norms of the columns. The condition number of the matrix is defined as $\kappa_2(\mathsf{P}) \;=\; \max\{\mathsf{W}\}/\min\{\mathsf{W}\}$. An orthonormal matrix is one which has the property that $\mathsf{A}^T\mathsf{A} \;=\; \mathsf{I}$, its condition number is 1. Such matrices are also known as *unary* matrices

Chapter 4

Experimental Set-up, Results and Discussion

4.1 Construction of the Experimental Apparatus

An experimental apparatus was constructed for the acquisition of image under controlled lighting conditions. The following hardware were utilised for this experiment: a camera; four white light sources; a rigid frame to support the camera and the light sources; and of course text samples.

The figure shown in Fig 4.1 is a schematic diagram of the experimental set-up. For simplicity purpose only two of the light source are shown and a fixed distance of 0.5m for from the camera centre shown. An overview of the various components used is now given.

4.1.1 Camera specification

The camera used is the PULNiX®–TM– 6CN with view lenses of 35mm 50mm. The PULNiX® TM–TM– 6CN is a VGA format, high-speed monochrome camera. It's field of view covers 752×582 pixels. The camea is shown in Fig 4.2.

4.1.2 Lighting specification and configuration

Although it might not be obvious, the success of the experiment depends greatly on the lighting and its configuration with respect to the other components. Therefore, care must be taken during material selection, construction of the rigid structure and image acquisition in order to have optimal results. To simplify the computation the light sources were at equal horizontal distance from the camera at each x and y directions.

The lighting used was a Luxeon® 6-ring white LED shown in Fig 4.3 with typical luminous flux of 150Φv. The choice of using LED is due to the experimental requirement for a pointed source of light which is constant for the duration of the experiment.

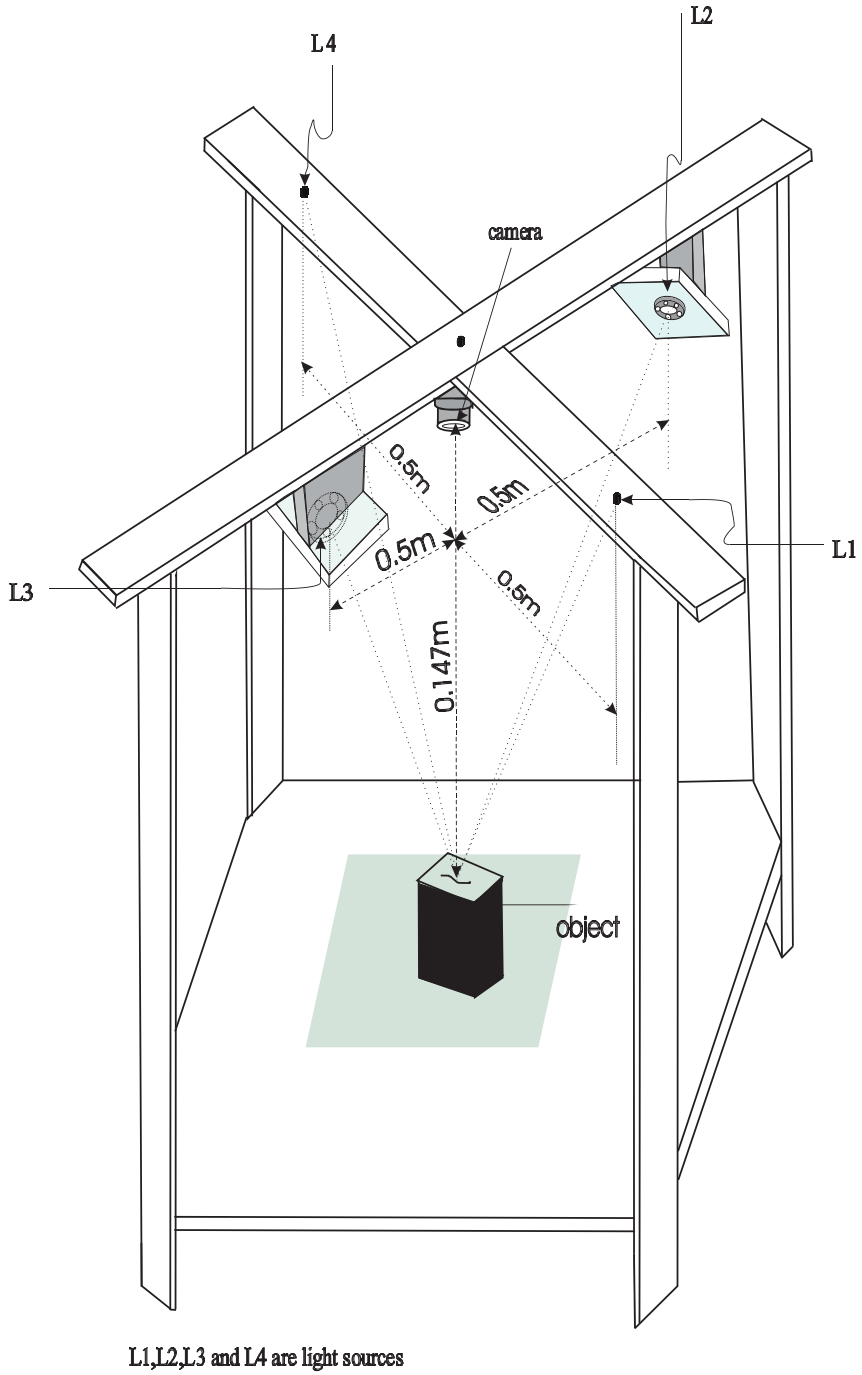


Figure 4.1: A schematic diagram of the experimental set-up.

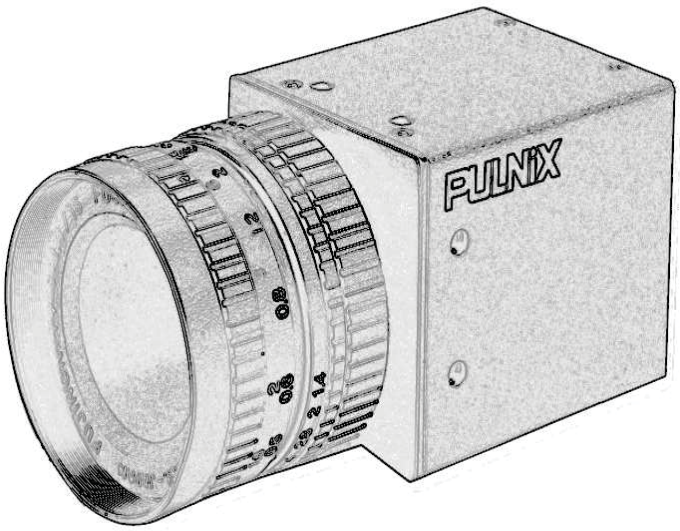


Figure 4.2: PUNiX TM – 6CN.

Figure 4.3: Luxeon® 6-ring LED (white)

4.1.3 Experimental procedure

A figure of the actual structure used for this experiment is shown in Fig 4.4. The object coordinate was chosen in a way that the plane of the camera lens is the (x, y) axis, this coordinate coincides with that of the test sample also. The positive z-axis is vertically upwards. The camera was positioned vertically above the test sample and the camera height maintained for each session of image capture, that is during each image/lighting confiuration.

The camera image plane was place at a considerable distance from the test object. This was to achieve the requirement of orthogonality between the object plane and the image plane. To reduce the possibility of ambient light affecting the result, care was taken to exclude this from the experiment.

The experiment was performed in two ways. First images were taken with the light sources positioned along the x and y axes and secondly when the light source were positioned in between these axes.

4.2 Surface Reconstruction

For the reconstruction of the surface a new method was developed and tested. The algorithm is given in chapter three. It is based on using discrete polynomial moments to reconstruct the surface. This method appeared better than any existing solution for surface approximation. Because of the problem of instability of discrete polynomials at higher degrees previous methods had to reconstruct surfaces in little patches. This problem appears to have been solved in [32]. Based on this result the algorithm shown in chapter three was developed.

4.3 Specular Pixels in Test Samples

In dealing with specular pixels three options were considered. The first is the case where no pixel is encountered. In this case the the pseudo-inverse is used to compute the surface normals by the use of Equation 3.24.

In the situation where a pixel in only one of the images is saturated, that pixel is excluded from the computation of the surface normal vector and the remaining three images are used, which is the normal situation, thus enabling complete reconstruction.

The third case is a situation where a pixel is found saturated in more than one of the images. In this situation that particular pixel surface normal is assumed to be vertical.

Figure 4.4: Picture of the experimental set-up

4.4 Presentation of Test result

The software used for this experiment was developed in Matab®. Various samples were tested using this software and the results are now presented.

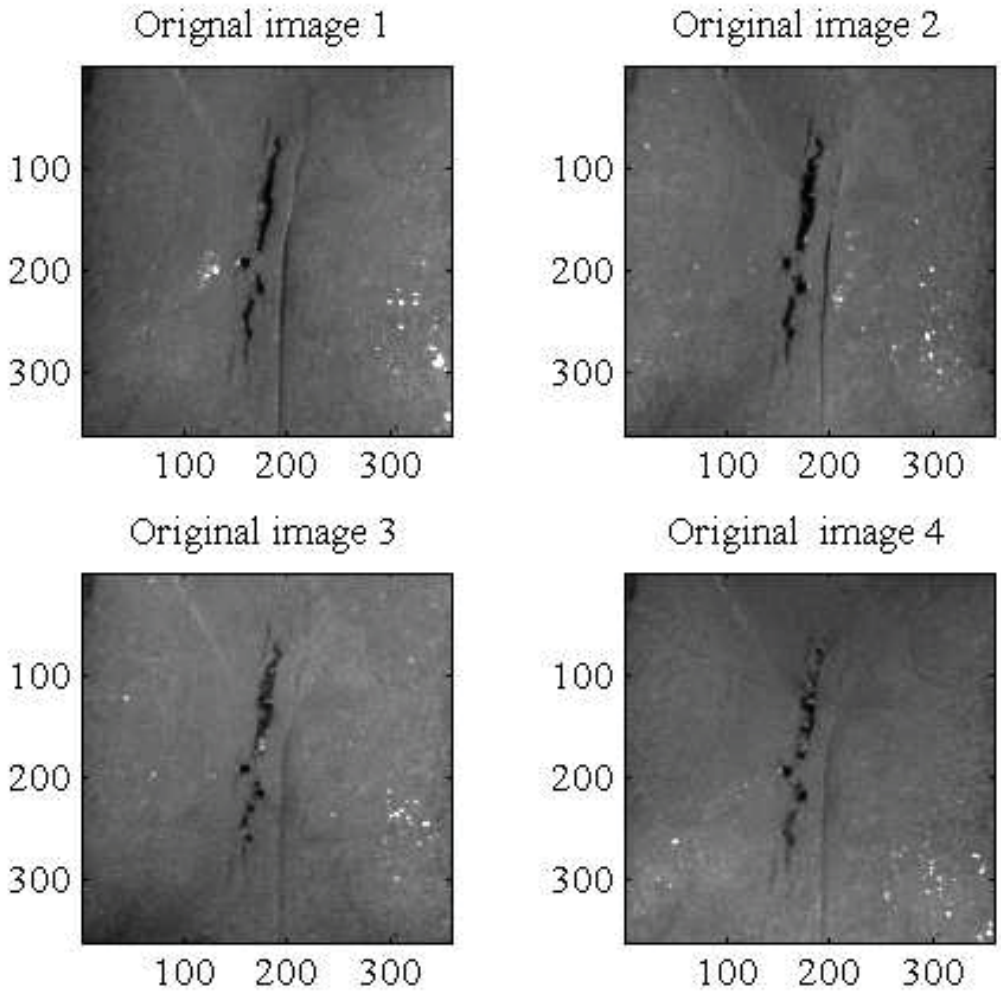


Figure 4.5: Original images

Fig 4.5 shows four acquired images of a steel block taken with the light positions varying as denoted by positions 1,2,3 and 4 as shown in Fig 4.1. Specular pixels in the images are easily identified.

Figure 4.6 shows the gradients of the image obtained using the four–light source photometric stereo and the pseudo–inverse of the light source vector. Notice how specularity in the images have been removed.

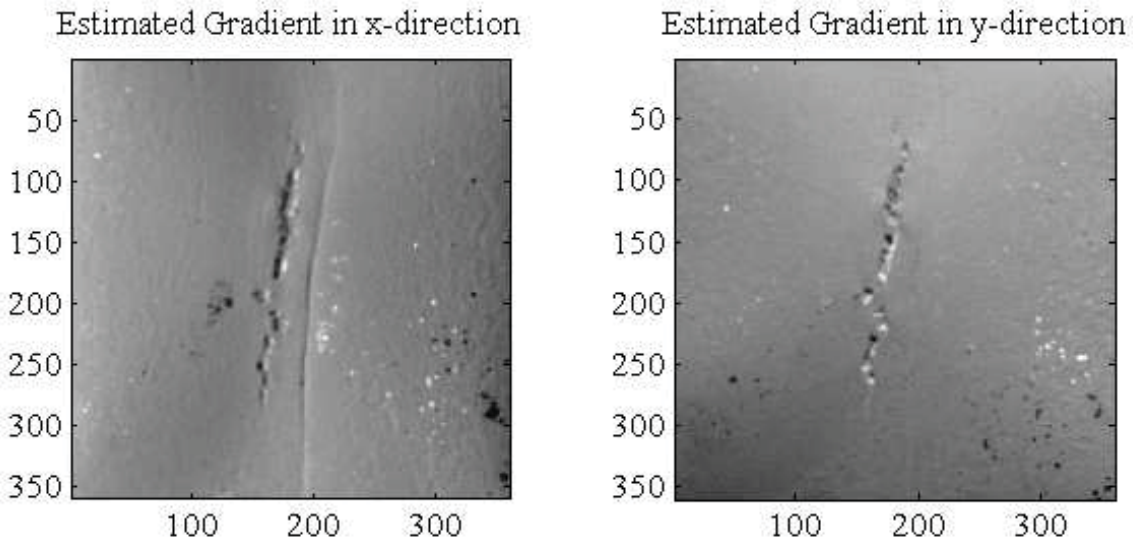


Figure 4.6: Obtained image Gradient using Photometric Stereo

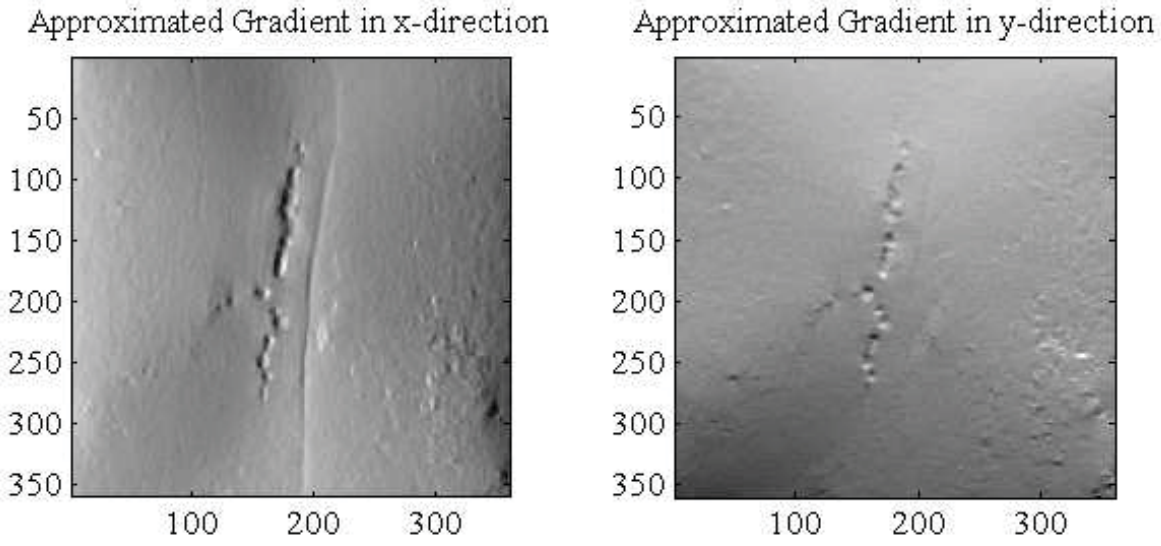


Figure 4.7: Reconstructed Gradients

Fig 4.7 shows the gradients of the image reconstructed using the new method. Compare these with those shown in Fig 4.6 to check the effectiveness of this method.

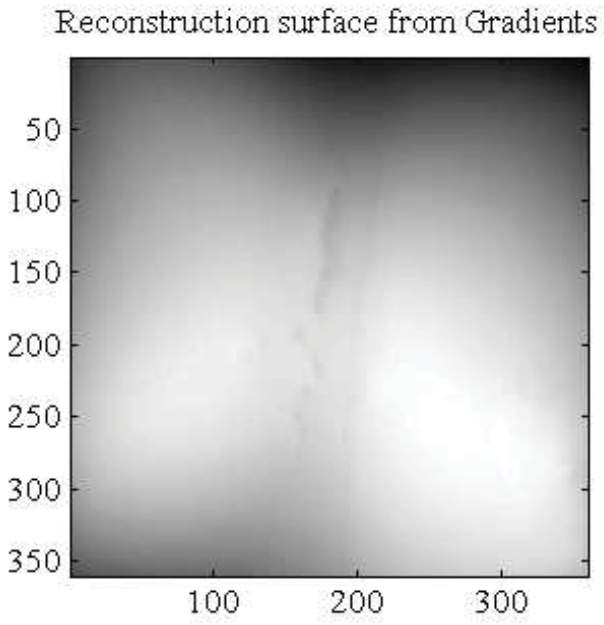


Figure 4.8: Reconstruction of Original Surface

Fig 4.8 shows the reconstructed image from discrete polynomial moments generated with the method discussed in section 3.11. The object axis was chosen in such a way that the vertical direction upwards was the z-direction. Therefore, points on the surface whose elevation are nearer the camera appear brighter than those farther.

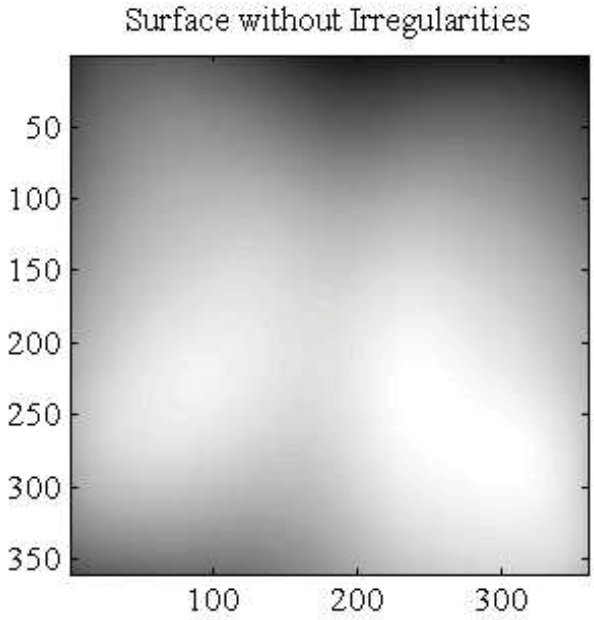


Figure 4.9: Reconstructed Surface without Defect

Figure 4.9 shows what the surface would have looked like without flaws.

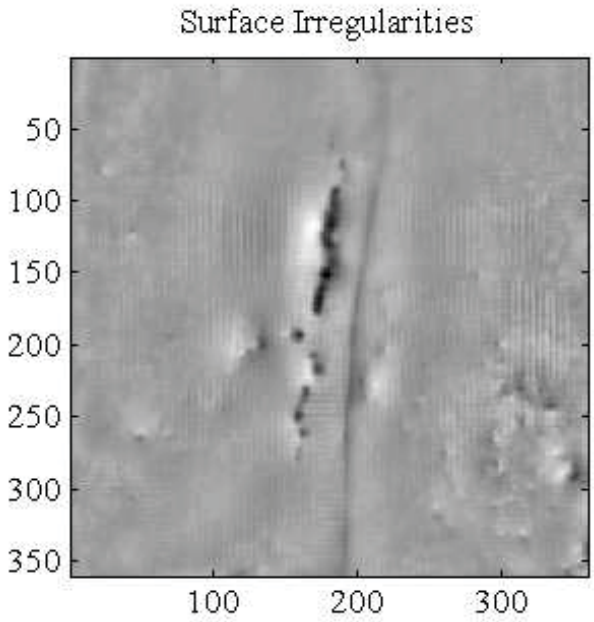


Figure 4.10: Extracted Surface Irregularities

Fig 4.11 shows the extracted flaws as surface heights. Note the heights in blue. These are

flaws, which are below the surface, that is into the material.

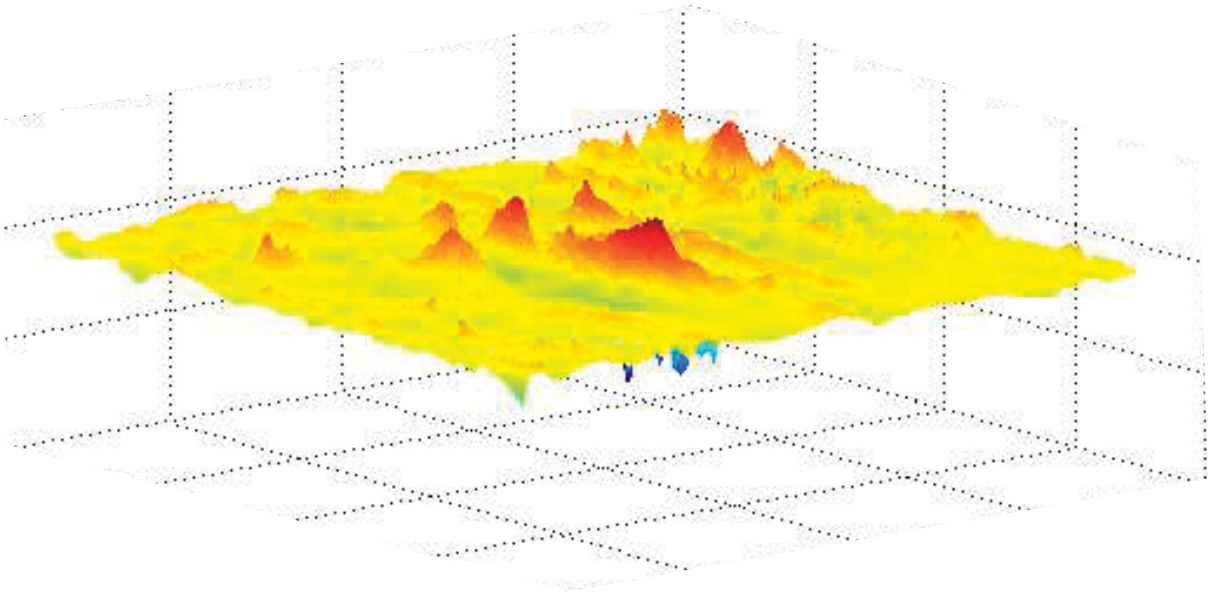


Figure 4.11: Extracted Surface Heights

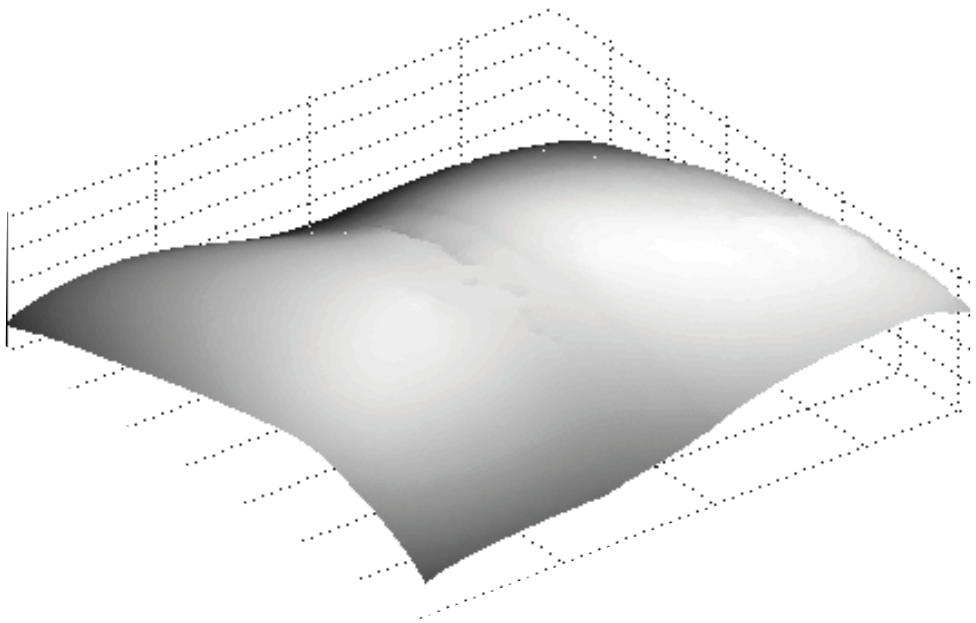


Figure 4.12: 3–D reconstructed Surface from moments

Fig 4.12 shows the reconstructed surface in three-dimension, while Fig 4.13 is a contour trace of the reconstructed surface in three-dimension

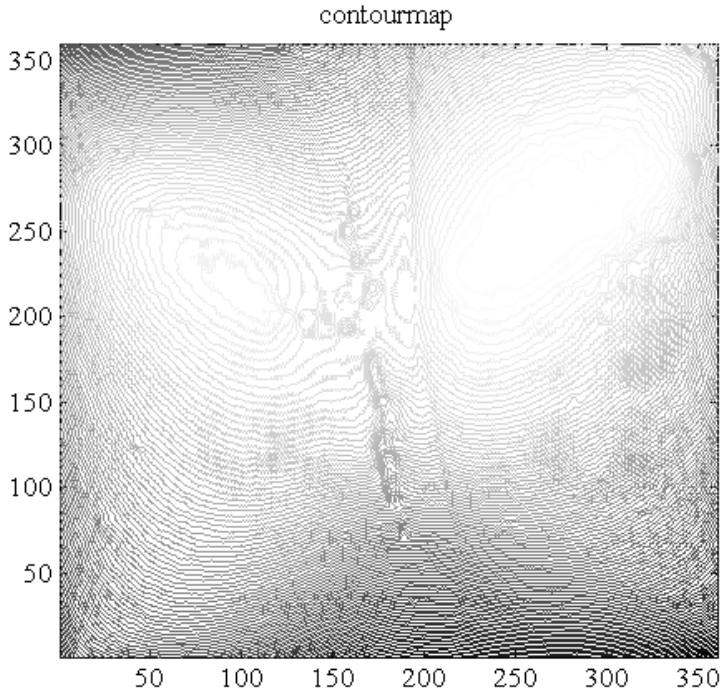


Figure 4.13: Contour trace of the reconstructed surface surface

4.5 More Results

Judging from the success of this method in its use to inspect metallic surface for flaws, attempt was made to see how effective it can be used for reconstructing metallic surfaces with engraved code. The obtained results are now presented.

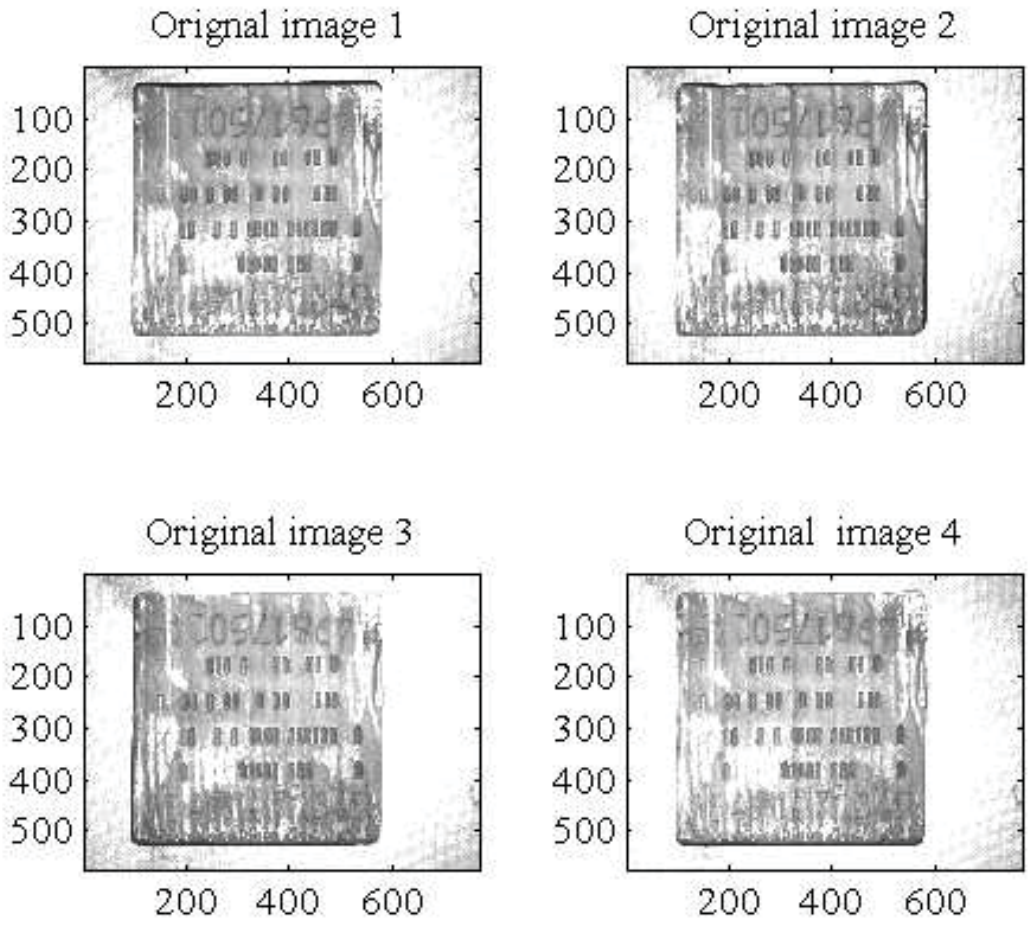


Figure 4.14: Original images

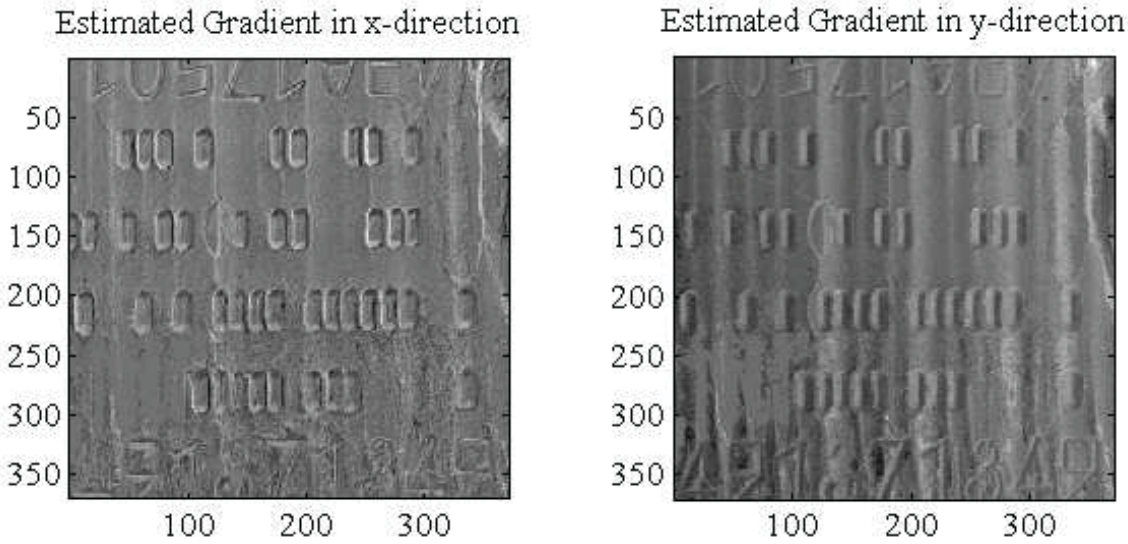


Figure 4.15: Obtain gradients using photometric stereo

Fig 4.15 shows the gradients computed from the four acquired images. Note how specularity in the images has been removed. Compare these images with those in Fig 4.16.

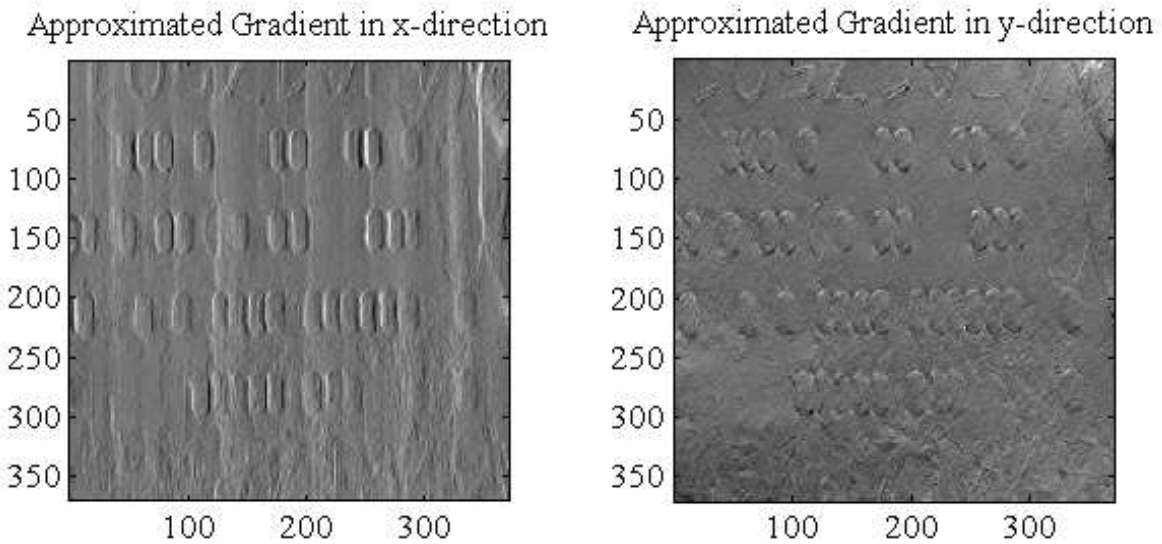


Figure 4.16: Reconstructed Gradients

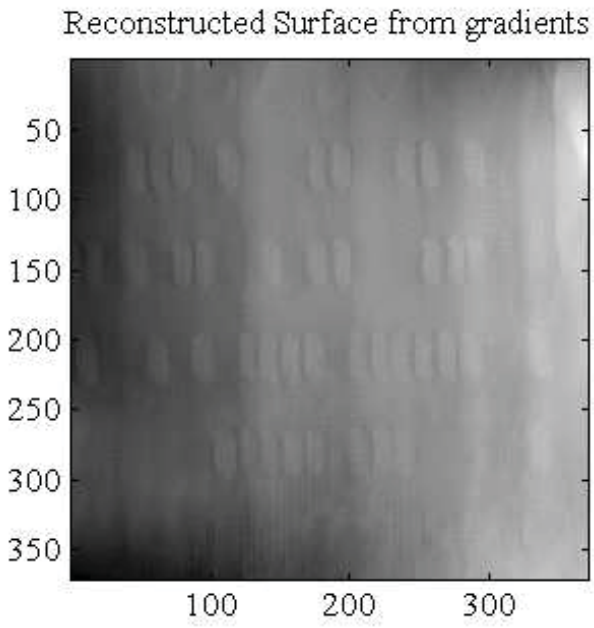


Figure 4.17: Reconstruction of Original Surface

Fig 4.17 shows the reconstructed surface from discrete moments. However, the numbers are no very clear. Fig 4.18 shows what the surface would have looked like without its roughness.

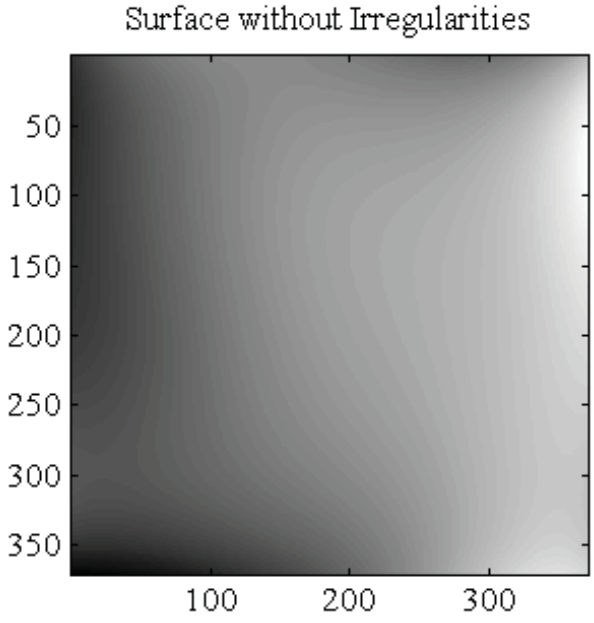


Figure 4.18: Reconstructed Surface without Defect

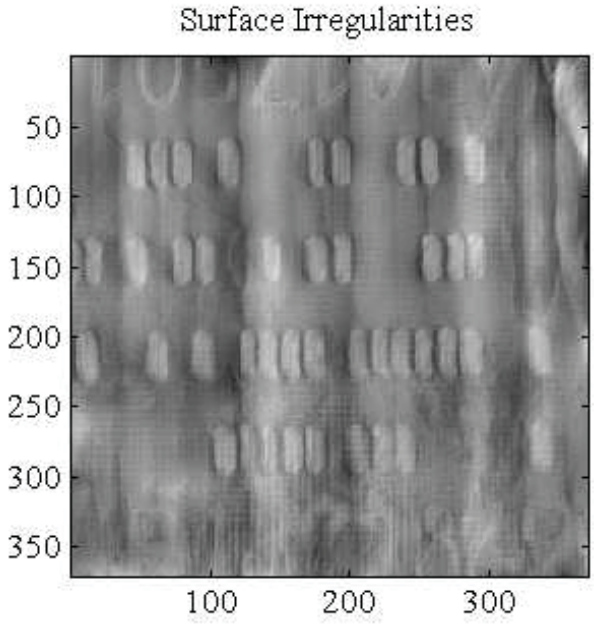


Figure 4.19: Extracted Surface Irregularities

Notice how the legibility of the number have been improved in Fig 4.19 after the flaws have been extracted. Fig 4.20 shows the extracted flaws as surface heights.

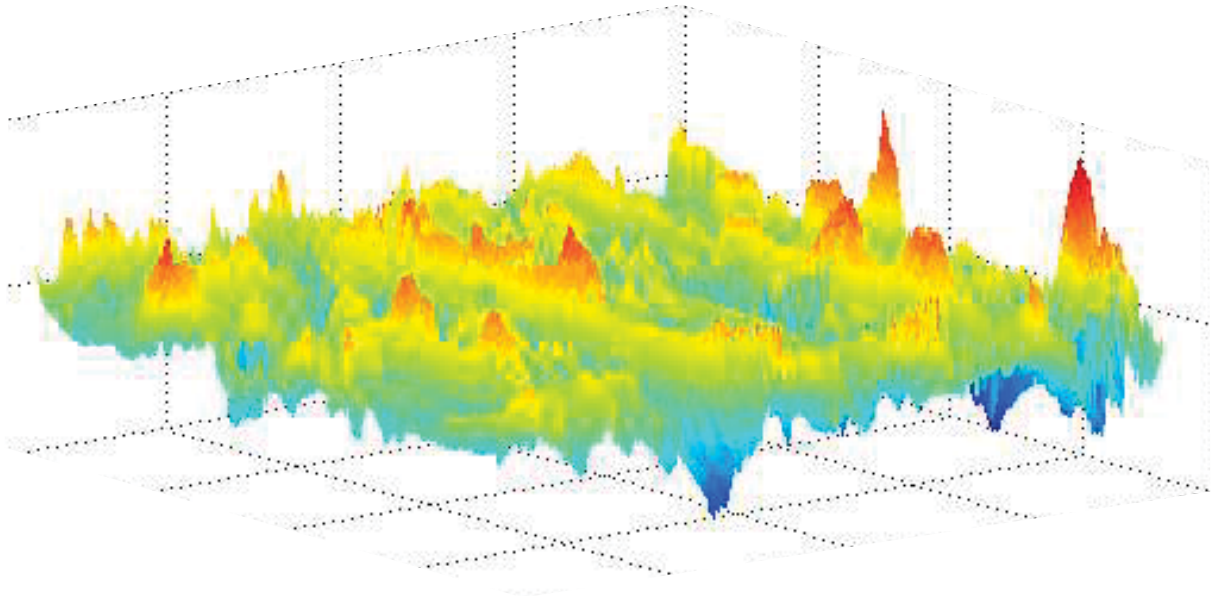


Figure 4.20: Extracted Surface Heights

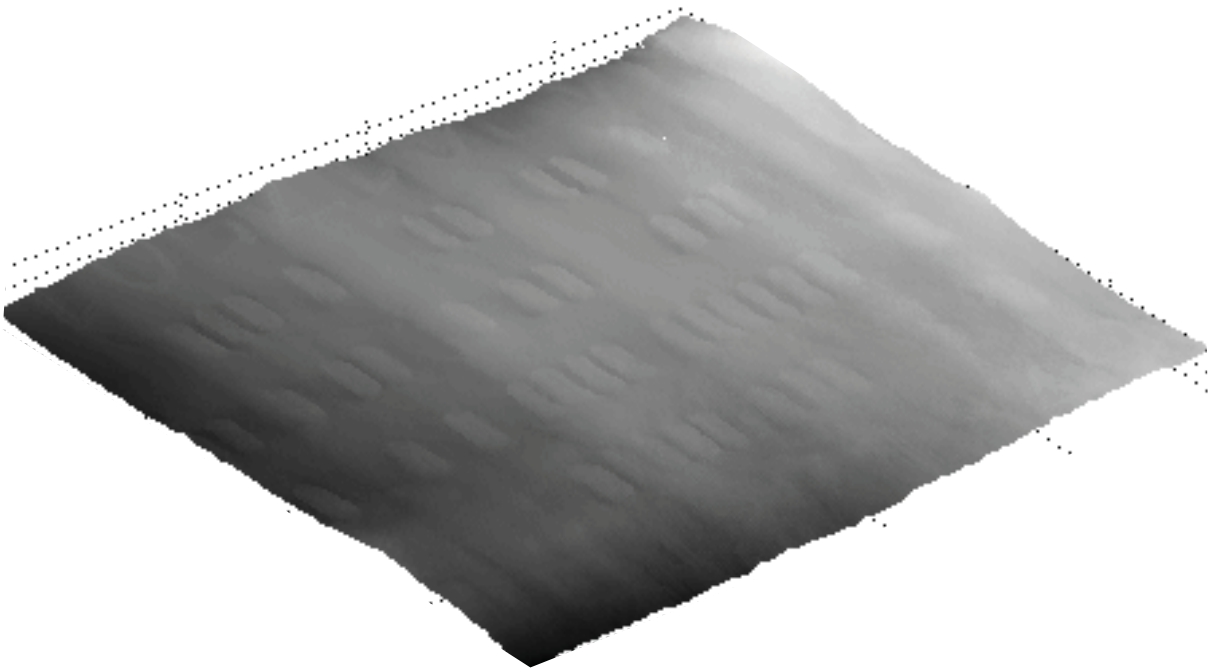


Figure 4.21: 3–D reconstructed Surface

Chapter 5

Summary and Conclusions

This thesis has shown that the use of four light source photometric stereo is possible and together with the pseudo-inverse of the light source matrix the problem of specularity often encountered in surface inspection can be solved.

This new algorithm has been shown to consider three special cases. The first case is a situation where no specular pixel exist, in which case the image gradients were computed normally by using the pseudo-inverse of the light source matrix. If on the other hand only a saturated pixel is encountered in one of the images, this particular pixel was eliminated from the computation and remaining three images were then used for gradients computations. If, finally, more than one pixels were found saturated then the the surface height of that particular pixel was set to vertical.

It has also been shown, in this thesis, the concept used in the new method of surface reconstruction from discrete polynomial moments. The moments were generated from the Vandermonde matrix and the computation reduced to a least square approximation solution. To obtain the coefficients of the moments the system of equations were reduced to Lyapunov equations, which are encountered in control engineering. As a precaution from having errors in solving this equation the matrices were partitioned before they were solved.

The required steps to be taken in order to be able to used this method can be summarised as follows:

1. Arrange for three or more images with different lighting positions.

2. Calculate image gradients I_x and I_y in the x and y diections respectively.
   — Eliminate saturated pixels and compute surface normal for the remaining three pixels.
   — When more than one pixels are saturated normalize I_x and I_y about the $z-$direction. i.e. by the use of four pixels use the pseudo-inverse of the source matrix, by three pixels use inverse of the source matrix, when two pixels are

saturated set the normal to vertical.

3. Apply the reconstruction algorithm to get the surface

In applying these methods the results have shown its effectiveness. The method of photometric stereo yielded the image gradients required for the surface reconstruction and when specular pixels were identified they were eliminated. The surface albedo were also evaluated from which the surface normals were obtained. A comparison of the reconstructed gradients and the estimated gradient further highlights the effectiveness of the reconstruction method. The surface defects were extracted and shown as surface heights and the quality of the reconstructed surface is a further proof of the performance of the reconstruction algorithm.

An inherent limitation of the photometric stereo method is that absolute height of object shapes cannot be reconstructed [33]. However, in situations where presence of flaws are to be identified and extracted, because of the numerical efficiency and superiority of the shown method, it could be incorporated in a real-time system in the industry for automatic surface inspection for identifying and extracting surface flaws.

Chapter 6

Outlook

6.0.1 Dynamic Photometric Stereo

The method used in this thesis to inspect metallic surfaces cannot be said to be all conclusive as there is room for future work. This method has been used subject to the fact that the inspected object is static. It would be interesting to investigate how this concept of four light-source photometric stereo behaves when applied to moving objects. This is what Smith and Smith [35] referred to as *Dynamic Photometric Stereo* (DPS). The first problem encountered here is how to acquire the images. Solving this problem has led to the idea of *multiplexing*. [35] gave a brief discussion of the various types of multiplexing. The types he identified are: temporal multiplexing; spatial multiplexing; and spectral multiplexing.

Brief definitions are now given for the various types. Temporal multiplexing represents an adaptation of the conventional static photometric stereo (SPS) to dynamic applications, in which separate lighting configurations are deployed and images are rapidly acquired at closely spaced intervals in time. Spatial multiplexing involves separate images of the same surface location being acquired at different points in space. image acquisition at the separate locations occurs simultaneously, therefore in order to register images between viewing positions the scan lines of the CCD(s) must be carefully synchronised with the velocity of the moving surface.

The SPS carried out in this thesis required switching the light on and off. In spectral multiplexing no switching of light on and off is required. Instead, three spectrally distinct light sources continuously illuminate the target objects from three different directions. A suitable RGB color camera acquires three-channel video images that subsequently are treated as three separate B&W images, one corresponding to each condition of illumination. An image/illumination configuration is called a '*channel*'. However, this method can be sensitive to decoloring of the surface, which may be interpreted as geometric variation [32]. Further investigation of this problem and proposals of addressing it would enhance the usefulness of this technique in the industry for two dimensional or three dimensional object texture classification.

Furthermore, [34] stated that surface reconstruction using moments of orthogonal polynomial basis is computational complex and therefore has been excluded from real- time implementations in software. [32] has shown that this is not true and real-time implementation is possible using these moments. Applying the conclusion reached in [32] to dynamic photometric stereo can be an issue to be looked into.

From the presented results it can be seen that an attempt was made to see the legibility of numbers and bar-codes engraved in an iron slab. Is it possible to apply the principle of photometric stereo to optical character recognition (OCR) or not? This is a question that could also be investigated because of its apparent usefulness.

6.0.2 Surface Reconstruction

The reconstruction method employed for this thesis is based on generating discrete polynomials from which its moments are derived. There exists now a new method of surface reconstruction in this Institute for Automation, which appears to reconstruct a required surface from its gradients quickly. This method is described in [43] and is based on the method of least square approximation. A brief explanation of this method is now given. The cost function between the require surface Z and its obtained gradients $\hat{\mathsf{Z}}$ is derived, which is then differentiated upon with respect to Z and the equation set equal to zero. The problem lies in solving this equation for the constant of differentiation.

The effectiveness of this method is based on the fast that for set of N distinct points the differentiated is exact and can be done to degree $d = N$ - 1 enabling the reconstruction of a full digital image. It is claimed that this method is numerically fast, which will make it suitable for on-line use. The four light photometric stereo can be used for obtaining the gradients and the reconstruction done with this method. It could also be tested for on-line use.

References

[1] Stan Birchfield. "An Introduction to Projective Geometry (for computer vision)" *Stanford University*, 1998 .

[2] M.L.Smith."Surface inspection techniques", *Professional Engineering Publishing Ltd*, 1st Edition, 2001.

[3] J. Wu. "Rotation Invariant Classification of 3D Surface Texture Using Photometric Stereo", Ph.D thesis *Heriot-Watt University*, 2003.

[4] J. L.Mundy, A. Zisserman."Projective Geometry for machine Vision", Appebdix to *Goemetric Invariance in Computer Vision, MIT Press,* Cambrigde, 1992.

[5] B. T. Phong. "Illumination for computed generated pictures", *Communications of the ACM*, Vol.18, No.6, June 1975.

[6] Robert J. Woodham. "Photometric Method for Determining Surface Orientation from Multiple images", *Optical Engineering*, Vol 19, No. 1, pp. 139–144, 1980.

[7] Homepage Istitute for computer based learning, Hariot-Watt University–http://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MARBLE/medium/shading/reflect.htm.

[8] B. K. P. Horn,. "Robot vision", *McGraw-Hill Book Company, The MIT Press,* 1986.

[9] F. Perkopf, "Automatic visual inspection of metallic surfaces", *VDI*, Reihe 8, Nr. 949, 2002.

[10] E.N Coleman, J. Ramesh. "Obtaining 3-dimensional shape of textured and specular surfaces using four-source photometry" *Jones And Bartlett Publishers, Inc. Physics-Based Vision: Principles And Practice*, pp. 180–199, 1992.

[11] A.El Gendy, A. Shalaby. "Improved Specularity Detection relying on a Photometric Stereo Technique", *GVIP Journal*, Vol 6, Issue 4, pp. 39–46, 2006.

[12] G.McGunnigle. "The classification of textured Surface under varying illuminating Direction", Ph.D thesis *Heriot-Watt University*, 1998.

[13] M. Eden, M. Unser, R. Leonadi. "Polynormial representation of Pictures", *Signal Processing*, Vol. 10, pp. 385–393, 1986.

[14] J. Shutler. Statistical Mooments

[15] M.K. Hu,. 1962, "Visual pattern recognition by moment invariants", *IEEE Transaction on Information Theory*, Vol 8, Issue 2, pp.(s): 179 - 187 pp. Feb 1962

[16] S. P. Prismall, M. S. Nixon, J. N. Carter. "Accurate object reconstruction by statistical moments", *International Conference on Visual Information Engineering, VIE 2003*, Volume 5, Issue 8 , pp. 29 - 32, July 2003.

[17] R. Mukundan, K.R. Ramakrishnan , "Moment functions in image analysis: theory and applications", *World Scientific*, Singapore, 1998.

[18] R. I. Harley,A. Zisserman. "Multiple view Geometry in Computer Vision", *Cambridge University Press* . 2nd Ed. , pp. 25-238, 2003.

[19] R. C. Love. "Surface Reflection Model Estimation from Naturally Illuminated Image Sequences", Ph.D thesis *University of Leeds*, 1996.

[20] Xiao D.He, K. E. Torrance, F. X. Sillion, D. P. Greemberg. "A comprehensive Physical Model for Light Reflection", *Computer Graphics,* Vol 25, No. 4, pp. 175–186 July 1991.

[21] Wikipedia the fre Encyclopedia

[22] Wolfram Mathwold, Eric Weisstein, Wolfram Researc

[23] G. W. Collins. "Fundamental Numerical methods and data analysis", *NASA Astrophysics Data System*, 1993, site: http://ads.harvard.edu/books/1990fnmd.book/.

[24] W. Gauschi. "Orthogonal Polynomials, Computation and Approximation", *Oxford University Press*, 2004.

[25] F. Ghorbel, S. Derrode, S. DhaHbi; R. Mezhoud. Reconsruction with geometric moments.

[26] J. Martinez, J. M. Porta, F. Thomas. "A matrix-Based Approach to the Image Moment Problem", *Journal of Mathematical Imaging and Vision*, Vol 26 , Issue 1-2, pp.105–113, November 2006.

[27] M. Pawlak. "Image Analysis by Moments: Reconstruction and Computational Aspects", *Oficyna Wydawncza Politechniki Wroclawskiej*, Wroclaw. 2006.

[28] M.R.Teague. "Image analysis via the general theory of moment", *Signal Processing*, Vol 87 , Issue 4 April 2007, pp. 687–708.

[29] P.T Yap,R. "Paramesran. An efficient method for the computation of legendre moments", *IEEE Transactions on Pattern Analysis and Machine Intelligence* , Vol 27 , Issue 12, pp. 1996–2002, December 2005.

[30] J. Shen, D. Shen. "Orthogonal legendre polynomials and their calculation", presented at the *13th International Conference on Pattern Recognition* (ICPR'96), Vol 2, pp. 241, 1996.

[31] T. Suesut, P. Schalk, P. O'Leary, I. Reindl, E. Fauster. "Real-time geometric surface inspection", presented at the *International Conference on Engineering, Applied Science and Technology*, November 2007.

[32] P. O'Leary. M. Harker. "Discrete polynomial moments for real-time geometric surface inspection", submitted for publication to: *Journal of Electronic Imaging*, August 6,2007

[33] K.Bang-Hwan, P. Rae-Hong. "Shape from Shading and photometric stero using surface approximation by Legendre polynomials", *Computer Vision and Image understanding*, vol. 66, pp. 255-270, 1997.

[34] L. Kotoulas, I. Andreadis. "Image analysis using moments",

[35] M.L. Smith, L.N. Smith. "Dynamic photometric stereo—a new technique for moving surface anylysis", *Image and Vision Computing*,vol 23, pp. 841-852, 2005.

[36] A.D. Marshal, R.R. Martin, "Computer Vision, Models, and Inspection", *World Scientific*, 1992.

[37] W. Gernot, "Thermoinductive Test Stand for Detection of Surface Cracks in Metallic Materials", *Diplomarbeit*, Institute for Automation, University of Leoben, Austria, 2004.

[38] K.Bang-Hwan, P. Rae-Hong. "Multiple-image Photometric Stereo using Surface Approximation by Legendre Polynomials", *Pattern Recognition*, Vol. 31 No. 8, pp. 1033-1047, 1998.

[39] P. Milanfar, W.C. Carl. "Moment–based Geometric Image Reconstruction",*IEEE* , vol , pp 825–859, 1994.

[40] A. Asano, "Radon Transformation and Projection Theorem", *Lecture Notes on Pattern Information Processing*, Division of Mathematical and Information Science, Department of Information Engineering, Faculty of Integrated Arts and Science, University of Hiroshima, Japan, 2006.

[41] J. Feldman. "The Classical Weierstrass Theorem", *Lecture Notes on Real Analysis II*, Department of Mathematics, University of British Columbia, Vancouver, 2001/2002.

[42] R. H. Bartels and G. W. Stewart. "Algorithm 432: Solution of the matrix equation AX + XB = C" , *Comm. ACM*, vol. 15, pp. 820-826, 1972.

[43] P. O'Leary, M. Harker."Least Square Surface Reconstruction from Measured Gradients". Submitted to CVPR 2008. Awaiting publication.

Appendix A

Matlab® Codes

A.1 Gradients and Albedo extraction

```
%
% Filename: gradient
%*****************************************************************************************
% Description:
% This code determines the intensity values of three images and with the position vector given determines
% the surface normal of the images and determine the values of p and q, where p is the image gradient in
% the x-direction and q is the image gradient in the y-direction. This code can also generate the surface albedo.
%
%   By              Godwin Ohenhen
%   Date:           December 14, 2007
%
% Version:          1.0
%
% © 2007, Institute for Automation, University of Leoben, Austria

% email: automation@unileoben.ac.at,    url: automation.unileoben.ac.at
%*****************************************************************************************
close all;
clear all;
clc;
%*****************************************************************************************
% read in images from file
%
[pic_1, mapa] =imread('C: Documents and Settings  Ohenhen My Documents My Pictures  Test Images
12_07_07  imageA3_12_07_07.bmp');
%
[pic_2, mapb] = imread('C: Documents and Settings  Ohenhen  My Documents  My Pictures  Test Images
12_07_07  imageB3_12_07_07.bmp');
%
[pic_3, mapc] = imread('C:  Documents and Settings Ohenhen My Documents  My Pictures Test mages
12_07_07 imageC3_12_07_07.bmp');
%
[pic_4, mapc] = imread('C: Documents and Settings  Ohenhen My Documents  My Pictures  Test Images
12_07_07  imageD3_12_07_07.bmp');
%
[ny, nx] = size( pic_1 );
cut = 370;
startx = round( (nx - cut)/2 );
starty = round( (ny - cut)/2 );
%
Ia = Ia(starty:starty+cut-1,startx:startx+cut-1) ;
Ib = Ib(starty:starty+cut-1,startx:startx+cut-1) ;
Ic = Ic(starty:starty+cut-1,startx:startx+cut-1) ;
Id = Id(starty:starty+cut-1,startx:startx+cut-1) ;
%%
%*****************************************************************************************
```

```
set up the position vector
%*****************************************************************************************
```

$$\mathsf{S} = 0.01 * \begin{bmatrix} s_{11} & s_{21} & s_{31}\ ;\cdots \\ s_{12} & s_{22} & s_{32}\ ;\cdots \\ s_{13} & s_{23} & s_{33}\ ;\cdots \\ s_{14} & s_{24} & s_{34} \end{bmatrix} ;$$

```
%
%*****************************************************************************************
      pinvS = pinv(S) ;
%*****************************************************************************************
PL = zeros(size(Ia)) ; QL = zeros(size(Ia)) ;
%
[m,n] = size( Ia ) ;
%
count = 0 ;
%
for i = 1:m
     for j = 1:n
          I = [Ia(i,j); Ib(i,j); Ic(i,j); Id(i,j)];
               %
          ind = find(I==254) ;
              %
          if isempty(ind)
               %
% No saturated pixels:
          %
N = pinvS * I ;
          %
        elseif length(ind) == 1
          %
% One saturated pixel:
          %
          rowind = find([1,2,3,4] =ind) ;
          %
          Ssub = S(rowind,:) ;
          %
          N = inv(Ssub)*I(rowind) ;
         %
        else
          %
% More than one saturated pixel:
          %
           N = [ 0; 0; -1 ] ;
           %
          count = count + 1 ;
          %
       end
          %
                N = pinvS * I ;
          PL(i,j) = –N(1)/N(3);
          QL(i,j) = –N(2)/N(3);
     end
end
%
P = PL ;
Q = QL ;
%
fig = figure;
subplot(2,2,1) ;
imagesc( Ia ) ;
axis image ;
colormap('gray') ;
title('Orignal image 1')
%
subplot(2,2,2) ;
imagesc( Ib ) ;
axis image ;
colormap('gray') ;
```

```
title('Original image 2');
%
subplot(2,2,3) ;
imagesc( Ic ) ;
axis image ;
colormap('gray') ;
title('Original image 3');
%
subplot(2,2,4) ;
imagesc( Id ) ;
axis image ;
colormap('gray') ;
title('Original image 4')
```

A.2 Surface Reconstruction

```
% Filename: reconstruction
%
% This code reconstructs the surface of the object using Polynomial basis fuction, the Vandermonde
% polynomail basis function. For this code we require the gradients obtained from the images.
%
%   By              Matthew Harker
%   Date:           December 14, 2007
%
% Version:          1.0
%
% ©2007, Institute for Automation, University of Leoben, Austria
% email: automation@unileoben.ac.at,    url: automation.unileoben.ac.at
%*********************************************************************************
close all;
clear all;
clc;
%*********************************************************************************
% Load the code for calculating the image gradient from a set of four photometric stereo images.
%
gradient;
%
Ix = P;
Iy = – Q;
%

[ny, nx] = size( Ix ); cut = 370;

% startx = round( (nx - cut)/2 );

starty = round( (ny - cut)/2 );

%
Ix = Ix(starty:starty+cut-1,startx:startx+cut-1) ;

Iy =Iy(starty:starty+cut-1,startx:startx+cut-1) ;

%

[m,n] = size(Ix) ;

%%

x = (1:n)' ;
y = (1:m)' ;

Evaluating the integration range of the Legendre Polynomial P(x) in the limit[-1,1] %
x = 2*(x-min(x))/(max(x)-min(x))-1;
y = 2*(y-min(y))/(max(y)-min(y))-1;

%
```

```
d = 450 ;

%

%****************** Set up the Vandermonde matrices: ******************************

XV = ones(size(x));
YV = ones(size(y));

XVI = x;
YVI = y;

XVD = zeros(size(x));
YVD = zeros(size(y));

%The Vandemonde matrix is of the form

%      X_v = [ 1   x11   x12...   x2d ]
%            [ 1   x21   x22...   x2d ]
%            [ .    .      .       .  ]
%            [ .    .      .       .  ]
%            [ 1   xn1   xn2...   xnd ].
% The same is done for Y_v.
%
for k = 1: d
          %
          XV = [XV, x.^k];
          YV = [YV, y.^k];
          XVD = [XVD, k * x.^(k - 1)];
          YVD = [YVD, k*y.^(k - 1)];
          %
end
%
XV(:,1:2) = [ones(size(x)), x];
YV(:,1:2) = [ones(size(y)), y];
%
%
XVD(:, 1:2) = [zeros(size(x)), ones(size(x))];
YVD(:, 1:2) = [zeros(size(y)), ones(size(y))];
%
[Qx,Rx] = qr( XV, 0 ) ;
[Qy,Ry] = qr( YV, 0 ) ;
%
XV = Qx; YV = Qy;
%
[dXVx,dXVy] = gradient(XV);
XVD = dXVy;
%
[dYVx,dYVy] = gradient(YV); YVD = dYVy;
%
% The Lyapunov matrix equation is of the form AX + BX = C
%
% This section solve for X using the Lyapunov's equation where;
%
A = YVD'*YVD;
B = XVD'*XVD;
C = -YV'*Ix*XVD - YVD'*Iy*XV;
%
%
As = A(2:end,2:end);
%
Bs = B(2:end,2:end);
% Cs =C(2:end,2:end);
%
c0 = C(1,1); c1 = C(1,2:end)'; c2 = C(2:end,1); cs = C(2:end,2:end);
%
Ms = lyap(As,Bs,Cs);
%
```

```
% Partition the matrix M
%The matrix is reduced to the form M = [ 0  x1'; x2  Ms ], % where Ms is the value of X obtained.
%
x1 = -Bs1; x2 = -As2;
%
M = [0, x1'; x2, Ms];
%
t = 88;
%****************************** Estimated Results ************************************
%
MT = M - fliplr(triu(fliplr(M),d+1-t));
ML =fliplr(triu(fliplr(M), d+1-t));
%
px = YV*M*XVD' ;       % approximated gradient in the x-direction
py = YVD*M*XV' ;       % approximated gradient in the y-direction
IIL = YV*M*XV' ;       % the reconstucted surface from gradients
Sp = YV*ML*XV' ;       % estimated surface without irregularities
Sr = YV*MT*XV' ;       % estimated surface irregularities

%****************************** Results Output ***********************************
%
figure
%
subplot(1,2,1);
imagesc(Ix);
colormap(gray);
axis image;
title('Estimated Gradient in x-direction');
%
subplot(1,2,2);
imagesc(Iy);
colormap(gray);
axis image;
title('Estimated Gradient in y-direction');
%
figure
%
subplot(1,2,1);
imagesc(px);
colormap(gray);
axis image;
title('Approximated Gradient in x-direction');
%
subplot(1,2,2);
imagesc(py);
colormap(gray);
axis image;
title('Approximated Gradient in y-direction');
%
fig = figure;
imagesc(IIL);
colormap(gray);
axis image;
title('Reconstructed surface from Gradients');
%
figure
%
subplot(1,3,1);
imagesc(IIL);
colormap(gray);
axis image;
title('Reconstructed Surface from gradients');
%
figure
%
subplot(1,3,2);
imagesc(Sp);
```

```
colormap(gray);
axis image;
title('Surface without Irregularities');
%
figure
subplot(1,3,3);
imagesc(Sr);
colormap(gray);
axis image;
title('Surface Irregularities');
%
figure

surf(IIL, 'linestyle', 'none');
axis image;
axis ij colormap('gray');
title('Contour Diagram of the Surface');
%
figure surf(YV*MT*XV','linestyle','none');
title('Surface Heights')
%
figure
%
contour(Sr );
axis image;
axis ij
colormap('gray');
title('Contour Diagram of the Surface');
```