Uncalibrated Photometric Stereo for 3D Surface Texture Recovery

Ivan Rabascall

May 2003

RM/02/02

Contents

List of Figures

Principal Symbols and Abbreviations

Symbol	**Description**	**Section first introduced**
I	Image data matrix	2.1
N	Unit vector surface normal	2.1
L	Unit vector light-source direction	2.1
τ	Tilt angle of the illumination vector	2.4.3
σ	Slant angle of the illumination vector	2.4.3
S(x,y)	Surface as a function	2.4.4
p	Facet slope in the x-direction	2.4.4
q	Facet slope in the y-direction	2.4.4
ρ	Albedo	2.4.4
T	Scaled surface normal	2.4.4
R(x,y)	Reflectance map	2.4.5
$\hat{p}$	Estimated facet slope in the x-direction	3
$\hat{q}$	Estimated facet slope in the y-direction	3
p	Number of pixel	3
f	Number of frame	3
U	*p*x*f* matrix of SVD	3
Σ	*f*x*f* diagonal matrix of SVD	3

V	*fxf* matrix of SVD	3
U‘	*p*x3 of U	3
Σ‘	3x3 of Σ	3
V‘	3x*f* of V	3
$\hat{S}$	Pseudo surface matrix	3
$\hat{L}$	Pseudo light-source matrix	3
$\hat{s}$	Pseudo surface vectors	3
A	Arbitrary invertible 3x3 matrix	3
B	Symmetric matrix $B = AA^T$	3
R	Rotation matrix	3
n	Number of pixel of the image	5
r	Absolute average of *p* and *q*	7

Abbreviation	**Description**	**Section first introduced**
SVD	Singular value descomposition	2.5.1
SNR	Signal-noise ratio	5
RMSE	Root mean square error	5
dB	Decibels	5

Acknowledgements

First of all, I would specially like to thank to my supervisor, Dr Mike Chantler for his support, and guidance in this dissertation.

Also I would like to extent my thanks to the Oficina de Relacions Internacionals of Universitat Rovira I Virgili, and David Riano from E.T.S.E and Mike Chantler from Heriot-Watt University for making my exchange possible between both universities.

I am very grateful to all people from the Texture Lab, especially Andy and Dong for his support and significant help, but mainly for our friendship, also I will miss the basketball matches with Dong.

But I also would like to thanks to all my friends from Edinburgh, who has been helping to enjoy in this fantastic city.

Finally, but not less important, to all my people in Barcelona, including my family and my friends.

Abstract

This dissertation presents a new photometric stereo method for estimating the surface normal and the surface reflectance of objects without a priori knowledge of the light-source direction or the light-source intensity. In this method, assuming only that the object's surface is Lambertian, the surface normal, and the surface reflectance, the light-source direction, and the light-source intensity can be determined simultaneously.

This method does not rely on any smoothness assumptions for these parameters. Furthermore, as the number of intensity images increases, the surface normal, the surface reflectance, the light-source direction, and the light-source intensity errors become smaller even if the intensity images are taken in a noisy environment.

Many experiments are carried out in order to assess the accuracy of this method. The investigation has been checked considering different kinds of surfaces.

Chapter 1
Introduction

1.1 Motivation & Objectives

The motivation of this dissertation stemmed from the necessity to find a reliable method to estimate the surface normal and the surface reflectance of objects without a priori knowledge of both, the light-source direction and the light-source intensity.

Photometric Stereo is a method for shape estimation with several kinds of applications and resources, mainly used for machine vision techniques. However the problem is that the majority of the previous methods need a priori knowledge of these both parameters, with the possibly added disadvantage that it means, therefore the effectiveness of this method is very important.

1.1 Document Organization

This dissertation may be divided into three main parts. In the first part, comprising chapter 2, a theoretical background concerning the photometric stereo is presented. In the second part, comprising chapters 3 and 4, the problem of unconstrained photometric stereo is strictly faced. In the third part, chapter 5, the experiments and conclusions are explained.

Chapter 2 includes the theory and comparison of constrained and unconstrained photometric stereo. For this reason, is divided in two main parts. The first part is a brief exposition about constrained photometric stereo. Although, in the second part, the theory of unconstrained photometric stereo is described. After that, in Chapter 3, the main problem of this dissertation is described, and the solution is strictly explained step by step, following with Chapter 4, where the pseudo code of the algorithm is given. Finally, Chapter 5 contains a list of experiments and conclusions about the efficiency of the algorithm explained before.

Chapter 2
Background Theory

2.1 Introduction

The technique of photometric stereo allows us to form surface description from several images of the same surface imaged under various illumination directions, varying the direction of the incident illumination between successive images, while holding the viewing direction constant.

Each illumination condition will have its own unique reflectance map, and a given point's intensity will vary accordingly. Therefore, each image defines a unique set of possible orientations.

On the other hand, photometric techniques use several images, imaged under a different illumination conditions, with their own specific, reflectance map (*figure 1.1*). The facet's derivatives, which are invariant, will lie at the intersection of this set of contours. Since two contours may overlap at more than one point, at least three images are required to resolve ambiguities in all cases. Mathematically, this may be expressed as follows:

$$i = n \cdot l$$

Where *i* is the level intensity, *n* is the unit surface normal, and *l* is the unit light-source direction. Now consider the same facet illuminates three times with different illuminant directions, and express the surface normal in terms of surface gradient.

$$\begin{pmatrix} i_1 \\ i_2 \\ i_3 \end{pmatrix} = \begin{pmatrix} n_x \\ n_y \\ n_z \end{pmatrix} \cdot \begin{pmatrix} l_{11} & l_{12} & l_{13} \\ l_{21} & l_{22} & l_{23} \\ l_{31} & l_{32} & l_{33} \end{pmatrix} \tag{1.1}$$

2.2 Differences between Constrained and Unconstrained PS

Unconstrained Photometric Stereo means that we do not have any priori knowledge of the light-source direction or the light-source intensity. However, in constrained Photometric Stereo the light-source direction is not an unknown, and for this reason will be easier to determine the surface normal and the surface reflectance.

2.3 Use of PS to Capture Bump Maps

Photometric Stereo can be used to capture the bump maps of real textures was introduced by [Woodham80]. Three images of a texture captured under known illumination conditions allow the model to be solved for the corresponding gradient maps and albedo image.

In [Blinn78] the concept of bump mapping was introduced as a technique for perturbing the normal of a smooth surface.

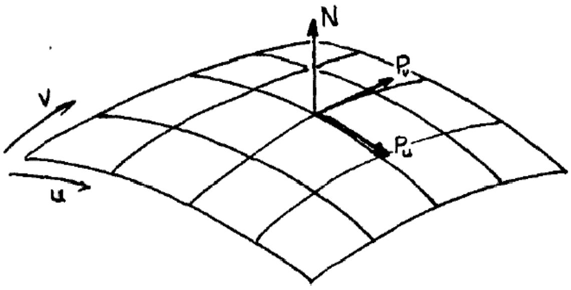

Figure 1: Definition of Normal Vector

2.4 Constrained Photometric Stereo

2.4.1 In 3D Shape

Photometric Stereo [Horn86] is a monocular 3D shape recovery method, that relies on a few images of the same scene taken under different lighting conditions. There are usually two processing steps: First, the direction of the normal to the surface is estimated at each visible point [Woodham80]. The set of normal directions, also known as the needle diagram, is then used to determine the 3D surface itself.

For a Lambertian surface, the normal direction and the albedo of the surface at each point can be determined locally using three images, obtained with different illumination directions. For neutral colored surfaces, the use of color allows photometric stereo recovery from a single image. This result can be extended to well-behaved colored surfaces [Drew87]. Without using color, reducing the number of images is possible only by sacrificing the locality of the normal direction computation, taking spatial considerations into account. At the limit, shape from shading [Horn86] requires a single image, but then solving for the normal direction or 3D location of any point requires integration of data from all over the image.

The local estimation of the normal directions in photometric stereo is the key to its numerical stability. In determining the 3D surface, each visible point contributes one unknown (depth) but two constraints, the components of the normal direction. The problem is thus highly overdetermined, hence extremely robust with respect to noise. The classic approach to computing shape from needle diagram data is via the calculus of variations, but direct integration [Wu88] and frequency-domain methods [Frankot88] have also been developed.

In spite of the quality of 3D recovery possible with photometric stereo, its fundamental noise immunity, the absence of the correspondence problem and the low cost of the setup, photometric stereo is seldom applied in practice. Photometric stereo can be used to accurately recover local shape details, but cannot always be trusted for global shape measurements.

3D shape recovery using photometric stereo is based on knowing (or estimating) the directions of the light sources and also requires some assumptions regarding their relative intensity, beam crossectional uniformity and presence of ambient light.

2.4.2 Surface Description

A description of a surface can be made on several levels. A single parameter may be sufficient to characterize a surface for some purposes, whereas in other cases a much greater degree of description is required. A first level of description seeks to estimate some property of the surface with a single parameter, i.e. height or gradient, with a single parameter. On a second level, a statistical model such as the histogram, is applied to the variation of height or gradient, which provides a more visual comprehension of the surface's characteristics. At a third level, there are those techniques which incorporate spatial interaction such as the Power Spectral Density (PSD) or the Autocorrelation Function (ACF).

2.4.3 Defining Illumination Direction

There are a number of different variables which can be used to define the direction of a light-source relative to a surface. These are used in photometric scheme since it uses images taken under different illumination conditions and should be defined for clarity.

If the illuminated plane is assumed to have been mounted in the x-y plane and is perpendicular to the z-axis, the direction of the illumination source can be defined by two angles. These are known as *slant* and *tilt*. The *tilt* angle is a measure of the angle that the illuminant direction makes with the x-axis in the x-y plane. The *slant* angle is that which the illuminant direction makes with the z-axis (*Figure 2*).

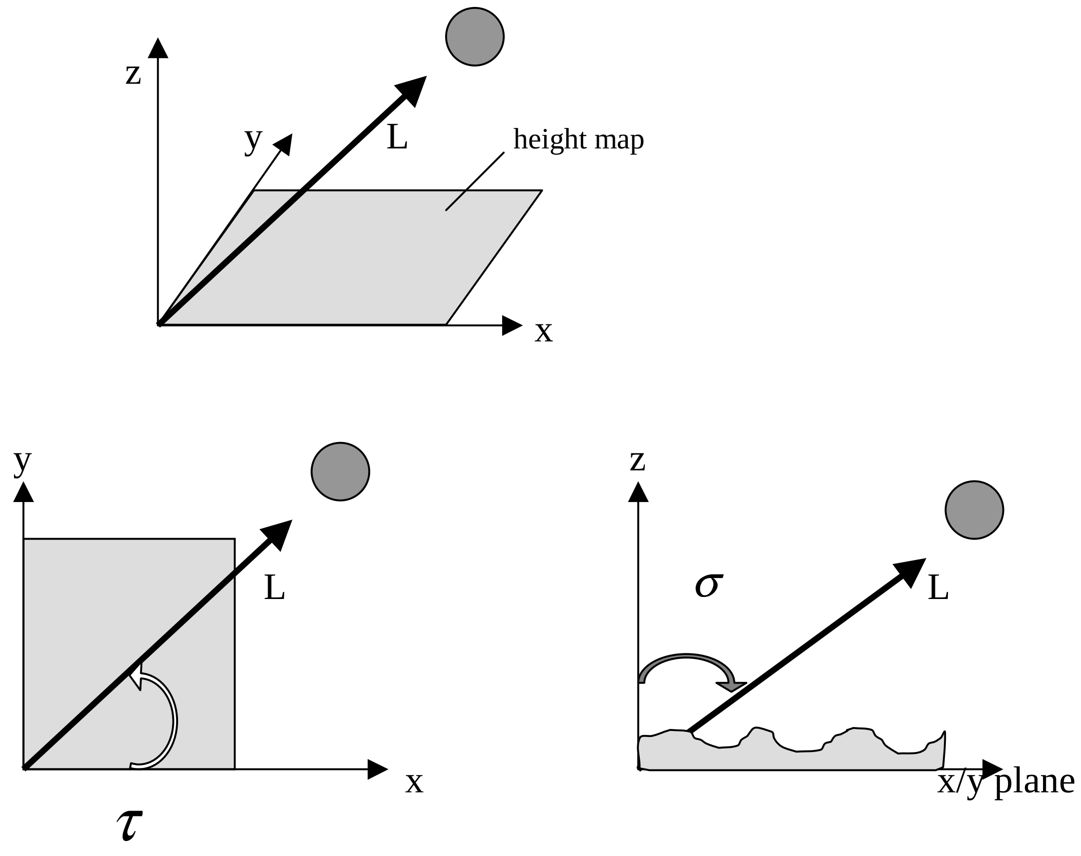

Figure 2: Definition of illumination direction in terms of tilt angle τ and slant angle σ.

2.4.4 Formulation

If an approximately flat texture plane coincides with the x-y plane, the surface *S* can be described as a height function as follows:

$$z=S(x,y)$$

The surface gradients in the x and the y directions for a facet on such a surface are therefore as follows:

$$p = \frac{dz}{dx} \qquad q = \frac{dz}{dy}$$

where $-1 < p,q < +1$

Two tangents perpendicular to this facet can then be written in vector form as [1,0,p] and [0,1,q]. The vector normal to the facet, *N*, is found by taking the cross-product of these tangents:

$$N = [p,q,-1]$$

Once normalized this becomes:

$$N = \frac{1}{\sqrt{p^2 + q^2 + 1}} [p,q,-1]$$

If the facet is illuminated by a light-source, the unit illumination vector *L*, which points away from the surface is written:

$$L = \lfloor l_x, l_y, l_z \rfloor$$

Defined in terms of a polar co-ordinate system this becomes:

$$L = [\cos\tau \sin\sigma, \sin\tau \cos\sigma, \cos\sigma]$$

where τ =tilt angle and σ =slant angle

Then:

$$\begin{bmatrix} p_1 & p_2 & \cdots & p_n \\ q_1 & q_2 & \cdots & q_n \\ -1 & -1 & \cdots & -1 \end{bmatrix} \cdot [lx \quad ly \quad lz] = [i_1 \quad i_2 \quad \cdots \quad i_n]$$

Lambert states that intensity is proportional to the cosine of the angle between the illumination vector *L* and the surface facet normal *N*, i.e. the dot product of the two vectors. The intensity of an illuminated surface facet captured by a camera at some distance from the texture along the z-axis can therefore be written as follows:

$$i = \rho(L \cdot N)$$

where ρ is the albedo

and $|N| = 1$, $|L| = 1$

Albedo is the ratio of the light reflected by a body to the light received by it. Albedo values range from 0 (pitch black) to 1 (perfect reflector).

$$albedo = \frac{light\ \ reflected}{light\ \ in}$$

In the photometric stereo scheme being considered three illuminations conditions are used producing three images of the texture. With regard to a particular surface facet three intensities are determined.

$$i_k = \rho(L_k \cdot N)$$

where k=1,2,3

Written in matrix form this becomes:

$$i = \rho\begin{bmatrix} lx & ly & lz \end{bmatrix}\begin{bmatrix} nx \\ ny \\ nz \end{bmatrix} = \rho[lx \cdot nx + ly \cdot ny + lz \cdot nz]$$

Or:

$$\begin{bmatrix} i_1 \\ i_2 \\ i_3 \end{bmatrix} = \rho \begin{bmatrix} lx_1 & ly_1 & lz_1 \\ lx_2 & ly_2 & lz_2 \\ lx_3 & ly_3 & lz_3 \end{bmatrix} \cdot \begin{bmatrix} nx \\ ny \\ nz \end{bmatrix}$$

Or the same:

$$I = \rho([L] \cdot N)$$

This can be re-written as:

$$[L]^{-1} I = \rho N$$

In [Barsky01] is defined as vector *T*, scaled surface normal:

$$T = \rho N$$

Then:

$$T = [L]^{-1} I$$

Solving this equation yields the surface gradients and the albedo as follows:

$$p = -\frac{t_1}{t_2} \qquad q = -\frac{t_2}{t_3}$$

$$\rho = \sqrt{t_1^2 + t_2^2 + t_3^2}$$

Hence three images defining surface topography and reflection are output from this algorithm: the *p* and *q* maps are the equivalent of a bump map and the albedo image or the proportion of the incident light reflected back diffusely.

2.4.5 Applications of Photometric Stereo

Photometric stereo can be used in two ways. First, photometric stereo is a general technique for determining surface orientation at each image point. For a given image point (x,y), the equations characterizing each image can be combined to determine the corresponding gradient (p,q).

Second, photometric stereo is a general technique for determining object points that have a particular surface orientation. This use of photometric stereo corresponds to interpreting the basic image-forming as one equation in the unknowns x and y. For a given gradient (p,1), the equations characterizing each image can be combined to determine corresponding object points (x,y). This second use of photometric stereo is appropriate for the so-called industrial "bin-of-parts" problem. The location in an image of key object points is often sufficient to determine the position and orientation of a known object on a table or conveyor belt so that the object may be grasped by an automatic manipulator.

A particularly useful special case concerns object points whose surface normal directly faces the viewer (i.e., object points with p=0 and q=0). Such points form an unique class of image points whose intensity value is invariant under rotation of the illumination direction about the viewing direction. Object points with surface normal directly facing the viewer can be located without explicitly determining the reflectance map R(p,q). The value of R(0,0) is not changed by varying the direction of illumination, provided only that the phase angle g is held constant.

Photometric Stereo also can be used for specular surfaces if extended light sources with spatially varying brightness are used, also to obtain surface curvature, surface depth when a point light source is relatively near the surface and near the camera. Also, this technique has been practically applied to scanning electron microscopic images.

2.4.6 Advantages

- Photometric stereo is better than traditional stereo when the surface gradient is to be found, working on smooth surfaces with few discontinuities, or when applied to surfaces with uniform surface properties.
- Since the images are obtained from the same point of view, there is no difficulty identifying corresponding points in the two images. This is the major computational task in traditional stereo.
- Under appropriate circumstances, the surface reflectance facto can be found because the effect of surface orientation on image intensity can be removed. Traditional stereo provides no such capability.
- Describing object shape in terms of surface orientation is preferable in a number of situations to description in terms of range or altitude above a reference plane.

2.5 Unconstrained Photometric Stereo

2.5.1 Introduction

A new photometric-stereo method for estimating the surface normal and the surface reflectance of objects without a priori knowledge of the light-source direction or the light-source intensity is proposed in [Hayakawa94]. In this method, assuming only that the object's surface is Lambertian, the surface normal, and the surface reflectance, the light-source direction, and the light-source intensity can be determined simultaneously. For example, we can estimate these four parameters from intensity images that are obtained under a light-source arbitrarily moved by a human. This method does not rely on any smoothness assumptions for these parameters. Furthermore, as the number of intensity images increases, the surface normal, the surface reflectance, the light-source direction, and the light-source intensity errors become smaller even if the intensity images are taken in a noisy environment.

The foundation of this method is similar to that of a factorization method for shape and motion estimation from image streams [Tomasi92]. In this method, we construct a p X f image data matrix I from p pixel image intensity data though f frames by moving a light-source arbitrarily. Under the Lambertian assumption, the image data matrix can be written as the product of two matrices S and L, with S representing the light characteristics (light-source direction and intensity). Using this formulation, we show that the image data matrix I has a rank of 3. On the basis of this observation, we use a singular-value decomposition (SVD) technique and one of the following constraints to factorize the image data matrix. One constraint is the constraint of surface reflectance. This constraint can be used when there at least 6 pixels in which the relative value of the light-source intensity. The idea of applying the SVD technique to observe image intensities to separate them into fundamental components was introduced previously [Sato93][Funt93].

2.5.2 Assumptions

In this method we assume that the object's surface is Lambertian. A Lambertian surface is a surface with perfectly matte properties, which means that these surfaces reflect light with equal intensity in all directions, and hence appear equally bright from all directions. For a given surface, the brightness depends only on the angle θ between the direction of the light-source *L* and the surface normal *N* (*figure 3*).

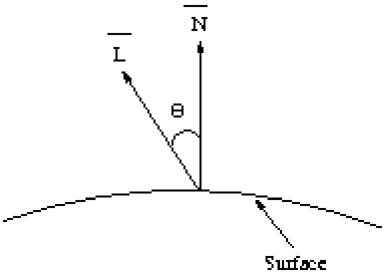

Figure 3: Lambertian Reflection

2.5.3 Advantages

The present method also describes how to deal with shadow regions. In an intensity image, there are two types of shadows: self-shadows and cast shadows. Previous methods could not deal with either shadow type without relying on certain assumptions. In the image data matrix, we select an initial submatrix having no shadowed data. We can then estimate the surface normal reflectance in shadow regions by growing a partial solution obtained from the initial submatrix.

The effectiveness of this method is demonstrated through performance analysis and through a laboratory experiment on Lambertian reflectance objects. Furthermore this method is applied to scenes an architectural model of the ATR building that are photographed with a hand-held camera.

Chapter 3
Algorithm Unconstrained Photometric Stereo

In this chapter, we will explain an algorithm to factorize image data matrix *I* into surface matrix *S* and light-source matrix *L* [Hayakawa94].

- First of all, we must to define the image data matrix I . Assuming that we measure the image intensity data i at p pixels through f frames by moving only a light-source. We write the image intensity data i into a p x f matrix I , with 1 row/pixel and 1 column/frame:

$$I = \begin{bmatrix} i_{11} & \cdots & i_{1f} \\ \vdots & \cdots & \vdots \\ i_{p1} & \cdots & i_{pf} \end{bmatrix}$$

- Compute the SVD of the image data matrix, assuming that $p \geq f$, the matrix can be decomposed into a p x f matrix U , a diagonal f x f matrix Σ, and an f x f matrix V:

$$I=U\Sigma V$$

Where $U^T U$ = V^T V = V V^T $= E$, where E is the f x f identity matrix. Σ is a nonnegative diagonal matrix whose diagonal entries are the singular values $\sigma_1 \geq \sigma_2 \geq \ \dots \ \geq \sigma_f \geq 0$ sorted in nonincreasing order. This is the SVD of the matrix *I*. Focusing only on the first three columns of *U*, the first 3 x 3 submatrix of Σ, and the first three rows of V, we can partition these matrices as follows:

$$U = \begin{bmatrix} \underbrace{U^{'}}_{3} & \underbrace{U^{''}}_{f-3} \end{bmatrix} \} \, p$$

$$\Sigma = \begin{bmatrix} \Sigma^{'} & 0 \\ \underbrace{0}_{3} & \underbrace{\Sigma^{''}}_{f-3} \end{bmatrix} \begin{matrix} \} \, 3 \\ \} \, f-3 \end{matrix}$$

$$V = \begin{bmatrix} V^{'} \\ \underbrace{V^{''}}_{f} \end{bmatrix} \begin{matrix} \} \, 3 \\ \} \, f-3 \end{matrix}$$

- Define the pseudo surface matrix $\hat{S}$ and the pseudo light-source matrix $\hat{L}$ as:

$$\hat{S} = U'\left(\pm \, [\Sigma']^{1/2}\right)$$

$$\hat{L} = \left(\pm \, [\Sigma']^{1/2}\right) V'$$

There are two different signs for the pseudo solutions. They correspond to the solutions in the right-handed and the left-handed coordinate systems. We choose the pseudo surface matrix $\hat{S}$ that corresponds to the 3 x 3 matrix whose determinant's sign is positive, and we also select the paired pseudo light-source matrix $\hat{L}$ with the chosen pseudo surface matrix $\hat{S}$.

- Due to the pseudo surfaces matrices $\hat{S}$ and $\hat{L}$ are different from S and L in general, we must find the matrix A such that

$$S = \hat{S}\, A$$
$$L = A^{-1}\, \hat{L}$$

But, this matrix is impossible to find without any knowledge of the surface and light characteristics. Then, we use the following useful constraint:

  - (C1) we can find at least 6 pixels in which the relative value of the surface reflectance is constant or known.

To find the matrix A using this constraint, first of all, we extract $p' (\geq 6)$ pseudo surface vectors $\hat{s}$ of matrix $\hat{S}$. Knowing that:

$$\hat{s}_k^T AA^T \hat{s}_k = 1 , \qquad k = 1, \ldots , p'$$

Then, we introduce the symmetric matrix $B = AA^T$, we can rewrite as:

$$\hat{s}_k^T B \hat{s}_k = 1 \ , \qquad\qquad k = 1, \ldots , \ p'$$

Therefore:

$$\underbrace{[abcefi]}_{B} \underbrace{\begin{bmatrix} x^2 & & x^2 \\ 2xy & & 2xy \\ 2xz & \cdots\cdots\cdots & 2xz \\ y^2 & & y^2 \\ 2yz & & 2yz \\ z^2 & & z^2 \end{bmatrix}}_{K} = \underbrace{[1\cdots\cdots1]}_{K}$$

Once B is determined, we can find the matrix A taking the SVD of B. $B = W\Pi W^T$, where W is diagonal and Π is orthonormal. Finally, we let $A = W[\Pi]^{1/2}$.

- Finally, compute the surface matrix S and light-source matrix L as:

$$S = \hat{S} \ A$$

$$L = A^{-1} \hat{L}$$

Chapter 4
Algorithm for Absolute Orientation

The unconstrained photometric stereo algorithm described in the previous chapter represents the surface matrix *S* and light-source matrix *L* in an arbitrary 3D coordinate system. But, if we know three surface normal or three light-source directions that are represented in the viewer-oriented coordinate system, then these matrices can be aligned to the viewer-oriented coordinate system by solution of an absolute orientation problem [Horn87].

- First of all, we let the coordinates of the three points in each of the two coordinates system be $r_{l,1}$, $r_{l,2}$, $r_{l,3}$ and $r_{r,1}$, $r_{r,2}$, $r_{r,3}$, respectively.

- Construct:

$$x_l = r_{l,2} - r_{l,1}$$

- Then

$$\hat{x}_l = x_l / \|x_l\|$$

is a unit vector in the direction of the new *x* axis in the left-hand system.

- Now let

$$y_l = (r_{l,3} - r_{l,1}) - [(r_{l,3} - r_{l,1}) \cdot \hat{x}_l]\hat{x}_l$$

the component of $(r_{l,3} - r_{l,1})$ perpendicular to $\hat{x}$. The unit vector

$$\hat{y}_l = y_l / \|y_l\|$$

is in the direction of the new *y* axis in the left-hand system.

- To complete the triad, we use the cross product

$$\hat{z}_l = \hat{x}_l \times \hat{y}_l$$

- This construction is now repeated in the right-hand system to obtain $\hat{x}_r$, $\hat{y}_r$ and $\hat{z}_r$.

- Now adjoin column vectors to form the matrices M_l and M_r as follows:

$$M_l = |\hat{x}_l \hat{y}_l \hat{z}_l| \ , \qquad M_r = |\hat{x}_r \hat{y}_r \hat{z}_r|$$

- The rotation matrix is given by

$$R = M_r M_l^T$$

- Finally, we must find the surface matrix *S* and light-source matrix *L* in an absolute coordinate system such that

$$L=L'\ R$$

$$S=S'\ R$$

Chapter 5
Design of Implementation

This chapter is concerned with the algorithm described in chapter 3 and 4, actually is a summary of the implementation in pseudo code with attention on the important steps of this algorithm. This design is divided in different parts, as is shown in the following diagram.

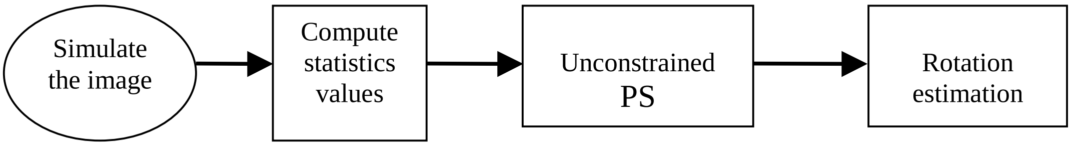

Figure 4: Block diagram of the design of the method.

1. Simulate the image (Appendix A.1)

First of all, we must to create the image that will use to test the technique. For this reason, we have developed an algorithm that is able to simulate an image dependent on the number of pixels, number of frames and roughness of the surface that we wish to use.

```
Define parameters of the image as n_pixels,
total_frames, p, q, slant, tilt.

Random to simulate p and q

While n_pixel<==total_n_pixels loop
    While n_frame<total_frames loop
         Image(pixel)=-p*cos(tilt)*sin(slant)-
         q*sin(tilt)*sin(slant)+cos(slant)
         N_frame++
         Tilt+60dg
    End
    N_pixel++
    Compute surface matrix (S)
    Random to simulate p and q
end
```

1.1 Add noise to the image (Appendix A.1.1)

Is exactly the same as the algorithm described above, but in this case, after computing the image we must to add the noise.

```
Define parameters of the image as n_pixels,
total_frames, p, q, slant, tilt.

Random to simulate p and q

While n_pixel<==total_n_pixels loop
    While n_frame<total_frames loop
         Image_noise(pixel)=I(I,j)+random(k)
         N_frame++
         Tilt+60dg
    End
    N_pixel++
    Compute surface matrix (S)
    Random to simulate p and q
end
```

2. Compute statistics values (Appendix A.2)

This part is concerned to the statistics calculations as the signal of the image, the signal-to-noise ratio (SNR) or the roughness.

```
%Compute the p and q average (add all the p and q and
%later we divide by number of pixels)

while I<n_total_pixels loop
    P_average=P_average+p_of_pixel(I,1)
    q_average=q_average+q_of_pixel(I,2)
    I++
End
P_average=P_average/n_total_pixels
q_average=q_average/n_total_pixels


%Compute signal(p) and signal(q)
```

$$signal(p)=\frac{\sum_{x,y}\left(p(x,y)-\overline{p}\right)^2}{n}$$

```
while I<n_total_pixels loop
    signal_p=signal_p+(p_of_pixel(I,1)-p_average)^2
    signal_q=signal_q+(q_of_pixel(I,2)-q_average)^2
    I++
End
signal_p=signal_p/n_total_pixels
signal_q=signal_q/n_total_pixels

%Compute signal
signal=(signal_p+signal_q)/2

%Compute root mean square error (RMSE)
```

$$RMSE=\sqrt{\frac{\sum\left[p(x,y)-\hat{p}(x,y)\right]^2}{n^2}}$$

```
While I<=n_total_pixels loop
```

```
            Res=res+((real_surface_matrix(row,col)-
            (estimated_surface_matrix(row,col))^2
End
RMSE=sqrt(res/n_pixel^2)

%Compute signal-to-noise ratio (SNR)
```

$$SNR = abs(20\log_{10}(signal/noise))$$

```
SNR=abs(20*log10(signal/rmse))
```

3. Unconstrained PS (Appendix A.3)

The implementation of the algorithm to factorize image data matrix *I* into surface matrix *S* and light-source matrix *L* [Hayakawa94] is as follows:

```
Load the image data matrix (I)
[U,E,V] = svd of I
U'= px3 of U
E'= 3x3 of E
V'= 3xf of V
If Determinant_of_U'*+[E']$^{1/2}$ > 0 then
    Pseudo_surface=U'*+[E']$^{1/2}$
    Pseudo_light-source=+[E']$^{1/2}$*V'

Else if Determinant_of_U'*-[E']$^{1/2}$ > 0 then
    Pseudo_surface=U'*-[E']$^{1/2}$
    Pseudo_light-source =-[E']$^{1/2}$*V'

End
Compute B matrix
[U2,E2,V2] = svd of B
A=U2*[E2]$^{1/2}$
Surface_matrix=pseudo_surface*A
Light-source_matrix=A$^{-1}$*pseudo_light-source_matrix
```

4. Rotation estimation (Appendix A.4)

The implementation of the algorithm to denote the surface matrix *S* and light-source matrix *L* in an absolute coordinate system [Horn87], is as follows:

```
Let coordinates of 3 points left-hand system (pseudo
light-source_matrix)
Compute unit vector in x axis ($\hat{x}_l$)
Compute unit vector in y axis ($\hat{y}_l$)
Compute unit vector in z axis ($\hat{z}_l$)
Let coordinates of 3 points right-hand system (light-
source-matrix)
Compute unit vector in x axis ($\hat{x}_r$)
Compute unit vector in y axis ($\hat{y}_r$)
Compute unit vector in z axis ($\hat{z}_r$)
Compute the rotation matrix R
Surface_matrix_after_rot=surface_matrix*R
Light-source_matrix_after_rot =light-source_matrix*R
```

Chapter 6
Experiments and Results

In this chapter, we will check the reliability of the method explained before [Hayakawa94] to compute the surface matrix *S* and the light-source matrix *L*. Consequently, to test it, we will develop several kinds of experiments and examples.

Mainly, the accuracy of this process will depend on the image: the number of pixels, the smoothness or roughness and the quantity of noise of the image.

Furthermore, in order to define the roughness we have introduced a new symbol *r*. This parameter express the absolute average of *p* and *q*, and is computed by:

$$r = \frac{|\overline{p}| + |\overline{q}|}{2}$$

In fact, to know the consistency of this technique, we use the signal-to-noise ratio. SNR measures are estimates of the quality of a reconstructed image compared with an original image. It is the ratio between the input signal and the noise. SNR in decibels (dB) is computed by using:

$$SNR = abs(20\log_{10}(signal/noise))$$

Where the signal is computed by using:

$$signal = \frac{\sum_{x,y} \left(p(x,y) - \overline{p}\right)^2}{n}$$

Actually, to know the noise, we must to compute the Root Mean Square Error (RMSE) as follow:

$$RMSE=\sqrt{\frac{\sum[p(x,y)-\hat{p}(x,y)]^2}{n^2}}$$

Assuming $p(x,y)$ is the source image that contains *n* by *n* pixels and $\hat{p}(x,y)$ is the reconstructed image.

Therefore, we will build some different kinds of analysis, where the RMSE are exposed in each case to trial the influence in the noise.

6.1 Varying the number of pixels.

In this section, we prove the trustworthiness of the application dependent on the number of pixels of the image source. After that, we fix the surface smoothness and only varying the size of the image will test how influence it in the noise.

The results are shown belove:

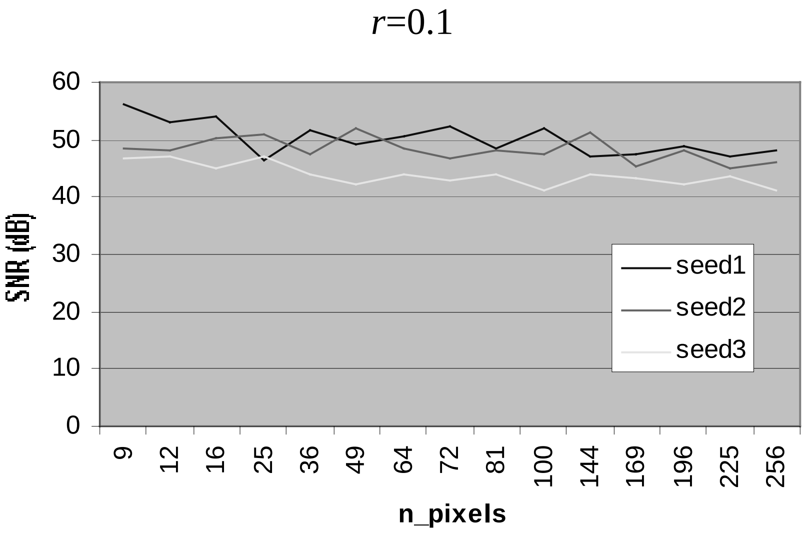

Figure 5: Graph of Signal-to-Noise Ratio against number of pixels.

This chart reveals that the number of pixels do not have a significantly influence in the final noise. Although, we work with a high number of pixels, the accuracy of the image is almost the same than with a few pixels.

However, this case has only been tested with 256 (16x16) pixels as a maximum number of pixels, due to the limitation in computing time of Matlab, but probably with a bigger matrix could demonstrate that the accuracy would be incremented.

6.2 Varying the roughness.

In this section, we try the reliability of the method dependent on the roughness of the surface. Thus, we fix the number of pixels and only varying the smoothness of the image will test how is significant it in the noise.

Therefore, to run this test is necessary to implement a short algorithm to create a different kind of roughness in each pixel of the image. Nevertheless, in this trial the parameters *p* and *q* are computed by a random number multiplied by a constant *k*. Thus, the constant *k* will vary each time to check the influence in the precision of the image. Consequently, as bigger is the constant higher is the value of *r*, or that is the same, the surface is more irregular.

$$p = k * random\ (seed)$$

$$q = k * random\ (seed)$$

Owing to the existence of a random number and this can generate a non useful values, is appropriate to run different trials only modifying the seed value in the random, and after that estimate the average of all of them.

The results are:

n_pixels=16

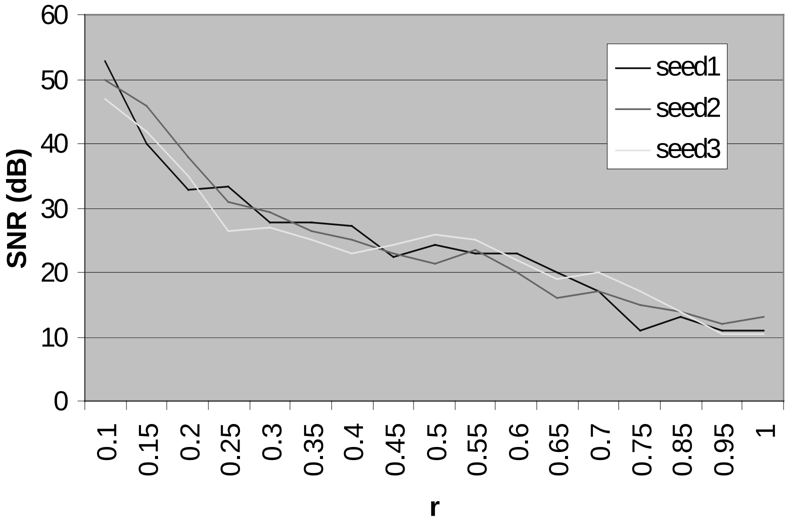

Figure 6: Graph of Signal-to-Noise Ratio against roughness.

This graph is expressed in dB, noise in the output, against *r*. Furthermore, we can observe three different kinds of seed.

Consequently, as is revealed in the chart, the point with less noise is when the roughness is lower, this indicates that for low values of *r*, the signal-noise ratio is quite superior, subsequently the noise is lower and the accuracy of the image become much better. Moreover, the precision of the image descends relatively faster as the image is bumpier.

6.3 Varying the quantity of noise.

In this section, we try the reliability of the method dependent on the quantity of noise that contains the image. Therefore, to design this experiment we have had to insert some amount of noise in the simulate image. Thus, the modified image is just a random number multiplied by a constant number added in each pixel of the image. Due to work with a random number always can generate values out of the ordinary, is recommended to use different seed values. Thus:

$$image\_with\_noise(i,j) = image(i,j) + random(seed) * k$$

The results are expressed in the ratio between the initial signal and the noise between the two images, with and without noise, against the ratio between the initial signal and the total noise produced.

And the results are:

r=0.5, *number_pixels*=16

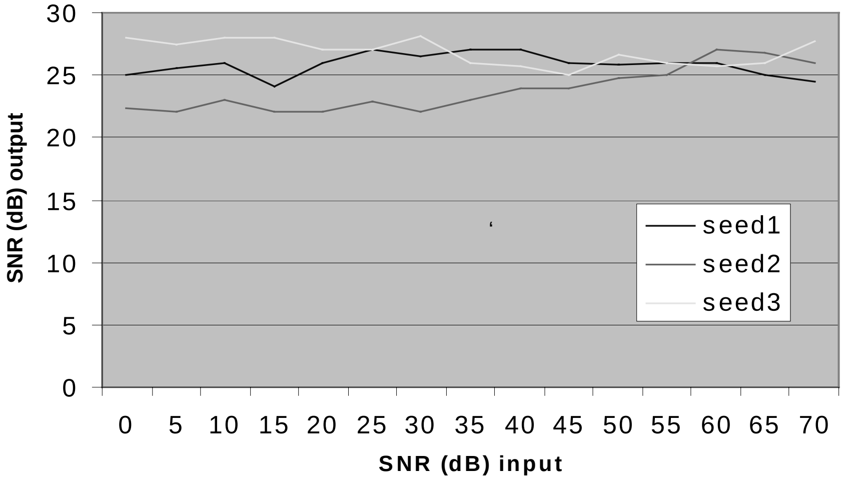

Figure 7: Graph of Signal-to-Noise Ratio input against the output.

Chapter 7
Conclusions and future work

7.1 Conclusions

This dissertation describes the study of the unconstrained photometric stereo, exactly the estimation of the surface normal and the surface reflectance of objects, but the most important of this method is that this estimation is without a priori knowledge of the light-source direction.

After different kinds of test, the conclusion to extract, is that this method is quite reliable and could be used in the estimation of the anterior parameters. But, moreover this method proves that an increase of the image roughness have a negative effect on image estimation.

7.2 Future Work

The unconstrained photometric stereo method presented in this dissertation is oriented to small image, with a few pixels, is due to the limitation of Matlab in computing time. Consequently, an important improvement would be to expand the algorithm to run over another kind of platform, to test the accuracy with more numbers of pixels.

Furthermore, in order to prove the restrictions and the authentic consistency of this technique, we would use it with a real image, since it have only been tested with a simulated images, in any case were real.

Therefore, this research is not still finished, and different improvements should be investigated in a near future.

Appendix A: Matlab source code.

Appendix A.1: Simulate the image.

```
%%  p and q computed with a random with k
clear

%parameters of the image
rows=4      %number of pixels=rows*cols
cols=4
total_frames=6

n_frame=0
n_pixel=1
i=1
j=1
Sreal=0

    %%to build the matrix
slant=45/180*pi
tiltdg=0
tilt=tiltdg/180*pi
signal=0
p_avrg=0
q_avrg=0

    %random to simulate p and q
k=0.1
r=1
randn('seed',1)
p=k*randn(1)
q=k*randn(1)
while (p>r | p<-r) | (q>r | q<-r)
    p=k*randn(1)
    q=k*randn(1)
end
```

```
while n_pixel<=rows*cols
    while n_frame<total_frames
         ixel=-p*cos(tilt)*sin(slant)-
         q*sin(tilt)*sin(slant)+cos(slant)
         I(i,j)=pixel
         j=j+1
         n_frame=n_frame+1
         tiltdg=tiltdg+60
         tilt=tiltdg/180*pi
    end
    temp=[sqrt(p^2+q^2+1)]

    if Sreal==0
         Sreal=[p/temp,q/temp,-1/temp]
    else
         Sreal=[Sreal;p/temp,q/temp,-1/temp]
    end
    %random to simulate p and q
    r=1
    p=k*randn(1)
    q=k*randn(1)
    while (p>r | p<-r) | (q>r | q<-r)
        p=k*randn(1)
        q=k*randn(1)
    end
    slant=45/180*pi
    tiltdg=0
    tilt=tiltdg/180*pi
    n_pixel=n_pixel+1
    i=i+1
    n_frame=0
    j=1
end
```

Appendix A.1.2: Add noise.

```
% to add noise in the simulate image I
n_frame=0
n_pixel=1
i=1
j=1
k=1
randn('seed',1)
```

```
while n_pixel<=rows*cols
    while n_frame<total_frames
         I_noise(i,j)=I(i,j)+randn(1)*k
         j=j+1
         n_frame=n_frame+1
     end
    n_pixel=n_pixel+1
    i=i+1
    n_frame=0
    j=1
end
```

Appendix A.2: Compute statistics values.

```
%%compute the p and q in each pixel
i=1
while i<=rows*cols
    p_and_q(i,1)=sqrt((Sreal(i,1))^2/(Sreal(i,3))^2)
    p_and_q(i,2)=sqrt((1/(Sreal(i,3))^2)-
    ((Sreal(i,1))^2/(Sreal(i,3))^2)-1)
    i=i+1
end

%%compute the p and q average in each pixel
i=1
while i<=rows*cols
    p_avrg=p_avrg+p_and_q(i,1)
    q_avrg=q_avrg+p_and_q(i,2)
    i=i+1
end
p_avrg=p_avrg/(rows*cols)
q_avrg=q_avrg/(rows*cols)

%%compute VAR(p)
i=1
var_p=0
while i<=rows*cols
    var_p=var_p+(p_and_q(i,1)-p_avrg)^2
    i=i+1
end
var_p=var_p/(rows*cols)

%%compute  VAR(q)
i=1
```

```
var_q=0
while i<=rows*cols
    var_q=var_q+(p_and_q(i,2)-q_avrg)^2
    i=i+1
end
var_q=var_q/(rows*cols)
```

%%compute VAR

```
VAR=(var_p+var_q)/2
```

%to compute root mean square error (RMSE)

```
[rows2,cols2]=size(Sreal)
i=1
j=1
res=0
while i<=rows2
    while j<=cols2
         res=res+(Sreal(i,j)-SafterRot(i,j))^2
           j=j+1
    end
    j=1
    i=i+1
end
rmse=sqrt(res/rows2)     %9pxl 3x3 3^2=9=rows
SNR_dB=abs(20*log10(signal/rmse))
```

Appendix A.3: Unconstrained algorithm.

```
[pxls,frms]=size(I)          %simulated image

[U,E,V]=svd(I,0)              %SVD of the image data matrix
```

%we prepare the matrices to compute the matrices $\hat{S}$ and $\hat{L}$

```
Y=V'
Ucut=U(1:pxls,1:3)
Ecut=E(1:3,1:3)
Vcut=Y(1:3,1:frms)

a=sqrt(Ecut(1,1))
```

```
b=sqrt(Ecut(2,2))
c=sqrt(Ecut(3,3))

spos=Ucut*[a,0,0;0,b,0;0,0,c]
sneg=Ucut*-[a,0,0;0,b,0;0,0,c]
spos3x3=spos(1:3,1:3)
sneg3x3=sneg(1:3,1:3)
detpos=det(spos3x3)
detneg=det(sneg3x3)

         %we decide the sign of the solution
         %and we compute the pseudo surface matrix ŝ (s_hat)
         %and the pseudo light-source matrix L̂ (l_hat)

if detpos>0
    s_hat=spos
    l_hat=[a,0,0;0,b,0;0,0,c]*Vcut
elseif detneg>0
    s_hat=sneg
    l_hat=-[a,0,0;0,b,0;0,0,c]*Vcut
end

         %to compute the matrix A
x=l_hat(1,1)
y=l_hat(2,1)
z=l_hat(3,1)
C1=[x^2,2*x*y,2*x*z,y^2,2*y*z,z^2]'
x=l_hat(1,2)
y=l_hat(2,2)
z=l_hat(3,2)
C2=[x^2,2*x*y,2*x*z,y^2,2*y*z,z^2]'
x=l_hat(1,3)
y=l_hat(2,3)
z=l_hat(3,3)
C3=[x^2,2*x*y,2*x*z,y^2,2*y*z,z^2]'
x=l_hat(1,4)
y=l_hat(2,4)
z=l_hat(3,4)
C4=[x^2,2*x*y,2*x*z,y^2,2*y*z,z^2]'
x=l_hat(1,5)
y=l_hat(2,5)
z=l_hat(3,5)
C5=[x^2,2*x*y,2*x*z,y^2,2*y*z,z^2]'
x=l_hat(1,6)
y=l_hat(2,6)
z=l_hat(3,6)
```

```
C6=[x^2,2*x*y,2*x*z,y^2,2*y*z,z^2]'

C=[C1,C2,C3,C4,C5,C6]
D=[1,1,1,1,1,1]
B1=D/C
a=B1(1,1)
b=B1(1,2)
c=B1(1,3)
e=B1(1,4)
f=B1(1,5)
i=B1(1,6)

B=[a,b,c;b,e,f;c,f,i]

[U2,E2,V2]=svd(B,0)

a=sqrt(E2(1,1))
b=sqrt(E2(2,2))
c=sqrt(E2(3,3))

A=U2*[a,0,0;0,b,0;0,0,c]          %matrix A computed

S=s_hat*A                          %surface matrix S
LbeforeRot=inv(A)*l_hat           %light-source matrix L
```

Appendix A.4: Rotation algorithm.

```
%%%%%%%%%%   to compute the L  %%%%%%%%%%%%%%%
 Lfirst=l_hat

  %we let the coordinates of the three points in the left-hand system
 r_l1=[Lfirst(1,1),Lfirst(2,1),Lfirst(3,1)]
 r_l2=[Lfirst(1,2),Lfirst(2,2),Lfirst(3,2)]
 r_l3=[Lfirst(1,3),Lfirst(2,3),Lfirst(3,3)]

 x_l=r_l2-r_l1
 length_x_l=sqrt(x_l(1,1)^2+x_l(1,2)^2+x_l(1,3)^2)
 x_l_hat=x_l/length_x_l
```

```
aux_l=r_l3-r_l1
temp=aux_l(1,1)*x_l_hat(1,1)+aux_l(1,2)*x_l_hat(1,2)+au
x_l(1,3)*x_l_hat(1,3)
y_l=aux_l-temp*x_l_hat
length_y_l=sqrt(y_l(1,1)^2+y_l(1,2)^2+y_l(1,3)^2)
y_l_hat=y_l/length_y_l

z_l_hat=cross(x_l_hat,y_l_hat)
  %we already have the triad in the left-hand system

  %we let the coordinates of the three points in the right-hand system
r_r1=[LbeforeRot(1,1),LbeforeRot(2,1),LbeforeRot(3,1)]
r_r2=[LbeforeRot(1,2),LbeforeRot(2,2),LbeforeRot(3,2)]
r_r3=[LbeforeRot(1,3),LbeforeRot(2,3),LbeforeRot(3,3)]

x_r=r_r2-r_r1
length_x_r=sqrt(x_r(1,1)^2+x_r(1,2)^2+x_r(1,3)^2)
x_r_hat=x_r/length_x_r

aux_r=r_l3-r_l1
temp=aux_r(1,1)*x_r_hat(1,1)+aux_r(1,2)*x_r_hat(1,2)+au
x_r(1,3)*x_r_hat(1,3)
y_r=aux_r-temp*x_r_hat
length_y_r=sqrt(y_r(1,1)^2+y_r(1,2)^2+y_r(1,3)^2)
y_r_hat=y_r/length_y_r

z_r_hat=cross(x_r_hat,y_r_hat)
  %we already have the triad in the right-hand system

   %we construct the matrices $M_l$ and $M_r$
M_l=[x_l_hat',y_l_hat',z_l_hat']
M_r=[x_r_hat',y_r_hat',z_r_hat']

R=M_r*M_l'                        %rotation matrix R

   %finally we find the surface and light-source matrix in an absolute
  %coordinate system
LafterRot=LbeforeRot'*R
LafterRot=LafterRot'


%%%%%%%%%%   to compute the S  %%%%%%%%%%%%%%%
Sfirst=s_hat

  %we let the coordinates of the three points in the left-hand system
r_l1=[Sfirst(1,1),Sfirst(2,1),Sfirst(3,1)]
```

```
r_l2=[Sfirst(1,2),Sfirst(2,2),Sfirst(3,2)]
r_l3=[Sfirst(1,3),Sfirst(2,3),Sfirst(3,3)]

x_l=r_l2-r_l1
length_x_l=sqrt(x_l(1,1)^2+x_l(1,2)^2+x_l(1,3)^2)
x_l_hat=x_l/length_x_l

aux_l=r_l3-r_l1
temp=aux_l(1,1)*x_l_hat(1,1)+aux_l(1,2)*x_l_hat(1,2)+au
x_l(1,3)*x_l_hat(1,3)
y_l=aux_l-temp*x_l_hat
length_y_l=sqrt(y_l(1,1)^2+y_l(1,2)^2+y_l(1,3)^2)
y_l_hat=y_l/length_y_l

z_l_hat=cross(x_l_hat,y_l_hat)
```

%we already have the triad in the left-hand system

%we let the coordinates of the three points in the right-hand system

```
r_r1=[SbeforeRot(1,1),SbeforeRot(2,1),SbeforeRot(3,1)]
r_r2=[SbeforeRot(1,2),SbeforeRot(2,2),SbeforeRot(3,2)]
r_r3=[SbeforeRot(1,3),SbeforeRot(2,3),SbeforeRot(3,3)]

x_r=r_r2-r_r1
length_x_r=sqrt(x_r(1,1)^2+x_r(1,2)^2+x_r(1,3)^2)
x_r_hat=x_r/length_x_r

aux_r=r_l3-r_l1
temp=aux_r(1,1)*x_r_hat(1,1)+aux_r(1,2)*x_r_hat(1,2)+au
x_r(1,3)*x_r_hat(1,3)
y_r=aux_r-temp*x_r_hat
length_y_r=sqrt(y_r(1,1)^2+y_r(1,2)^2+y_r(1,3)^2)
y_r_hat=y_r/length_y_r

z_r_hat=cross(x_r_hat,y_r_hat)
```

%we already have the triad in the right-hand system

%we construct the matrices M_l and M_r

```
M_l=[x_l_hat',y_l_hat',z_l_hat']
M_r=[x_r_hat',y_r_hat',z_r_hat']

R=M_r*M_l'                              %rotation matrix R
```

%finally we find the surface and surface matrix in an absolute
%coordinate system

```
SafterRot=SbeforeRot'*R
```

References

- Barsky01 S.Barsky and M.Petrou, “Colour Photometric Stereo: Simultaneous Reconstruction of Local Gradient and Colour of Rough Textured Surfaces”, 2001.

- Blinn78 J.Blinn, “Simulation of Wrinkled Surface”, *Computer Graphics*, pp. 286-292, 1978.

- Drew87 M.S. Drew, “Photometric Stereo without Multiple Images”, *SPIE*, Vol.3016, pp. 369-380, 1997.

- Frankot88 R.T. Frankot and R.Chellappa, “A Method for Enforcing Integrability in Shape from Shading Algorithms”, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. 10, pp. 439-451, 1988.

- Funt93 B.V.Funt and M.S.Drew, “Color Space Analysis of Mutual Illumination”, *IEEE Trans. Pattern Anal. March. Intell. 15*, pp. 1319-1326, 1993.

- Hayakawa94 H. Hayakawa, “Photometric Stereo Under a Light Source with Arbitrary Motion”, *Optical Society of America*, vol. 11, no. 11, 1994.

- Horn86 B.K.P.Horn, “Robot Vision”, *MIT Press*, 1986.

- Horn87 B.K.P.Horn, “Closed-form Solution of Absolute Orientation using Unit Quaternions”, *J.Opt. Soc. Am. A 4*, pp. 629-642, 1987.

- Horovitz01 I. Horovitz and N. Kiryati, “Bias Correction in Photometric Stereo using Control Points”, *VMV*, 2001.

- McGunnigle98 G.McGunnigle, “The Classification of Texture Surfaces under varying Illumination Direction”, *PhD Tesis*, Dept. of Computing and Electrical Engineering, Heriot-Watt University, 1998.

- Tomasi92 C.Tomasi and T. Kanade, “Shape and Motion from Image Streams Under Orthography: a Factorization Method”, *Int. J. Comput. Vision 9*, pp. 137-154, 1992.

- Sato93 Y.Sato and K.Ikeuchi, “Temporal-color Space Analysis of Reflection”, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (Institute of Electrical and Electronics Engineers, New York), pp. 570-576, 1993.

- Spence02 A. Spence, “The Use of Photometric Stereo-Based Techniques to Capture the Bump Maps of Real Textures for Visualization Applications”, *First Year Report*, Dept. of Computing and Electrical Engineering, Heriot-Watt University, 2002.

- Wu88 Z. Wu and L. Li, “A Line Integration based Method for Depth Recovery from Surface Normals”, *Comput. Vision Graphics Image Process.*, Vol.43, pp. 53-66, 1988.

- Woodham80 R.J.Woodham, “Photometric Method for Determining Surface Orientations from Multiple Images”, *Optical Engineering*, Vol.19, pp.139-144, 1980.