CLASSICAL AND DEEP LEARNING BASED PHOTOMETRIC STEREO TECHNIQUES

A Project Report

submitted by

S.K.BHARATH

in partial fulfilment of requirements
for the award of the dual degree of

BACHELOR OF TECHNOLOGY AND MASTER OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

May 10, 2017

THESIS CERTIFICATE

This is to certify that the thesis titled **CLASSICAL AND DEEP LEARNING BASED PHOTOMETRIC STEREO TECHNIQUES**, submitted by **S.K.BHARATH**, to the Indian Institute of Technology, Madras, for the award of the degree of **Dual Degree(B.Tech and M.Tech) in Electrical Engineering**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. A.N.Rajagopalan
Research Guide
Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 3rd May 2017

ACKNOWLEDGEMENTS

I would like to thank my parents, friends, family for their sacrifices and making positive impact on my life. I had the good fortune of being taught by talented and loving teachers who encouraged me to pursue my dreams. I am grateful to IIT Madras, the institution, that let me explore my passions in fields ranging from sports to robotics. My seniors deserve credit for their mentorship over the past five years. This work would have been impossible without the groundwork laid by Mahesh Mohan, Doctoral Candidate at the IPCV lab. Most importantly,I thank Prof.A.N.Rajagopalan for giving me the opportunity to work at IPCV lab and the freedom he afforded me to explore the process of research.

ABSTRACT

KEYWORDS: Photometric Stereo, SEM, Scanning Electronic Microscopy, Deep Learning, Gradient, SVD, Support Vector Decomposition, Convolutional Neural Networks, CNN

Photometric stereo, which deals with recovering depth from multiple images of an object captured from different lighting directions, is a problem studied extensively over the past few decades. Traditional approaches to solve this problem use priors to overcome the bas-relief ambiguity that arises due to ill-posedness of the problem. In this work, two novel techniques are discussed to avoid the ambiguity. The first technique is a classical approach that is only applicable to scenes containing symmetric objects. The second technique is a deep-learning based extension of the first technique that applies to scenes containing general space. These approaches outperform classical techniques by converting them into a two-step procedure to solve for light source directions and surface normals without ambiguity.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

SEM	Scanning Electron Microsccope
PS	Photometric Stereo
CNN	Convolutional Neural Network

NOTATION

r	Radius, m
α	Angle of SEM detector with x-axis
β	Half Angle of cone formed by SEM detectors and the sample
L	Light Matrix
N	Unit surface normal Matrix
A	Albedo Matrix
λ	Regularization hyper-parameter

CHAPTER 1

NOVEL CLASSICAL PHOTOMETRIC STEREO TECHNIQUE

In this chapter, we present a Photometric Stereo based technique to estimate depth of objects.

1.1 Recovering depth from SEM Images

1.1.1 Scanning Electron Microscope(SEM)

Recovering depth information from images has become hugely significant over the past few decades. Potential applications in the fields of areas ranging from medical imaging to autonomous vehicles has made it an interesting area of research.

Traditional optical microscopy has been developed in the 17th century has led to a vastly improved understanding of the microbial world. The use of visible light to magnify smaller and smaller objects ran into trouble due to the physical limitations relating the wavelength of the objects and the size of the sample being magnified. SEM or Scanning Electron Microscope imaging deals with magnification of objects smaller than a nanometer. SEM shines a beam of electrons on the sample instead of photons as is the case in an optical microscope. The electrons are absorbed and reflected back by the substrate and are captured by the detector. The intensity of the image thus formed is a function of the energy and the density of the electrons reflected.

SEM imaging has since been used widely in the fields of chemistry, physics and molecular biology. The invention of transistor and its miniaturization ever since have made SEM imaging an indispensable tool to study the properties of semiconductors and in their inspection. One of the interesting applications is to understand the topology of the object under inspection. Such an understanding could enable the automated detection and classification of defects in the semiconductor.

The setup of the SEM makes it possible to image the object of interest using electron beams that illuminate the scene. The reflected light is captured by detectors placed in different directions. This makes the shading information in the images under varying lighting conditions an obvious cue to find the depth of the object. This technique is referred to as Photometric Stereo. The amount of noise and the variation of intensity of electron beams however makes the task of finding the depth information a difficult task. Traditional Photometric Stereo techniques fall short in uniquely characterizing depth and the unknown light directions simultaneously because of the ill-posedness of the problem. We aim to overcome that by making assumptions about the symmetry of the objects. The main contribution of this paper is the proposed two-step approach to the problem of uncalibrated Photometric Stereo. In the first step we identify the directions of light source for each image. In the second step, the surface normal(and therefore the depth) is computed. We show that our method removes the ambiguity that other approaches suffer.

1.1.2 Image based depth recovery

Popular techniques

Various cues embedded in the images have been exploited for the purpose of extracting the information about the depth. The classic stereo based depth estimation [1] uses the disparity as a cue to estimate depth. Depth from Defocus [2] extracts depth based on the amount of defocus. It has also been shown that depth can be found from an image under motion blur [3]. In recent years, a notable volume of work has shown the application of deep learning for the purpose of finding depth from just one image in absence of the above mentioned cues. Works [4],[5] and [6] are examples of this approach.

Photometric Stereo

Photometric Stereo is based on the shading information in images under various lighting conditions. When a stationary scene that is of interest is illuminated by various sources of light and assuming a Lambertian surface, the intensity observed by a detector such as a camera sensor is proportional to the dot product between the surface normal and the light direction. The surface normals can then be integrated to find the depth map of

the scene.

1.2 Background Knowledge

1.2.1 Lambertian Reflectance Model

Human vision is based on perception of light reflected off objects. It is hence important to study the behavior of objects under incident light. Many objects around us do not vary in their intensity as we look at them from various angles. This is because these objects reflect light in all directions. This type of reflection is referred to as diffuse reflection. Lambertian reflectance model is a popular model used to explain diffuse reflection.

Lambertian reflectance model assumes that the intensity of an object I under a light source is proportional to the dot-product of the surface normal$\hat{n}$ of the object and the light vector L. The magnitude of the light vector denotes the intensity of the light source and the direction indicates the direction of incident light. Further, according to the model, the intensity observed is independent of where the sensor is positioned. The constant of proportionality in the relation described is known as albedo a and is a property of the material. Albedo can be understood as the amount of incident light that a surface reflects. Note that dot-product can be negative when the angle between incident light and surface normal is obtuse. This doesn't make physical sense since intensity observed by a sensor cannot be negative. Eq 1 is mathematical representation of the lambertian model

$$I = max(a(\hat{n}.\vec{l}), 0) \tag{1.1}$$

We often omit the non-linearity introduced by introducing a comparison operation for the sake of ease.

While a surface that adheres to the lambertian assumption doesn't exhibit directional preference as it reflects light, some surfaces reflect light predominantly in a certain direction. Such surfaces create specular artifacts in the images. Fig 1.1 contrasts the two phenomena

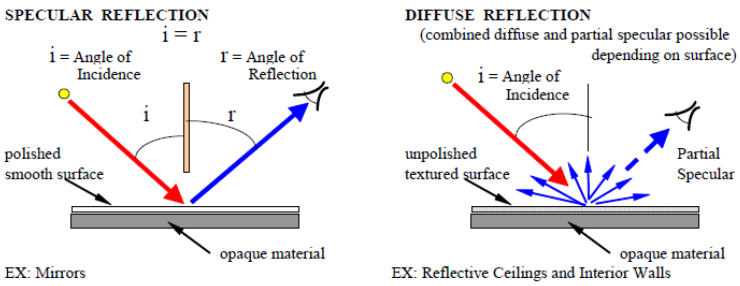

Figure 1.1: Lambertian vs Specular reflection

1.2.2 Photometric Stereo Setup

Mathematical Formulation

The discussion on behavior of objects under incident light makes it clear that a same object behaves differently when light is incident on it from different directions. Photometric Stereo exploits this difference in behavior to estimate the surface normals and hence the depth.

A typical Photometric Stereo setup involves multiple light sources, a detector(camera) and the object. We use the matrix L of order $3 \times k$ to represent the light source direction where k is the number of light sources. $i = 1, 2, 3..., k$ is used as an index to refer to individual light sources. The object is imaged by the detector under the illumination of each light source. The matrices N and A of orders $p \times 3$ and $p \times 1$ respectively represents the matrix of unit surface normals and albedo respectively for each of the p pixels in the image. Finally, I is the $p \times k$ matrix used to denote the matrix of intensities of each pixel under each light source. Index $j = 1, 2, 3..., p$ is used to refer to individual pixels. It is assumed that the object and detector are stationary during the course of experiment so as not to introduce motion blur in the images. Under the Lambertian assumption, we have:

$$I = A \odot (N \cdot L) \tag{1.2}$$

The equation is further simplified by merging the element wise product between albedo and unit surface normals. M is used to represent this new matrix. $M = A \odot N$.

Eq 2 now simplifies into:

$$I = M \cdot L \tag{1.3}$$

where

$$M = A \odot N \tag{1.4}$$

the element wise product is taken across x,y,z components of the normal for each pixel.

Uncalibrated and Calibrated Photometric Stereo

From the discussion above, Photometric Stereo boils down to studying techniques to extract the matrix M given matrix I. Photometric Stereo is broadly classified into calibrated and uncalibrated Photometric Stereo. Under Uncalibrated Photometric Stereo, the Light Source Matrix L is assumed to be unknown. Under Calibrated Photometric Stereo, it assumed that the Light Source Matrix L is known a priori. Common sense suggests that Calibrated Photometric Stereo is a problem much simpler to solve than the problem of Uncalibrated Photometric Stereo. We study the problem of Uncalibrated Photometric Stereo for SEM Images in this work.

1.2.3 SEM Imaging Setup

SEM Images are formed by reflected electrons rather than reflected photons. However, we can apply the same principles used for images formed by reflected photons. A highly simplified version of SEM Imaging Setup is shown in fig 1.2 . In case of SEM Imaging, the equivalent of Light Vectors lie along the surface of a cone. L is in this case a 4×3 matrix. The light matrix can be characterized by just two angles α and β as shown in the fig 1.3 . β is the half angle of the cone and α is the position w.r.t x-axis.

The light matrix in case of a SEM Imaging setup boils down to 1.3. Moreover, the

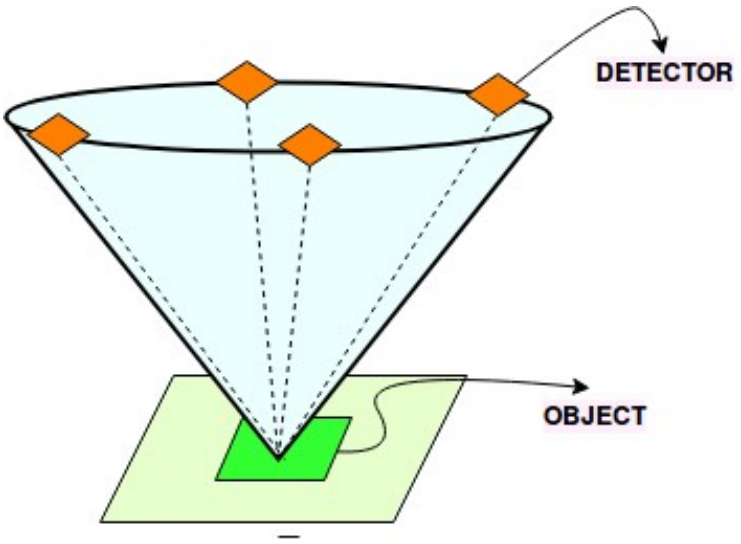

Figure 1.2: Simplified SEM Imaging Setup

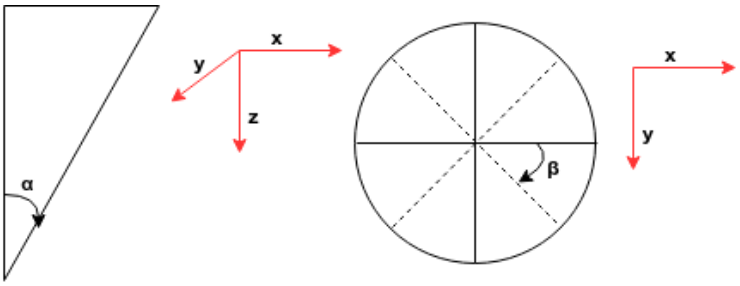

Figure 1.3: α and β parameters

parameter β is known. So the only unknown in the light matrix is the parameter α.

$$L = \begin{bmatrix} cos(\alpha)sin(\beta) & sin(\alpha)sin(\beta) & -cos(\alpha)sin(\beta) & -sin(\alpha)sin(\beta) \\ sin(\alpha)sin(\beta) & -cos(\alpha)sin(\beta) & -sin(\alpha)sin(\beta) & cos(\alpha)sin(\beta) \\ cos(\beta) & cos(\beta) & cos(\beta) & cos(\beta) \end{bmatrix} \tag{1.5}$$

1.3 Previous Work

We discuss some of the most popular popular approaches to solve the problem of Photometric Stereo in this section. Broadly speaking, the techniques can be divided into Classical and Learning based techniques. We will introduce classical techniques in this section and delay the discussion on Learning based techniques till the following chapter. Note that the discussion is limited to Uncalibrated Photometric Stereo under Lambertian assumption.

1.3.1 Classical Methods

Classical techniques use techniques of linear algebra to solve the problem of Photometric Stereo. 1.3 shows the relationship between matrices I, M and L. Remember that the matrix M is a element wise product between albedo matrix A and unit surface normal matrix N. This implies that once M is estimated, A is the magnitude of each row of M and N is the unit vector corresponding to each row of M.

$$A_j = ||M_j||_2 \tag{1.6}$$

$$N_j = \frac{M_j}{||M_j||_2} \tag{1.7}$$

where j is the index used to identify a pixel as described in section 1.2.2

Singular Value Decomposition

Observe that the matrices M and L are of rank 3. This implies that according to eqn 1.3, matrix I is also of rank 3. But the rank of matrix M is never strictly equal to 3 in practice because of added noise and other non-linearities.

This property of rank of matrices M and L is exploited in many classical techniques. Singular Value Decomposition(SVD) is a technique used to factorize a matrix and helps us find the best low rank estimate of a matrix. SVD decomposes any matrix A into three matrices U, S and V such that

$$A = U \cdot S \cdot V^T \tag{1.8}$$

The columns of matrix U are the eigenvectors of matrix AA^T. The columns of matrix V are the eigenvectors of matrix A^TA. The matrix S is a diagonal matrix whose entries are referred to as singular values of the matrix A. Further, singular values are the square root of eigenvalues of AA^T.

Low rank estimate of a matrix

As mentioned above, in practice, the rank of the matrix I is never equal to 3. So, in order to enforce lambertian reflectance assumption, we find a matrix $\bar{I}$ that is close to I but is of rank 3. Eckart-Young theorem shows that the $\bar{I}$ is obtained by retaining only the top-3 singular values of I and zeroing out the lower singular values. In other words, without loss of generality, if the diagonal elements of S are in a descending order, then

$$\bar{I} = \sum_{i=1}^{3} S_{ii} U_i \otimes V_i \tag{1.9}$$

where U_i and V_i are the i^{th} columns of matrices U and V respectively. $\otimes$ represents the outer-product of the two column vectors. S_{ii} denotes the i^{th} largest singular value of the matrix I.

Let us define a matrix $\bar{S}$ which is identical to S except for all singular values other the top-3 singular values being replaced with zeros. Similarly, we define matrices $\bar{U}$ and $\bar{V}$ of orders $P \times 3$ and $3 \times n$ that contain just the first three columns of matrices U

and V. This means that

$$\bar{I} = \bar{U} \cdot \bar{S} \cdot \bar{V}^T \tag{1.10}$$

Classical techniques

The matrix $\bar{I}$ is now decomposed into matrices M and L that are unknown such that

$$\bar{I} = M \cdot L \tag{1.11}$$

Equations 1.10 and 1.11 imply that

$$M \cdot L = \bar{U} \cdot \bar{S} \cdot \bar{V}^T \tag{1.12}$$

Consider two 3×3 matrices C and D. From Eqn 1.12, we can say that

$$M = \bar{U} \cdot C \qquad L = D \cdot \bar{V}^T \qquad s.t \qquad C \cdot D = \bar{S} \tag{1.13}$$

Now the problem is simplified into solving for matrices C and D rather than solving for much larger unknown matrices M and L. The constraint connecting C and D is the equation $C \cdot D = \bar{S}$.

Obviously, there are infinitely many ways to construct two matrices whose product equals a diagonal square matrix $\bar{S}$. An obvious solution to this problem lies in imposing additional constraints by using reasonable priors.

In [7], Yuille et al, use integrability constraint to solve for matrices C. While this doesn't uniquely identify the matrix C, it reduces the number of unknowns from 9 to 3. This transformation to with C can be reduced to is called Generalized Bass Relief(GBR) ambiguity. The ambiguity matrix G is given by

$$G = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ \nu & \tau & \lambda \end{bmatrix} \tag{1.14}$$

G is then solved for assuming some prior knowledge on the direction of light sources.

In [8], QuÃl'au et al propose a prior on Total Variation of depth. This translates to imposing a smoothness prior on the depth map of the object. In [9], Alldrin et al. impose a prior over the entropy of albedo to eliminate the ambiguity.

1.4 Our Method

1.4.1 Problems with previous works

As pointed out in section 1.2.2, calibrated Photometric Stereo is a much simpler problem to solve. If we know the matrix L a priori, the system of over constrained equations given by $I = M \cdot L$ can be solved to find the matrix M. In absence of any knowledge of matrix L, we are forced to use prior information to find matrices M and L.

Each of the priors described above rely on vaguely defined rules formed by inspection and observation. The Total Variation prior is a product of observation that in most images, depth doesn't vary abruptly too often. We give mathematical meaning to these vaguely defined rules in the form of priors. But a prior doesn't necessarily result in accurate results in all scenarios. In our case, it is possible to construct an object or a scene that leads to poor results when Total Variation prior technique is used.

In this work, we discuss and develop exact techniques for the problem of Uncalibrated Photometric Stereo for SEM Imaging setup without using any priors under certain assumptions discussed in sections that follow.

1.4.2 Our approach

Our work hinges on converting the Uncalibrated Photometric Stereo problem into a two step process. In the first step, we find the light matrix L without ambiguity. In the second step we simply solve the over-constrained set of equations $I = M \cdot L$ to estimate M and hence N. Figure 1.4 represents the difference in approach between classical techniques and our novel method.

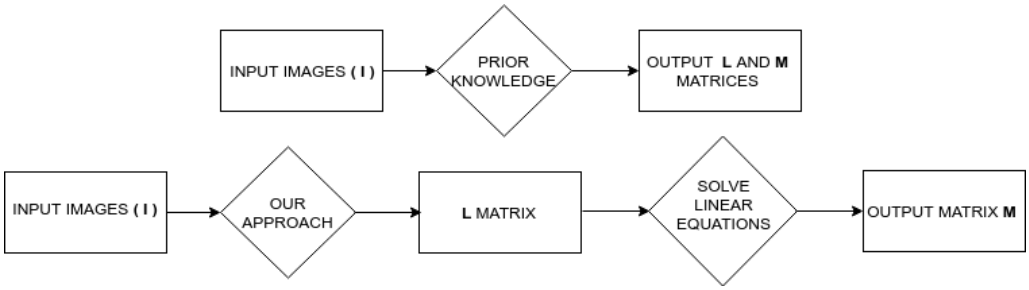

Figure 1.4: Classical approach and Our Approach

1.4.3 Assumptions

In the work, we assume that the objects in the figure are limited to symmetric objects. Specifically, we assume that the objects in the scene are either flat surfaces or spherical surfaces. Under these assumptions, the light matrix L can be estimated accurately without any need for priors. Fig 1.5 shows a set of 4 images with multiple spheres that are imaged under lighting conditions similar to those in SEM Imaging setup discussed previously.

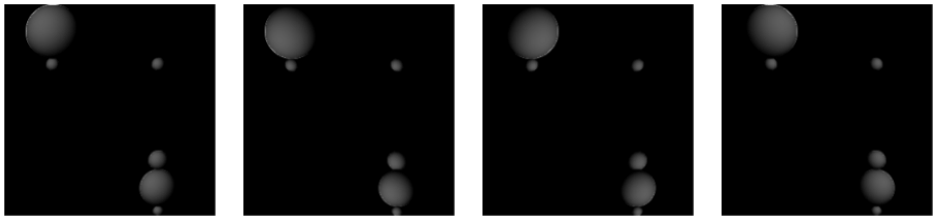

Figure 1.5: A sample of 4 synthetic images of spheres

1.4.4 Mathematical formulation

Consider a light vector L_j corresponding to the j^{th} light source. Each light vector is a column vector that has x,y,z components.

$$L_j = \left[L_j^x, L_j^y, L_j^z\right] \tag{1.15}$$

From eq 1.15, we see that the light direction can be decomposed into L_j^x , L_j^y and L_j^z. In the special case of SEM imaging, L_j^z which indicates the cosine of angle β is known as mentioned in section 1.2.3. So, if we find the other two components of light directions, L_j^x and L_j^y ,the light vector direction is fully determined.

The symmetry of the objects under consideration makes the computation of L_j^x and L_j^y extremely easy. As described in [12], light direction can be identified for a sphere without any prior knowledge about it's position or radius. Without loss of generality, consider a sphere centered at along the axis of camera that is illuminated by a light source with light vector L_j. Fig 1.6 shows such a sphere and Fig 3(b) shows the gradient vectors in x and y directions. Simple observation reveals that these vectors are distributed symmetrically w.r.t the light vector's direction. So, the average of all the gradient vectors over the image will point in the direction of the 2d-vector $\left[L_j^x, L_j^y\right]$.

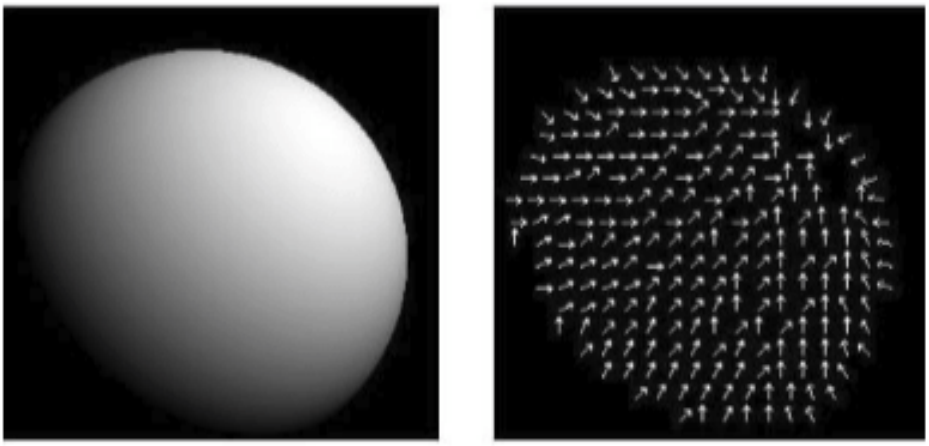

Figure 1.6: Image of a sphere and the image gradient vectors

Intuition and arguments based on symmetry should convince this simple relationship between direction of light vector and the average of gradients. Refer [12] for the formal proof.

Images span two physical dimensions. But we used I_j to denote a row matrix containing the image intensities. In other words, I_j is formed by reshaping the image into a one dimensional matrix. Let $\mathcal{I}_j$ denote the two dimensional version of the row vector I_j. $\mathcal{G}_j$ be the gradient of the image $\mathcal{I}_j$ captured using the source j. The relationship between light direction and the gradients can be described using equations 1.16 and 1.17. Eq 1.16 describes the definition of gradient while eq 1.17 describes the light vector components L_j^x and L_j^y as a function of $\mathcal{G}_j$.

$$\mathcal{G}_j = \nabla \mathcal{I}_j \tag{1.16}$$

$$[L_j^x, L_j^y] = sin(\beta) \times \frac{\sum_{i=1}^{p} \mathcal{G}_j}{||\sum_{i=1}^{p} \mathcal{G}_j||_2} \tag{1.17}$$

Here the sum runs over gradients at all p pixels. The sum of gradients in the numerator is divided by the magnitude of gradient in the denominator in order to normalize. The normalized sum of gradients is then multiplied with $sin(\beta)$ in accordance with light vectors in a SEM Setup as described in section 1.2.3.

With this the entire light vector L_j of the j^{th} light source is fully determined. The same process is repeated for all light sources to construct the light matrix L

$$[L_j^x, L_j^y, L_j^z] = \left[sin(\beta) \times \frac{\sum_{i=1}^{p} \mathcal{G}_j}{||\sum_{i=1}^{p} \mathcal{G}_j||_2}, cos(\beta) \right] \tag{1.18}$$

Once we identify the light matrix, we turn to the problem of finding the matrix M. The relationship between intensity, surface normals and the light direction is given by eq 1.2.and more concisely by equations 1.3 and 1.4. The matrix M is computed by solving the system of linear equations $I = M \cdot L$. If L^+ denotes the pseudo inverse of the matrix L, then M is given by:

$$M = I \cdot L^+ \tag{1.19}$$

1.5 Results

1.5.1 Robustness to Noise

SEM Images are formed by enhancing faint signals of reflected electrons. Hence, SEM Images are characterized by high noise levels and hence low SNR. In this section, we study the effect of noise on the calculation of light source directions using our method.

The computation of light vector direction using average of gradients makes it robust to noise assuming that the noise is symmetrically distributed about it's mean. In fig 1.7 we show synthetically generated images with varying levels of noise.

We computed the light vectors in images containing spheres using our method and

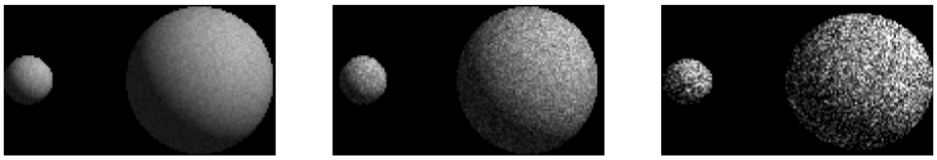

Figure 1.7: Images of spheres with increasing levels of noise

TV-prior method. We steadily increase noise level and compute the angular deviation θ between true light direction $\hat{l}_{true}$ and computed light direction $\hat{l}_{calc}$. Our method gives robust results compared to TV-prior based method. Eq 18 describes the computation of θ

$$\theta = cos^{-1}(\hat{l}_{true} \cdot \hat{l}_{calc}) \tag{1.20}$$

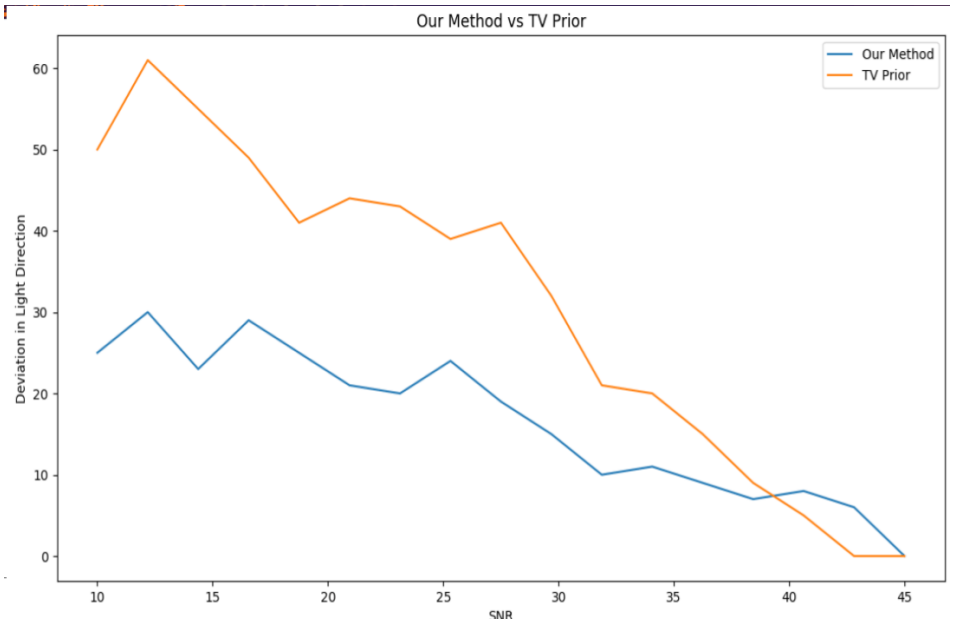

Figure 1.8: Angular deviation between ground truth and computed vector

The height of objects in the scene is an important parameter used in semiconductor verification process to detect and classify manufacturing defects. We analyze the ratio of radii of two spheres under increasing levels of noise. Our method gives superior results compared to TV-prior based method as seen in fig 1.9 .

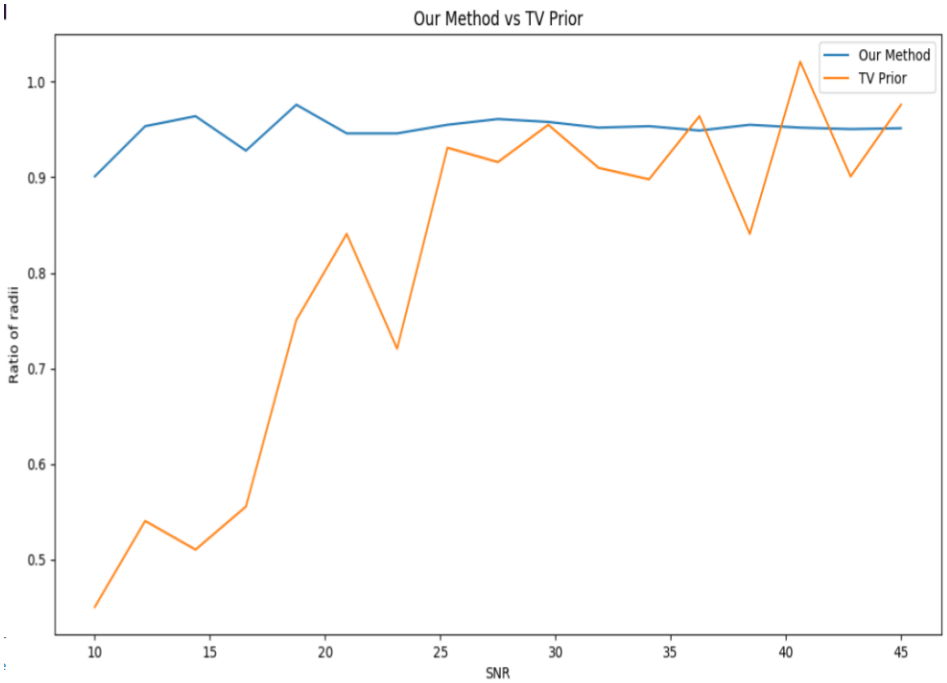

Figure 1.9: Ratio of radii for our method and TV-Prior based method

1.5.2 Results on the SEM Dataset

Here we present results of our method on actual SEM Dataset. The four images from the SEM detectors are shown in fig 1.10 .

The output of our method is shown in fig 1.11 .

It is easy to see that the spheres in the input image closely correspond to the spheres in the depth map. This demonstrates the effectiveness of our method in images with multiple spheres.

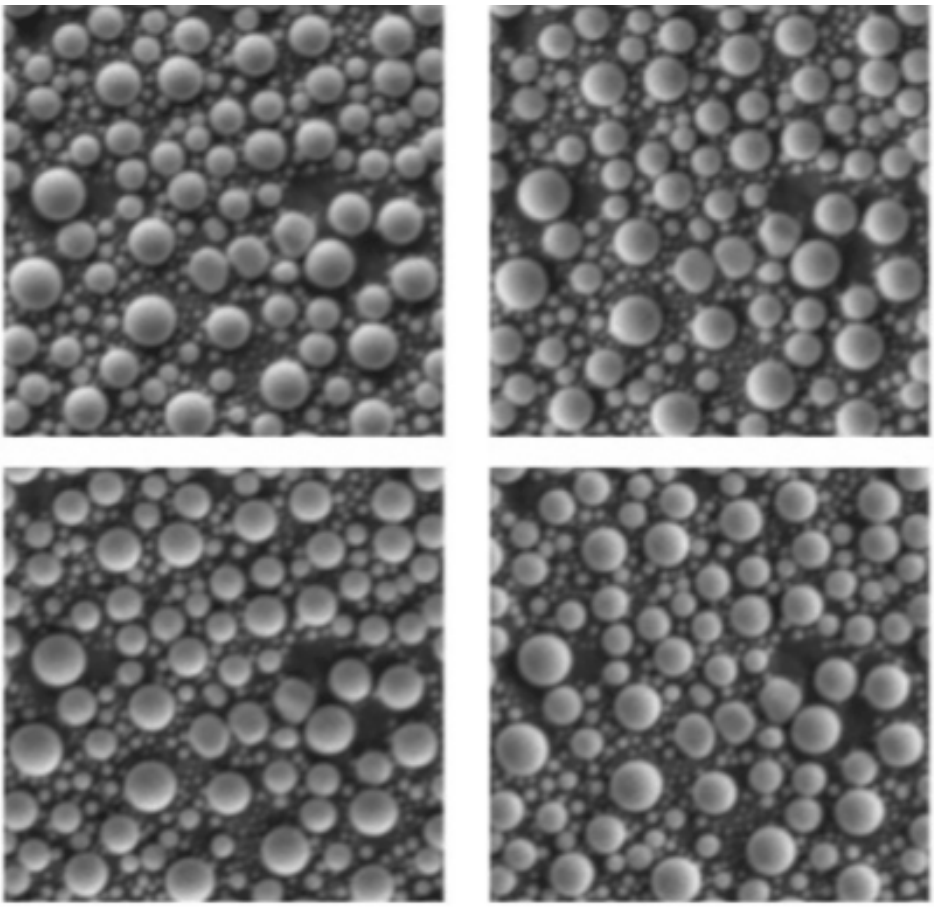

Figure 1.10: Data from SEM containing spheres

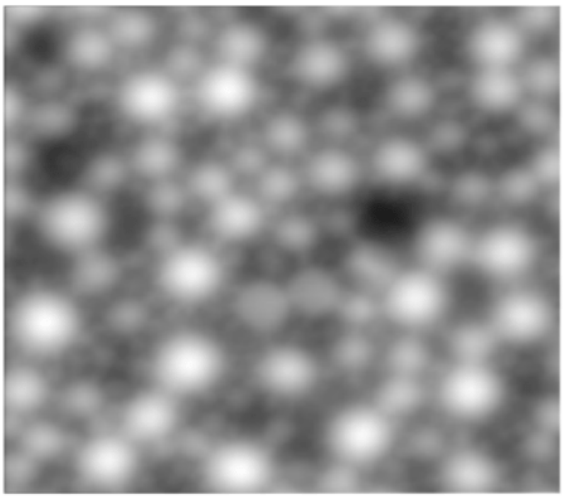

Figure 1.11: Normalized depth map from our method

CHAPTER 2

NOVEL DEEP LEARNING BASED PHOTOMETRIC STEREO TECHNIQUES

In this chapter, we present the second part of the thesis, a method to solve the problem of uncalibrated photometric stereo using our Deep Learning based technique.

2.1 Background Knowledge

The discussion on lambertian reflection, Photometric Stereo and SEM Imaging setup in sections 1.2.1 , 1.2.2 and 1.2.3 are worth revisiting as we proceed with our discussion on Deep Learning based techniques.

2.1.1 Machine Learning

Machine Learning is the study of techniques that leverage data to give machines the ability to make decisions rather than through hard coded rules. A machine learning system is first trained using available data. The trained system is then tested on previously unseen examples known as test data. Machine Learning systems are prone to over performing on seen data and under performing on unseen data. This problem is known as over-fitting. The aim of the system is to produce good results on unseen data. The goal of machine learning is generalization over unseen data.

A typical machine learning system involves a vector of inputs $\mathbf{x_i}$ and a vector of outputs $\mathbf{y_i}$ corresponding to the input $\mathbf{x_i}$. The goal is to estimate the transformation or the function $\mathbf{f} : \mathbf{x} \rightarrow \mathbf{y}$ corresponding to the system that transforms input $\mathbf{x_i}$ to input $\mathbf{y_i}$.

Obviously the function of the underlying system that converts $\mathbf{x_i}$ to $\mathbf{y_i}$ is unknown. Hence, we perform a guided search in the space of a class of functions to get closer to the true function. One example of such class of functions could be a function that is

linear in the input feature vector $\mathbf{x_i}$. Such classes of functions are characterized by set of weights $\mathbf{w}$. So the estimated output $\mathbf{y_i^{est}}$ is given by

$$\mathbf{y_i^{est}} = f(\mathbf{x_i}, \mathbf{w}) \tag{2.1}$$

The unknown set of weights $\mathbf{w}$ characterize the function. We define a loss function $L(w)$ that tells us how well our function behaves on unseen data. The best estimate of $\mathbf{w}$ is the weights that correspond to lowest possible loss. In other words,

$$w_{optimal} = \min_{w} L(w) \tag{2.2}$$

A good choice of $L(w)$ is the mean squared error between estimated output y_i^{est} and true output y_i.

$$L(w) = \sum_{x} p(x)||y_{est} - y||_2^2 \tag{2.3}$$

or

$$L(w) = \sum_{x} p(x)||f(w, x) - y||_2^2 \tag{2.4}$$

where $p(\mathbf{x})$ is the probability that $\mathbf{x}$ is the input. The sum is over all possible $\mathbf{x}$. This formulation of $L(\mathbf{w})$ is problematic since we do not know output $\mathbf{y}$ for all possible $\mathbf{x}$. A more practical formulation is to restrict the summation only to seen or training examples. This formulation results in:

$$L(w) = \sum_{i=1}^{i=n} ||f(w, x_i) - y_i||_2^2 \tag{2.5}$$

While this problem is better that formulation given by eqn 2.4 , it is skewed to produce excellent results on training data and that need not necessarily generalize to unseen data. This could be because out function f agrees with outliers in training examples. A characteristic of functions that agree with outliers is that some of their weights $\mathbf{w}$ are very large in magnitude. This is plausible because outliers lie far away

from the bulk of the data. In order for our function to stay closer to the outliers, some of the weights need to be abnormally large. This suggests that a good way to avoid over-fitting is by introducing a new term into our loss function $L(w)$ that discourages large weights. Our modified loss function becomes:

$$L(w) = \sum_{i=1}^{i=n} ||f(w, x_i) - y_i||_2^2 + \lambda \times ||w||_2^2 \tag{2.6}$$

where λ is a hyper-parameter that decides the balance between competing interests.

$$L(w) = \underbrace{\sum_{i=1}^{i=n} ||f(w, x_i) - y_i||_2^2}_{\text{data loss}} + \underbrace{\lambda \times ||w||_2^2}_{\text{regularization loss}} \tag{2.7}$$

There are various choices for the regularization loss term. L2, L1 and a combination of both have been studied.

The optimal weight vector $\mathbf{w}$ is obtained by minimizing the loss function given by eq 2.7 . We begin with randomly initialized weights and we iteratively reduce the loss function. This is accomplished by various methods. The most commonly used method is gradient descent. We move the weight vector in the direction opposite to the direction of gradient of $L(\mathbf{w})$ w.r.t $\mathbf{w}$,i.e, $\nabla_{\mathbf{w}} L(\mathbf{w})$.

$$\mathbf{w_t} \leftarrow \mathbf{w_{t-1}} - \alpha \times \nabla_{\mathbf{w}_{t-1}} L(\mathbf{w}_{t-1}) \tag{2.8}$$

α is the learning rate.

2.1.2 Deep Learning

Machine learning has had great successes in producing sate-of-the-art results in various fields such as as risk management, biometrics, medicine, computer vision etc. But very often, raw data from systems is unsuitable for training by machine learning algorithms. Data pre-processing is often needed to pick relevant features and discard the rest. It is often very unclear which features to include and which ones to discard. This is particularly a problem for image data.

Deep Learning addresses the problem by turning the machine learning system into a hierarchy of layers. The initial layers process the data and pick out useful features that are complex transformations of input vector $\mathbf{x}$. The later layers combine these useful features to estimate the output $\mathbf{y}$. Fig 2.1 shows the hierarchy. The layers in-between are known as hidden layers. Each hidden layer contains a number of nodes known as neurons.

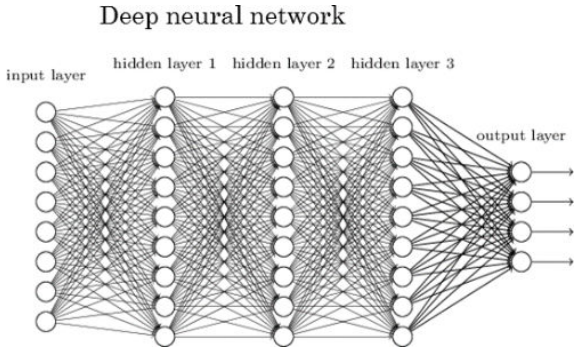

Figure 2.1: Deep Learning framework *Source:Internet*

Each neuron computes a non-linear transformation of input vector to produce the output. Each neuron is associated with a set of weights $\mathbf{w}$ and a bias b. The output of the i^{th} neuron is given by:

$$y_i = f(\mathbf{w} \cdot \mathbf{x}^T + b) \tag{2.9}$$

The function f is non-linear. Popular choices of f are sigmoid, tanh and ReLU as shown in equations 2.10, 2.11 and 2.12 respectively.

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.10}$$

$$f(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{2.11}$$

$$f(x) = \max(0, x) \tag{2.12}$$

If the problem is a regression problem, we use a loss function similar to the one we used in classic Machine Learning techniques as described in eqn 2.7 . But if the

problem is a classification problem, we use sigmoid cross-entropy loss to compute the loss. Sigmoid is a function used to convert the output at the last layer to probabilities of input belonging to each class. If y_i where $i = 1, 2, 3, ...n$ is the output of i^{th} output node, then the probability that input $\mathbf{x}$ belongs to class i is given by

$$P(i|\mathbf{x}) = \frac{e^{y_i}}{\sum_{j=1}^{n} e^{y_i}} \tag{2.13}$$

and the cross-entropy loss is defined as:

$$L(w) = \frac{1}{N} \times \sum_{i=1}^{n} (y_i \times y_i^{true} + (1 - y_i) \times (1 - y_i^{true}) \tag{2.14}$$

Just like in case of classic machine learning techniques, we compute the gradient of loss w.r.t each unknown weight and bias to update the weights.

$$\mathbf{w_t} \leftarrow \mathbf{w_{t-1}} - \alpha \times \nabla_{\mathbf{w}_{t-1}} L(\mathbf{w}_{t-1}) \tag{2.15}$$

But computation of gradient $\nabla_{\mathbf{w}} L(\mathbf{w})$ is much harder because of the depth of the network. It is impractical to write down closed form expressions for the gradient. However, we can exploit the hierarchal nature of the network to simplify the computation of gradients. The gradients of loss w.r.t weights at one layer are a function of gradients at the next layer. This algorithm, known as the back propagation algorithm helps us update weights effectively.

One disadvantage of Deep Networks in comparison with traditional techniques is the large number of weights involved. Networks typically have millions of parameters to be trained which is rarely the case in traditional techniques. However, this potential drawback is overcome in recent years by the availability of large datasets that help us train deep networks effectively. Moreover, deep networks are a lot slower compared to traditional algorithms which usually involve a simple dot product followed by a non-linearity. This problem is overcome by massive parallelization using GPUs.

Convolutional Neural Networks

Convolutional Neural Networks or CNNs are a special modification of Deep Networks that have broken records in performing several image processing tasks such as object detection, semantic segmentation, depth estimation etc. CNNs are a modified version of Deep Networks that make use of the fact that features in images are local. Local structures combine to produce higher order patterns. For example, different parts of the face are positioned in a certain pattern to produce a human face. In Deep Networks, each output node is a function of each of its input node. But these connections are modified into local connections in a CNN. Moreover the weights involved are same for all input patches at a node. This property is known as weight-sharing. The fig 2.2 .

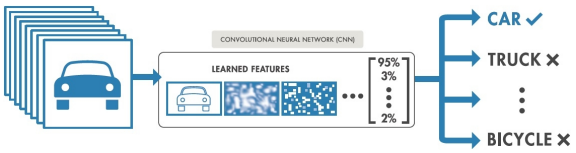

Figure 2.2: CNN framework *Source:Internet*

Pooling layers are used to reduce the size of input image and introduce translational invariance. The last few layers are fully connected layers like in the case of Deep Learning framework. The back-propagation algorithm used for Deep Learning algorithms is slightly modified for CNNs.

2.2 Previous Work

Santo et al. [10] describe a solution to compute surface normals in a per-pixel fashion. While this deals with the problem of cast shadows in a novel way, it assumes that the light- directions are known a priori making it unsuitable for our application. Tang et al. [11] uses a generative model to describe the distribution of surface normals of human-face data and is of little use in SEM based applications with widely varying object shapes unlike in case of human face data.

2.3 Our Approach

Our approach in Chapter 1 is only applicable to the case of symmetric objects such as spheres. Can we predict accurately, the directions of light source if the objects in the scene are not symmetric? As shown in equations 1.17 and 1.20, the light direction boils down to the direction of sum of gradients in the image under assumptions of symmetry. In case of non symmetric objects, it is plausible to imagine that the function relating the direction of light source L_j with the image I_j is far more complex than sum of gradients. We use a CNN to estimate this complex function f.

$$L_j = f(I_j) \tag{2.16}$$

2.3.1 Mathematical Formulation

The input to our network is an image of a scene that is captured under a light source. The output is the light vector that illuminates the object. This is shown in fig 2.3 .

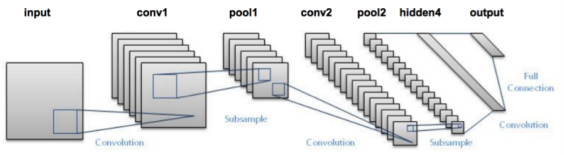

Figure 2.3: Our framework *Source:Internet*

The details of various parameters of the network are summarized in table 2.1 .

Table 2.1: Network Structure and Parameters

Input Data	Synthetically Generated
Activation	ReLU
No of Conv Layers	7
No of FC Layers	3
No of weights	10 mi
Loss	L2 loss with L2 regularization
Learning	Adam Optimizer

2.4 Results

We used downloaded data from Staford's Photometric Stereo Database. We relighted the objects with lighting from randomly chosen directions to generate test data. This is a regression problem. So the loss function is given by:

$$L(w) = \sum_{i=1}^{N} ||\vec{l}_i^{est} - \vec{l}_i^{true}||_2^2 \tag{2.17}$$

Some of the input images to the network are shown below.

Figure 2.4: Synthetic Data relighted from different directions

The training is performed with parameters mentioned in table 2.1 and drop-out regularization with probability of 0.5. The networks trains very well producing excellent results on test data. The average angular variation between $\vec{l}_i^{est}$ and $\vec{l}_i^{true}$ is about 0.5^o. The training and test loss at different iterations during training process are shown in fig 2.5 .

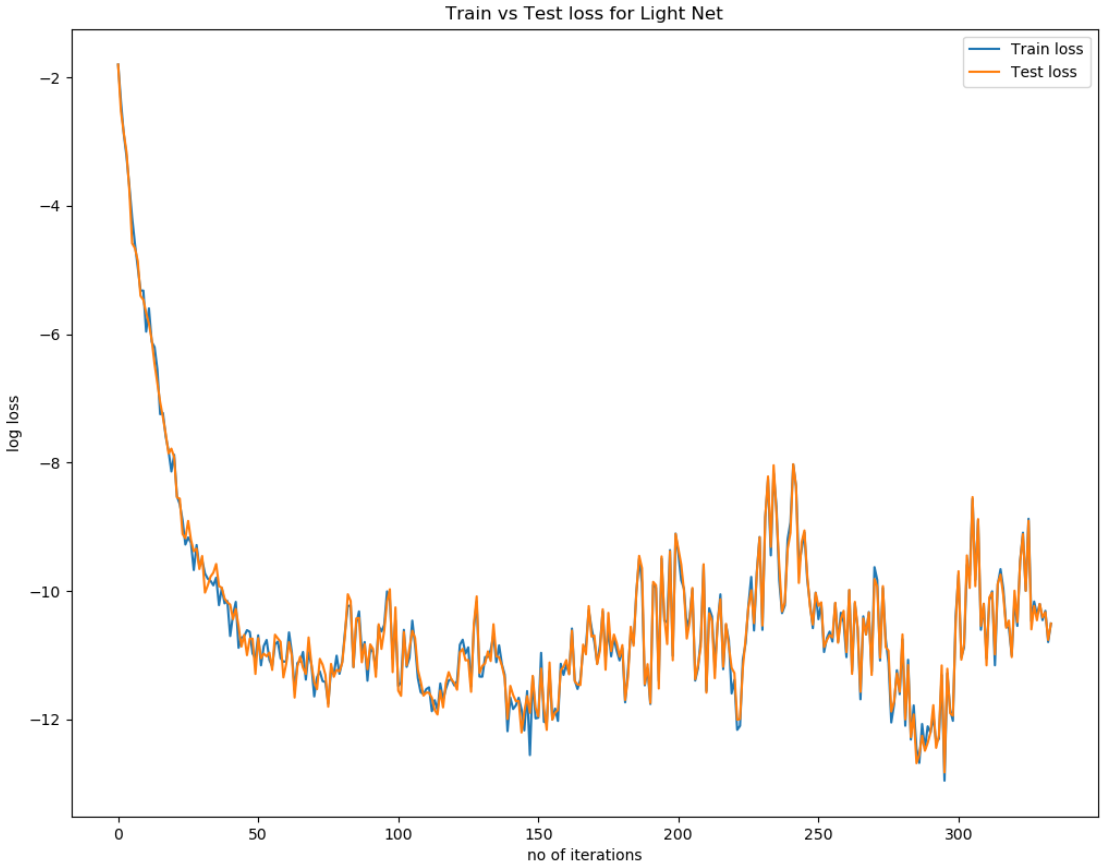

Figure 2.5: Training and Test loss

CHAPTER 3

Future work

The work shown in this chapter cannot be applied to SEM Images directly since the ground truth light directions is not known. A way forward to use the power of deep learning in the case of SEM images is to use semi-supervised learning approach. Such an approach would have to simultaneously estimate matrices L and M and use reasonable priors to propagate the loss. The framework for such a system is shown in figure 2.6 .

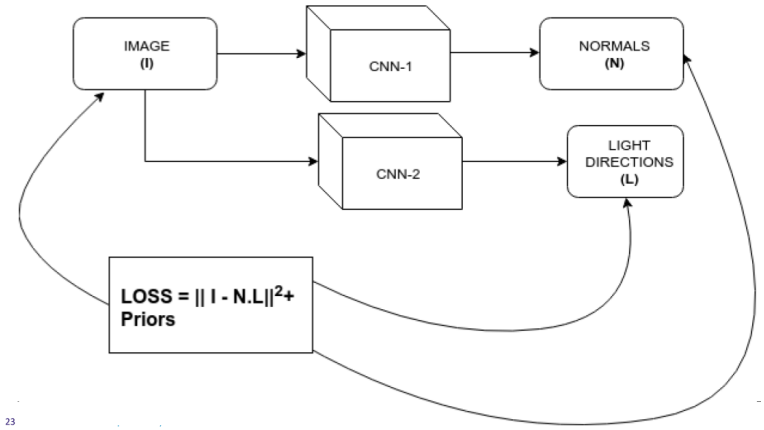

Figure 3.1: A framework for semi-supervised learning

References

1. Lucas and Kanade An iterative image registration technique with an application to stereo vision *IJCAI'81*,Volume-2, Pages 674-679, (1981)

2. Subbarao and Surya Depth from defocus: A spacial domain Approach *IJCV*,Volume 13, Isuue 3,pp 271-294, (1992).

3. Paramanand and Rajagopalan Depth From Motion and Optical Blur with an unscented kalman filter *IEEE transactions on image processing*,vol.21, no. 5, (2012).

4. Saxena Sun and Andrew Make3D: Learning 3D Scene Structure from a Single Still Image *IEEE Transactions on PAMI*,Volume: 31, Issue :5 ,(2009).

5. Eigen, Puhrsch, and Fergus Depth map prediction from a single image using a multi-scale deep network. *NIPS*, (2014).

6. Garg, Kumar, Carniero and Reid Unsupervised CNN for Single ViewDepth Estimation: Geometry to the Rescue *ECCV*, Lecture Notes in Computer Science, vol 9912. Springer, Cham(2016).

7. Yuille and Snow Shape and Albedo from Multiple Images using Integrability *CVPR*, (1997).

8. Queau, Lauze and Durou Solving the Uncalibrated Photometric Stereo Problem Using Total Variation , ().

9. Alldrin, Mallick and Kriegman Resolving the Generalized Bas-Relief Ambiguity by Entropy Minimization CVPR, (2007).

10. Santo, Samejima,Sugano, Shi and Matshushita Deep Photometric Stereo Network *ICCV Workshop*, (2017).

11. Tang, Salakhatdinov and Hinton Deep Lambertian Networks *arXiv:1206.6445 [cs.CV]*.

12. Dosselman and Yang Determining Light Direction in Spheres using Average Gradient *Technical Report TR-CS*, (2009).

LIST OF PAPERS BASED ON THESIS

1. S.K.Bharath , Prof.A.N.Rajagopalan , Raj Kuppa Ambiguity-Free Photometric Stereo fro symmetric objects *Neoterix*, (2018).