ABSTRACT

Title of dissertation: Scene Analysis under Variable Illumination using Gradient Domain Methods

Amit Agrawal, Doctor of Philosophy, 2006

Dissertation directed by: Professor Rama Chellappa
Department of Electrical and Computer Engineering

The goal of this research is to develop algorithms for reconstruction and manipulation of gradient fields for scene analysis, from intensity images captured under variable illumination. These methods utilize gradients or differential measurements of intensity and depth for analyzing a scene, such as estimating shape and intrinsic images, and edge suppression under variable illumination. The differential measurements lead to robust reconstruction from gradient fields in the presence of outliers and avoid hard thresholds and smoothness assumptions in manipulating image gradient fields.

Reconstruction from gradient fields is important in several applications including shape extraction using Photometric Stereo and Shape from Shading, image editing and matting, retinex, mesh smoothing and phase unwrapping. In these applications, a non-integrable gradient field is available, which needs to be integrated to obtain the final image or surface. Previous approaches for enforcing integrability have focused on least square solutions which do not work well in the presence of

outliers and do not locally confine errors during reconstruction. I present a generalized equation to represent a continuum of surface reconstructions of a given non-integrable gradient field. This equation is used to derive new types of feature preserving surface reconstructions in the presence of noise and outliers. The range of solutions is related to the degree of anisotropy of the weights applied to the gradients in the integration process.

Traditionally, image gradient fields have been manipulated using hard thresholds for recovering reflectance/illumination maps or to remove illumination effects such as shadows. Smoothness of reflectance/illumination maps is often assumed in such scenarios. By analyzing the direction of intensity gradient vectors in images captured under different illumination conditions, I present a framework for edge suppression which avoids hard thresholds and smoothness assumptions. This framework can be used to manipulate image gradient fields to synthesize computationally useful and visually pleasing images, and is based on two approaches: (a) gradient projection and (b) affine transformation of gradient fields using cross-projection tensors. These approaches are demonstrated in the context of several applications such as removing shadows and glass reflections, and recovering reflectance/illumination maps and foreground layers under varying illumination.

Scene Analysis under Variable Illumination using Gradient Domain Methods

by

Amit Agrawal

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2006

Advisory Committee:
Professor Rama Chellappa, Chairman/Advisor
Professor Larry Davis
Professor Prakash Narayan
Assistant Professor Min Wu
Dr. Ramesh Raskar, Mitsubishi Electric Research Labs (MERL)

DEDICATION

To my parents

ACKNOWLEDGMENTS

I would like to thank

My advisor Prof. Rama Chellappa for his constant guidance, support and encouragement. He introduced me to the interesting problem of integrability. His positive outlook and cheerful attitude has always inspired and guided me.

Dr. Ramesh Raskar, Mitsubishi Electric Research Labs (MERL), for his constant guidance and supervision during the last two years. His incessant enthusiasm and optimism towards research has always encouraged me and given me a new outlook on research.

My committee members, Prof. Min Wu, Prof. Prakash Narayan and Prof. Larry Davis for wonderful suggestions and discussions.

Prof. Jack Tumblin, Northwestern University, with whom I had the pleasure of collaborating on several projects and having several interesting discussions during the last two years. Our gradient camera project motivated me to pursue the gradient integration problem.

Prof. Shree K. Nayar, Columbia University, for this help and guidance on several projects.

Dr. Chandra Shekhar and Dr. Reuven Meth for their supervision at UMD.

Joe Marks, (Director,MERL), MERL members and interns for their help during several projects including Jeoren van Baar, Wojciech Matusik, Paul Beardsley,

Hideaki Nii, Karhan Tan and others.

MERL and Army Research Lab (ARL) for supporting my research in Advanced Decision Architectures Consortium under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0009, along with Dr. B. Chadrasekaran, Dr. Sue Archer, Cathi Brents, Larry Tocarcik, Dr. Phil David and Jeff Dehart.

Satya Mallick, Holger Winnemoller, Dan Goldman, Abhijit Ogale, and Gang Qian for providing datasets and images.

Colleagues at CFAR, MERL and MIT for their friendship and several stimulating discussions especially Aravind Sundaresan, Neel Joshi, Yuanzhen Li and many others.

My friends at Maryland and Boston for all the good times I had: Manish Shukla, Rahul Ratan, Anuj Rawat, Vishal Khandelwal, Tarun Pruthi, Gaurav Agarwal, Rishi Chaturvedi, Sumit Chandwani, Sachin Garg, Lavanya Sharan, Shruti Kapoor and many others.

And above all, my parents for their unending love and support and all the sacrifices they made, my sister, Smita for her love and smiles and my brother, Anuj for his humor and sharing my responsibilities.

Finally, I thank God for giving me strength and courage throughout my studies and leading me to its completion.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Scene analysis from intensity images is the central goal of computer vision. This analysis could be in terms of estimating shape of objects in the scene, building 3D models, recovering intrinsic scene properties such as reflectance of scene objects and scene illumination, object recognition, motion estimation and so on. This dissertation presents methods for reconstructing and manipulating gradient fields of intensity and depth for scene analysis, such as estimating shape and intrinsic images, from intensity images captured under varying illumination.

This research address issues of noise and outliers in reconstruction from gradient fields. It also provides a framework for manipulations of image gradient fields that avoid hard thresholds and smoothness assumptions, to obtain computationally useful and visually pleasing images. In this chapter, I first introduce the related problems and the developed methods, and discuss the motivation of using gradients. Then, I present the outline for rest of the dissertation.

1.1 The Problem

This dissertation focuses on two main problems. The first problem is encountered while reconstructing an image or surface from a 2D gradient field. Reconstruction from gradient fields is important in several applications including shape

estimation using Photometric Stereo (PS) [1, 2, 3, 4, 5] and Shape from Shading (SfS) [6, 7, 8, 9, 10, 11, 12, 13], high dynamic range compression [14], phase unwrapping [15, 16, 17, 18] and image editing [19], matting [20], and fusion [21, 22]. Typically, an estimate of gradient field is available, and the final image or surface is obtained by integrating the available gradient field.

For example, techniques such as PS utilizes multiple images of a scene captured under varying illumination to first estimate a surface normal/gradient field using a reflectance model (e.g., Lambertian reflectance model [23]). The estimated surface gradient field needs to be integrated to obtain the final shape. In image based manipulations such as image editing [19], the gradient fields of one or more images are modified to achieve the desired goal, and the final image is obtained by integrating the manipulated gradient field. The Retinex [24, 25, 26, 27] algorithm for estimating lightness from images assumes that illumination is slowly varying in image and reflectance has sharp edges. By removing strong gradients from the intensity gradient field, the resulting gradient field can be integrated to obtain the lightness values [24, 28].

The gradient field of a scalar surface should be integrable, i.e., the integral along any closed loop should be equal to zero. Inherent noise and ambiguities in the estimation process, and linear/non-linear manipulations of gradient fields results in **non-integrable** gradient fields. Common approaches for enforcing integrability such as solving the Poisson equation [11, 29] and Frankot-Chellappa algorithm [30] are based on least squares. It is well known that least square (LS) approaches are not robust, i.e., they are sensitive to outliers and favor smooth solutions. This results in

loss of features in shape reconstruction (using PS and SfS) and leads to contouring, pinching and color artifacts in images obtained by integrating manipulated image gradient fields. In addition, LS approaches do not have the property of local error confinement. Errors in the estimated gradient field propagates to the entire surface during reconstruction and are not confined locally.

The second problem is related to the manipulation of gradient fields itself. Variable illumination poses a problem for computer vision algorithms and it is desirable to remove the effects of illumination such as shadows [31] and to recover an intrinsic image [32, 33, 34, 35, 36] of the scene. Traditionally, image gradient fields have been manipulated using hard thresholds for estimating reflectance/illumination (as in Retinex [24]) or for removing shadows by setting gradients corresponding to shadow pixels to zero [31]. However, threshold estimation is difficult and a single threshold cannot account for illumination variations across the entire image. Often, smoothness assumptions on reflectance/illumination maps are used in these algorithms. These smoothness assumptions are not valid at reflectance and illumination boundaries. Scene analysis from a single image is a challenging problem and it is desirable to avoid these assumptions and hard thresholds.

1.2 The Approach

In this dissertation, the above two problems are tackled as follows. We wish to obtain **feature preserving reconstruction** from gradient fields in the presence of noise and outliers to handle the first problem. This is addressed by analyzing the

problem of 2D integration in several ways. Firstly, it is useful to think about the gradient integration problem in terms of robust estimation to handle outliers. This is possible because the presented analysis shows that all gradients are not required for integration. One could **selectively choose the gradients for integration** using information from the given gradient field itself, to locally confine errors.

Secondly, it is shown that the solutions for surface reconstruction lie in a high dimensional space and the least square solution is one of them. Thus, one can try to find other robust solutions. Least square approaches are *isotropic*, assigning equal weights to all the gradients (input data). Anisotropic approaches for image restoration [37, 38, 39], motion estimation and optical flow [40, 41], as well as robust estimation in early vision problems [42, 43] are widely popular. Inspired by the success of these algorithms, this research shows that by an **anisotropic treatment of gradients**, one can preserve sharp features in the reconstruction.

The second problem of manipulating image gradient fields is addressed using illumination invariants and possible use of active illumination. In several scenarios, such as a fixed surveillance camera, multiple images from the same viewpoint are available. How can we utilize information from multiple images and discount the illumination variations? Image intensities are not invariant under illumination changes and it has been shown that discriminative illumination invariants do not exist [44]. However, the direction of image intensity gradient remains more or less stable with the illumination changes [44]. To manipulate the gradient field, the information from the direction of image gradients is utilized for **edge suppression**. This enables removal of edges from an image using another image based solely on

the intensity gradient direction and can handle large illumination variations in the images.

Active illumination can sometimes be used to acquire multiple images from the same viewpoint. Digital cameras are equipped with external *flash*, which provides an easy way to introduce additional illumination and acquire an image within a short exposure time [45, 46]. By taking a pair of images, one captured under flash illumination and the other without flash (only using the ambient illumination), one can get more information about the scene. Subsequent analysis of intensity gradient directions can then be used for edge suppression.

1.3 Why Gradients?

A basic question to ask is: why use gradients? Gradients are more noisy than the input images, so how can gradient domain methods be favored?

Two scenarios need to be considered in answering this important question: applications where gradients are *estimated* and applications where gradients are *computed* and subsequently manipulated. The first scenario occur in problems such as Photometric Stereo and Shape from Shading as discussed above. The surface gradient field is *estimated* from intensity images using a reflectance model (usually Lambertian). The estimated gradient field will have noise and possibly outliers, due to noise in the input intensity images, inaccuracies in modeling (non-Lambertian surfaces) and possible ambiguities [47]. Thus, noise in the estimated gradient field is unavoidable, and it is important to discount noise and outliers to get good re-

constructions. This is one of the main goals of this research and the anisotropic treatment resulting from the use of generalized equation address this issue.

The second scenario occur in applications such as image gradient field manipulations, where the gradient fields of one or more images are *computed*, and then manipulated to achieve a desired goal. The final image is obtained by integrating the manipulated gradient field. Several issues need to be addressed here. Firstly, the process of computing gradients should be *numerically compatible* with the process of integration. By compatibility, I mean the following. Suppose I take an image, compute its gradient field and integrate it back without any modifications. I should be able to get back the original image up to an unknown constant of integration. The reconstruction algorithm should be *exact*. In other words, the difference of the original and the integrated image should be a constant all over the image, if no modifications are made to the gradient field. If forward finite differences are used for computing gradients, the corresponding integration should reflect this fact. If instead, central differences are used, the integration process should change accordingly. This subtle fact is often ignored. Thus, even without any modifications of the gradient field, the integrated image will differ from the original image and it becomes difficult to analyze the effect of modifications themselves. The reconstruction algorithms that are presented in this dissertation are exact, and do not introduce any additional artifacts.

Secondly, even if compatibility is achieved, manipulation of noisy gradients can be problematic. Assuming IID Gaussian noise in image intensities with mean 0 and variance σ^2, the noise in gradients will have variance $2\sigma^2$ (assuming finite differ-

ences for computing gradients). But it is important to note that twice the number of gradients are available. For an $N \times N$ grid, there are N^2 intensity values, but $2N^2$ gradients (x and y gradients). Thus, if all gradients are used judiciously, noise can be handled. However, more important fact is that manipulations presented in this dissertation use the information from the direction of the gradients. The magnitude of the intensity gradients as well as image intensities are sensitive to changes in illumination, but the direction of the intensity gradients remains stable [44]. Previous research on local structure tensors [48, 49] have used neighborhood support and color information for reliable estimation of the intensity gradient direction [50]. The affine transformation approach presented in this dissertation for manipulating gradient fields handle noise by estimating the cross-projection tensors using the local structure tensors.

Another advantage of using gradient domain methods is that the integration process used to obtain the final image automatically provides smoothing. For example, this is useful in applications such as seamless image stitching. Thus, smoothness assumptions can be avoided. Finally, rapid improvement in resolution and reduction in sensor noise of digital cameras also favor gradient based manipulations. Coupled with better algorithmic approaches and correct numerical procedures, this dissertation show results from image gradient field manipulations that are computationally useful and visually pleasing.

1.4 Contributions

In this dissertation, I present ideas for reconstructing gradient fields and manipulating image gradient fields under varying illumination. This dissertation makes the following contributions:

- **Gradient Field Integration** The idea that all gradients are not required for integration and least square based Poisson solver is only one of the several solutions in the space of all possible solutions. The minimal set of gradients required to integrate a 2D gradient field correspond to the spanning tree of the underlying 2D planar graph.

- **Feature Preserving Surface Reconstructions** A generalized equation for surface reconstruction from non-integrable gradient fields, based on controlling the degree of anisotropy of weights assigned to the gradients. The key idea is to replace gradients by functions of gradients arising from valid error functionals. Common approaches such as Poisson solver [29] and Frankot-Chellappa algorithm [30] are special cases of this equation.

- **Edge Suppression** A framework to remove edges in an image based on edge information in another image for image gradient field manipulations. The key idea is to use the direction of the intensity gradient as a constraint in manipulating image gradient fields.

- The idea of obtaining the information about the direction of image intensity gradients by taking another image using **flash as active illumination**.

This dissertation contributes several new gradient domain algorithms for manipulating and reconstructing gradient fields. These include:

- **α-surface** algorithm for surface reconstruction that can provide tradeoff between smoothness and robustness using the single parameter α.
- **Affine transformation of gradient fields** using diffusion tensor to obtain feature preserving reconstructions.
- **Gradient Projection** for removing edges by taking a vector projection of the image gradient onto the image gradient of another image.
- Affine transformation of image gradient fields using **cross-projection tensors** for edge suppression under varying illumination.

1.5 Thesis Outline

The rest of the dissertation is organized as follows. In Chapter 2, I discuss the problem of surface reconstruction from gradient fields, analyze the solution space and show limitations of least square approaches. I present an algebraic solution for enforcing integrability that achieves local error confinement. The ideas presented in this chapter are extended in Chapter 3 to derive a generalized equation for surface reconstruction from gradient fields. I derive several new algorithms using this equation and provide comparisons with previous approaches. Chapter 4 presents a framework for edge suppression using gradient projection and affine transformation of gradient fields. I introduce cross-projection tensors for affine transformation and

show results on several applications such as foreground layer recovery, removing shadows and glass reflections and recovering intrinsic images. Finally, I conclude and outline future directions and applications of this work.

Chapter 2

Surface Reconstruction from Gradient Fields: Analysis and Algebraic Approach

In this chapter, I analyze the space of all possible reconstructions from a gradient field. I show that least square approaches do not work well in the presence of outliers and do not have the property of local error confinement. I then present an algebraic approach for enforcing integrability, which has the property of local error confinement. The key idea resulting from this analysis is that all gradients are not required for integration. By identifying erroneous gradients using curl of the given gradient field, they can be discarded during integration which leads to a better solution.

2.1 Introduction

The notion of integrability arises whenever a surface has to be reconstructed from a gradient field. In several computer vision problems such as SfS [6, 7, 8, 10, 11, 12] and PS [1, 2, 3, 4, 5], an estimate of the gradient field is available. The gradient field is then integrated to obtain the desired 2D surface (shape). However, the estimated gradient field often has non-zero curl making it non-integrable. Reconstruction from gradient fields is also important in several other applications such mesh editing [51], retinex [24, 28], high dynamic range compression [14], phase

unwrapping [15, 16, 17, 18] and image editing [19], matting [20] and fusion [21, 22]. In these algorithms, the gradient fields of one or more images are manipulated to achieve the desired goal and the final image is obtained by a 2D integration of the manipulated gradient field.

The gradient field of a scalar surface should have zero curl or it should be integrable. The integral along any closed loop (path) should be equal to zero and the reconstruction should not depend on the choice of the integration path. In practice, the estimated gradient field is rarely integrable due to the inherent noise in the estimation process, or manipulation of gradient fields. In addition, ambiguities in the solution and ill-posed problems often lead to non-integrable gradient fields.

2.1.1 Related Work

Researchers have addressed the issue of enforcing integrability typically specific to the problem at hand. Previous methods have used the integrability constraint during the estimation of surface (or surface normals) in PS, SfS and Shape from Texture as in [11, 52]. In these methods, integrability is enforced as a constraint to regularize the solution or to remove the inherent ambiguities.

In **SfS** [6, 7, 8, 10, 11, 12], integrability was enforced as a constraint in the minimization routine in [11, 53]. Frankot & Chellappa [30] project the non-integrable gradient field on to a set of integrable slopes using the Fourier basis functions. However, their method is dependent on the choice of basis functions. Several variants of this approach have been proposed by either choosing a different basis

function [54] (cosine functions) or using a redundant non-orthogonal set of basis functions (shapelets) [55]. In [29], a direct analytical solution based on solving the Poisson equation was proposed, for obtaining the integrable gradient field closest to the given gradient field in the least squares sense. As shown in Section 2.3.3, this method ignores the information in the curl and finds a zero-curl field which has the same divergence as the given non-integrable field. The method also lacks the property of local error confinement. For a survey of SfS algorithms see [12].

Photometric Stereo [1, 2, 3, 4, 5] use multiple images obtained under different illumination directions to recover the surface gradients. In [56], belief propagation in graphical networks was used to enforce integrability for SfS and PS problems. In [57, 58], the integrability constraint was used to remove the bas-relief ambiguity in estimation of shape and albedo from multiple images. Calibration information about the illumination geometry is obtained by imposing integrability constraint in [59]. In [60], the integrability constraint was used along with rank constraints in a minimization routine to estimate surface albedo and normals. Other examples include [52, 61]. The idea of enforcing integrability has also been used in flow field visualization [62, 63] by decomposing the given field into curl-free and divergence-free parts.

Noise reduction in images is a topic commonly addressed in image restoration techniques. Several methods based on solving partial differential equations (PDE) such as anisotropic diffusion [37], shock filters [64] and energy based methods [65] (see [66] for detailed analysis and algorithms) have been proposed for restoring an image while preserving edges or sharp features. Inspired by the success of

these approaches, I show how to incorporate robust estimation, regularization and anisotropic diffusion in the gradient integration problem.

2.1.2 Shape from Shading

SfS attempts to extract 3D shape from a single image. It is assumed that the reflectance map is given or is known. Let $\mathbf{s}$ denote the direction of the light source given by

$$\mathbf{s} = \frac{1}{\sqrt{(}1 + p_s^2 + q_s^2)}(-p_s, -q_s, 1)^T, \tag{2.1}$$

for some p_s and q_s. Let Z be the surface height and $p = \frac{\partial Z}{\partial x}$, $q = \frac{\partial Z}{\partial y}$. The surface normal $\mathbf{n}$ is then given by [11]

$$\mathbf{n} = \frac{1}{\sqrt{(}1 + p^2 + q^2)}(-p, -q, 1)^T. \tag{2.2}$$

For a Lambertian surface, the image intensity $I(x, y)$ is modeled as [11]

$$\begin{aligned} I(x, y) &= \rho(x, y)\mathbf{n}(x, y) \cdot \mathbf{s} \\ &= \rho \frac{1 + pp_s + qq_s}{\sqrt{(}1 + p^2 + q^2)\sqrt{(}1 + p_s^2 + q_s^2)}, \end{aligned} \tag{2.3}$$

where ρ is the albedo of the surface and $\cdot$ denotes dot product. Usually, ρ is assumed to be constant over the surface. If the albedo is constant, one can also write the image irradiance equation as

$$\begin{aligned} I(x, y) &= R(p, q) \\ &= \frac{1 + pp_s + qq_s}{\sqrt{(}1 + p^2 + q^2)\sqrt{(}1 + p_s^2 + q_s^2)}, \end{aligned} \tag{2.4}$$

where $R(p, q)$ is also called the reflectance map. Given an estimate of the light source direction, the problem then reduces to estimating the surface gradients p and q from the image intensity I.

2.1.3 Photometric Stereo

Photometric Stereo tries to estimate the 3D shape of an object from images taken from the same viewpoint but illuminated from distant point light sources from multiple directions. A minimum of three images are required to estimate the shape with the constraint that the direction of light sources should not be coplanar. If $I_1 \dots I_n$ are n images captured under distant point light sources whose directions are given by $\mathbf{s}_1 \dots \mathbf{s}_n$, then assuming a Lambertian reflectance model, one can write

$$I_i(x, y) = \rho(x, y)\mathbf{n}(x, y) \cdot \mathbf{s}_i \quad i = 1 \dots n. \tag{2.5}$$

If $\mathbf{s}_i$'s are known, then for each pixel, one can write a linear system for the scaled albedo $\mathbf{a}(x, y) = \rho(x, y)\mathbf{n}(x, y)$ by stacking the image intensities on top of each other

$$\begin{bmatrix} \mathbf{s}_1^T \\ \mathbf{s}_2^T \\ \vdots \\ \mathbf{s}_n^T \end{bmatrix} \mathbf{a}(x, y) = \begin{bmatrix} I_1(x, y) \\ I_2(x, y) \\ \vdots \\ I_n(x, y) \end{bmatrix}. \tag{2.6}$$

This linear system is solved to estimate the scaled albedo $\mathbf{a}$. The albedo, ρ, can be obtained as the norm of $\mathbf{a}$.

$$\rho(x, y) = |\mathbf{a}(x, y)|. \tag{2.7}$$

And the surface normal $\mathbf{n}(x, y)$ is given by

$$\mathbf{n}(x, y) = \frac{\mathbf{a}(x, y)}{|\mathbf{a}(x, y)|}. \tag{2.8}$$

Given $\mathbf{n}(x, y) = [n_x, n_y, n_z]^T$, the surface gradients can be obtained as $p = -n_x/n_z$, $q = -n_y/n_z$. From now on, I will focus on the problem of obtaining Z, once an estimate of the gradient field (p, q) is known.

2.2 Problem Statement

Let $Z(x, y)$ be a $2D$ real valued scalar function defined on a $H \times W$ rectangular grid (y, x) of image pixels. Let $\{p(y, x), q(y, x)\}$ denote the given non-integrable gradient field over this grid. Define the curl and divergence operators as [67]

$$C_{p,q} = \texttt{curl}(p, q) = \frac{\partial p}{\partial y} - \frac{\partial q}{\partial x} \tag{2.9}$$

$$L_{p,q} = \texttt{div}(p, q) = \frac{\partial p}{\partial x} + \frac{\partial q}{\partial y} \ . \tag{2.10}$$

Given $\{p, q\}$, the goal is to obtain a surface Z. Let $\{Z_x, Z_y\}$ denote the gradient field of Z. A common approach is to minimize the least square error functional given by [11, 29]

$$J(Z) = \int \int \left((Z_x - p)^2 + (Z_y - q)^2 \right) dxdy \ . \tag{2.11}$$

The Euler-Lagrange equation corresponding to (2.11) gives

$$\begin{aligned} \frac{\partial J}{\partial Z} - \frac{\partial}{\partial x}\frac{\partial J}{\partial Z_x} - \frac{\partial}{\partial y}\frac{\partial J}{\partial Z_y} &= 0 \\ \therefore \frac{\partial}{\partial x}(Z_x - p) + \frac{\partial}{\partial y}(Z_y - q) &= 0 \\ \therefore \frac{\partial^2 Z}{\partial x^2} + \frac{\partial^2 Z}{\partial y^2} &= \frac{\partial p}{\partial x} + \frac{\partial q}{\partial y} \\ \therefore \nabla^2 Z &= \texttt{div}(p, q) \end{aligned} \tag{2.12}$$

This is the ***Poisson equation*** [11, 29], where $\nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ is the Laplacian operator. This method is referred to as the Poisson solver. The integrable field is found by differentiating the estimated surface Z. Thus, $\{Z_x, Z_y\}$ is the integrable gradient field corresponding to the given gradient field $\{p, q\}$.

2.3 Space of All Solutions

Suppose Z denote the surface which is obtained by integrating the given gradient field using some algorithm. Let $\{Z_x, Z_y\}$ denotes the gradient field of Z. Thus, $C_{Z_x,Z_y} = 0$ everywhere. One can always write

$$\{Z_x, Z_y\} = \{p, q\} + \{\epsilon_x, \epsilon_y\} \ , \tag{2.13}$$

where $\{\epsilon_x, \epsilon_y\}$ denotes the *residual gradient field* which is added to the given non-integrable field to make it integrable. Applying the curl operator in (2.13),

$$0 = C_{Z_x,Z_y} = C_{p,q} + C_{\epsilon_x,\epsilon_y} \quad \texttt{or} \quad C_{\epsilon_x,\epsilon_y} = \quad -C_{p,q} \ . \tag{2.14}$$

$C_{p,q}$ is known from the given non-integrable field $\{p, q\}$. To enforce integrability, consider the smallest loop made up of 4 square connected pixels, $(y, x), (y, x + 1), (y + 1, x)$ and $(y + 1, x + 1)$ (see Figure 2.1(b)). The integral along this loop can be written as

$$C_{p,q}(y + \frac{1}{2}, x + \frac{1}{2}) = p(y + 1, x) - p(y, x) + q(y, x) - q(y, x + 1) \ , \tag{2.15}$$

where the notation $(y+\frac{1}{2}, x+\frac{1}{2})$ has been used to emphasize that the curl is calculated for that loop. Writing (2.15) for the residual gradient field in (2.14), we get

$$\epsilon_x(y + 1, x) - \epsilon_x(y, x) + \epsilon_y(y, x) - \epsilon_y(y, x + 1) = -C_{p,q}(y + \frac{1}{2}, x + \frac{1}{2}) \ , \tag{2.16}$$

for one particular loop. For an $H \times W$ image, the number of loops will be equal to $M = (H - 1) \times (W - 1)$, the number of x gradients will be equal to $K_x = H(W - 1)$ and the number of y gradients will be equal to $K_y = (H - 1)W$. Let

$K = K_x + K_y = 2HW - H - W$. Writing the above equation for all the loops in the image and stacking all of them on top of each other (in lexicographical order) we get

$$\mathbf{Ax} = \mathbf{b} \ , \tag{2.17}$$

where $\mathbf{b}$ is a $M \times 1$ vector containing the negative of the curl values $C_{p,q}$ for all the loops, $\mathbf{x}$ is a $K \times 1$ vector containing all the unknown gradient values corresponding to the residual gradient field. The matrix $\mathbf{A}$ is a $M \times K$ sparse matrix, where each row corresponds to a loop equation and has only 4 non-zero entries: two $+1$'s and two -1's according to (2.16). Note that any other bigger loop can be written as a linear combination of these basic 4 pixel loops and will not give linearly independent rows in matrix $\mathbf{A}$. Thus, it is sufficient to consider just the 4 pixel loops.

2.3.1 General Solution for Residual Gradient Field

We have $K - M = HW - 1 \geq 0$. For example, for a 3×3 grid, $K = 12$ and $M = 8$. Thus, the matrix $\mathbf{A}$ has a null space of dimension $n = K - M = HW - 1$. Suppose $\{x_b^1, x_b^2, \ldots, x_b^n\}$ denote an ortho-normal basis set for the null space of $\mathbf{A}$. A general solution x_g is given by the sum of a particular solution x_p satisfying $\mathbf{A}x_p = \mathbf{b}$, and a homogeneous solution x_h satisfying $\mathbf{A}x_h = \mathbf{0}$. The homogeneous solution lies in the null space and can be written as the weighted sum of the basis functions

$$x_h = \sum_{i=1}^{n} \beta_i x_b^i \ , \tag{2.18}$$

where the coefficients β can be obtained by the dot-product of x_h with the basis functions

$$\beta_i =< x_h, x_b^i >,\ i = 1 \ldots n. \tag{2.19}$$

Thus, any solution is of the form

$$x_g = x_p + \sum_{i=1}^{n} \beta_i x_b^i \ . \tag{2.20}$$

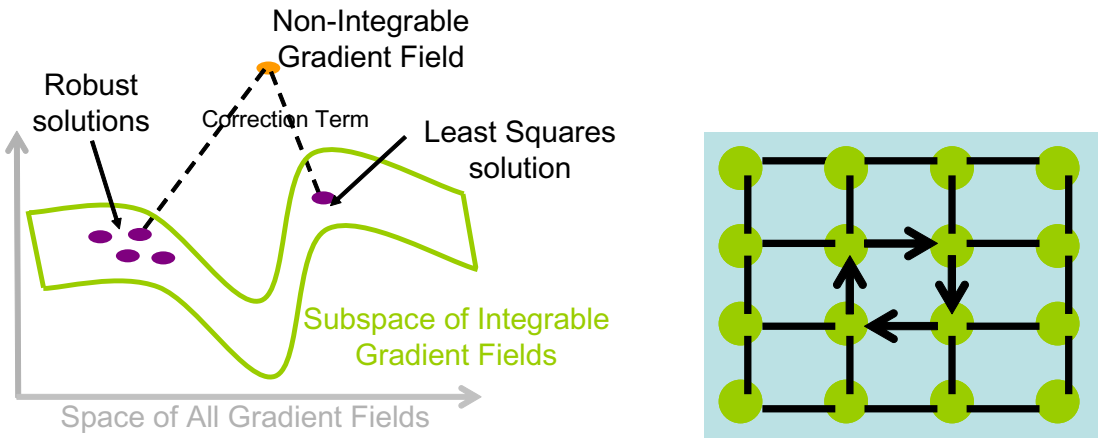
Now if we define a space using the coefficients β_i's, then any correction field will lie in this space. The dimension of this space is equal to the null space of the matrix $\mathbf{A}$. Thus any solution lies in a space R^{HW-1}. Note that all integrable gradient fields lie in the null space of $\mathbf{A}$. Thus each x_b^i corresponds to an integrable gradient field. Interestingly, [68] also showed that the feasible space is of dimension $HW - 1$. However, their analysis is based on the choice of a specific basis function and projections on to this set of basis. In addition, they define the feasible space in terms of the integrable gradient fields, but we define it in terms of the coefficients β_i.

2.3.2 Least Square Solution in the Space of All Solutions

The Poisson solver tries to find that surface Z which minimizes the following least square cost function

$$J(Z) = \int\int \left((Z_x - p)^2 + (Z_y - q)^2\right) dxdy = \int\int (\epsilon_x^2 + \epsilon_y^2) dxdy \ . \tag{2.21}$$

The Euler-Lagrange equation gives the Poisson equation: $\nabla^2 Z = \texttt{div}(p, q)$. Thus, it tries to find that solution for which the residual gradient field has the minimum



(a) (b)

Figure 2.1: Space of all solutions. (a) Poisson solver finds the solution corresponding to the minimum norm residual gradient field, but this may not be robust. (b) The 2D graph corresponding to a sample 4×4 grid. Nodes correspond to the value of the surface at the grid points and gradients correspond to the edges.

norm (see Figure 2.1(a)). Hence, the Poisson solver is also equivalent to

$$\min_{x_g} \quad x_g^T x_g \quad \texttt{subject to} \quad Ax_g = b = \texttt{curl}(p, q) \ . \tag{2.22}$$

Let x_{LS} be the residual gradient field for the Poisson solver (solution of the above equation). Now x_{LS} is a particular solution for the above system of equations as it satisfies $Ax_{LS} = b$. So a general solution can be written as

$$x_g = x_{LS} + \sum_{i=1}^{n} \beta_i x_b^i \ . \tag{2.23}$$

Proposition 1. *If the general solution x_g is defined as in* (2.23)*, the solution given by the Poisson solver lies at the origin of R^{HW-1} independent of the choice of basis functions x_b^i.*

Proof. From the orthogonality principle, the least square error is orthogonal to the solution space. Thus $\beta_i =< x_{LS}, x_b^i >= 0$ for all i, independent of the choice of basis functions. □

2.3.3 Interpretation of Poisson Solver in terms of Curl-Divergence Space

Note that

$$\texttt{div}(Z_x, Z_y) = \frac{\partial Z_x}{\partial x} + \frac{\partial Z_y}{\partial y} = \nabla^2 Z = \texttt{div}(p, q). \tag{2.24}$$

Thus, the Poisson solver enforces integrability by finding a zero curl gradient field which has the *same* divergence as the given non-integrable gradient field.

One can visualize a curl-divergence space as shown in Figure 2.2. A vector in this space represents a gradient field. All integrable (zero-curl, irrotational) gradient fields lie along the real (X) axis. All divergence free (solenoidal) gradient fields lie along the imaginary (Y) axis.

Let $\overrightarrow{OA_1}$ denotes an integrable gradient field with divergence L. Suppose the estimated gradient field is given by $\overrightarrow{OA_2}$ with divergence L_1 and curl C_1. The residual (error) gradient field is then given by $\overrightarrow{A_1A_2}$. As discussed earlier, reconstruction using the Poisson solver will give a solution which has the same divergence as the given non-integrable field $\overrightarrow{OA_2}$. Thus, the integrable field given by the Poisson solver will be $\overrightarrow{OA_3}$. It is important to note that the divergence free part (curl) of $\overrightarrow{OA_2}$ is completely ignored during the reconstruction. The approach that I present in this chapter tries to utilize the information in curl by estimating the residual gradient field.

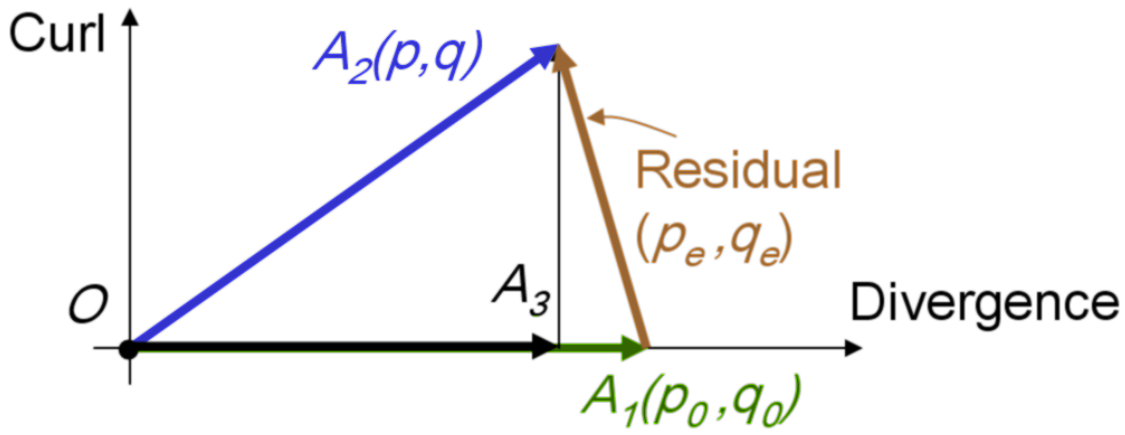


Figure 2.2: The Curl-Divergence space. A vector in this space represents a gradient field. Vectors along X axis (OA_1) represents zero curl (integrable) gradient fields. Usually, a non-integrable field OA_2 is given as an estimate of OA_1. The residual gradient field is given by A_1A_2. Enforcing integrability by solving the Poisson equation gives OA_3 as the integrable gradient field. By estimating the residual gradient field A_1A_2, one can use the information in the curl to move from OA_2 to OA_1 to get a better estimate.

2.4 Drawback of Least Square (LS) Approaches

2.4.1 Lack of Robustness

It is well known that a least square solution does not perform well in the presence of outliers. Consider the surface shown in Figure 2.3(a), which consists of a ramp and several peaks. Gaussian random noise and uniformly distributed outliers were added to the gradient field of this surface. The reconstructed surface from the noisy gradient field using the Poisson solver is shown in Figure 2.3(b). However, if we knew the locations of the outliers, we could use the rest of the gradients to perform the integration. The corresponding reconstruction is shown in Figure 2.3(c). It is clear that a better solution can be obtained by removing outliers. Thus, gradient integration can be thought of as a robust estimation problem.

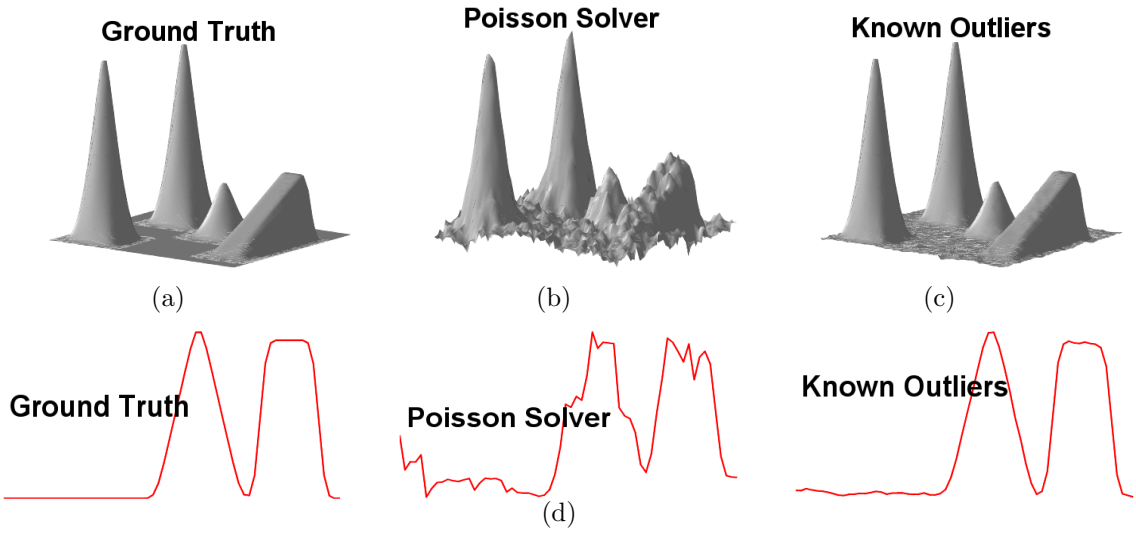


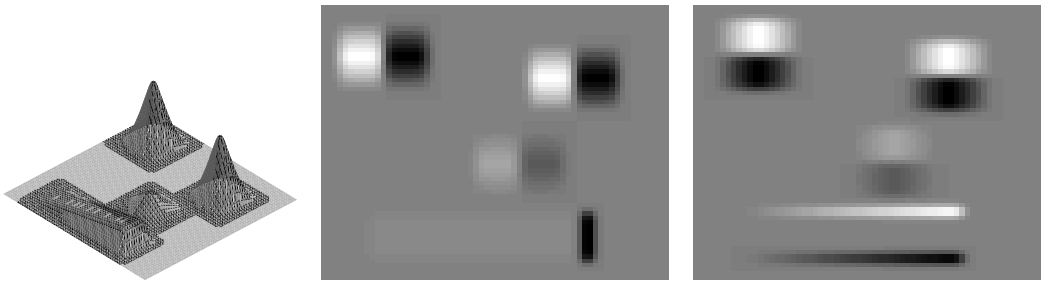
Figure 2.3: Effect of outliers in 2D integration. (a) True surface. (b) Gaussian noise ($\sigma = 0.02$g, g = maximum gradient magnitude) and uniformly distributed outliers were added to the gradients of this surface. Reconstruction using the Poisson solver. Mean Square Error (MSE) = 10.81. (c) If the location of outliers were known, rest of the gradients can be integrated to obtain a much better estimate. MSE = 0.211. (d) One-D height plots for a scan line across the middle of grid.

However, in practical scenarios, the location of the outliers will not be known. To obtain robust solutions, the information in the given gradient field itself need to be utilized.

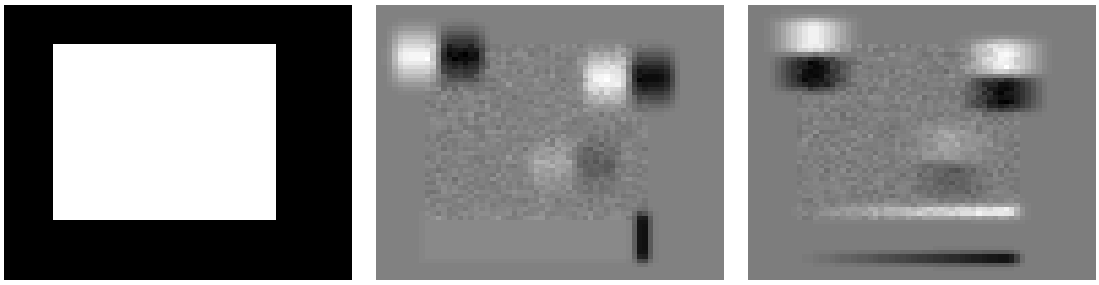
2.4.2 Lack of Local Error Confinement

Another drawback with LS approach is the lack of local error confinement. While reconstruction, errors in the gradient field are propagated throughout the entire surface. Thus, if only a part of the gradient field is in error, it would be desirable to contain the errors in the reconstructed surface to that region only.

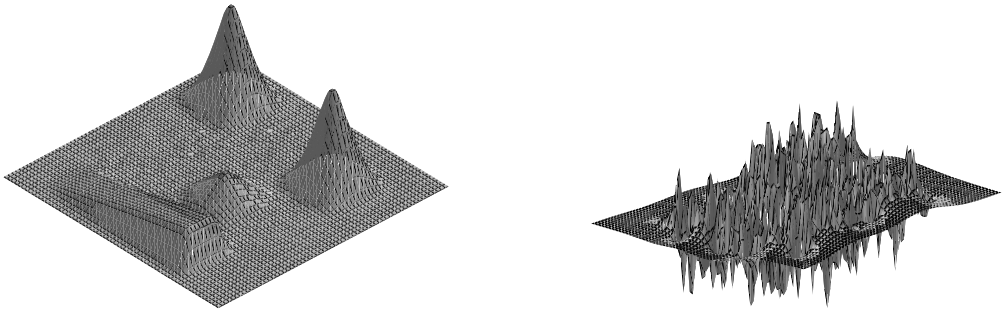
An example will help clarify the point. Figure 2.4(a) shows a synthetic surface. We add noise to a central region of its gradient field as shown in Figure 2.4(d). The



(a) True Surface (b) g_x (c) g_y



(d) Mask for Noise (e) Noisy g_x (f) Noisy g_y



(g) Least Square Reconstruction (h) Corresponding Error

Figure 2.4: Lack of local error confinement in the reconstruction using the Poisson solver. (a,b,c) A synthetic surface with ramp and peaks and the gradient field $\{g_x, g_y\}$ of the surface. (d) Mask region where noise was added to the original gradient field. (e,f) Modified gradients. (g,h) Reconstruction from the noisy gradient field and the corresponding error. Note that the error is not confined to the mask region.

reconstruction from the noisy gradient field is shown in Figure 2.4(g). Note that the error is not confined to the mask region. Thus, errors in the gradient field have propagated to the entire surface. We would like the error to be confined to the mask region. In the next section, I present an algorithm to achieve local error confinement.

2.5 An Algebraic Solution

In this section, I present an algebraic solution for curl correction which has the property of local error confinement. In Section 2.3, while discussing the space of all possible solutions, I formulated curl correction as estimation of the *residual gradient field.* For completeness, I again present the curl equation. Let $\{p, q\}$ be the given gradient field. The curl equation can be written as

$$C(y + \frac{1}{2}, x + \frac{1}{2}) = p(y + 1, x) - p(y, x) + q(y, x) - q(y, x + 1) \tag{2.25}$$

In terms of residual gradient field, the above equation can be written as

$$\epsilon_x(y + 1, x) - \epsilon_x(y, x) + \epsilon_y(y, x) - \epsilon_y(y, x + 1) = -C_{p,q}(y + \frac{1}{2}, x + \frac{1}{2}) \ , \tag{2.26}$$

To enforce integrability, we wish to find an integrable gradient field with zero curl all over. In other words, if we can solve for the residual gradient field, it can be subtracted from the given gradient field to get the integrable gradient field. Thus, curl correction is equivalent to solving for the residual gradient field $\{\epsilon_x, \epsilon_y\}$.

We begin by observing that for an integrable gradient field, curl is zero everywhere and hence the residual gradient field is zero everywhere (trivial solution). So,

we consider only those box loops (of 4 pixels) which have non-zero curl. Thus, if $C_{p,q}(y+\frac{1}{2}, x+\frac{1}{2})$ is non-zero, we consider the gradients in the residual gradient field for that loop to be non-zero and those gradients need to be estimated. Note that in absence of any auxiliary information, all four gradients in the loop are considered to be erroneous.

Suppose K loops have non-zero curl in the image. For each such loop, we can write (2.26). Stacking all such equations on top of each other, we get

$$A\mathbf{x} = b, \tag{2.27}$$

where $\mathbf{x}$ denotes the unknown residual gradient field values that needs to be estimated, b is a $K \times 1$ vector, containing the negative of the curl values for all these loops. The matrix A is a sparse matrix, with each row having only 4 non-zero entries, two $+1$'s and two -1's corresponding to left hand side of (2.26).

To solve the above linear system, A must have full rank. This raises the following questions.

- How do we know when the matrix A has full rank?
- How do we solve for this system when it is not full rank?

In the next section, I present a graph analogy to understand the problem.

2.5.1 Graph Analogy

Using a graph analogy, I outline conditions under which the system of equation $Ax = b$ is solvable. Graph analogy has also been used for residue-cut algorithm [16]

for phase unwrapping. We assume Dirichlet boundary conditions, i.e. the surface values are known at the boundaries. However, the discussion holds for Neumann boundary conditions as well.

We define an undirected graph $G = (V, E)$ on the image plane. Each node in the graph corresponds to a pixel in the image (including boundary nodes). Each node in the interior (not on boundary) has four edges, connecting it to nodes in north, south, east and west directions. For nodes on boundary, those at the corners of the image have two edges and the rest have three edges each. Each edge in the graph between two nodes represent the gradient p or q between the nodes.

With the above terminology, given a non-integrable gradient field, one knows the value of the graph edges and the value of graph nodes at the boundary points (from Dirichlet conditions). The goal is to integrate the gradient field or to find the value of nodes (pixels) in the interior of the image.

If the estimated gradient p or q is erroneous at a node (or the curl is non-zero for that loop), we **break** the corresponding edge in the graph. Thus, when curl is non-zero everywhere, all the edges in the graph (except those between boundary points) will be broken.

Proposition 2. *If the broken edges are such that the graph remains connected, the corresponding system of equation $Ax = b$ can be solved. If however, the graph breaks into n pieces, rank deficit of A is equal to $n - 1$.*

If the graph remains connected, one can reach any node of the graph starting from some boundary node. Thus for each node (pixel), one can always find at

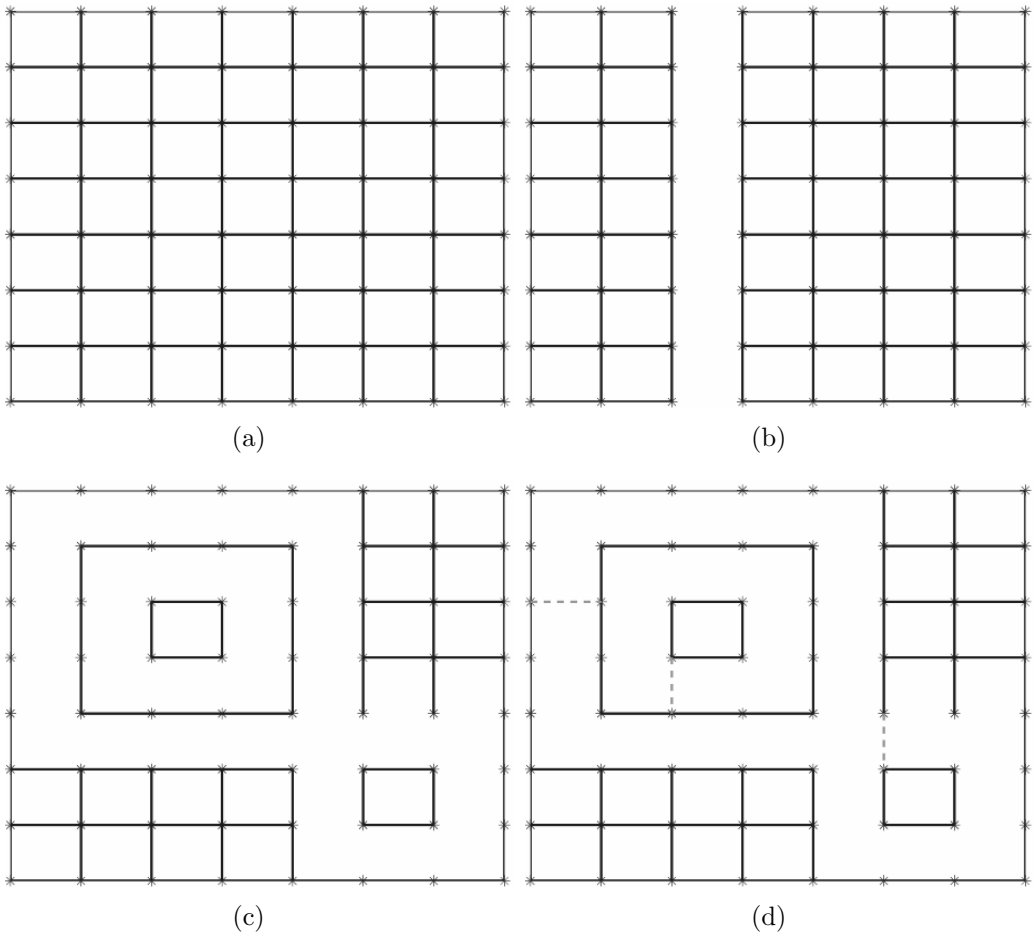


Figure 2.5: Graph analogy and rank considerations. (a) A sample graph on a 8×8 grid. Lines denote edges and dots denote nodes. (b) A configuration such as this cannot happen as boundary edges are assumed to be known. (c) A particular configuration of broken edges where the graph is broken into $n = 4$ parts. Total number of broken edges $= 34$, $\texttt{rank}(A) = 31$. rank deficit $= 34 - 31 = 3 = n - 1$. (d) If 3 more links are known to connect the graph (dashed edges), number of unknowns reduces to $31 = \texttt{rank}(A)$.

least one integration path from some boundary pixel. Thus, the node value can be obtained for all the pixels. This in turn implies that the value of broken edges can be obtained. Thus the system $Ax = b$ has to be full rank.

If however the graph breaks into n pieces, the minimum number of edges that are needed to connect it is $n - 1$. If we have known the edge value of $n - 1$ more edges, we could have solved the system. Thus, the rank of A will be $n - 1$ less than required. Figure 2.5 shows an example.

Thus, the basic idea is to identify the given gradients as good or bad based on the non-zero curl values for each loop. We would like to integrate only using the good gradients. Considering the gradient field as a graph, we break all the edges corresponding to the bad gradients. If the graph remains connected, integration can be done using the good gradients. Otherwise, we need to connect the graph by assuming that some bad gradients are good.

2.5.1.1 Connecting the Graph

We need to find the minimum number of edges that should be joined so that the graph gets connected. To connect the graph, we define an edge weight for each edge and find that *minimal* set of edges which connects the graph and has the minimum total weight. For edges corresponding to the x gradient between nodes $V_{y,x+1}$ and $V_{y,x}$ and corresponding to the y gradient between nodes $V_{y,x}$ and $V_{y+1,x}$, the edge weight is defined as the curl value of the loop.

2.5.2 Algorithm Outline

The complete algorithm can be specified as follows. Given a non-integrable gradient field (p, q),

1. Obtain the curl of the given gradient field all over the image. Form the image graph. For each edge in graph, assign a weight as described in the previous section. Assign all the boundary nodes to set B_2.

2. Identify nodes corresponding to loops where curl is greater than some threshold ($\tau = 10^{-2}$). Assign all such nodes to set B_1 if it is not a boundary node. Assign the remaining nodes to set B_2. Note that when the curl is non-zero all over, B_2 will contain just the boundary nodes and B_1 will contain the remaining nodes.

3. Break all the edges connecting any node in B_1 to any node in B_2. Also break all the edges between the nodes in B_1.

4. **Finding minimal set of edges to join**

   While B_1 is not empty

   (a) Find the shortest path from any node in B_2 to a node in B_1. Let V_{i_1,j_1} be the node in B_1 and V_{i_2,j_2} be the node in B_2 corresponding to this path.

   (b) Remove V_{i_1,j_1} from B_1 and put in B_2.

   (c) Join the edge connecting V_{i_1,j_1} and V_{i_2,j_2}.

5. Now the graph will be connected. For all the edges still broken, form the

equation $A\mathbf{x} = b$. Solve for $\mathbf{x}$ and subtract $\mathbf{x}$ from the corresponding locations in the given gradient field.

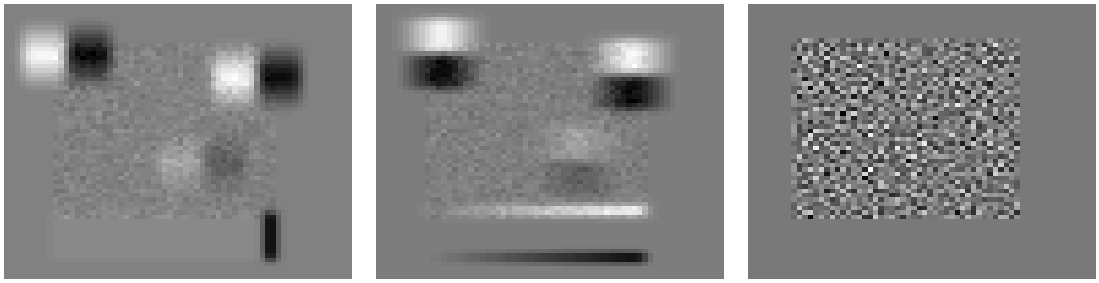
6. Now we have obtained an integrable gradient field. The underlying 2D surface can be obtained by integrating using a Poisson solver.

2.5.3 Results

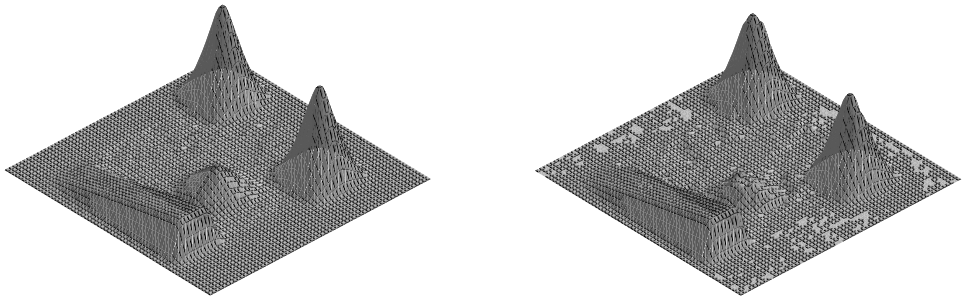
2.5.3.1 Local Error Confinement

We first show that errors in the reconstruction can be confined locally using the algebraic approach. Figure 2.6 shows the same noisy gradient field as shown in Figure 2.4, along with its curl values. We find all loops where the curl is non-zero and break the corresponding edges in the graph. For this example, the number of broken edges was 4004, which formed 1770 disconnected segments. To connect the graph, $1770 - 1 = 1769$ gradients were assumed to be good. The values of the rest of $4004 - 1769 = 2235$ erroneous gradients were obtained by solving the linear system as described above. The final integrable gradient field was obtained by subtracting the residual gradient field from the given noisy gradient field.

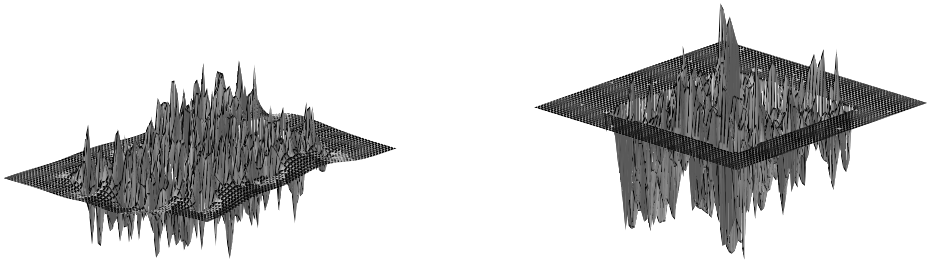
Figure 2.6 compares the reconstruction using the Poisson solver and using the algebraic approach. Note that using the algebraic approach, the error in the reconstructed surface is confined to the mask region where noise was added.



(a) Noisy g_x (b) Noisy g_y (c) Curl



(d) LS Reconstruction using Poisson solver (e) Reconstruction using algebraic approach



(f) Error for LS Reconstruction (g) Error for Proposed Reconstruction

Figure 2.6: Local error confinement. (a,b) Noisy gradient field as shown in Figure 2.4. (c) Curl of the noisy gradient field. (d,e) Comparison of LS reconstruction with the curl correction method. (f,g) Corresponding error with respect to the true surface. Note that in our method, the error is confined to the mask region where noise is added.

2.5.3.2 Shape from Shading

For SfS, we implemented the algorithm by Brooks and Horn [13] and extended it to incorporate our integrability method. This algorithm assumes a Lambertian reflectance model for the surfaces. In this method, at each iteration, new estimates of the surface gradients $\begin{bmatrix} \widehat{p} \\ \widehat{q} \end{bmatrix}_{k+1}$ are obtained from the previous estimates $\begin{bmatrix} \widehat{p} \\ \widehat{q} \end{bmatrix}_{k}$ as

$$\begin{bmatrix} \widehat{p} \\ \widehat{q} \end{bmatrix}_{k+1} = \begin{bmatrix} \widehat{\widehat{p}} \\ \widehat{\widehat{q}} \end{bmatrix}_{k} + \lambda(I - R) \begin{bmatrix} R_x \\ R_y \end{bmatrix} \tag{2.28}$$

where $\widehat{\widehat{p}}_k$ and $\widehat{\widehat{q}}_k$ denotes the smoothed values of $\widehat{p}_k$ and $\widehat{p}_k$ respectively, I is the input image, R is the reflectance map and R_x, R_y denotes its derivatives.

In [30], Frankot & Chellappa proposed to impose integrability at each iteration of the above algorithm by projecting the non-integrable gradients on to the Fourier basis functions. Similar to them, we impose integrability as follows. At each iteration, we first find the new update using the above equation. Then we find the integrable field using our method and use the integrable field in the next iteration.

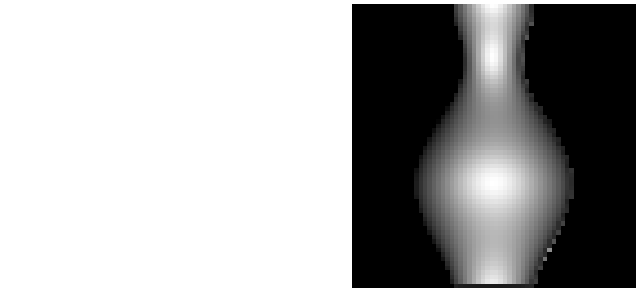
Figure 2.7 shows a synthetic example for Vase and a real example for the Lena image. For vase, we use the illumination direction $(0.05, 0.05, 1)$ to generate the image using a Lambertian reflectance model. For Lena, the illumination direction $(1.5, 0.866, 1)$ was used as suggested in [12]. For both examples, 50 iterations of Brooks & Horn algorithm were performed. Figures 2.7(b) and 2.7(d) compare the reconstructed surfaces using the Poisson solver and the proposed method. Note that in the vase image, enforcing integrability using the Poisson solver does not

properly maintain the boundaries of the object, whereas in the reconstruction using our method, the boundaries are well maintained. For the Lena image, a better visual reconstruction is obtained as shown in Figure 2.7(d).

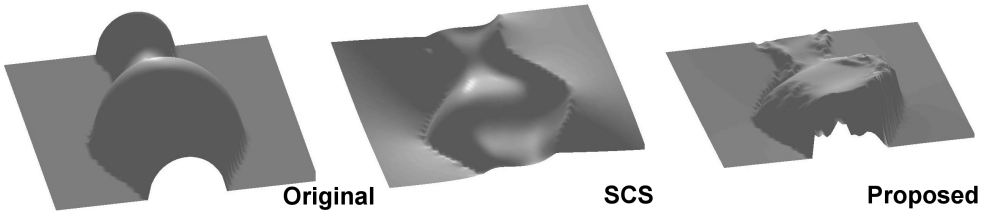
2.5.3.3 Photometric Stereo

For PS, we present results using the Mozart depth map and the Yale face database B ($YaleB01_{P00}$). For the Mozart dataset, five images were generated assuming Lambertian reflectance model using the ground truth depth map. For the Yale database, all 64 images corresponding to the frontal pose for the first subject were used. We assume that the light source directions are known. A simple LS approach was used to estimate the surface normal (n_x, n_y, n_z) at each pixel. The gradient field was obtained as $p = -n_x/n_z$, $q = -n_y/n_z$.

Figure 2.8(a) shows four of the five input images generated using the Lambertian model for the Mozart face, whose true depth map is shown in Figure 2.8(e). Figure 2.8(b) shows the curl of the estimated gradient field. Figure 2.8(e)(center) shows the reconstruction from the estimated gradient field using the Poisson solver. As described in Section 2.5.2, we find those edges which can potentially give rise to non-zero curl values. Nodes connecting these edges (set B_1) are shown in Figure 2.8(c) in white. It is clear that the resultant graph will not be connected. We then find the minimal set of edges to join so as to connect the graph. Figure 2.8(d) shows the final configuration of the connected edges. Note that in this graph, all nodes can be reached using the edges present in the graph. For edges not present

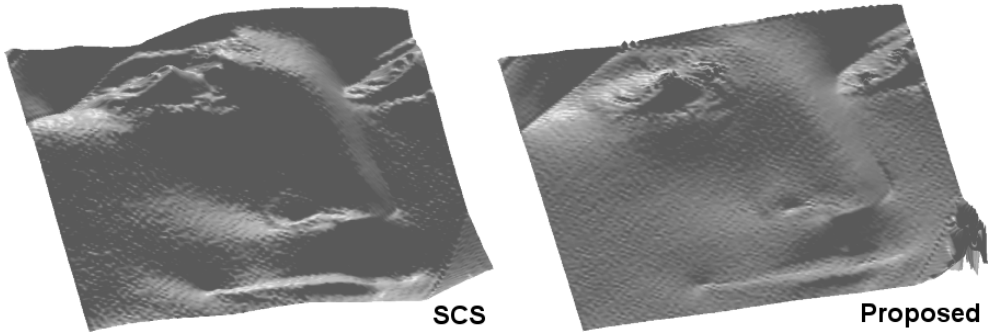


(a)



(b)

(c)



(d)

Figure 2.7: SfS on Vase and Lena image. (a) Synthetic vase image generated using $(0.05, 0.05, 1)$ as the illumination direction. (b) Original depth map, reconstructed surface using the Poisson solver and using the proposed algorithm. Note that in the Poisson solver reconstruction, the surface is globally distorted and boundaries are not preserved. (c) Lena Image. (d) Reconstructed surface using the Poisson solver and using the proposed algorithm. There are distortions along the lips and top left of the reconstructed surface using the Poisson solver.

between the nodes, the system of equation $Ax = b$ was solved. The gradient field was then updated using the x values. Figure 2.8(e) shows the reconstruction using curl corrected gradient field which is much better than using the Poisson solver. The percentage depth error between the true depth map and the reconstructed depth map was 4.26 using Poisson solver and 2.7 using our method.

Figure 2.9(a) shows four (out of sixty four) input images for one of the subjects in frontal pose from Yale database. Figure 2.9(b) shows the curl of the estimated gradient field. Figures 2.9(c) and 2.9(d) show the reconstruction using the Poisson solver and using our method. Note that the reconstruction using our method is much better especially along the face boundaries. Poisson solver can have global distortions since the gradients at any location affect the reconstruction all over the image. In contrast, our method can locally confine errors, does not have any global distortions and all features (sharp gradients) are preserved. Also notice the transitions between the right cheek and the back plane which is smoothed out in the reconstruction using Poisson solver.

2.6 Summary

In this chapter, I analyzed the space of all possible solutions for surface reconstruction from gradient fields. I showed that least square approaches such as Poisson solver do not give feature preserving reconstructions in the presence of noise and outliers and do not have the property of local error confinement. Using a graph analogy, I presented an algebraic approach for curl correction and enforcing inte-

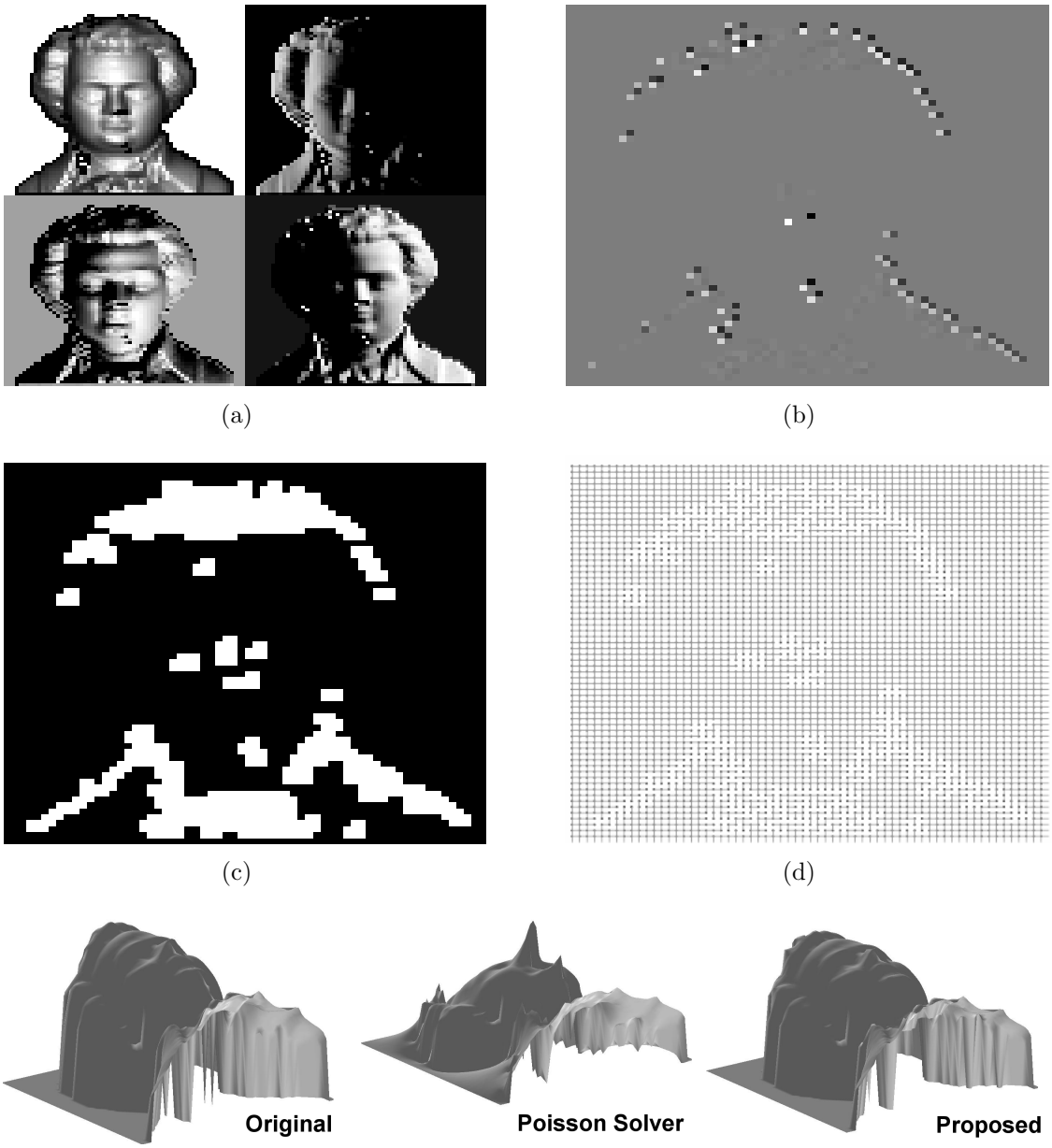
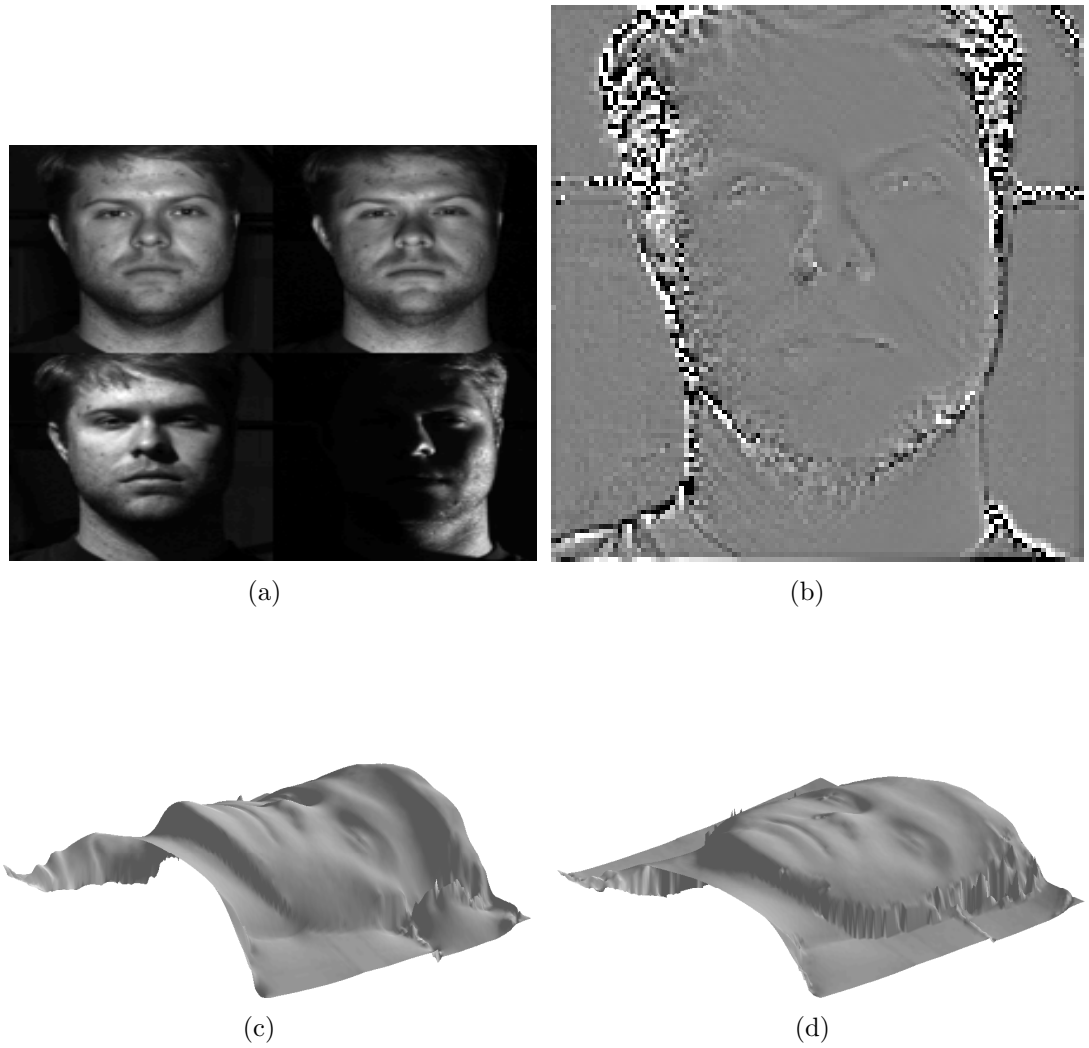


Figure 2.8: Photometric Stereo on Mozart dataset. (a) Sample images. (b) Curl values for estimated gradient field. (c) Initial nodes in set B_1 (Section 2.5.2 step 3). (d) Final image graph which is connected. Nodes are represented by dots and edges by lines. Edges not present can be solved for by forming $Ax = b$. (e) Original depth map, reconstructed surface using the Poisson solver and using the algebraic approach.



(a) (b)

(c) (d)

Figure 2.9: Photometric Stereo on Yale face database. (a) Four out of sixty four input images. (b) Curl values of the estimated gradient field. (c) Reconstructed depth map using the Poisson solver. Global distortions are present. (d) Reconstructed depth map using our method. Our method exploits information in curl often ignored in gradient reconstruction. This brings in high gradients that are smoothed out in the Poisson solver. In addition, the reconstruction has a local error confinement property so that errors in the gradient field do not create global distortions in the reconstructed surface.

grability, which provides local error confinement. Using this analysis, I describe a generalized equation for robust reconstruction from gradient fields in the following chapter.

Chapter 3

A General Framework for Surface Reconstruction from Gradient Fields

In this chapter, I describe a general framework for surface reconstruction from gradient fields. Firstly, I show that reconstruction from gradient fields can be considered as a robust estimation problem and traditional approaches such as RANSAC [69] are computationally prohibitive due to the high dimensionality of the problem. Then I describe a generalized equation that lead to several new algorithms for feature preserving reconstructions in the presence of noise and outliers. I show that previous approaches such as Poisson solver and Frankot-Chellappa algorithm are special cases of this framework. The key idea behind the generalized equation is to replace gradients by suitable functions of gradients arising from general error functionals. These functions control the degree of anisotropy of the weights assigned to the gradients in the integration process, resulting in feature preserving reconstructions.

3.1 Robust Estimation

Techniques for robust estimation includes the well-known RANSAC algorithm and M-estimators [70]. Firstly, I show that applying RANSAC to gradient integration is computationally prohibitive. To apply RANSAC, we need to find the

minimum number m of gradients required for integration. For example, if we want to estimate a line from 2D points, we need $m = 2$ points.

Proposition 3. *For a surface defined over a $H \times W$ grid, the minimum number of gradients required for integration is $m = HW - 1$. However, integration cannot be done using any such set of m gradients. These m gradients should form a spanning tree of the 2D planar graph defined on the grid.*

Proof. Define a 2D graph over the grid, where the nodes correspond to the value of the surface at each pixel and gradients correspond to the edges (see Figure 2.1). To be able to integrate, each pixel (node) should be reachable using some integration path. Since a spanning tree is a minimal configuration which spans all nodes, the gradients should be in that configuration. For HW nodes, the number of edges in any spanning tree is $HW - 1$, hence $m = HW - 1$.

In Section 2.3, it was shown that the subspace of all gradient fields is of dimension $HW - 1$, by enforcing integrability using curl correction. The above proposition is an alternative argument using graph analogy.

3.1.1 RANSAC Gradient Integration (Computationally Prohibitive)

RANSAC works by randomly selecting a set of minimum data points m and finding the number of inliers using a given tolerance level τ. This is repeated T times and the set having the maximum number of inliers is used to estimate the parameters. A naive RANSAC based approach to surface reconstruction can be as follows:

- Find a random spanning tree of the 2D planar graph on the grid.
- Integrate using the gradients corresponding to the edges in the spanning tree. Find the number of gradient inliers using the solution given an error tolerance τ.
- Repeat T times and choose that spanning tree using which maximum number of inliers are obtained.

In [69], it was shown that to ensure with probability γ that at least one of the random selections is an error-free set of m data points, one must make at least T selections, where T is given by

$$T = \log(1-\gamma)/\log(1-w^m). \tag{3.1}$$

In (3.1), w is the probability that a particular data point is an inlier. However, T becomes extremely large as the size of the grid is increased. For example, assuming $w = 0.95$, even for a 16×16 grid ($m = 255$), to ensure a probability $\gamma = 0.95$, $T = 1.43 * 10^6$. Thus, a random selection process for choosing the inliers set is practically impossible for decent grid sizes.

3.2 A General Framework

A general solution can be obtained by minimizing the following n^{th} order error functional

$$J = \int\int E(Z, p, q, Z_{x^a y^b}, p_{x^c y^d}, q_{x^c y^d}, \ldots)\, dxdy\ , \tag{3.2}$$

where E is a continuous differentiable function, a, b, c and d are non-negative integers such that $a + b = k$, $c + d = k - 1$ for some positive integer k, $Z_{x^a y^b} = \frac{\partial^k Z}{\partial x^a \partial y^b}$,

$p_{x^c y^d} = \frac{\partial^{k-1} p}{\partial x^c \partial y^d}$, $q_{x^c y^d} = \frac{\partial^{k-1} q}{\partial x^c \partial y^d}$ and the above equation includes terms corresponding to all possible combinations of a, b, c and d for all k, $1 \leq k \leq n$. Restricting to first order derivatives ($n = 1$), we will consider error functionals of the form

$$J = \int \int E(Z, p, q, Z_x, Z_y) dx dy. \tag{3.3}$$

The Euler-Lagrange equation gives

$$\frac{\partial E}{\partial Z} - \frac{d}{dx}\frac{\partial E}{\partial Z_x} - \frac{d}{dy}\frac{\partial E}{\partial Z_y} = 0 \quad \texttt{or} \quad \frac{\partial E}{\partial Z} = \texttt{div}(\frac{\partial E}{\partial Z_x}, \frac{\partial E}{\partial Z_y}) \ . \tag{3.4}$$

Consider the following form for $\frac{\partial E}{\partial Z_x}$ and $\frac{\partial E}{\partial Z_y}$

$$\frac{\partial E}{\partial Z_x} = f_1(Z_x, Z_y) - f_3(p, q), \quad \frac{\partial E}{\partial Z_y} = f_2(Z_x, Z_y) - f_4(p, q) \ , \tag{3.5}$$

where $f_i : \mathbf{R} \times \mathbf{R} \rightarrow \mathbf{C}, i = 1 \ldots 4$ are different functions. Note that these functions cannot be arbitrary as they should satisfy $\frac{\partial^2 E}{\partial Z_x \partial Z_y} = \frac{\partial^2 E}{\partial Z_y \partial Z_x}$. This implies that

$$\frac{\partial f_1(Z_x, Z_y)}{\partial Z_y} = \frac{\partial f_2(Z_x, Z_y)}{\partial Z_x} \ . \tag{3.6}$$

Substituting (3.5) into (3.4) and bringing all Z terms on one side, we get the **generalized equation**

$$\texttt{div}(f_1(Z_x, Z_y), f_2(Z_x, Z_y)) - \frac{\partial E}{\partial Z} = \texttt{div}(f_3(p, q), f_4(p, q)) \ . \tag{3.7}$$

The generalized equation replaces the gradients by functions of gradients. In the next section, I show that previous solutions such as the Poisson solver and Frankot-Chellappa algorithm (in general, projection onto continuous basis functions) can be derived using the generalized equation (3.7). Then I present novel solutions using the above equation. In all solutions, I assume Neumann boundary conditions given by $\nabla Z \cdot \widehat{\mathbf{n}} = 0$.

3.3 Previous Solutions as Special Cases

3.3.1 Poisson Solver (Spatially Invariant Isotropic Weights)

The Poisson equation

$$\texttt{div}(Z_x, Z_y) = \texttt{div}(p, q). \tag{3.8}$$

can be obtained from (3.7) by substituting (see Table 3.1)

$$\begin{cases} \frac{\partial E}{\partial Z} = 0 \\ f_1(Z_x, Z_y) = Z_x \\ f_2(Z_x, Z_y) = Z_y \\ f_3(p, q) = p \\ f_4(p, q) = q. \end{cases} \tag{3.9}$$

Equation (3.6) is satisfied as both sides are zero. Thus, Poisson equation is the simplest possible case of the generalized equation, where the functions f_i's do not modify the gradients.

3.3.1.1 Numerical Solution

Let $u = \texttt{div}(p, q)$. Using finite differences and vectoring the 2D matrices in lexicographical order, the Poisson equation can be discretized [29] to give

$$\mathbf{LZ} = \mathbf{u}, \tag{3.10}$$

where $\mathbf{u} = [u(1,1), \ldots, u(H,W)]^T$ and the matrix $\mathbf{L}$ is the sparse Laplacian matrix[1] of size $HW \times HW$. Each row of $\mathbf{L}$ has -4 at the diagonal entry and four 1's corresponding to the isotropic Laplacian kernel ∇^2. Z can be obtained as

$$\mathbf{Z} = \mathbf{L}^{-1}\mathbf{u}. \tag{3.11}$$

3.3.2 Reconstruction using Basis Functions

Frankot-Chellappa (FC) algorithm reconstructs the surface Z by projecting $\{p, q\}$ on the set of integrable Fourier basis functions. Let $\mathcal{F}(s(x,y))$ denote the Fourier transform of $s(x,y)$[2]. Given $\{p,q\}$, Z is obtained as [30]

$$Z = \mathcal{F}^{-1}(-j\frac{\xi_x\mathcal{F}(p) + \xi_y\mathcal{F}(q)}{\xi_x^2 + \xi_y^2}) \ . \tag{3.12}$$

Let $\phi(x, y, \xi_x, \xi_y) = e^{j(\xi_x x + \xi_y y)}$. We have

$$\phi_x = j\xi_x\phi, \qquad \phi_y = j\xi_y\phi. \tag{3.13}$$

Substituting

$$\begin{cases} \frac{\partial E}{\partial Z} = 0 \\ f_1(Z_x, Z_y) = \mathcal{F}(Z_x)\phi \\ f_2(Z_x, Z_y) = \mathcal{F}(Z_y)\phi \\ f_3(p,q) = \mathcal{F}(p)\phi \\ f_4(p,q) = \mathcal{F}(q)\phi \end{cases} \tag{3.14}$$

[1] The Laplacian matrix needs to be modified at the boundary according to the boundary conditions.

[2] $\mathcal{F}(s(x,y)) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} s(x,y)e^{-j(\xi_x x + \xi_y y)}dxdy$

in (3.7), we get

$$\begin{aligned}
&\texttt{div}(\mathcal{F}(Z_x)\phi, \mathcal{F}(Z_y)\phi) = \texttt{div}(\mathcal{F}(p)\phi, \mathcal{F}(q)\phi) \ , \\
\therefore \quad & j\xi_x\mathcal{F}(Z_x) + j\xi_y\mathcal{F}(Z_y) = j\xi_x\mathcal{F}(p) + j\xi_y\mathcal{F}(q) \ , \\
\therefore \quad & -(\xi_x^2 + \xi_y^2)\mathcal{F}(Z) = j\left(\xi_x\mathcal{F}(p) + \xi_y\mathcal{F}(q)\right) \\
\therefore \quad & \mathcal{F}(Z) = \frac{-j\left(\xi_x\mathcal{F}(p) + \xi_y\mathcal{F}(q)\right)}{(\xi_x^2 + \xi_y^2)} \ .
\end{aligned}$$

which is equivalent to (3.12). The *projection* on the Fourier basis functions is implicit in the above definition of f_i's which transforms the gradients as weighted basis functions ϕ, the weights being equal to the Fourier transform coefficients. One can generalize this approach to use any set of ortho-normal basis functions ϕ. Kovesi's [55] algorithm is in a similar spirit while using a redundant set of non-orthogonal basis functions.

In the next section, I show how these functions f_i's can be changed to obtain a continuum of solutions. Intuitively, in solving the Poisson equation, the Laplacian matrix $\mathbf{L}$ is obtained by using a *spatially invariant isotropic* kernel (∇^2) which gives equal weights to gradients. This results in the Poisson solver being non-robust and favoring smoothness. To obtain robust solutions, we modify the Laplacian matrix by using *spatially varying anisotropic* kernel depending on local shape, or residual gradient field.

3.4 New Solutions

Now I present several new algorithms based on the previous framework, by changing the degree of anisotropy of weights applied to the gradients in the integra-

tion process.

3.4.1 $\boldsymbol{\alpha}$-surface: Anisotropic Scaling using Binary Weights

In Section 3.1.1, it was shown that random selection process for the set of inlier gradients is computationally prohibitive due to the high dimensionality of the gradient integration problem. If there is a problem related rationale for choosing the set of inliers, one could use a deterministic selection process instead of a random one as argued in [69]. In a general estimation problem like fitting a line, each data point is independent and there are no *structural constraints*. For 2D integration, integrability enforces a structural constraint. Also, since the goal is to fit a surface, there is an inherent smoothness involved (at regions separated by discontinuities). Thus, one can decide an initial spanning tree using a deterministic process (as shown in Section 3.4.1.1).

Suppose we decide an initial spanning tree, claiming all gradients corresponding to the edges in this spanning tree to be inliers. We define α-surface as an iterative scheme, where at each iteration, based on the tolerance level α, all gradients for which the residual term is less than α are added to the inliers set. Formally, let S denote the set containing the gradients corresponding to the edges in the initial spanning tree. For an $\alpha \geq 0$, α-surface is given by

- Initialize: Integrate using the gradients in the set S to get Z^0. $k \leftarrow 1$.
- At iteration k: Compute Z_x, Z_y as the gradients of Z^{k-1}.
- If $|\epsilon_x| = |Z_x - p| \leq \alpha$ and Z_x not in S, add Z_x to set S. If $|\epsilon_y| = |Z_y - q| \leq \alpha$

and Z_y not in S, add Z_y to set S. Let n be the number of new additions to set S.

- Integrate using the gradients in S to obtain Z^k.
- Terminate if $n = 0$, else $k \leftarrow k + 1$.

Note that the gradients are not removed from S in the above scheme because the minimal configuration of spanning tree must be satisfied. The parameter α decides between outliers and inliers. If $\alpha = 0$, gradients corresponding to the edges in the initial spanning tree are considered as inliers and are used for integration. As α is increased, more gradients are used for integration. At a large value of α, all gradients will be treated as inliers and the solution becomes equivalent to that given by the Poisson solver. **By changing α, one can trace a path in the solution space**, where one end is the solution based on a minimal data configuration and the other end is the solution based on using all the data. Thus, α-surface is a weighted approach, where the weights are 1 for gradients in S (used for integration) and 0 otherwise. If we define

$$b_x(x,y) = \begin{cases} 1 & \text{if } Z_x \in S, \\ 0 & \text{otherwise.} \end{cases} \qquad b_y(x,y) = \begin{cases} 1 & \text{if } Z_y \in S, \\ 0 & \text{otherwise,} \end{cases} \tag{3.15}$$

then the error functional J for each iteration of α-surface can be written as

$$J = \int\int b_x(Z_x - p)^2 + b_y(Z_y - q)^2 dxdy \ . \tag{3.16}$$

The corresponding Euler-Lagrange equation is

$$\mathtt{div}(b_x Z_x, b_y Z_y) = \mathtt{div}(b_x p, b_y q). \tag{3.17}$$

Thus, the gradient fields $\{Z_x, Z_y\}$ and $\{p, q\}$ are *scaled* using the binary weights b_x and b_y in an anisotropic manner.

3.4.1.1 Determining Initial Spanning Tree

An easy way to obtain an initial spanning tree is to assign a weight to each edge and find the minimum spanning tree (MST). In Section 2.5, we used the curl values as weights on the edges to connect the graph by finding the set of links with minimum total weight. We experimented with two types of edge weights: one based on curl values and other based on gradient magnitude. We found that assigning gradient magnitude as weights gives better results compared to curl values. For results presented in Sect. 3.5, we use gradient magnitude as weights.

3.4.1.2 Determining α

Suppose that the gradients are corrupted by additive Gaussian noise (independent and identically distributed) $\mathbf{N}(0, \sigma^2)$. In discrete domain, curl values can be obtained by considering the smallest loop made up of 4 square connected pixels, $(y, x), (y, x+1), (y+1, x)$ and $(y+1, x+1)$ (see Fig. 2.1(middle)). The integral along this loop is

$$C_{p,q}(y, x) = p(y+1, x) - p(y, x) + q(y, x) - q(y, x+1) \ . \tag{3.18}$$

Using the above equation, the mean and variance of $C_{p,q}$ will be 0 and $4\sigma^2$ respectively (in practice, variance can be higher due to outliers). We estimate σ as

$$\sigma = \sqrt{(\sigma_C^2)/4}, \tag{3.19}$$

where (σ_C^2) denote the estimated variance of $C_{p,q}$ using the given gradient field $\{p, q\}$. We use $\alpha = 1.5\sigma$.

3.4.1.3 Numerical Solution

Let $u_b = \texttt{div}(b_x p, b_y q)$. $\texttt{div}(b_x Z_x, b_y Z_y)$ can be written as $\nabla_b^2 Z$, where ∇_b^2 is the *weighted Laplacian kernel* (Fig. 3.1, ∇_w^2 with b's as weights). This weighted kernel is applied at each pixel to calculate the weighted Laplacian matrix $\mathbf{L}_b$ and the weighted divergence u_b. Z is obtained as

$$\mathbf{Z} = \mathbf{L}_b^{-1}\mathbf{u}_b. \tag{3.20}$$

Note that the matrix $\mathbf{L}_b$ is guaranteed to be invertible since the set S contains the gradients corresponding to some spanning tree (minimal configuration). Next I show how to generalize the inlier/outlier weighting scheme to approaches based on continuous weights.

3.4.2 Anisotropic Scaling using Continuous Weights

3.4.2.1 M-estimators

M-estimators reduce the effect of outliers by replacing the squared error residual $\rho(.) = (.)^2$ by another function of residuals. Here ρ is a symmetric, positive-definite function with a unique minimum at zero, and is chosen to be less increasing than square. Several functions such as Huber, Cauchy, Tuckey and those based on L^p norm have been proposed [71, 70]. The error function corresponding to M-estimators

can be minimized using an iterative re-weighted least squares approach [72], where at each iteration a new error function is defined as

$$J = \int\int w(\epsilon_x^{k-1})(Z_x - p)^2 + w(\epsilon_y^{k-1})(Z_y - q)^2 dxdy \ . \tag{3.21}$$

The weights ($w_x = w(\epsilon_x^{k-1}), w_y = w(\epsilon_y^{k-1})$) at iteration k depends on the residual at iteration $k-1$ using the function ρ. The Euler-Lagrange equation of (3.21) gives

$$\texttt{div}(w_x Z_x, w_y Z_y) = \texttt{div}(w_x p, w_y q). \tag{3.22}$$

This is similar to the α-surface method described above, except that the weights are continuous. Z can be obtained as

$$\mathbf{Z} = \mathbf{L}_w^{-1}\mathbf{u}_w. \tag{3.23}$$

3.4.2.2 Regularization

Ill-posed problems in computer vision are often solved by **regularization** (e.g. estimating optical flow [73]). The Poisson solver can be regularized by modifying the error function as

$$J(Z) = \int\int \left((Z_x - p)^2 + (Z_y - q)^2\right) + \lambda(\phi(Z_x) + \phi(Z_y))dxdy \ , \tag{3.24}$$

where the second term is the regularization term using the function ϕ. Common examples include $\phi(s) = \sqrt{1+s^2}$ and $\phi(s) = \log(1+s^2)$ [66, 65]. The Euler-Lagrange equation of the above error functional gives

$$\texttt{div}(Z_x, Z_y) + (\lambda/2)\texttt{div}(\phi'(Z_x), \phi'(Z_y)) = \texttt{div}(p, q). \tag{3.25}$$

In terms of (3.7), this corresponds to (Table 3.1)

$$\begin{cases} \frac{\partial E}{\partial Z} = 0 \\ f_1(Z_x, Z_y) = Z_x + \frac{\lambda}{2}\phi'(Z_x) \\ f_2(Z_x, Z_y) = Z_y + \frac{\lambda}{2}\phi'(Z_y) \\ f_3(p, q) = p \\ f_4(p, q) = q. \end{cases} \tag{3.26}$$

Minimizing (3.24) is difficult due to the error functional being non-linear. Using the principle of half-quadratic minimization (see Appendix A and [65] for details), one can introduce auxiliary variables $w = (w_x, w_y)$. Minimizing (3.24) is then equivalent to the following iterative minimization

- $Z^0 \equiv 0$. $k \leftarrow 1$. Repeat until convergence
- $w_x^k = \phi'(Z_x^{k-1})/(2Z_x^{k-1})$, $w_y^k = \phi'(Z_y^{k-1})/(2Z_y^{k-1})$
- Solve for Z^k: $\quad \nabla^2 Z^k + \lambda \mathtt{div}(w_x^k Z_x^k, w_y^k Z_y^k) = \mathtt{div}(p, q)$

The equation for solving Z^k can be rewritten as

$$(\nabla^2 + \lambda \nabla_{w^k}^2) Z^k = \mathtt{div}(p, q), \tag{3.27}$$

where $\nabla_{w^k}^2$ is the weighted Laplacian kernel (Fig. 3.1). The solution is given by

$$\mathbf{Z}^k = (\mathbf{L} + \lambda \mathbf{L}_{w^k})^{-1} \mathbf{u}. \tag{3.28}$$

3.4.3 Affine Transformation of Gradients using Diffusion Tensors

Image restoration from noisy images has been a classical problem in image processing. Anisotropic diffusion [37] and energy minimization methods [65, 66] are some of the common approaches for image restoration. Weickert [38] proposed a generalization of divergence based equation for image restoration, given by

$$I_t = \texttt{div}(D\nabla I), \tag{3.29}$$

where

$$D(y,x) = \begin{bmatrix} d_{11}(y,x) & d_{12}(y,x) \\ d_{21}(y,x) & d_{22}(y,x) \end{bmatrix} \tag{3.30}$$

is a 2×2 symmetric, positive-definite matrix at each pixel (a field of diffusion tensors). In image restoration, image intensities are modified using the above equation. For the problem of surface reconstruction, let us consider generalizing the Poisson equation using D as

$$\texttt{div}(D \begin{bmatrix} Z_x \\ Z_y \end{bmatrix}) = \texttt{div}(D \begin{bmatrix} p \\ q \end{bmatrix}) \ . \tag{3.31}$$

The above equation is the Euler-Lagrange equation of the following valid error functional

$$J(Z) = \int\int d_{11}(Z_x - p)^2 + (d_{12} + d_{21})(Z_x - p)(Z_y - q) + d_{22}(Z_y - q)^2 dxdy \tag{3.32}$$

and can be written as

$$\texttt{div}(d_{11}Z_x + d_{12}Z_y, d_{21}Z_x + d_{22}Z_y) = \texttt{div}(d_{11}p + d_{12}q, d_{21}p + d_{22}q) \ . \tag{3.33}$$

Note that (3.33) can be obtained from (3.7) by substituting (Table 3.1).

$$\begin{cases} \frac{\partial E}{\partial Z} = 0 \\ f_1(Z_x, Z_y) = d_{11} Z_x + d_{12} Z_y \\ f_2(Z_x, Z_y) = d_{21} Z_x + d_{22} Z_y \\ f_3(p, q) = d_{11} p + d_{12} q \\ f_4(p, q) = d_{21} p + d_{22} q \end{cases} \tag{3.34}$$

Thus, Diffusion corresponds to the function f_i's being **affine** in their arguments. The gradients are scaled *and* linearly combined. The symmetry of the tensor D comes directly from the fact that (3.6) must be satisfied, leading to $d_{21} = d_{12}$. The positive-definiteness criteria is required to avoid ill-conditioning in the numerical solution obtained from discretization. Although, I loosely call this scheme as Diffusion, there is no notion of time or iteration in this scheme. Let

$$u_D = \texttt{div}(d_{11} p + d_{12} q, d_{21} p + d_{22} q). \tag{3.35}$$

Equation (3.33) can be written as

$$\nabla_D^2 Z = u_D, \tag{3.36}$$

where ∇_D^2 denotes the weighted Laplacian kernel based on the diffusion tensor D (Fig. 3.1). The solution is given by

$$\mathbf{Z} = \mathbf{L}_D^{-1} \mathbf{u}_D \tag{3.37}$$

3.4.3.1 Obtaining Diffusion Tensor

Several schemes for obtaining diffusion tensor that preserves edges [66](Eq. 3.60) and coherence [38] have been proposed. We use an edge-preserving diffusion tensor similar to edge preserving tensor defined in [66] as follows. At each pixel, we first obtain a 2×2 matrix H by convolving component wise $\begin{bmatrix} p^2 & pq \\ pq & q^2 \end{bmatrix}$ with a Gaussian kernel $K_\sigma(x,y) = \frac{1}{2\pi\sigma^2}\exp(-\frac{x^2+y^2}{2\sigma^2})$.

$$H(x,y) = \begin{bmatrix} A(x,y) & B(x,y) \\ B(x,y) & C(x,y) \end{bmatrix} \tag{3.38}$$

where

$$\begin{aligned} A(x,y) &= \int\int p(x-u,y-v)^2 K_\sigma(u,v)dudv \\ B(x,y) &= \int\int p(x-u,y-v)q(x-u,y-v)K_\sigma(u,v)dudv \\ C(x,y) &= \int\int q(x-u,y-v)^2 K_\sigma(u,v)dudv \end{aligned} \tag{3.39}$$

Let $\mu_1 \geq \mu_2$ denote the eigen-values of H and $\mathbf{v}_1$, $\mathbf{v}_2$ denote its eigen-vectors.

$$H = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 \end{bmatrix} \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \end{bmatrix}. \tag{3.40}$$

The eigen-values of H are modified following [66](Eq. 3.60).

$$\begin{cases} \lambda_2 = 1 \\ \lambda_1 = \begin{cases} 1 & \text{if } \mu_1 = 0, \\ \beta + 1 - \exp(-3.315/\mu_1^4) & \text{if } \mu_1 > 0 \end{cases} \end{cases} \tag{3.41}$$

D is obtained from the eigen-vectors of H and the new eigen-values as

$$D = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_1 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \end{bmatrix}. \tag{3.42}$$

Here $\beta = 0.02$ to ensure positive-definiteness of D.

The tensor D transforms the gradient field so as to remove large gradients. Note that the left hand side of (3.31) is also modified by D. Intuitively, reconstruction is done only using the low gradients information. For example, consider a horizontal ramp in the surface starting from depth value zero to a large depth value. One can integrate *across* the ramp using large x gradients, or one can integrate *along* the ramp using smaller x gradients. The affine transformation approach effectively integrates along the ramp by transforming the gradient field to remove large gradients.

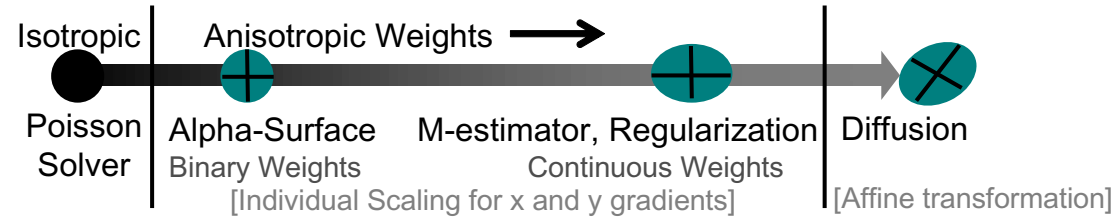
3.4.4 Discussion

The weighted solutions can be explained as follows. In a weighted solution, the Laplacian matrix is obtained using a *spatially varying anisotropic* kernel based on weights. This is in contrast with a spatially invariant isotropic kernel used in the Poisson equation. for M-estimators, the weights depend on the residual error, while in Diffusion and Regularization, it depends on the underlying surface. In Regularization, the divergence $\mathbf{u}$ of the given gradient field is not modified according to the weights, while it is modified in other approaches (see Table 3.1).

3.4.4.1 Exactness of the Solution

Let us compare the algorithms in terms of exactness of the solution. By exactness, we mean that if there was no noise in $\{p, q\}$, i.e., if $Z_x = p$ and $Z_y = q$, will the reconstruction algorithm give the original Z back (up to an unknown constant)? This can be analyzed using the error function for different approaches. The LS based Poisson solver is exact because if $Z_x = p$ and $Z_y = q$, the error function $J = 0$. Similarly, M-estimators, α-surface and affine transformation are exact. However, Regularization is not exact as the original cost function is modified. For $\lambda > 0$, even if $Z_x = p$ and $Z_y = q$, $J \geq 0$ for Regularization. The Regularization approach does not give good solutions as compared to other approaches as it does not minimize the correct error function.

Although I described a range of solutions, the choice of using a particular algorithm for a given application remains an open problem. In general, for smooth surfaces without discontinuities, LS approaches may give good solutions while handling noise. With sharp features on a surface, diffusion and alpha-surface gives better feature preserving reconstructions in the presence of noise and outliers. In all the above solutions, $\frac{\partial E}{\partial Z} = 0$. This framework could also be used when the Z values are known at some control points or while combining sparse depth and orientation information to obtain a surface [74, 75, 76] by utilizing the $\frac{\partial E}{\partial Z}$ term.



$$\nabla^2 \equiv \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad \nabla_w^2 \equiv \begin{bmatrix} 0 & w_y(y-1,x) & 0 \\ w_x(y,x-1) & -\sum & w_x(y,x) \\ 0 & w_y(y,x) & 0 \end{bmatrix}$$

$$\nabla_D^2 \equiv$$

$$\begin{bmatrix} 0 & d_{22}(y-1,x)+d_{21}(y-1,x) & -d_{21}(y-1,x) \\ d_{11}(y,x-1)+d_{12}(y,x-1) & -\sum & d_{11}(y,x)+d_{21}(y,x) \\ -d_{12}(y,x-1) & d_{22}(y,x)+d_{12}(y,x) & 0 \end{bmatrix}$$

∇^2: Isotropic kernel, ∇_w^2: Anisotropic kernel, ∇_D^2: Diffusion kernel

Figure 3.1: A continuum of solutions can be derived by changing f_i's in the generalized equation (3.7). At one end is the Poisson solver which gives equal weights to all the gradients, resulting in a spatially invariant isotropic Laplacian kernel ∇^2. *Individual scaling* of the gradients using spatially varying weights (binary for α-surface, continuous for M-estimator and Regularization) results in anisotropic kernel ∇_w^2 ($\sum$ denotes the sum of neighboring values). In Diffusion, x and y gradients are scaled *and* linearly combined, resulting in an affine transformation of gradients. This results in diffusion kernel ∇_D^2.

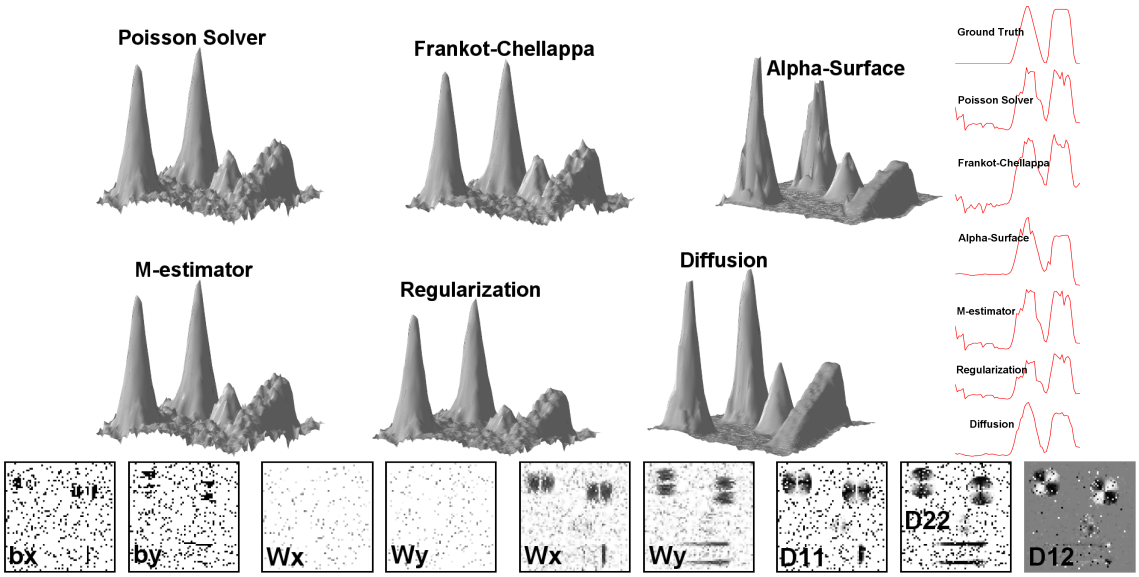


Figure 3.2: Reconstruction in the presence of noise and outliers (Ramp-Peaks). (Top two rows) (Left) Reconstructed surfaces using various algorithms. (Right) One-D height plots for a scan line across the middle of grid for various solutions. (Bottom row) x and y gradient weights for the last iteration of α-surface, M-estimator & Regularization. Last three images shows d_{11}, d_{22} & d_{12} for Diffusion. (white= 1, black= 0) except for d_{12} (white= 0.5, black= -0.5). Notice that α-surface and Diffusion give much better results compared to other approaches.

3.5 Experimental Validation

We compare[3] Poisson solver, Frankot-Chellappa (FC) algorithm, α-surface, M-estimator using Huber function, Diffusion and Regularization using $\phi(s) = \sqrt{1+s^2}$, $\lambda = 10$. Table 3.1 gives summary of the functions f_i's and the equation for each algorithm.

3.5.1 Ramp-Peaks

Figure 3.2 shows the reconstructed surfaces using various algorithms from the noisy gradient field of the synthetic surface shown in Figure 2.3. Note that the

[3]Matlab code is available at `http://www.umiacs.umd.edu/~aagrawal/`, [77].

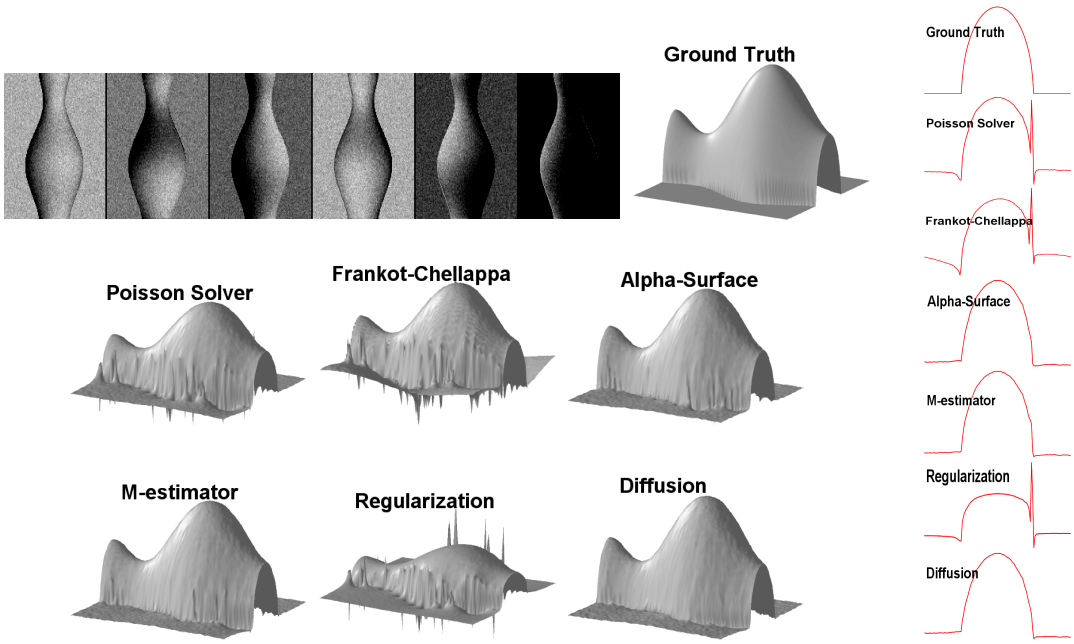


Figure 3.3: Photometric Stereo on Vase. (Top row) Noisy input images and the true surface. (Next two rows) Reconstructed surfaces using various algorithms. (Right column) One-D height plots for a scan line across the middle of Vase. Better results are obtained using α-surface, Diffusion and M-estimator as compared to Poisson solver, FC and Regularization.

surface reconstructed using α-surface and Diffusion are much better than those reconstructed using other approaches.

We also present results on calibrated photometric stereo using synthetic and real images. The synthetic images were generated using the Lambertian reflectance model under distant point light sources. We first estimate the surface normals (n_x, n_y, n_z) at each pixel. The gradient field is then obtained as $p = -n_x/n_z$, $q = -n_y/n_z$. Pixels where the surface normal cannot be estimated (being in shadow in most of the images) give rise to outliers. Table 3.2 gives the mean square error between the estimated surface and the true surface for synthetic datasets using various algorithms, along with some of the properties of these algorithms.

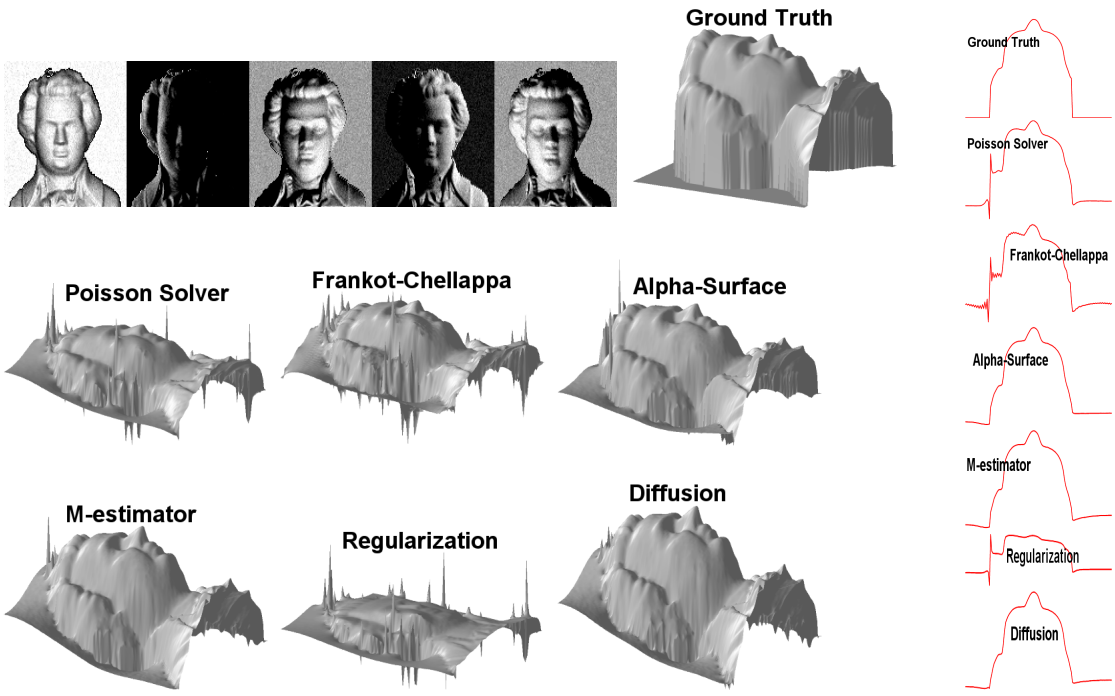


Figure 3.4: Photometric Stereo on Mozart. Top row shows noisy input images and the true surface. Next two rows show the reconstructed surfaces using various algorithms. (Right Column) One-D height plots for a scan line across the Mozart face. Notice that all the features of the face are preserved in the solution given by α-surface, Diffusion and M-estimator as compared to other algorithms.

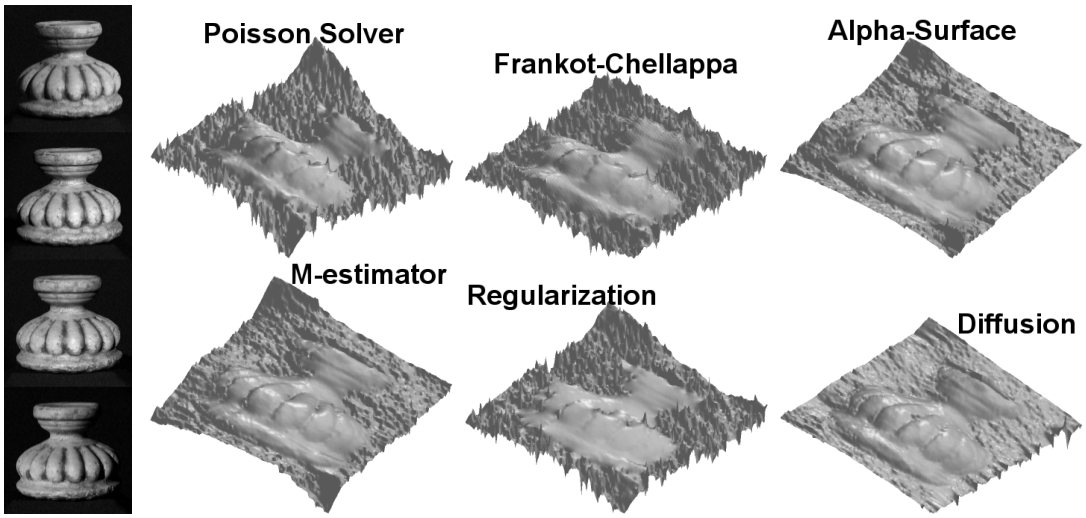


Figure 3.5: Photometric Stereo on Flowerpot. Left column show four real images of a flowerpot. Right column show the reconstructed surfaces using various algorithms. The reconstructions using Poisson solver and FC algorithm are noisy and all features (such as fine variations on the top of the flowerpot) are not recovered. Diffusion, α-surface and M-estimator methods discount noise while recovering all the salient features.

3.5.2 Vase

Six images generated using the Vase depth map are shown in Figure 3.3. We add Gaussian random noise (σ = 10% of maximum intensity) to the images. In addition, we also add small amount of uniformly distributed noise to the light source directions. The reconstructed surfaces using various algorithms are shown in Figure 3.3. α-surface, Diffusion and M-estimator give better shape estimates compared to the rest of algorithms.

3.5.3 Mozart

Five images generated using the Mozart depth map are shown in Figure 3.4. Gaussian random noise (σ = 5% of maximum intensity) was added to the images.

Table 3.1: A continuum of solutions can be obtained by changing f_i's in the generalized equation (3.7), which control the anisotropy of the weights applied to the gradients. In weighted solutions, the Laplacian matrix is obtained using a *spatially varying anisotropic* kernel based on weights. This is in contrast with a spatially invariant isotropic kernel used in the Poisson solver. In M-estimators, the weights depend on the residual error, while in Diffusion and Regularization, they depend on the underlying surface.

Algorithm	f_i's corresponding to (3.7), $\frac{\partial E}{\partial Z} = 0$				Equation
	$f_1(Z_x, Z_y)$	$f_2(Z_x, Z_y)$	$f_3(p, q)$	$f_4(p, q)$	
Poisson solver	Z_x	Z_y	p	q	$\mathbf{LZ} = \mathbf{u}$
Frankot-Chellappa	$\mathcal{F}(Z_x)\phi$	$\mathcal{F}(Z_y)\phi$	$\mathcal{F}(p)\phi$	$\mathcal{F}(q)\phi$	(3.12)
$\boldsymbol{\alpha}$-surface	$b_x Z_x$	$b_y Z_y$	$b_x p$	$b_y q$	$\mathbf{L}_b\mathbf{Z} = \mathbf{u}_b$
M-estimators	$w_x Z_x$	$w_y Z_y$	$w_x p$	$w_y q$	$\mathbf{L}_w\mathbf{Z} = \mathbf{u}_w$
Regularization	$Z_x + \frac{\lambda}{2}\phi'(Z_x)$	$Z_y + \frac{\lambda}{2}\phi'(Z_y)$	p	q	$(\mathbf{L} + \lambda\mathbf{L}_w)\mathbf{Z} = \mathbf{u}$
Diffusion	$d_{11}Z_x + d_{12}Z_y$	$d_{21}Z_x + d_{22}Z_y$	$d_{11}p + d_{12}q$	$d_{21}p + d_{22}q$	$\mathbf{L}_D\mathbf{Z} = \mathbf{u}_D$

Table 3.2: Mean square errors (MSE) for synthetic data sets for various algorithms along with some of their properties.

	Poisson-solver	FC	α-surface	M-estimator	Regularization	Diffusion
Ramp-Peaks	10.81	11.20	2.65	9.49	5.35	2.26
Vase	294.46	239.62	22.20	15.14	164.98	2.78
Mozart	2339.24	1316.66	219.72	359.12	806.85	373.72
Exact	Yes	Yes	Yes	Yes	No	Yes
Iterative	No	No	Yes	Yes	Yes	No
MATLAB Run-time (128×128 grid)	0.09s	0.03s	4.03s	24.57s	16.75s	4.67s

The reconstructed surfaces using various algorithms are also shown in Figure 3.4. While the discontinuities in the shape are smeared in solutions using the Poisson solver, FC and Regularization, these are preserved in reconstructions obtained from α-surface, Diffusion and M-estimator approaches.

3.5.4 Flowerpot

Figure 3.5 shows results on calibrated photometric stereo using four real images of a flowerpot. Notice that LS solutions (Poisson solver and FC algorithm) are noisy and do not recover all features (such as the fine variations on the top of the flowerpot). Diffusion, α-surface and M-estimator approaches recovers all salient features while discounting noise.

3.6 Summary

In this chapter, I presented a general framework for surface reconstruction from gradient fields, based on controlling the anisotropy of weights assigned to gradients during the integration. I showed that previous solutions such as Poisson solver and Frankot-Chellappa algorithm are special cases of this framework. Using the generalized equation, I derived a continuum of solvers: α-surface (binary weights), where α allows tradeoff between smoothness and robustness, Regularization and M-estimators (continuous weights) and Diffusion (affine transformation on gradients). The new approaches give consistently better feature preserving reconstructions in the presence of noise and outliers.

Chapter 4

Edge Suppression by Gradient Field Manipulations

In this chapter, I present a framework for edge suppressing operation on images by manipulating image gradient fields. Edge-suppression can be a useful tool in a class of problems. These problems involve analyzing images of the same scene under variable illumination. Traditionally, edge suppression is achieved by setting image gradients to zero based on thresholds. A common application is in the Retinex problem [24, 28], where the illumination map is recovered by suppressing the reflectance edges, assuming that it is slowly varying.

I present two new approaches for edge-suppressing operations on images: (a) gradient projection and (b) affine transformation of image intensity gradient fields. I introduce **cross-projection tensors** for affine transformations. The key idea is to utilize the information from the direction of the intensity gradients, in images captured under varying illumination. These approaches can avoid hard thresholds and smoothness assumptions on reflectance/illumination and can be used for image gradient field manipulations. These approaches are demonstrated in the context of several applications such as (a) recovering the foreground layer under varying illumination, (b) estimating intrinsic images in non-Lambertian scenes, (c) removing shadows from color images and obtaining the illumination map, and (d) removing glass reflections.

4.1 Introduction

Image formation depends on shape and reflectance of the objects in the scene and scene illumination. Scene analysis involves, for example, factoring the image to recover the reflectance or illumination map. In techniques that use local per-pixel operations, a common approach is to preserve (or suppress) image gradients at known locations so that in the recovered map, corresponding edges and textures are preserved (or suppressed). For instance, the Retinex algorithm by Land and McCann [27] assumes reflectance to be piece-wise constant (Mondrian scenes) and illumination to be smooth. Horn [24] proposed to manipulate the image gradient field under these assumptions, by setting large derivatives corresponding to the reflectance edges to zero using thresholds. By integrating the modified gradient field, one can recover the illumination map.

However, a single threshold for the entire image cannot account for illumination and reflectance variations across the image. The approaches presented here provide a principled way of removing scene texture edges from images as compared to thresholding (or zeroing the corresponding gradients). No assumptions on ambient lighting and smoothness of the reflectance or illumination maps (as in [24]) are made and explicit shadow masks (as in [31]) are not used.

Scene analysis from a single image is a challenging task. We use more than one image under variable illumination for recovering the maps and show techniques that work under natural as well as active illumination variations. For natural illumination, we use an approach proposed by Weiss [32], which uses multiple images for

estimating intrinsic images and improve on the estimation of illumination maps. For active illumination, we use the attached flash unit in digital cameras to introduce additional illumination in the scene. These additional images are used to extract reliable information about scene texture edges, thus avoiding hard thresholds and assumptions on smoothness of reflectance/illumination maps.

4.1.1 Related Work

Intrinsic images were proposed as a useful mid-level scene description by Barrow and Tenenbaum [78]. The observed image is considered to be the product of a reflectance image and an illumination image [32, 24]. Decomposing a given image into intrinsic images is an ill-posed problem. Funt *et al.* [28] extended the Retinex problem to color images, again using thresholds but correcting for the non-zero curl of the modified gradient field. Kimmel *et al.* [25] proposed a variational approach assuming reflectance is smooth and illumination is close to input images. However, smoothness of illumination is not valid at shadows and specularities. A learning based approach was used in [33] to separate illumination and reflectance edges. However, the training depends on the given illumination direction and is difficult to generalize to other directions. Impressive results were shown by Finlayson *et al.* [31] for removing shadows from a single color image, by projecting the 2D log-chromaticities along an invariant direction. However, their approach requires imaging under Planckian lights (daylight is a close approximation). In addition, they have an explicit shadow mask for zeroing edges corresponding to shadows.

Recently, Weiss [32] proposed to use multiple images of a scene under changing illumination for estimating intrinsic images. A probabilistic approach, based on maximum-likelihood (ML) estimation was proposed in [32], assuming the scene to be Lambertian. However, for **non-Lambertian scenes**, the estimated reflectance image does not accurately represent the scene reflectance and some portion of the scene reflectance will be included in the illumination images [35]. Matsushita *et al.* [35] proposed to remove the scene texture edges from the illumination images using a manually specified threshold. The approach presented here provides a natural way to remove scene texture edges from the intensity image by transforming the image gradient field, thus avoiding the thresholding altogether.

Background subtraction is used to segment moving regions in image sequences taken from a static camera [79, 80]. There exists vast literature on background modeling using adaptive/non-adaptive Gaussian mixture models and its variants. See review by Piccardi [81] and references therein. Layer separation in the presence of motion has been discussed in [82, 83]. I show how mutual edge-suppression can be effectively used for **foreground extraction** of opaque layers. The presented gradient-based approach relies on local structure rather than absolute intensities and can handle significant illumination variations across images. This approach can also be used to remove complex scene structures such as **reflection layers** due to glass. For example, while photographing through glass, flash images usually have undesirable reflections of objects in front of the glass. Methods for reflection removal include changing polarization or focus [84, 85], Independent Component Analysis (ICA) [86] and using local features [87].

Local structure tensors and **diffusion tensors** derived from them have been used for spatio-temporal image processing and optical flow [88], and PDE based image regularization [66, 50, 89, 90]. These approaches are based on modifying the image intensities using the non-linear diffusion equation

$$I_t = \mathtt{div}(D\nabla I), \tag{4.1}$$

where `div` denotes the divergence operator, ∇I is the image gradient and D denotes the diffusion tensor. In comparison, my approach is a gradient domain approach based on transforming the gradient field ∇I using suitably defined projection tensors.

4.2 Gradient Projection

I first explain the technique of **gradient projection** (GP) and then show that it can also be described by an affine transformation of the gradient field. The goal is to capture properties of intensity images that remain invariant under change of illumination (**illumination invariants**). We use the observation that the orientation of image gradients remains stable under variable illumination, if the gradients are due to local changes in reflectance and geometric shape and not due to local changes in illumination [91, 92]. Chen *et al.* [44] showed that for an object with Lambertian reflectance, discriminative functions that are invariant to illumination do not exist. However, to a large extent, the direction of the image gradients is insensitive to changes in illumination. To utilize information from multiple images, we use the *direction of the image gradients as a constraint* for gradient field ma-

nipulation. This is achieved by taking a **vector projection** of the image gradient vector from one image onto the corresponding image gradient vector of another image. This projection results in a gradient field that preserves common edges in both the images.

Suppose A and B denote the two input images and ∇A, ∇B denote the gradient field of these images. Let $\rightarrow$ denote the operation of taking the vector projection. The modified (projected) gradient field $\nabla A'$ is obtained from ∇A and ∇B as

$$\nabla A' = \nabla A \rightarrow \nabla B \tag{4.2}$$

The projected gradient field constraints the direction of the gradients in image A to be the same as those in image B.

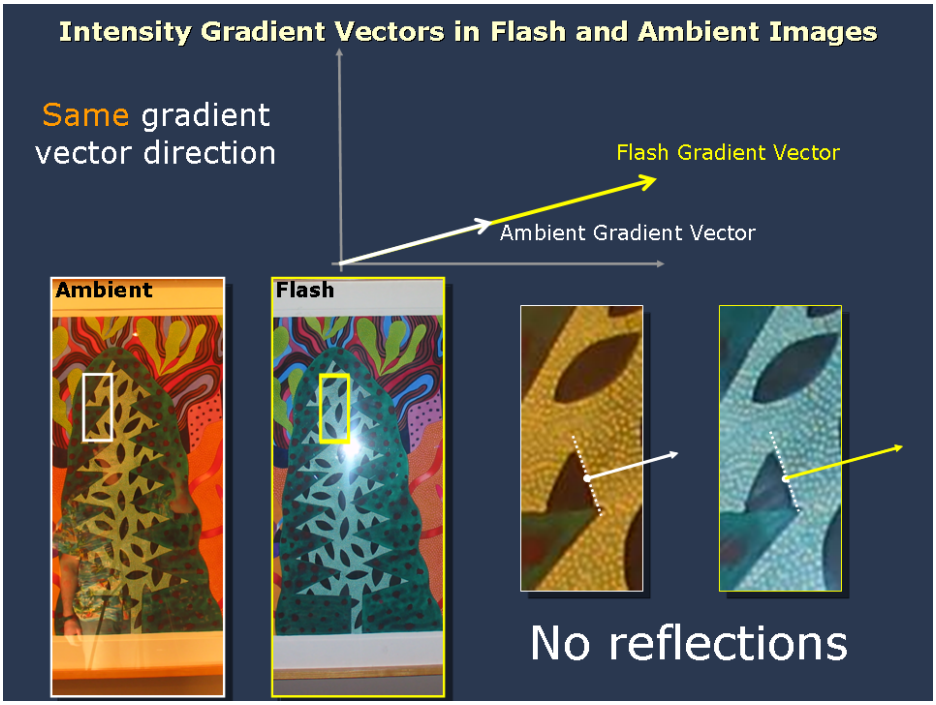
Let us look at an example to understand the concept. Consider the problem of removing photography artifacts from digital images. Figure 4.1 show two images of a painting. The top left image was taken under ambient illumination (referred to as the *ambient image*). Due to the glossy surface of the painting, the ambient image has reflections of the photographer. The zoomed region shows the reflections of the hands and face of the photographer. An easy way to remove these reflections is to take another image under flash illumination (referred to as the *flash image*) using the on-board flash of the camera and a short exposure time ($\approx$ 4 ms). Due to the short exposure time, the effect of ambient illumination is reduced. However, flash may result in a hotspot (specularity) at the center of the image as shown in Figure 4.1. The goal here is to use information from the flash image to remove the

Figure 4.1: Glass reflections. Two images of a painting taken under ambient and flash illumination. The zoomed in region shows the self-reflection of the photographer in the ambient image.

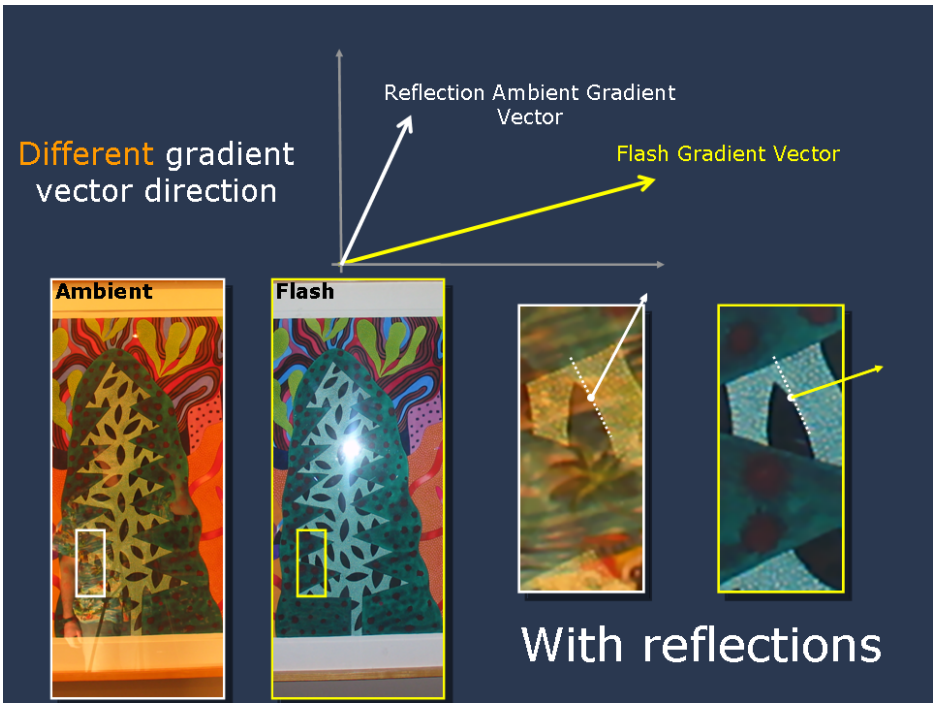
reflections in the ambient image and to possibly recover the reflection layer.

Let us compare the intensity gradient vectors in flash and ambient images at different regions. For regions with no reflections, we speculate that the intensity gradient direction will be same in both the images as shown in Figure 4.2(a). At regions where reflections are present in the ambient image, the direction of intensity gradient vectors in the ambient image will be different from those in the flash image (see Figure 4.2(b)).

To remove reflections from the ambient image, we constrain the direction of the ambient image gradients using the flash image. This is done by taking a vector projection of the ambient image gradient onto the flash image gradient. If α and Φ



(a)



(b)

Figure 4.2: Relationship of intensity gradient vectors in flash and ambient images at various image regions. (Top) The direction of the image gradient vector will be same in flash and ambient images at regions where artifacts (reflections) are not present. (Bottom) Due to reflections in the ambient image, the direction of the intensity gradients in the ambient image is perturbed at those regions.

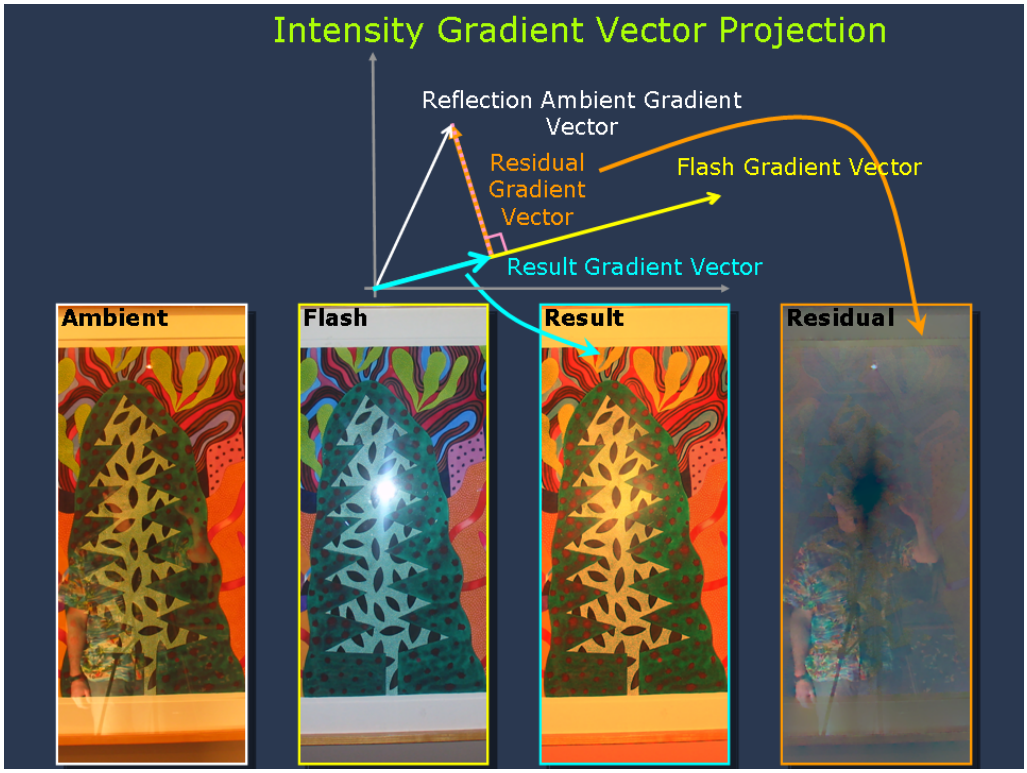


Figure 4.3: Gradient Projection. To constrain the direction of the image gradient using the flash image, a vector projection of the ambient image gradient on to the flash image gradient is taken. Reflections are removed in the projected gradient field. The residual gradient field (orthogonal component) corresponds to the reflection layer.

denote the ambient and flash images respectively, the projected gradient field $\nabla\widetilde{\alpha}$ is obtained as

$$\nabla\widetilde{\alpha} = \nabla\alpha \rightarrow \nabla\Phi \tag{4.3}$$

Figure 4.3 shows that the reflection free image $\widetilde{\alpha}$ can be obtained by integrating the projected gradient field $\nabla\widetilde{\alpha}$[1]. In addition, the reflection layer can be obtained by integrating the orthogonal gradient field given by $\nabla\alpha - \nabla\widetilde{\alpha}$.

Figure 4.4 shows another example on removing glass reflections from a flash

[1]In practice, projection cannot be done at flash hotspot due to lack of information. High dynamic range (HDR) images might be useful in such cases.

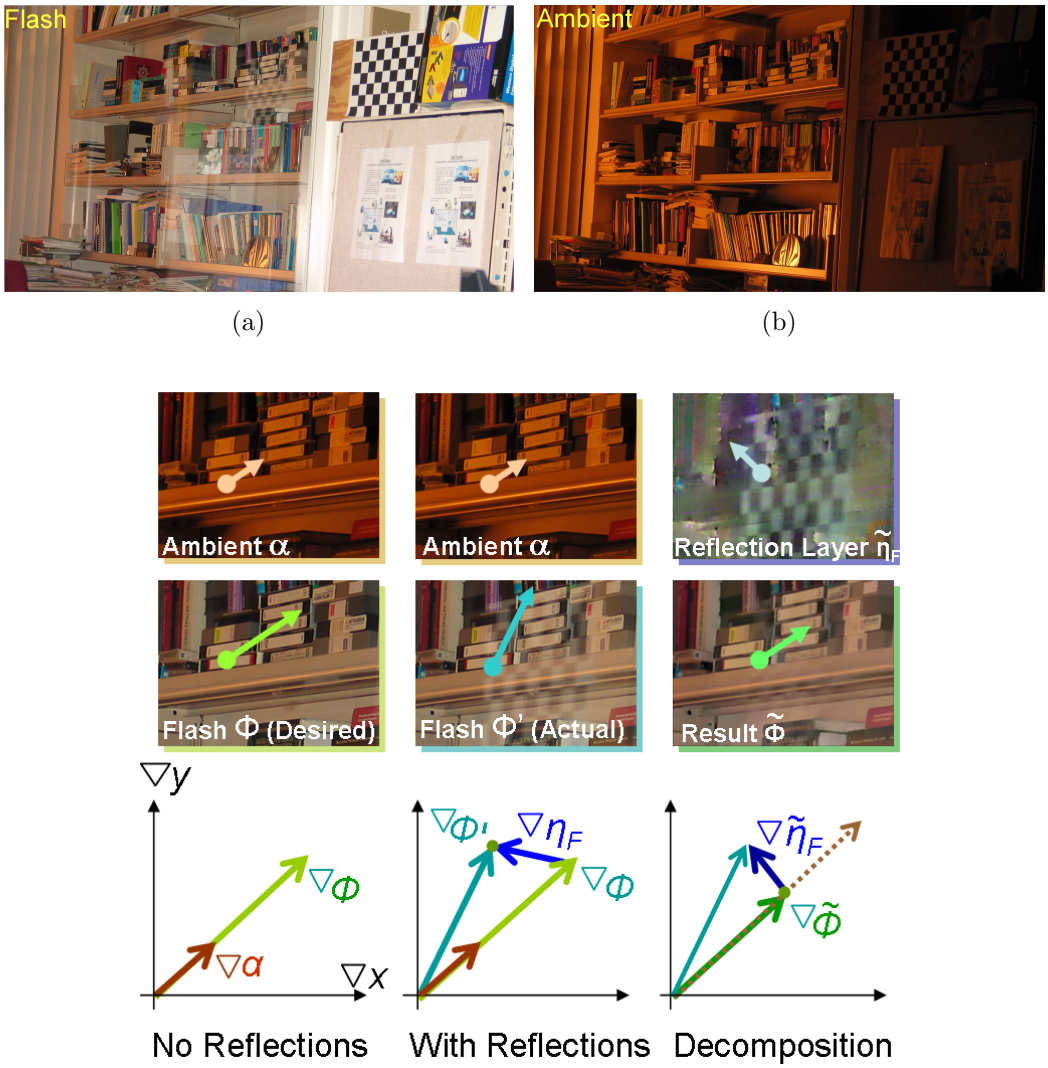


Figure 4.4: Overview of the reflection removal approach. (a) Image of an office scene under flash illumination. (b) Image under ambient illumination. (c) The glass reflections in the flash image can be removed by taking a vector projection of the flash image intensity gradient vector onto the ambient image intensity gradient vector, and integrating the projected gradient field.

image (Φ') using an ambient (no-flash) image (α). The top row in Figure 4.4 show images of an office scene taken under ambient and flash illumination, where the camera is looking through a glass window. The ambient image has low contrast and although using a flash improves the contrast, it brings in reflections of the checkerboard outside the office. We also captured a ground-truth flash image Φ without the checkerboard. With no reflections, we expect the flash and ambient image intensity gradients to be aligned in the same direction. In the presence of reflections in the flash image, its gradient vectors are perturbed. To remove the reflections from the flash image using the ambient image, we take a vector projection of $\nabla\Phi'$ on to $\nabla\alpha$ to get the projected gradient field $\nabla\widetilde{\Phi}$:

$$\nabla\widetilde{\Phi} = \nabla\Phi' \rightarrow \nabla\alpha \tag{4.4}$$

The reflection free flash image $\widetilde{\Phi}$ can be obtained by integrating the projected gradient field $\nabla\widetilde{\Phi}$. The reflection layer ($\widetilde{\eta}_F$) can be obtained by integrating the orthogonal gradient field

$$\nabla\widetilde{\eta}_F = \nabla\Phi' - \nabla\widetilde{\Phi} \tag{4.5}$$

For comparison, we show the ground truth flash image Φ. Note that in this example, the reflections are in the flash image while in the previous example, the reflections were in the ambient image.

4.3 Local Structure Tensors

Let $I(x, y)$ be an intensity image and $\nabla I = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$ denote the gradient vector of I at each pixel. The smoothed structure tensor $\mathbf{G}_\sigma$ is defined as [50]

$$\mathbf{G}_\sigma = (\nabla I \nabla I^T) * K_\sigma = \begin{bmatrix} g_x^2 & g_x g_y \\ g_x g_y & g_y^2 \end{bmatrix} * K_\sigma, \tag{4.6}$$

where $*$ denotes convolution and K_σ is a normalized 2D Gaussian kernel of variance σ. The matrix $\mathbf{G}_\sigma$ can be decomposed as

$$\mathbf{G}_\sigma = V \Sigma V^T = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \end{bmatrix}, \tag{4.7}$$

where $\mathbf{v}_1, \mathbf{v}_2$ denote the eigen-vectors corresponding to eigen-values λ_1, λ_2 respectively and $\lambda_2 \leq \lambda_1$. The eigen-values and eigen-vectors of $\mathbf{G}_\sigma$ give information about the local intensity structures in the image [66]. For homogeneous regions, $\lambda_1 = \lambda_2 = 0$. If $\lambda_2 = 0$ and $\lambda_1 > 0$, it signifies the presence of an intensity edge. The eigen-vector $\mathbf{v}_1$ (corresponding to the higher eigen-value λ_1) corresponds to the direction of the edge.

For the problem of image restoration based on diffusion process, Weickert [39, 90] proposed a generalization of the divergence based equation given by (4.1), where D is a field of *diffusion tensors*. At each pixel, $D(x, y)$ is a 2×2 symmetric, positive definite matrix. Weickert proposed to design the diffusion tensors D by selecting its eigen-vectors $\mathbf{u}_1, \mathbf{u}_2$ and eigen-values μ_1, μ_2 based on the eigen-values

and eigen-vectors of $\mathbf{G}_\sigma$. D is then obtained as

$$D = \begin{bmatrix} D_{11} & D_{12} \\ D_{12} & D_{22} \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 \end{bmatrix} \begin{bmatrix} \mu_1 & 0 \\ 0 & \mu_2 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1^T \\ \mathbf{u}_2^T \end{bmatrix}.$$

Several designs for obtaining D have been proposed for coherence enhancing diffusion [66, 90], edge enhancing diffusion [66], color image restoration, in-painting, and magnification [89]. Usually, D is obtained from the given image I. All these approaches modify the image intensities using the diffusion equation (4.1). In the next section, I show how to obtain *projection tensors* and discuss the properties and applications of affine transformation of the gradient field ∇I of an image using them.

4.4 Self-Projection Tensors

Now I show that taking a projection can also be defined by an affine transformation of the gradient field using self-projection tensors. This analysis will lead us to the main idea of cross-projection tensors: to estimate these tensors from a second image and apply them to the given image to suppress edges.

As discussed in Section 4.3, the eigen-vector $\mathbf{v}_1$ of the structure tensor matrix $\mathbf{G}_\sigma$ corresponds to the direction of the local edge. Suppose we define the self-projection tensor D^{self} as

$$\begin{aligned} &\mathbf{u}_1 = \mathbf{v}_1 \quad \mathbf{u}_2 = \mathbf{v}_2, \quad \mu_1 = 0 \quad \mu_2 = 1, \\ &D^{self} = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \end{bmatrix} \end{aligned} \tag{4.8}$$

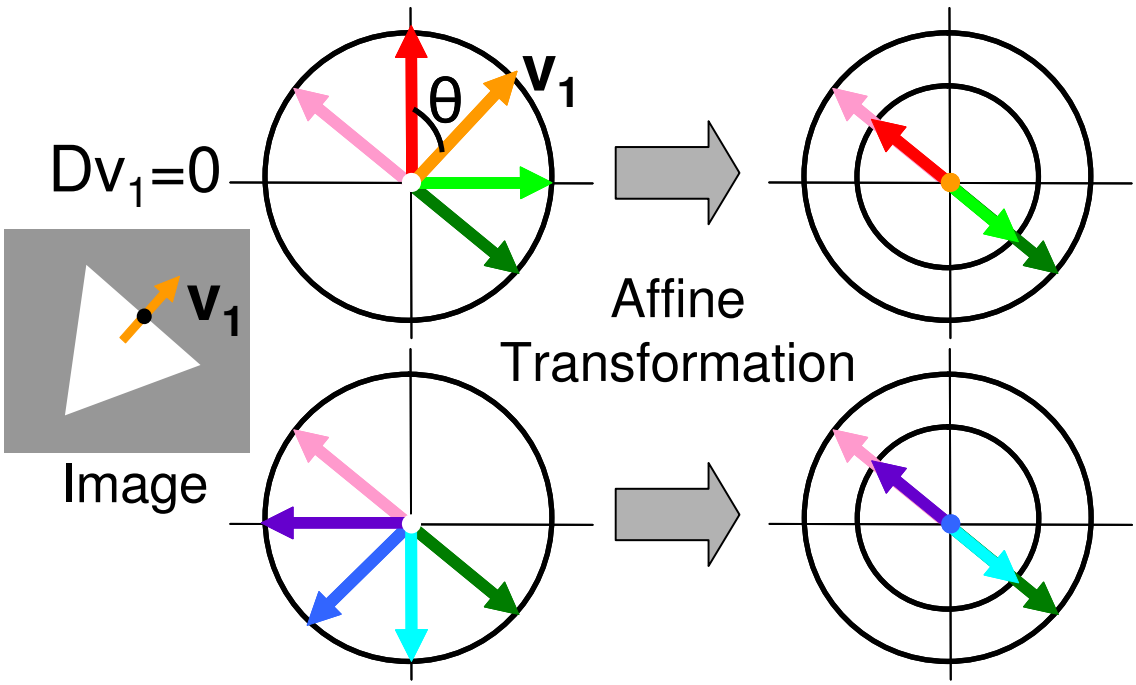


Figure 4.5: Visualizing affine transformation on gradient vectors. At each pixel in an image, $\mathbf{v}_1$ corresponds to the direction of the local edge. After affine transformation using D^{self}, any vector gets projected along the direction orthogonal to $\mathbf{v}_1$.

It is easy to see that an affine transformation of the image gradient using D^{self} will remove the local edge.

$$\begin{aligned} D^{self}\mathbf{v}_1 &= \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1^T \\ \mathbf{v}_2^T \end{bmatrix} \mathbf{v}_1 \\ &= \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}. \end{aligned} \tag{4.9}$$

Figure 4.5 shows the effect of transforming gradient vectors using D^{self}. All vectors are projected along the direction *orthogonal* to the local gradient vector $\mathbf{v}_1$. Thus, we can establish the following relationship.

Transforming a vector using D^{self} is equivalent to projecting along the orthog-

onal direction of the local gradient vector.

Figure 4.6 shows the reconstructed images obtained by integrating the transformed gradient field of the Lena image using D^{self}. To handle noise, it is useful to have a larger spatial support by using $\sigma > 0$ for reliable estimation of the direction of the local edge. In that case, although the estimated $\mathbf{v}_1$ may not lie in the null space of D^{self}, the affine transformation can still remove the dominant edges from the gradient field ∇I. Figure 4.6 shows that with $\sigma = 0$, the reconstructed image I' is zero everywhere. With $\sigma = 0.5$, I' has most of its edges removed.

4.4.1 Limitations of Gradient Projection

The gradient projection approach cannot handle homogeneous regions and introduces color artifacts (see Figure 4.14). This is because it does not include neighborhood support for gradient direction estimation, which is unstable in the presence of noise and low frequency regions. In addition, the projection is done for each channel separately which leads to color artifacts. In the next section, I show how to estimate **cross-projection tensors**. This approach combines information spatially (using $\sigma > 0$) and across channels to handle noise in gradient direction estimation and has no color artifacts.

4.5 Cross-Projection Tensors

Now I show how to remove the scene texture edges from an image by transforming its gradient field using cross projection tensors obtained from a second image

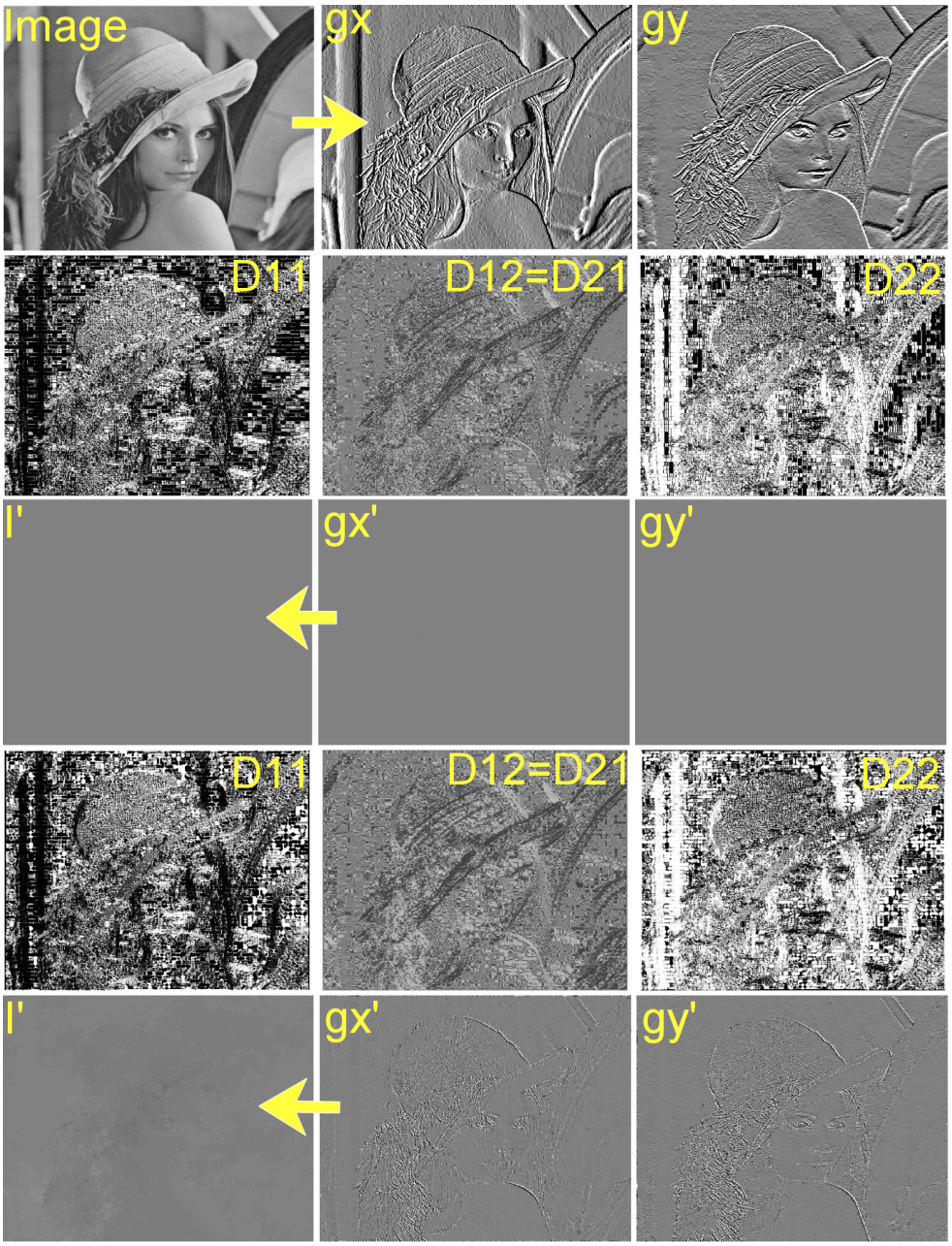


Figure 4.6: Affine transformation of image gradient field using D^{self} for different values of σ. (Top row) Lena image and the corresponding gradient field $\{g_x, g_y\}$. (Second row) Components D_{11}, D_{12} and D_{22} of D^{self} with $\sigma = 0$. (Third row) Transformed gradients g'_x, g'_y, and the image I' reconstructed from them. g'_x, g'_y and I' are zero all over. (Last two rows) Components of the projection tensor, modified gradient field and the reconstructed image corresponding to D^{self} using $\sigma = 0.5$. Even if $\sigma > 0$, all dominant edges are removed. A non-zero σ incorporates spatial information over the neighborhood for better estimation of cross projection tensors in the presence of noise. D_{11} and D_{22} are between $[0, 1]$. D_{12} is between $[-1, 1]$.

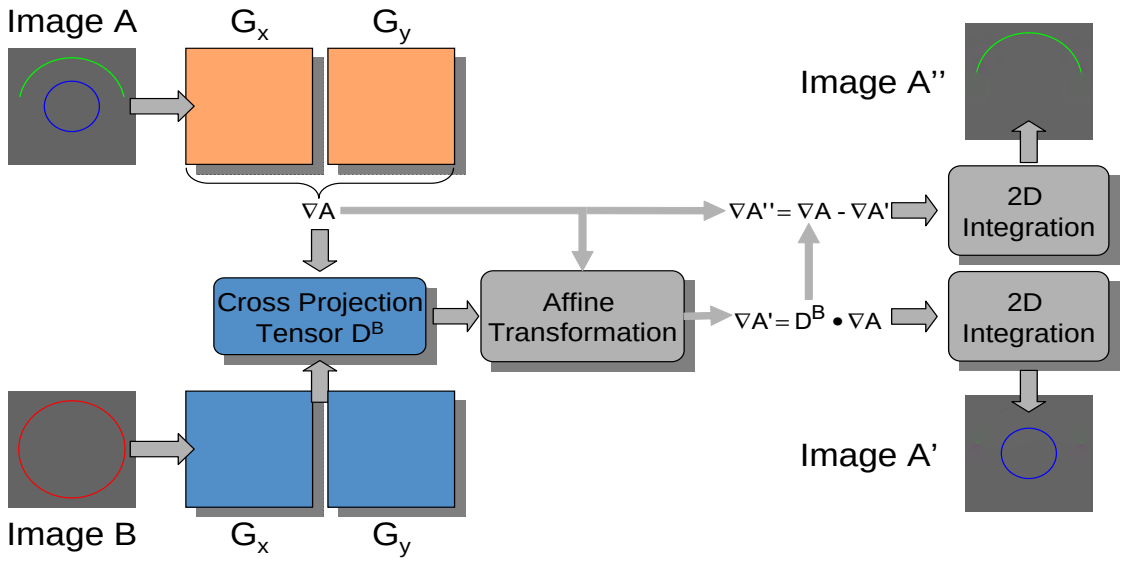


Figure 4.7: Suppressing edges in an image A using another image B by affine transformation of gradient field using cross projection tensors. The cross projection tensor D^B is obtained using images. The gradient field ∇A is transformed using D^B to give $\nabla A'$, removing all those edges from A which are present in B. Reconstruction from $\nabla A'$ gives an image A', with all the corresponding edges suppressed. Reconstruction from the difference gradient field $(\nabla A - \nabla A')$ gives an image A'', which preserves those edges in A which are also present in B.

of the same scene (see Figure 4.7). The final image is obtained by a 2D integration of the modified gradient field.

Let A and B denote the two images. Let $\mathbf{G}_\sigma^A$ and $\mathbf{G}_\sigma^B$ denote the smoothed structure tensors for images A and B respectively. The eigen-values and eigen-vectors of $\mathbf{G}_\sigma^A$ and $\mathbf{G}_\sigma^B$ will be denoted by superscripts A and B respectively. The technique for obtaining the cross projection tensor D^B is explained now. Note that by transforming ∇A with D^B, we wish to (a) remove all edges from A which are present in B, and (b) retain all edges in A which are not in B. To obtain D^B, we use the following rules:

- $\mathbf{u}_1 = \mathbf{v}_1^B, \quad \mathbf{u}_2 = \mathbf{v}_2^B.$

- If B is homogeneous ($\lambda_1^B = 0$)
  - If A is also homogeneous ($\lambda_1^A = 0$), set $\mu_1 = \mu_2 = 0$. This results in $D(x,y) = \begin{bmatrix} 0 & 0 \\ & \\ 0 & 0 \end{bmatrix}$ for that pixel.
  - If A is not homogeneous ($\lambda_1^A > 0$), set $\mu_1 = \mu_2 = 1$. This results in $D(x,y) = \begin{bmatrix} 1 & 0 \\ & \\ 0 & 1 \end{bmatrix}$ and edges which are in A but not in B can be retained.
- Else, if there is an edge in B ($\lambda_1^B > 0$), remove that edge by setting $\mu_1 = 0$, $\mu_2 = 1$.

In practice, due to noise and gradient estimation using finite differences, a small non-zero value is used as a threshold to check for homogeneity. One might think that the above homogeneity threshold needs to vary across the image, if the image has spatially varying illumination. Since we take into account the direction of the edge, we do not need spatially adaptive thresholds. Figure 4.8 shows such an example where illumination is spatially varying in the images. Also note that there are no other thresholds in this scheme.

4.5.1 Combining Information across Color Channels

The above formulation can be used for gray scale images. A naive way of handling color images would be to obtain the cross projection tensor for each channel and transform the gradient field in each channel separately. However, this scheme introduces color artifacts in the final reconstructed image as the projection tensor

does not utilize information across channels. To this end, we obtain a common cross projection tensor for all the channels by estimating a common $\mathbf{G}_\sigma$ matrix as [50]

$$\mathbf{G}_\sigma = (\sum_{i=1}^{3}(\nabla I_i \nabla I_i^T)) * K_\sigma, \tag{4.10}$$

where i denote the color channel.

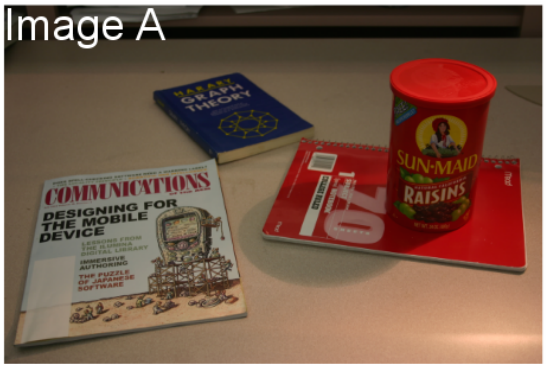
4.6 Applications

I show applications in recovering the foreground layer under varying illumination, estimating intrinsic images for non-Lambertian scenes, removing shadows from color images, recovering the illumination map, and removing glass reflections from images. In all experiments, $\sigma = 0.4$ is used.

4.6.1 Recovering Foreground Layer under Varying Illumination

Background subtraction and foreground layer recovery is a challenging problem in the presence of significant illumination variations. Consider the pair of images in the first row of Figure 4.8. Image A was captured with a foreground object (raisin box) illuminated from a table lamp on the right. Image B was captured with the table lamp on the left, but without the object. Notice the spatially non-uniform illumination in the images. Intensity based measures such as frame differencing cannot discount such illumination variations across images as shown in Figure 4.8(c).

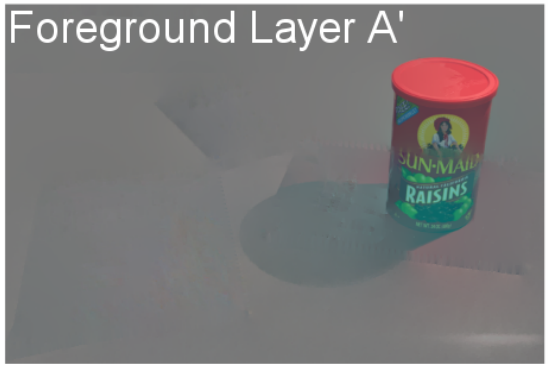
We compute the cross projection tensor D^B at each pixel using the background image B and transform the gradient field ∇A using D^B to obtain $\nabla A'$. The components of the cross-projection tensor are shown in Figure 4.9. This suppresses all the



(a) Image A

(b) Image B

(c) $|A - B|$



(d) Image A'

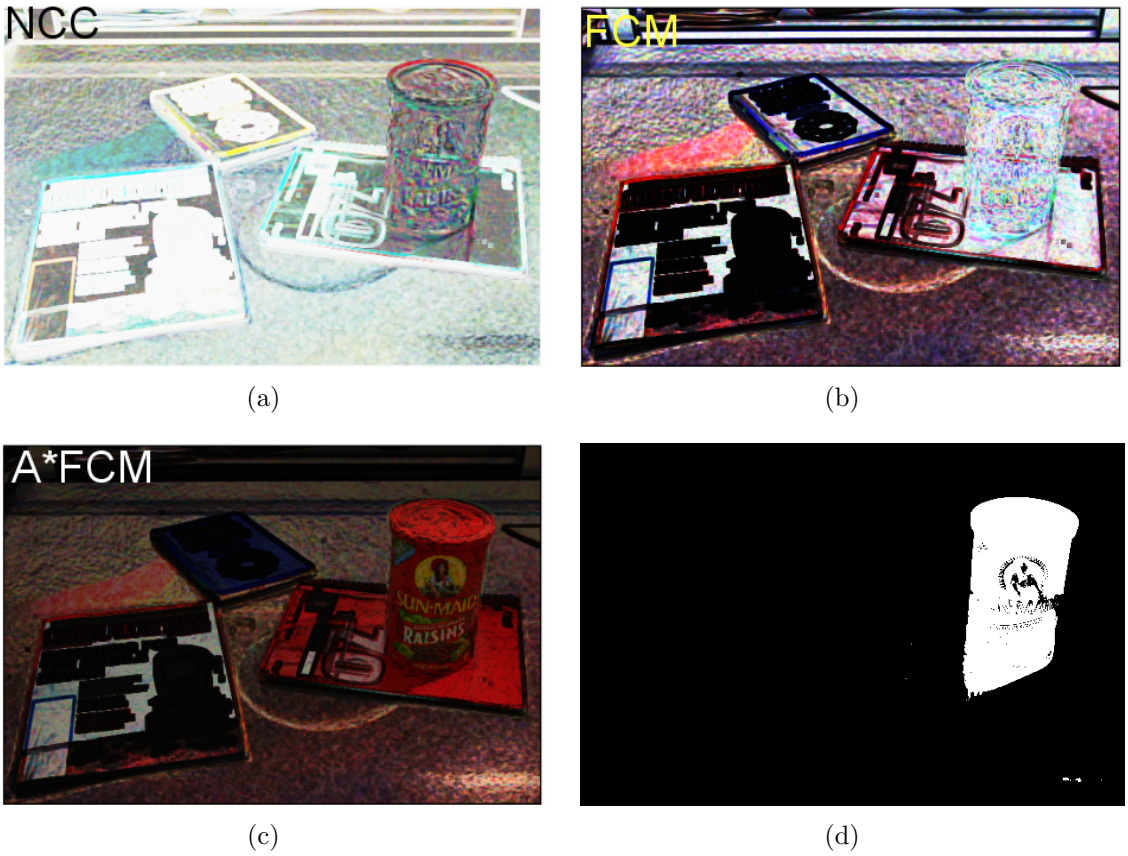
Figure 4.8: Recovering foreground layer under varying illumination. (a) Image A was captured with a foreground object (raisin box) under illumination from a table lamp on the right. (b) Image B was captured with the table lamp on the left but without the object. The images have spatially non-uniform illumination with respect to each other. (c) The absolute of the image difference $(A - B)$ is confounded by illumination variations and does not give information about the foreground object. (d) Foreground layer A' obtained using our approach. We remove those edges from image A which are present in image B, resulting in A' as the recovered foreground layer. Notice that A' is free of *all* scene texture edges apart from those due to the box. Although, the shadow of the box overlaps the texture on the red notebook, our method was able to remove all such texture edges *inside the shadow*. However, all object edges in A which coincide with the background edges in B will be removed.

(a) D_{11} (b) D_{12} (c) D_{22}

Figure 4.9: Components of the cross-projection tensor corresponding to the foreground layer example, shown as images. D_{11} and D_{22} are between $[0, 1]$ and D_{12} is between $[-1, 1]$.

texture edges corresponding to the background. The foreground layer is obtained by integrating $\nabla A'$. The recovered foreground layer is free of the background texture, even inside the shadow of the box. Notice that part of the foreground (red box) is similar in color to the background (red book). A color based differencing approach will fail at such regions. In addition, homogeneous regions on the foreground objects will leave holes for any local pixel intensity based approach. Our method is able to "fill-in" such regions by propagating information from edges during the integration of the modified gradient field. However, edges of the foreground object which align with the background edges (i.e., share the same gradient vector direction) are treated as part of the background and suppressed. Fortunately, as is well known, the likelihood of alignment of 1D features, such as edges, on two different objects is low. Nevertheless, in overlapping high frequency regions the likelihood is increased and some foreground edges may be lost. Notice how the top of the text "SUN-MAID" on the red box is smeared in A', as it overlaps with the red book binding in the image B.

For comparisons with our results, we computed the normalized cross correla-



(a) (b)

(c) (d)

Figure 4.10: Recovering foreground layer. (a) NCC image (between $[-1, 1]$). (b) Foreground confidence map (FCM) obtained as $1 - \texttt{abs}(\texttt{NCC})$. Ideally, the foreground confidence map should be high on the foreground object and low otherwise. (c) Product of image A with the foreground confidence map does not give good segmentation. (d) Binary segmentation map obtained by thresholding the obtained foreground layer using our approach.

tion (NCC) of image A and image B as

$$\texttt{NCC}(x, y) = \frac{\texttt{Cov}(x, y)}{\sqrt{\texttt{Var}(A(x, y))\texttt{Var}(B(x, y))}}, \tag{4.11}$$

where Var denotes local variance over 5×5 neighborhood and Cov denotes the covariance of the local patches in A and B. Cov is given by

$$\texttt{Cov}(x, y) = \overline{A(x, y)B(x, y)} - \overline{A(x, y)} \times \overline{B(x, y)}, \tag{4.12}$$

where the bar denotes averaging over a local 5×5 neighborhood. The NCC image as shown in Figure 4.10 is low around the box region and high on other regions.

However, NCC can only give a qualitative description of a foreground layer. Note that at homogeneous regions, NCC is not defined. So it cannot differentiate a homogeneous region into foreground or background. Figure 4.10 also shows the *foreground confidence map* (FCM) obtained as $1 - \texttt{abs}(\texttt{NCC})$ and the product of image A with the foreground confidence map. Ideally, this product should give the foreground layer. However, the correlation measure cannot discount the illumination variations whereas our method can easily recover the foreground layer.

4.6.1.1 Obtaining Foreground Mask

Usually, in background subtraction, a binary mask corresponding to the foreground is desired. Figure 4.10(d) shows the binary mask corresponding to the foreground object obtained by thresholding the foreground layer. Choosing the correct threshold for segmentation remains an open problem. However, our approach was able to successfully discount the illumination variations and thus can give better segmentations.

4.6.2 Recovering Illumination Images in Non-Lambertian Scenes

Weiss [32] proposed to decompose a set of N intensity images $I(x, y, t)_{t=0}^{N-1}$ obtained from a fixed view-point under changing illumination into a single reflectance image $R(x, y)$ and the corresponding illumination images $L(x, y, t)$ as:

$$I(x, y, t) = R(x, y)L(x, y, t) \quad t = 0 \ldots N - 1. \tag{4.13}$$

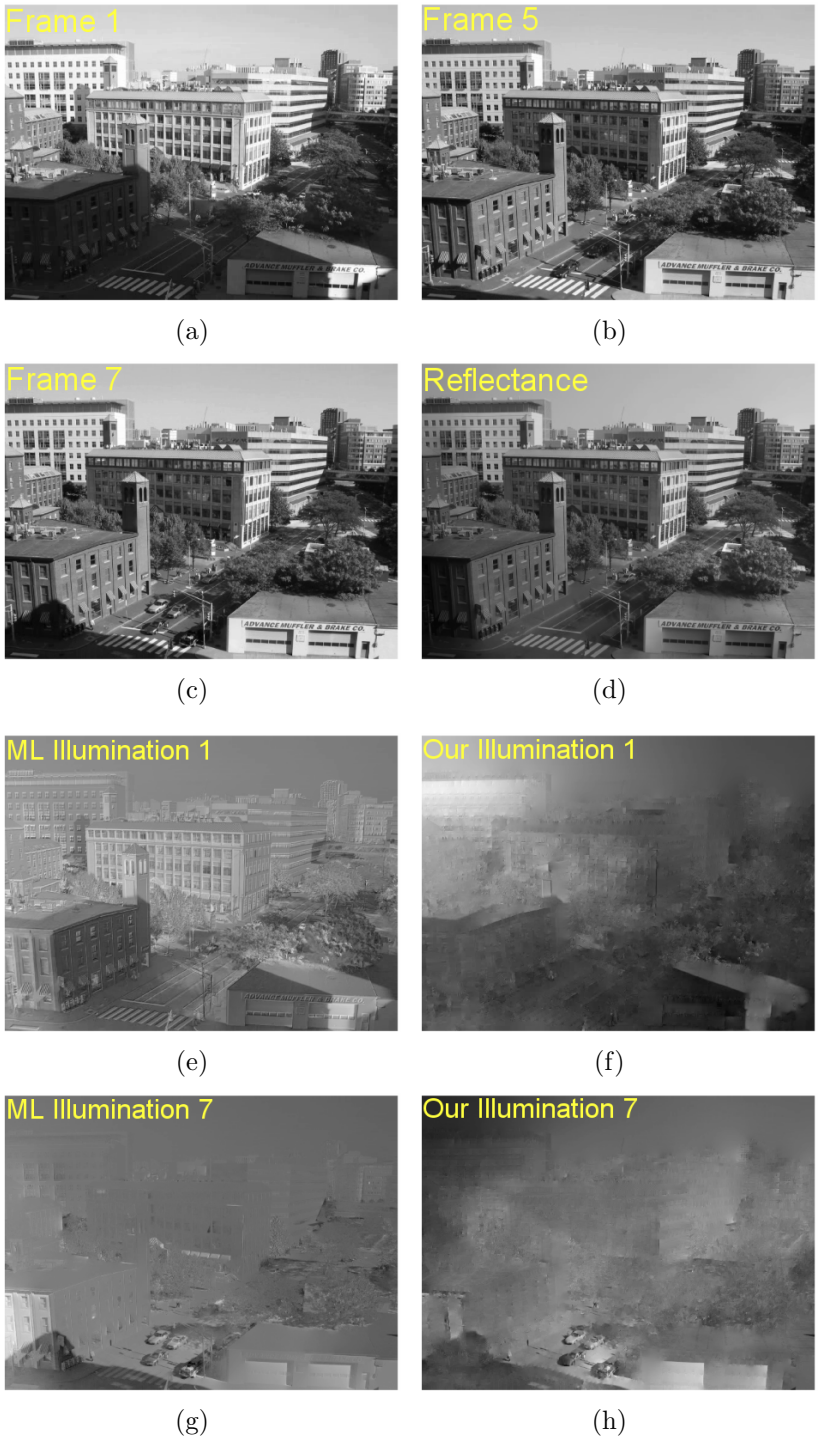


Figure 4.11: Recovering intrinsic images for an outdoor non-Lambertian scene. (a-c) Input images of an outdoor scene taken at different times of the day. (d) ML reflectance image. (e,g) Estimated illumination images using ML estimation. (f,h) Estimated illumination images using our approach. The scene texture edges (white stripes on the road) are visible in the ML illumination images. These are removed in our result while all the shadows are preserved. However, we make the usual assumption that illumination and reflectance edges do not coincide. All such illumination edges cannot be recovered.

Taking the logarithm of both sides, we get

$$i(x,y,t) = r(x,y) + l(x,y,t) \quad t = 0 \ldots N-1. \tag{4.14}$$

The method in [32] uses a prior that when derivatives filters f_n are applied to l, the output tends to be sparse. Assuming the filter outputs to be Laplacian distributed, the Maximum-Likelihood (ML) estimate of the filtered reflectance image $\widehat{r_n} = r * f_n$ is given by the median of the filtered images $i_n = i * f_n$ along the temporal axis. The filtered illumination images l_n can then be obtained as

$$l_n(x,y,t) = i_n(x,y,t) - \widehat{r_n}(x,y) \quad t = 0 \ldots N-1. \tag{4.15}$$

However, if the scene is not Lambertian, l_n will have some effect of scene texture edges. Matsushita *et al.* [35] proposed to remove the scene texture edges from l_n using a threshold T by setting

$$l_n(x,y,t) = \begin{cases} 0 & \text{if } |\widehat{r_n}(x,y)| > T, \\ l_n(x,y,t) & \text{otherwise.} \end{cases} \tag{4.16}$$

However, the threshold was manually specified in [35] and is difficult to generalize to different scenes. Our approach provides an elegant way of estimating the illumination images l by avoiding the two-step process which involves thresholding. We first estimate r using Weiss's method. For each image i, we then find the cross projection tensor D^r using r and i, and transform the gradient field ∇i using D^r. This will remove all the edges from i which are present in r. Thus,

$$\nabla l(x,y,t) = D^r \cdot \nabla i(x,y,t). \tag{4.17}$$

The illumination images $l(x, y, t)$ are obtained by integrating the resulting gradient field $\nabla l(x, y, t)$ for each t.

Figure 4.11 shows results on images of an outdoor scene taken under different times of the day. All illumination images have been shown with logarithmic non-linearity following [32]. Notice that the ML illumination images contains the effect of scene texture, especially white lines on the road surface. Using our approach, all such scene texture edges can be successfully removed from the illumination images while preserving shadows.

4.6.3 Removing Shadows from Color Images

We use a flash image F of the scene to remove shadows from the ambient (no-flash) image A. The flash and the ambient images were captured in quick succession using the remote capture utility with the camera mounted on a tripod. We obtain the cross projection tensor D^F using F and transform the gradient field ∇A using it. Figure 4.12 shows an example on a highly textured book. Notice that the recovered shadow free image A'' has no color artifacts and the recovered illumination map A' is free of strong texture edges on the face of the book. Figure 4.13 shows the components of the cross projection tensor obtained from the ambient and flash images of the book.

Figure 4.14 shows a challenging scenario where the hat on the mannequin cast shadows on the mannequin's face and neck. Usually, the ambient and flash images have different color tone due to the ambient illumination being yellow-reddish and

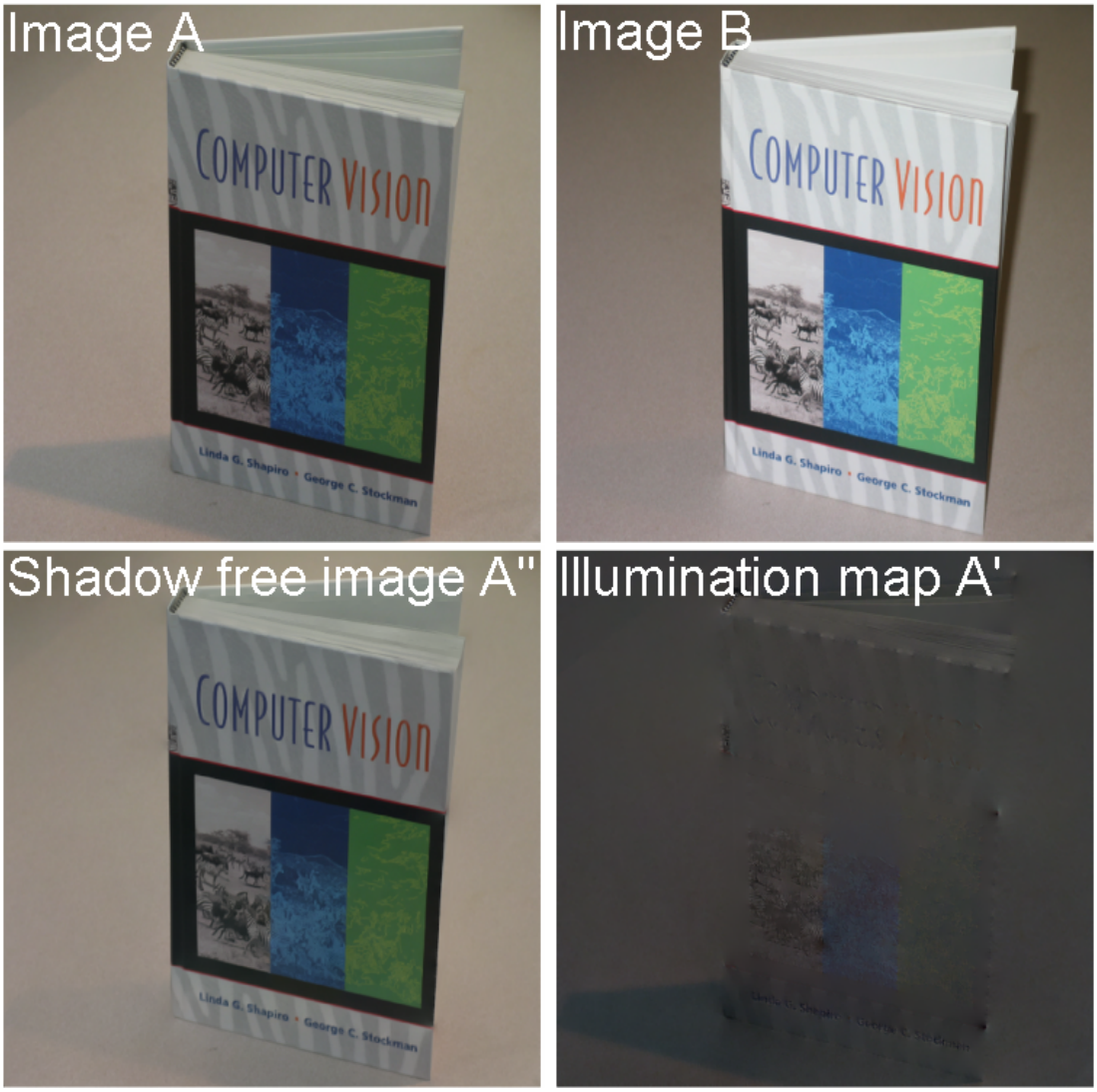


Figure 4.12: Ambient and flash images of a book on a table. We remove the edges from ∇A using ∇B to get $\nabla A'$, which is integrated to obtain the illumination map A'. Even though the face of the book is highly textured, A' does not have scene texture edges. Reconstruction from $\nabla A - \nabla A'$ gives the shadow free image A''.

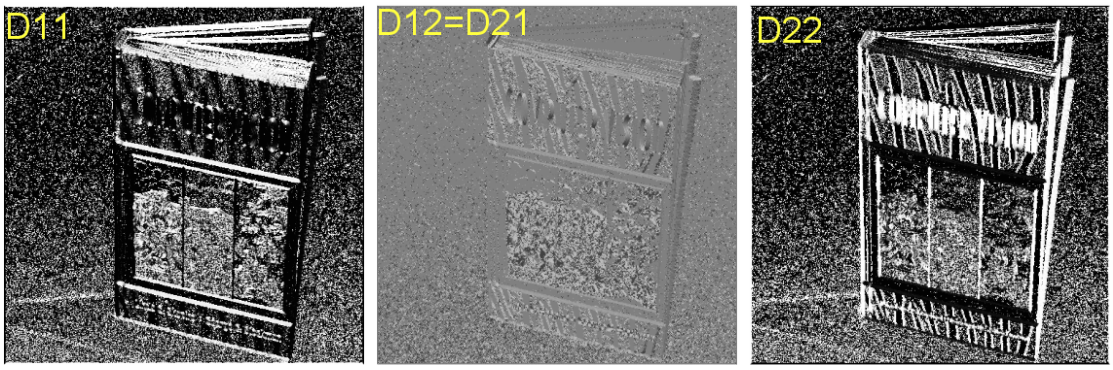


Figure 4.13: Components of the cross projection tensor obtained from the flash and no-flash images of the book. D_{11} and D_{22} are between $[0, 1]$ and D_{12} is between $[-1, 1]$.

the flash illumination being bluish. Figure 4.15 shows the components D_{11}, D_{12} and D_{22} of the cross projection tensor obtained from the ambient and flash images of the mannequin. Our algorithm requires no pre-processing or color calibration and has no color artifacts as compared to the result using gradient projection. One might think that the ratio image A/F could give the illumination map of the scene. However, the ratio image (shown in Figure 4.14) does not represent the illumination map due to the presence of flash shadows at depth discontinuities. The illumination map obtained by our approach better represents the diffuse ambient illumination.

4.6.4 Removing Glass Reflections

While photographing through glass in low light environments, an ambient image is usually of low quality and has low contrast. Using a flash improves the contrast, but it may result in reflections of objects in front of the glass. Figure 4.16 shows such an example, where the camera is looking into an office scene through a glass window. The flash image has undesirable reflections of the checkerboard

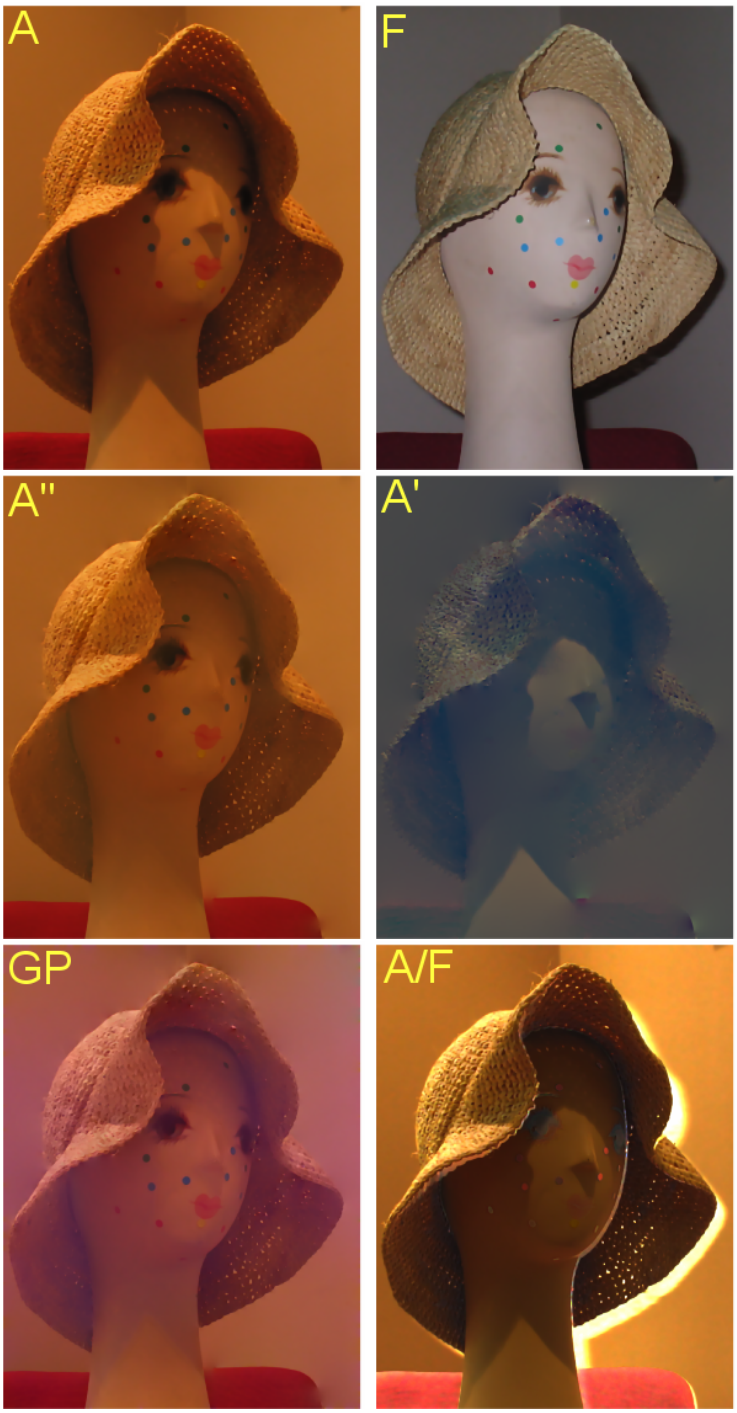


Figure 4.14: Removing cast shadows. (Top row) Ambient and flash images of a mannequin. The hat cast shadows on the mannequin's face and neck in the ambient image A. The flash image F is taken with a short exposure time. (Second row) Recovered shadow free image A'' and the illumination map A'. (Last row) Result using gradient projection has visible color artifacts. One cannot obtain the illumination map by taking the ratio A/F (shown on right) which is confounded by shadows due to flash at depth discontinuities. Notice that the color tones in the flash and ambient images are different due to automatic white balance setting during image acquisition. Despite that there are no color artifacts in our result.

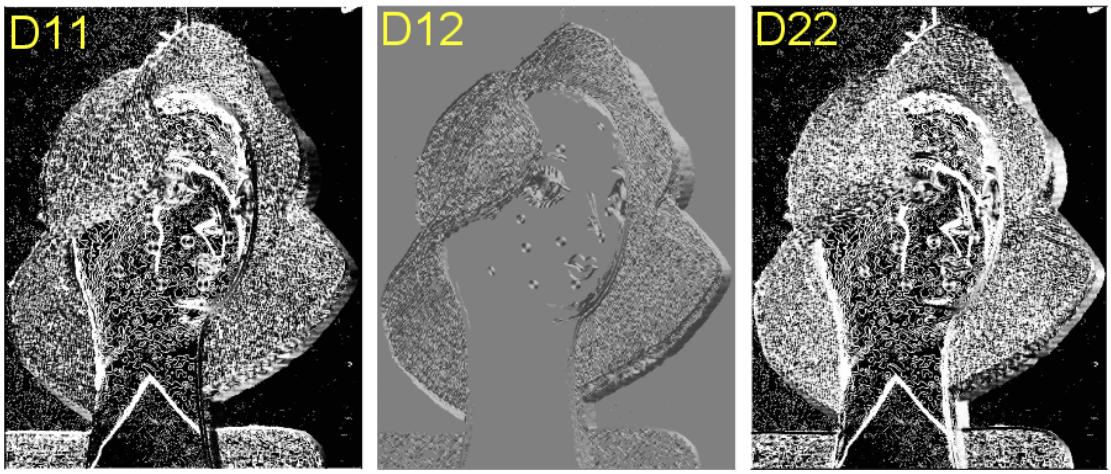


Figure 4.15: Components of the cross projection tensor obtained from the flash and no-flash images of the mannequin. D_{11} and D_{22} are between $[0, 1]$ and D_{12} is between $[-1, 1]$.

outside the glass window. We use the ambient image A to obtain the cross projection tensor D^A and transform the gradient field ∇F of the flash image F using it. The reflection layer is obtained by integrating $\nabla F'$ and the reflection free flash image is obtained by integrating $\nabla F - \nabla F'$. Figure 4.17 shows the components of the cross-projection tensor D^A. For this example, we repeat the affine transformation 5 times as the reflection layer has strong edges. In comparison, one can see a slight tinge of reflection remaining in the gradient projection result.

4.7 Summary

In this chapter, I presented two approaches for edge-suppressing operations on images: gradient projection and affine transformation of gradient fields using cross projection tensors. These approaches utilize the fact that the direction of the image intensity gradient is insensitive to changes in illumination to a large extent. The gradient projection approach does a vector projection of the image gradient of

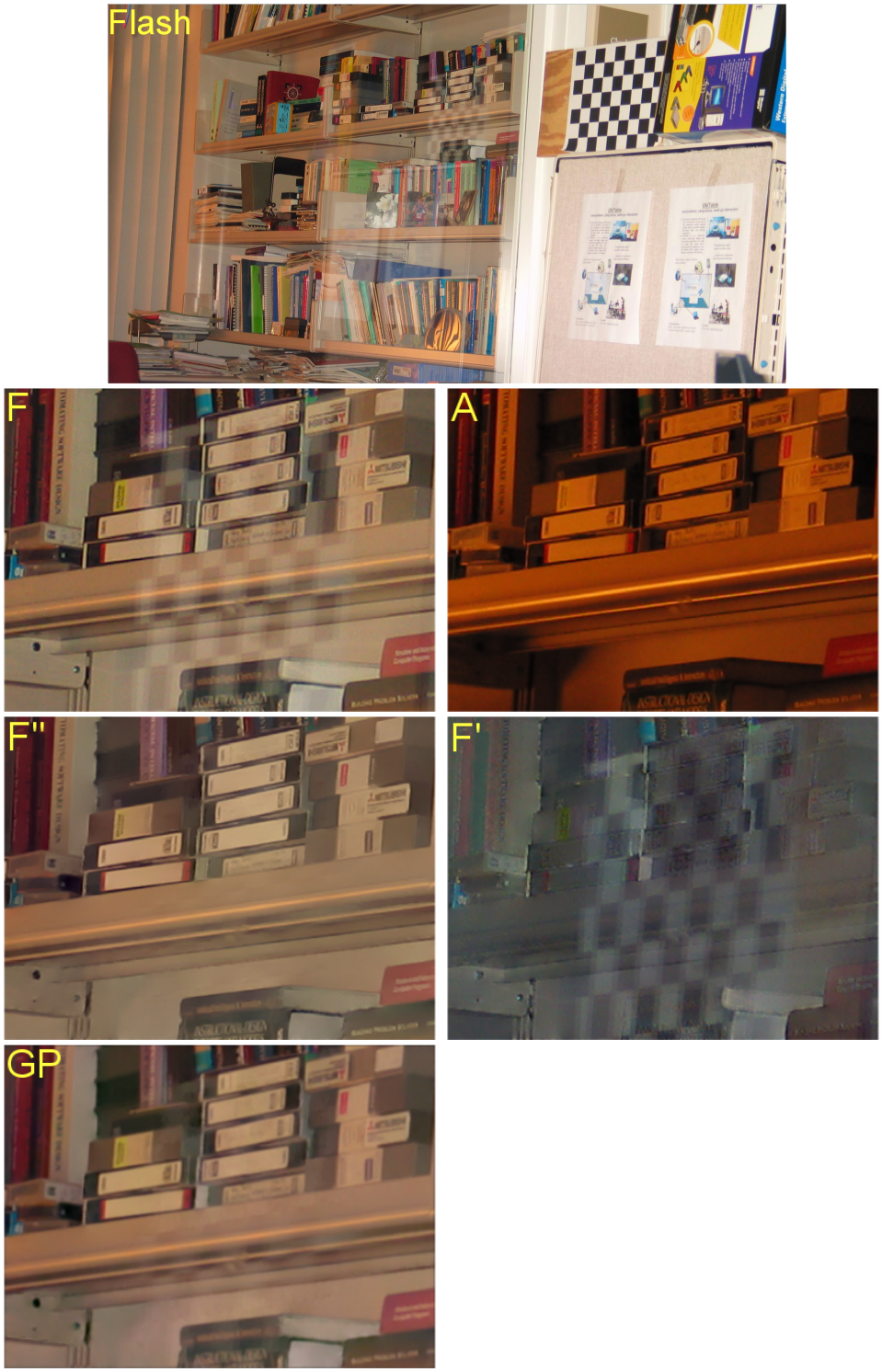


Figure 4.16: Removing glass reflections from a flash image using an ambient image. (Top row) Flash image F of an office scene through a glass window. The checkerboard outside the office results in reflections on the glass window. (Second row) Zoomed in flash and ambient images. (Third row) Recovered reflection layer F' and the reflection free image F''. (Last row) Result using gradient projection has a slight tinge of the reflection layer remaining along with a brownish hue (on top of books in the lower shelf).

(a) D_{11} (b) D_{12} (c) D_{22}

Figure 4.17: Components of the cross-projection tensor corresponding to the flash and no-flash images of the office scene.

one image onto the image gradient of another image. The projected gradient field preserves common edges in the two images. I showed that taking a projection is a special case of gradient field transformation using suitably defined projection tensors. I then described how to estimate cross-projection tensors using an image and apply them to transform the gradient field of another image for edge suppression.

Both these methods are local and requires no global analysis. In recovering the illumination map, the usual assumption that the scene texture edges do not coincide with the illumination edges was made. Hence, all such illumination edges cannot be recovered. Similarly, while extracting foreground layer, edges of the foreground object which exactly align with the background edges cannot be recovered. This may be handled by incorporating additional global information in designing the cross projection tensors, which remains an area of future work. In addition, image saturation, specular objects, and black objects will create problems due to the lack of reliable information. A fixed variance σ was used for estimating the structure tensor $\mathbf{G}_\sigma$, but an adaptive neighborhood scheme might improve results.

The approaches presented here can easily handle color images without the

need for any color calibration or white balancing. Applications on extracting foreground layer, removing shadows and glass reflections from images, recovering the illumination map, and estimating intrinsic images in non-Lambertian scenes were demonstrated.

Chapter 5

Conclusions

In this dissertation, I presented algorithms for reconstructing and manipulating gradient fields. In this chapter, I summarize the main ideas presented in this dissertation and suggest areas of future exploration.

5.1 Thesis Summary

Surface Reconstruction from Gradient Fields: Shape estimation using PS and SfS requires integrating the estimated non-integrable gradient field to obtain the final shape. In chapter 2, I analyzed the space of all possible reconstructions from a gradient field and showed that previous LS approaches do not perform well in the presence of outliers. In addition, these approaches do not have the important property of local error confinement. Using the idea that all gradients are not required for integration, I presented an algebraic approach for enforcing integrability. This approach can locally confine errors during reconstruction.

In Chapter 3, I presented a generalized equation which gives a range of feature preserving surface reconstructions. I showed that previous solutions such as the Poisson solver and Frankot-Chellappa algorithm are special cases of this framework. The range of solutions obtained from the framework are related to the degree of anisotropy of weights applied to the gradients during reconstruction. This frame-

work leads to a better understanding of the 2D gradient integration problem. I derived several new algorithms using this framework including α-surface and affine transformation of gradient field using diffusion tensors. These algorithms were shown to give significantly better feature preserving reconstructions in the presence of noise and outliers.

Edge Suppression using Gradient Field Manipulations: Variable Illumination poses problem for computer vision algorithms. In Chapter 4, I presented a framework for edge suppressing operation on images taken under varying illumination. Under illumination variations, although pixel intensities do not remain invariant, the direction of the image intensity gradient remains stable. I first presented the *gradient projection* technique to constrain the direction of the intensity gradient in one image using another image. This is achieved by a vector projection of the intensity gradient from one image onto the corresponding intensity gradient from the second image. The projected gradient field preserves common edges in the images. The orthogonal (residual) gradient field have common edges suppressed.

Then I showed that gradient projection approach is a special case of affine transformation of gradient field using suitable defined projection tensors. Using this analysis, I introduced cross-projection tensors. These tensors are obtained using another image of the scene taken under different illumination conditions and are used to transform the gradient field of the given image to achieve edge suppression. I showed how to use the on-board flash as an active illumination device to obtain the additional image. These approaches were demonstrated in the context of several applications such as recovering foreground layer, removing shadows in color images,

removing glass reflections and recovering intrinsic images in non-Lambertian scenes.

5.2 Future Directions

This work opens up several new possibilities in various areas. Below, I discuss applications and future directions in gradient domain image processing, surface reconstruction, and edge analysis.

5.2.1 Reconstruction from Gradient Fields

Other Vision and Graphics Applications: As discussed earlier, reconstruction from gradient fields is also important in other problems such as retinex, phase unwrapping, image editing, image matting, high dynamic range compression, mesh smoothing and seamless image stitching. In these methods, the gradient field of an image or multiple images is modified and the final image is obtained by a 2D integration of the modified gradient field. Least square reconstruction in these problems often lead to pinching and contouring artifacts in the final image. It would be desirable to apply the framework presented here to these problems.

Incorporating Prior Information: In PS and SfS, only the surface gradients are estimated from images. Prior work has been done on using control points in PS along with surface gradients [93]. The depth values at control points can be obtained using other methods such as laser scanning or stereopsis, and are used as constraints in the reconstruction. But even in these scenarios, usually LS reconstruction is obtained. The presented algorithms could be applied in these settings

for better reconstructions.

Integration under Gradient and Intensity Constraints: A similar and useful application in image editing would be to integrate an image gradient field using both gradients and image intensity constraints. Usually, in image editing applications, only the gradient field of images is manipulated. However, manipulation of gradient fields is less intuitive than manipulation of intensities. An extension would be to use the image intensities as constraints (hard or soft) during the integration process. In my framework, this could be done by including the $\frac{\partial E}{\partial Z}$ term, which at present is set to zero. This can give a totally new flavor to the end-users in image editing applications.

New Solutions for Surface Reconstructions: Although I presented a range of algorithms, the choice of using a particular algorithm remains an open question. In general, for smooth surfaces with no outliers in the gradient field, LS approaches can work well, while for a surface with discontinuities, α-surface and affine transformation gives better feature preserving reconstructions.

While my framework presented a range of solutions, further research can lead to new solutions not covered by my framework. I believe that even within a particular solution, there might be a class of solutions. For example, for the affine transformation based approach, I considered edge preserving diffusion tensor, but other tensors such as those used in coherence preserving image smoothing might be useful.

5.2.2 Gradient Field Manipulations

Depth Edges: In my framework, edge suppression was done on intensity edges but could be extended to other types of edges, such as depth edges. A simple and practically viable method for depth edge extraction was proposed by Raskar *et al.* [94] using a multi-flash camera. If depth edges are first extracted, then using the techniques presented in Chapter 4, they can be suppressed to obtain reflectance/illumination edges.

Cross-Diffusion Tensors: The idea of cross-projection tensors may have interesting applications in image restoration. Usually, in PDE based image restoration, diffusion tensors are estimated using the given image and applied to the same image. By estimating *cross-diffusion* tensors, one might be able to use other similar images for noise reduction in the given image.

Image Inpainting: The main focus in this work was to remove edges, but *inserting edges* could be a next step. Previous work in this direction includes [95, 96]. Image completion algorithms usually hallucinate pixel intensities but recovering image intensities under the guidance of a gradient field could be useful.

Background Subtraction: I have showed an application in foreground layer extraction using a single foreground and background image. Background subtraction is an important problem in computer vision and illumination variations are problematic. Usually, information from intensity and color is utilized. Incorporating my gradient domain method with intensity based methods could improve background subtraction. Multiple frames could be used to build distributions of direction of the

image gradients for foreground and background. These distributions can then be used for classifying edges as foreground/background. Also, one might be able to delay hard thresholds and decisions further down the algorithm pipeline.

5.2.3 Gradient Camera

Traditional digital cameras suffer from the dynamic range problem. Natural scenes have extremely high dynamic range (contrast ratio of $\approx 1 : 10^6$), while a traditional cameras's dynamic range is limited ($\approx 1 : 1000$). Thus, one cannot capture extremely bright and dark objects in the scene in a single image. Human vision tries to overcome this limitation by adapting to the local brightness level.

Many ingenious high dynamic range (HDR) photography methods merge multiple mutually-aligned images with different exposure settings [97, 98, 99], or have varied or self-adjusting gain. For example, the Smal [100] cameras AutoBrite self-adapting method reduces out-of-range contrasts before measurement, while Dalstar [101] and others extended A/D measurement abilities. A novel asynchronous binary camera for measuring wide-ranging intensity by variable pulse rate was described in [102]. Nayar et al. [103] proposed a suite of HDR techniques that includes spatially-varying exposures and adaptive pixel attenuation, and micro-mirror arrays to re-aim and modulate incident light on each pixel sensor Nayar [104]. Logarithmic intensity cameras also avoid saturation well [105], but their increased quantization error and noise can hide small contrasts.

A gradient camera can offer an alternative solution without the need for cap-

turing multiple images. The central idea is to sense intensity gradients rather than static intensities. A large dynamic range of the scene cannot fit into the low A/D range of the sensor. However, the distribution of gradients in natural images is strongly zero peaked [32]. Thus, most of the gradients are small in natural images. By locally adapting the A/D range to measure these gradients, the final image can be obtained by a 2D integration of the sensed gradients. A gradient camera can also offer low quantization error as the final image is obtained as a piece-wise linear approximation resulting from integration of gradients rather than a piece-wise constant approximation in a traditional camera. In addition, certain gradient based image processing algorithms could also be implemented on the chip subject to the processing power limitations. A prototype for gradient camera is described in [106].

In a practical gradient camera, even the sensed gradients will saturate due to very large intensity ratios. These, along with image sensor imperfections will lead to a non-integrable intensity gradient field. Our understanding of local error confinement and discontinuity preserving reconstructions could be used to obtain meaningful images from the sensed gradient field.

Appendix A

Half-Quadratic Minimization

Image restoration from noisy images has been a classical problem in image processing. There is a wide literature on edge-preserving regularization including anisotropic diffusion [37], total variation methods [107], and energy based methods [65, 66]. Given a noisy image u_0, edge preserving image restoration can be posed as the problem of finding u which minimizes the following energy

$$J(u) = \int\int (u - u_0)^2 + \lambda\phi(|\nabla u|) \quad dxdy, \tag{A.1}$$

where the first term is the data fidelity term and the function ϕ controls smoothing using the gradient magnitude $|\nabla u|$. ϕ should satisfy certain requirements as given in [65] to be edge-preserving, i.e., small gradients must be smoothed and large gradients must be preserved. Common examples include $\phi(s) = \sqrt{1+s^2}$, $\phi(s) = \log(1+s^2)$.

The problem of gradient integration can be posed as edge preserving regularization as follows. Given a gradient field (p, q), we wish to find a surface Z which minimizes

$$J(Z) = \int\int (Z_x - p)^2 + (Z_y - q)^2 + \lambda(\phi(Z_x) + \phi(Z_y)) \quad dxdy. \tag{A.2}$$

The Euler-Lagrange equation gives

$$\nabla^2 Z + \frac{\lambda}{2}\mathtt{div}(\phi'(Z_x), \phi'(Z_y)) = \mathtt{div}(p, q). \tag{A.3}$$

However, minimizing the energy as above is difficult because (A.3) is non-linear.

In [65], the principle of half-quadratic regularization was proposed to solve this problem by introducing auxiliary variables $b = (b^x, b^y)$. The above error function can be solved by minimizing the dual energy

$$J(Z) = \inf_{b^x, b^y} J^*(Z, b^x, b^y), \tag{A.4}$$

where J^* is given by

$$J^*(Z, b^x, b^y) = \int\int (Z_x - p)^2 + (Z_y - q)^2 + \lambda(b^x Z_x^2 + b^y Z_y^2) + \lambda(\psi(b^x) + \psi(b^y))dxdy. \tag{A.5}$$

The function ψ is strictly convex and decreasing and satisfies

$$\phi(t) = \inf_{0<w\leq M}(wt^2 + \psi(w)), \tag{A.6}$$

where $M = \lim_{t\to 0^+} \frac{\phi'(t)}{2t}$. The energy is then minimized by alternate minimizations over Z and b. It was also shown that given Z, the minimum for b occurs at

$$b^x = \phi'(Z_x)/(2Z_x), \qquad b^y = \phi'(Z_y)/(2Z_y). \tag{A.7}$$

Given b, the minimum for Z can be found by writing the Euler-Lagrange equation for (A.5) which gives

$$(\nabla^2 + \lambda\nabla_b^2)Z = \mathtt{div}(p, q), \tag{A.8}$$

where ∇_b^2 is the *weighted Laplacian* obtained by using the b values as weights. The normal Laplacian can be calculated as applying the **same** kernel $\begin{bmatrix} & 1 & \\ 1 & -4 & 1 \\ & 1 & \end{bmatrix}$ at each pixel. The weighted Laplacian is obtained by applying the following kernel at

pixel (y, x): $\begin{bmatrix} & b^y(y-1,x) & \\ b^x(y,x-1) & -\sum & b^x(y,x) \\ & b^y(y,x) & \end{bmatrix}$, where $\sum$ denotes the sum of the neighboring b values.

Thus, the regularization approach can be summarized as follows. In solving the Poisson equation, one applies a spatially invariant Laplacian kernel (with 4 at center and -1's around). While regularizing, this is changed to a spatially varying kernel depending on the underlying surface. If the surface is smooth, and have low gradients, an isotropic smoothing will be done. Otherwise, smoothing will be inhibited along the edges. The entire minimization procedure can be written as

- $Z^0 \equiv 0$
- Repeat until convergence
  - $b^{n+1} = \texttt{arg}\min_b[J^*(Z^n, b)]$
  - $Z^{n+1} = \texttt{arg}\min_Z[J^*(Z, b^{n+1})]$

Appendix B

Matlab Codes

In this section, I present Matlab code for edge suppression using gradient field transformations described in Chapter 4. This code along with Matlab codes for surface reconstruction algorithms discussed in Chapter 2 and Chapter 3 can be found at [77].

The following code takes as input two images, A and B which have been obtained from fixed view-point under different illumination conditions. It then estimates cross-projection tensors using these images, and transforms the gradient field of image A using the estimated tensors. The transformed gradient field is integrated to obtain an image A', with edges suppressed. The input images used in this code can also be found at [77].

```
THRESHOLD_SMALL = 1

% Read image A
imgstr = 'Mannequin/MannequinAmbient.png';
disp(sprintf('Reading image %s',imgstr))
Aimage = imread(imgstr); Aimage = double(Aimage);

% Read image B
imgstr = 'Mannequin/MannequinFlash.png';
disp(sprintf('Reading image %s',imgstr))
Bimage = imread(imgstr); Bimage = double(Bimage);

% Convert to YUV color space
Aimage = RGB2YUV(Aimage); Bimage = RGB2YUV(Bimage);

% find size of images
[ho,wo,ch] = size(Aimage); clear ch

% zero padding for 2D integration
PAD = 100
Aimage = padarray(Aimage,[PAD PAD],0,'both');
Bimage = padarray(Bimage,[PAD PAD],0,'both');
[H,W,CH] = size(Aimage);

disp('Finding cross projection tensor')
sigma = 0.4 %
ss = floor(6*sigma);
if(ss<=3)
    ss = 3;
end

% K_sigma in the paper; gaussian kernel for smoothing
ww = fspecial('gaussian',ss,sigma);

% find G_sigma for image A and image B
% T11, T12 and T22 are elements of matrix G_sigma
[T11,T12,T22,EigD_1,EigD_2,X1,X2,Y1,Y2] = ...
  ...  TensorAnalysis(Bimage,ww);
[T11_2,T12_2,T22_2,EigD_1_2,EigD_2_2,X1_2,X2_2,Y1_2,Y2_2]...
  =  TensorAnalysis(Aimage,ww);
clear T11_2 T12_2 T22_2 X1_2 X2_2 Y1_2 Y2_2
```

```
% L1 = mu2 , L2 = mu1
% initially set to 1,1 to retain all edges in image A
L1 = ones(H,W); L2 = ones(H,W);


%If there is an edge in Bimage, set mu1 = 0. mu2 = 1
%to remove that edge from image A

idx = find(EigD_2 > THRESHOLD_SMALL); L2(idx) = 0;

% if both A and B are homogeneous
idx = find(EigD_2 < THRESHOLD_SMALL & ...
... EigD_2_2 < THRESHOLD_SMALL);

L1(idx) = 0; L2(idx) = 0;

% Get cross projection tensor terms
D11 = L1.*(X1.^2) + L2.*(Y1.^2);
D12 = L1.*(X1.*X2) + L2.*(Y1.*Y2);
D22 = L1.*(X2.^2) + L2.*(Y2.^2);


% allocate some memory
Ap_color = zeros(H,W,3); App_color = zeros(H,W,3);

% do for each channel
for channel = 1:CH
    disp('==========================================')
    disp(sprintf('Processing channel = %d',channel))

    u = Aimage(:,:,channel);
    Ap = u;

    % find gradient field
    [gx,gy] = CalculateGradients(Ap,0);

    % Affine transformation using tensors
    gx1 = (D11.*gx + D12.*gy);
    gy1 = (D12.*gx + D22.*gy);
```

```
    % 2D Integration
    Ap = Integration2D(gx1,gy1,zeros(H,W));
    App = Integration2D(gx-gx1,gy-gy1,zeros(H,W));

    % save
    Ap_color(:,:,channel) = Ap;
    App_color(:,:,channel) = App;
end
clear Ap App gx gy gx1 gy1

% remove zero padding
Aimage = Aimage(PAD+1:end-PAD,PAD+1:end-PAD,:); Bimage =
Bimage(PAD+1:end-PAD,PAD+1:end-PAD,:); D11 =
D11(PAD+1:end-PAD,PAD+1:end-PAD,:); D12 =
D12(PAD+1:end-PAD,PAD+1:end-PAD,:); D22 =
D22(PAD+1:end-PAD,PAD+1:end-PAD,:); Ap_color =
Ap_color(PAD+1:end-PAD,PAD+1:end-PAD,:); App_color =
App_color(PAD+1:end-PAD,PAD+1:end-PAD,:);

% YUV to RGB transformation
Aimage = YUV2RGB(Aimage);
Bimage = YUV2RGB(Bimage);
Ap_color = YUV2RGB(Ap_color);
App_color = YUV2RGB(App_color);

% for display
Ap_color = Ap_color - min(Ap_color(:));
```

BIBLIOGRAPHY

[1] R.J. Woodham. Photometric method for determining surface orientation from multiple images. *OptEng*, 19(1):139–144, 1980.

[2] R. Basri and D. Jacobs. Photometric stereo with general, unknown lighting. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 374–381, 2001.

[3] D. Forsythe and J. Ponce. *Computer Vision: A modern approach*. Prentice Hall, 2001.

[4] K. Ikeuchi. Determining surface orientations of specular surfaces by using the photometric stereo method. *IEEE Trans. Pattern Anal. Machine Intell.*, 3(6):661–669, 1981.

[5] H. Hayakawa. Photometric stereo under a light source with arbitrary motion. *J. Opt. Soc. of America*, 11:3079–3089, 1994.

[6] K. Ikeuchi and B.K.P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17:141–184, 1981.

[7] R. Szeliski. Fast shape from shading. *Computer Vision, Graphics, Image Processing: Image Understanding*, 53:129–153, 1991.

[8] P.S. Tsai and M. Shah. Shape from shading using linear approximation. *Image and Vision Computing J.*, 12(8):487–498, 1994.

[9] B.K.P. Horn. *Robot Vision*. McGraw-Hill, 1986.

[10] B.K.P. Horn and M. Brooks, editors. *Shape from Shading*. The MIT Press, July 1989.

[11] B.K.P. Horn. Height and gradient from shading. *Int'l J. Computer Vision*, 5(1):37–75, 1990.

[12] R. Zhang, P.S. Tsai, J.E. Cryer, and M. Shah. Shape from shading: A survey. *IEEE Trans. Pattern Anal. Machine Intell.*, pages 690–706, 1991.

[13] M.J. Brooks and B.K.P. Horn. Shape and source from shading. In *Proc. Intl. Joint Conf. Artif. Intelli.*, pages 932–936, 1985.

[14] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. *ACM Trans. Graph.*, 21(3):249–256, 2002.

[15] G. Fornaro, G. Franceschetti, and R. Lanari. Interferometric SAR phase unwrapping using greens formulation. *IEEE Trans. Geosci. Remote Sens.*, 34:720–727, 1996.

[16] R. M. Goldstein, H. A. Zebker, and C. L. Werner. Satellite radar interferometry: Two-dimensional phase unwrapping. *Radio Science*, 23:713–720, 1988.

[17] M.A. Herraez, M.A Gdeisat, D.R. Burton, and M.J. Lalor. Robust, fast, and effective two-dimensional automatic phase unwrapping algorithm based on image decomposition. *Applied Optics*, 41(35):7445–7455, December 2002.

[18] H.A. Zebker and Y. Lu. Phase unwrapping algorithms for radar interferometry: residue-cut, least-squares, and synthesis algorithms. *J. Opt. Soc. of America*, pages 586–598, 1998.

[19] P. Perez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, 2003.

[20] J. Sun, J. Jia, C.K. Tang, and H.Y. Shum. Poisson matting. *ACM Trans. Graph.*, 23(3):315–321, 2004.

[21] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. In *Proc. European Conf. Computer Vision*, volume 4, pages 377–389, 2004.

[22] R. Raskar, A. Ilie, and J. Yu. Image Fusion for Context Enhancement and Video Surrealism. In *Proc. 3rd Int'l Symposium on Non-photorealistic Animation and Rendering*, pages 85–94, 2004.

[23] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(2):218–233, 2003.

[24] B.K.P Horn. Determining lightness from an image. *Comput. Graphics, Image Processing*, 3:277–299, 1974.

[25] R. Kimmel, M. Elad, D. Shaked, R. Keshet, and I. Sobel. A variational framework for retinex. *Int'l J. Computer Vision*, 52(1):7–23, 2003.

[26] E.H Land. The retinex theory of color vision. *Sci. American*, 237:108–128, 1977.

[27] E.H. Land and J.J. McCann. Lightness and retinex theory. *J. Opt. Soc. of America*, 61:1–11, 1971.

[28] B.V. Funt, M.S. Drew, and M. Brockington. Recovering shading from color images. In *Proc. European Conf. Computer Vision*, pages 124–132, May 1992.

[29] T. Simchony, R. Chellappa, and M. Shao. Direct analytical methods for solving poisson equations in computer vision problems. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(5):435–446, May 1990.

[30] R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Trans. Pattern Anal. Machine Intell.*, 10(4):439–451, July 1988.

[31] G. D. Finlayson, S. D. Hordley, and M. S. Drew. Removing shadows from images. In *Proc. European Conf. Computer Vision*, pages 823–836, 2002.

[32] Y. Weiss. Deriving intrinsic images from image sequences. In *Proc. Int'l Conf. Computer Vision*, pages 68–75, 2001.

[33] M.F. Tappen, W.T. Freeman, and E.H. Adelson. Recovering intrinsic images from a single image. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(9):1459–1472, September 2005.

[34] Y. Matsushita, S. Lin, S.B. Kang, and H.Y. Shum. Estimating intrinsic images from image sequences with biased illumination. In *Proc. European Conf. Computer Vision*, pages 274–286, 2004.

[35] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakauchi. Illumination normalization with time-dependent intrinsic images for video surveillance. *IEEE Trans. Pattern Anal. Machine Intell.*, 26(10):1336–1347, 2004.

[36] G.D. Finlayson, M.S. Drew, and C. Lu. Intrinsic images by entropy minimization. In *Proc. European Conf. Computer Vision*, volume 3, pages 582–595, 2004.

[37] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Machine Intell.*, 12(7):629–639, 1990.

[38] J. Weickert. *Anisotropic Diffusion in Image Processing*. PhD thesis, University of Kaiserslautern, Germany, January 1996.

[39] J. Weickert. A review of nonlinear diffusion filtering. In *Scale-Space Theory in Computer Vision, Lecture Notes in Comp. Science (Springer, Berlin)*, 1997.

[40] M. Black and P. Anandan. Robust dynamic motion estimation over time. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 296–302, Maui, Hawaii, June 1991.

[41] M. Black. *Robust Incremental Optical Flow*. PhD thesis, Yale Univ., 1992.

[42] M.J. Black and A. Rangarajan. On the unification of line processes, outlier rejection, and robust statistics with applications in early vision. *Int'l J. Computer Vision*, 19:57–91, 1996.

[43] M.J. Black, D.J. Fleet, and Y. Yacoob. Robustly estimating changes in image appearance. In *Computer Vision and Image Understanding*, volume 78, pages 8–31, 2000.

[44] H. Chen, P. Belhumeur, and D. Jacobs. In search of illumination invariants. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 254–261, June 2000.

[45] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. *ACM Trans. on Graphics*, 23(3):673–678, August 2004.

[46] G. Petschnigg, M. Agrawala, H. Hoppe, R. Szeliski, M. Cohen, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Trans. on Graphics*, 23(3):664–672, August 2004.

[47] P.N. Belhumeur, D.J. Kriegman, and A.L. Yuille. The bas-relief ambiguity. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 1060–1066, 1997.

[48] B. Jahne. Analytical studies of low-level motion estimators in space-time images using a unified filter concept. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 229–236, June 1994.

[49] H. Liu, R. Chellappa, and A. Rosenfeld. Accurate dense optical flow estimation and segementation using adaptive structure tensors and a parametric model. *IEEE Trans. Image Processing*, 12(10):1170–1180, 2003.

[50] D. Tschumperle. *PDE's Based Regularization of Multivalued Images and Applications*. PhD thesis, Universite de Nice - Sophia Antipolis, December 2002.

[51] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.Y. Shum. Mesh editing with poisson-based gradient field manipulation. *ACM Trans. Graph.*, 23(3):644–651, 2004.

[52] D.A. Forsyth. Shape from texture and integrability. In *Proc. Int'l Conf. Computer Vision*, pages 447–452, July 2001.

[53] Q. Zheng and R. Chellappa. Estimation of illuminant direction, albedo, and shape from shading. *IEEE Trans. Pattern Anal. Machine Intell.*, 13(7):680–702, 1991.

[54] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Machine Intell.*, 23(6):643–660, 2001.

[55] P. Kovesi. Shapelets correlated with surface normals produce surfaces. In *Proc. Int'l Conf. Computer Vision*, pages 994–1001, 2005.

[56] N. Petrovic, I. Cohen, B.J. Frey, R. Koetter, and T.S. Huang. Enforcing integrability for surface reconstruction algorithms using belief propagation in graphical models. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 743–748, 2001.

[57] P.N. Belhumeur, D.J. Kriegman, and A.L. Yuille. The bas-relief ambiguity. *Int'l J. Computer Vision*, 35(1):33–44, 1999.

[58] A. Yuille and D. Snow. Shape and albedo from multiple images using integrability. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 158–164, 1997.

[59] J. Fan and L.B. Wolff. Surface curvature from integrability. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 520–526, 1994.

[60] S.K. Zhou, R. Chellappa, and D.W Jacobs. Characterization of human faces under illumination variations using rank, integrability, and symmetry constraints. In *Proc. European Conf. Computer Vision*, volume 1, pages 588–601, 2004.

[61] A. L. Yuille, D. Snow, R. Epstein, and P.N. Belhumeur. Determining generative models of objects under varying illumination: Shape and albedo from multiple images using SVD and integrability. *Int'l J. Computer Vision*, 35(3):203–222, 1999.

[62] K. Polthier and E. Preuss. Variational approach to vector field decomposition. In *Proc. Eurographics Workshop on Scientific Visualization*, 2000.

[63] Y. Tong, S. Lombeyda, A.N. Hirani, and M. Desbrun. Discrete multiscale vector field decomposition. *ACM Trans. Graph.*, 22(3):445–452, 2003.

[64] S. Osher and L.I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM J. Numer. Anal.*, 27(4):919–940, 1990.

[65] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barluad. Deterministic edge-preserving regularization in computed imaging. *IEEE Trans. Image Processing*, 6(2):298–311, February 1997.

[66] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations*, volume 147 of *Applied Mathematical Sciences*. Springer-Verlag, 2002.

[67] D. Zwillinger. *Handbook of Differential Equations*. Academic Press, Boston, MA, 3 edition, 1997.

[68] B. Karacali and W. Snyder. Noise reduction in surface reconstruction from a given gradient field. *Int'l J. Computer Vision*, 60(1):25–44, 2004.

[69] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381–395, 1981.

[70] P.J. Huber. *Robust Statistics*. John Wiley and Sons, New York, 1981.

[71] F.R. Hampel, E.M. Ronchetti, P.J. Rousseeuw, and W.A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. Wiley, New York, 1986.

[72] Z. Zhang. M-estimators. http://www-sop.inria.fr/robotvis/personnel/zzhang/Publis/Tutorial-Estim/node25.html.

[73] B.K.P. Horn and B.G. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1981.

[74] I. Horovitz and N. Kiryati. Depth from gradient fields and control points: Bias correction in photometric stereo. *Image and Vision Computing*, 22:681–694, 2004.

[75] R. Kimmel and I. Yavneh. An algebraic multigrid approach to image analysis. *SIAM J. Sci. Comput.*, 24(4):1218–1231, 2003.

[76] M. Shao, T. Simchony, and R. Chellappa. New algorithms for reconstruction of 3-d depth map from one or more images. In *Proc. Conf. Computer Vision and Pattern Recognition*, June 1988.

[77] A. Agrawal. Matlab code for gradient domain algorithms. http://www.umiacs.umd.edu/∼aagrawal.

[78] H.G. Barrow and J.M. Tenenbaum. Recovering intrinsic scene characteristics from images. In *Computer Vision Systems*, pages 3–26, 1978.

[79] A.M. Elgammal, D. Harwood, and L.S. Davis. Non-parametric model for background subtraction. In *Proc. European Conf. Computer Vision*, pages 751–767, 2000.

[80] C. Stauffer and W.E.L. Grimson. Adaptive background mixture models for real-time tracking. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 2, pages 252–258, 1999.

[81] M. Piccardi. Background subtraction techniques: a review. In *Proc. IEEE SMC Intl. Conf. Systems, Man and Cybernetics*, October 2004.

[82] B. Sarel and M. Irani. Separating transparent layers through layer information exchange. In *Proc. European Conf. Computer Vision*, volume 4, pages 328–341, 2004.

[83] R. Szeliski, S. Avidan, and P. Anandan. Layer extraction from multiple images containing reflections and transparency. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 246–243, June 2000.

[84] S.K. Nayar, X.S. Fang, and T. Boult. Separation of reflection components using color and polarization. *Int'l J. Computer Vision*, 21(3):163–186, February 1997.

[85] Y.Y. Schechner, N. Kiryati, and R. Basri. Separation of transparent layers using focus. *Int'l J. Computer Vision*, 39(1):25–39, August 2000.

[86] H. Farid and E.H. Adelson. Separating reflections and lighting using independent components analysis. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 1262–1267, June 1999.

[87] A. Levin, A. Zomet, and Y. Weiss. Separating reflections from a single image using local features. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 306–313, June 2004.

[88] B. Jahne. *Spatio-Temporal Image Processing, Theory and Scientific Applications*, volume 751 of *Lecture Notes in Computer Vision*. Springer-Verlag, Berlin, Germany, 1993.

[89] D. Tschumperle and R. Deriche. Vector-valued image regularization with pdes : A common framework for different applications. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 651–656, June 2003.

[90] J. Weickert. Coherence-enhancing diffusion filtering. *Int'l J. Computer Vision*, 31:111–127, 1999.

[91] J. Lichtenauer, M. Reinders, and E. Hendriks. Influence of the observation likelihood function on particle filtering performance in tracking applications. In *Sixth IEEE Int'l. Conf. on Automatic Face and Gesture Recognition*, pages 767–772, May 2004.

[92] G.D. Finlayson, M.S. Drew, and C. Lu. Intrinsic images by entropy minimization. In *Proc. European Conf. Computer Vision*, volume 3, pages 582–595, 2004.

[93] I. Horovitz and N. Kiryati. Depth from gradient fields and control points: bias correction in photometric stereo. *Image Vision Comput*, 22(9):681–694, 2004.

[94] R. Raskar, K.H. Tan, R. Feris, J. Yu, and M. Turk. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. *ACM Trans. Graph.*, 23(3):679–688, 2004.

[95] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Trans. Image Processing*, 10(8):1200–1211, August 2001.

[96] A. Levin, A. Zomet, and Y. Weiss. Learning how to inpaint from global image statistics. In *Proc. Int'l Conf. Computer Vision*, pages 305–312, 2003.

[97] S. Mann and R. W. Picard. Being undigital with digital cameras: extending dynamic range by combining differently exposed pictures. In *Proc. of IS&T 48th annual conference*, pages 422–428, 1995.

[98] P.E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. *Journal of Computer Graphics*, 31(Annual Conference Series):369–378, August 1997.

[99] S.B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High dynamic range video. *ACM Trans. Graph.*, 22(3):319–325, 2003.

[100] Smal camera. http://www.smalcamera.com/, 2004.

[101] Dalstar, inc. http://www.pinnaclevision.co.uk, 2004.

[102] D. Yang, A.E. Gamal, B. Fowler, and H. Tianet. A 640×512 cmos image sensor with ultrawide dynamic range floating-point pixel-level ADC. *IEEE Journal on Solid State Circuits*, 34:1821–1834, 1999.

[103] S. K. Nayar and V. Branzoi. Adaptive dynamic range imaging: Optical control of pixel exposures over space and time. In *Proc. Int'l Conf. Computer Vision*, pages 1168–1175, 2003.

[104] S. K. Nayar, V. Branzoi, and T. Boult. Programmable imaging using a digital micromirror array. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 436–443, 2004.

[105] Fraunhofer IMS. One Mpixel CMOS HDR array, 2001.

[106] J. Tumblin, A. Agrawal, and R. Raskar. Why I want a gradient camera. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 103–110, June 2005.

[107] L.I. Rudin and S. Osher. Total variation based image restoration with free local constraints. In *Proc. Int'l Conf. Image Processing*, volume 1, pages 31–35, 1994.

[108] A. Agrawal, R. Raskar, and R. Chellappa. Edge suppression by gradient field transformation using cross-projection tensors. In *Proc. Conf. Computer Vision and Pattern Recognition*, June 2006.

[109] A. Agrawal, R. Raskar, and R. Chellappa. What is the range of surface reconstructions from a gradient field? In *Proc. European Conf. Computer Vision*, May 2006.

[110] A. Agrawal, R. Chellappa, and R. Raskar. An algebraic approach to surface reconstruction from gradient fields. In *Proc. Int'l Conf. Computer Vision*, volume 1, pages 174–181, October 2005.

[111] A. Agrawal, R. Raskar, S.K. Nayar, and Y. Li. Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Trans. Graphics (Proc. SIGGRAPH)*, 24(3):828–835, 2005.

[112] A. Agrawal and R. Chellappa. Robust ego-motion estimation and 3D model refinement using surface parallax. *IEEE Trans. Image Processing*, 15(5), May 2006.

[113] A. Agrawal and R. Chellappa. Fusing depth and video using rao-blackwellized particle filter. In *Proc. Int'l Conf. Pattern Recognition and Machine Intelligence*, pages 521–526, December 2005.

[114] A. Agrawal and R. Chellappa. 3D model refinement using surface-parallax. In *Proc. Int'l Conf. Acoustics, Speech and Signal Processing*, pages 285–288, Montreal, Canada, May 2004.

[115] A. Agrawal and R. Chellappa. Robust ego-motion estimation and 3D model refinement using depth based parallax model. In *Proc. Int'l Conf. Image Processing*, volume 4, pages 2483–2486, Singapore, October 2004.

[116] A. Agrawal and R. Chellappa. Moving object segmentation and dynamic scene reconstruction using two frames. In *Proc. Int'l Conf. Acoustics, Speech and Signal Processing*, volume 2, pages 705–708, Philadelphia, March 2005.

[117] A. Agrawal and R. Chellappa. Ego-motion estimation and 3D model refinement in scenes with varying illumination. In *Workshop on Applications of Computer Vision/MOTION*, pages 140–146, 2005.