

## Edge Suppression by Gradient Field Transformation using Cross-Projection Tensors

Amit Agrawal, Ramesh Raskar, Rama Chellappa

TR2006-058 June 2006

### Abstract

We propose a new technique for edge-suppressing operations on images. We introduce cross projection tensors to achieve affine transformations of gradient fields. We use these tensors, for example, to remove edges in one image based on the edge-information in a second image. Traditionally, edge suppression is achieved by setting image gradients to zero based on thresholds. A common application is in the Retinex problem, where the illumination map is recovered by suppressing the reflectance edges, assuming it is slowly varying. We present a class of problems where edge-suppression can be a useful tool. These problems involve analyzing images of the same scene under variable illumination. Instead of resetting gradients, the key idea in our approach is to derive local tensors using one image and to transform the gradient field of another image using them. Reconstructed image from the modified gradient field shows suppressed edges or textures at the corresponding locations. All operations are local and our approach does not require any global analysis. We demonstrate the algorithm in the context of several applications such as (a) recovering the foreground layer undervarying 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.

*IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*

This work may not be copied or reproduced in whole or in part for any commercial purpose. Permission to copy in whole or in part without payment of fee is granted for nonprofit educational and research purposes provided that all such whole or partial copies include the following: a notice that such copying is by permission of Mitsubishi Electric Research Laboratories, Inc.; an acknowledgment of the authors and individual contributions to the work; and all applicable portions of the copyright notice. Copying, reproduction, or republishing for any other purpose shall require a license with payment of fee to Mitsubishi Electric Research Laboratories, Inc. All rights reserved.



# What is the Range of Surface Reconstructions from a Gradient Field?

Amit Agrawal<sup>1</sup>, Ramesh Raskar<sup>2</sup>, and Rama Chellappa<sup>1</sup>

<sup>1</sup> Center for Automation Research, University of Maryland  
College Park, MD, USA 20742

{aagrawal, rama}@cfar.umd.edu

<sup>2</sup> Mitsubishi Electric Research Labs (MERL)

201 Broadway, Cambridge, MA, USA 02139

{raskar}@merl.com

**Abstract.** We propose a generalized equation to represent a continuum of surface reconstruction solutions of a given non-integrable gradient field. We show that common approaches such as Poisson solver and Frankot-Chellappa algorithm are special cases of this generalized equation. For a  $N \times N$  pixel grid, the subspace of all integrable gradient fields is of dimension  $N^2 - 1$ . Our framework can be applied to derive a range of meaningful surface reconstructions from this high dimensional space. The key observation is that the range of solutions is related to the degree of anisotropy in applying weights to the gradients in the integration process. While common approaches use isotropic weights, we show that by using a progression of spatially varying anisotropic weights, we can achieve significant improvement in reconstructions. We propose (a)  $\alpha$ -surfaces using binary weights, where the parameter  $\alpha$  allows trade off between smoothness and robustness, (b) M-estimators and edge preserving regularization using continuous weights and (c) Diffusion using affine transformation of gradients. We provide results on photometric stereo, compare with previous approaches and show that anisotropic treatment discounts noise while recovering salient features in reconstructions.

## 1 Introduction

Reconstruction from gradient fields is important in several applications such as photometric stereo (PS) and shape from shading (SfS) [1], mesh smoothing, retinex [2], high dynamic range compression [3], phase unwrapping, image editing, matting and fusion [4]. In gradient based algorithms, the gradient field of images is manipulated to achieve the desired goal and the final image is obtained by a 2D integration of the manipulated gradient field. In PS/SfS, surface normals/gradients are obtained first and the desired surface is obtained by integrating the gradient field. The gradient field of a 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 obtained 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.

Previous methods have used the integrability constraint during the estimation of surface (or surface normals) in PS, SfS and Shape from Texture as in [1][5]. In these methods, integrability is enforced as a constraint to regularize the solution or to remove the inherent ambiguities. For example, by enforcing integrability in uncalibrated PS, the ambiguity in shape estimation can be reduced to a generalized bas-relief transformation [6]. Another class of methods first estimate the gradient field and then apply integrability to estimate the surface as in [7][8][9][10]. We propose a general framework for surface reconstruction when a non-integrable gradient field is already provided.

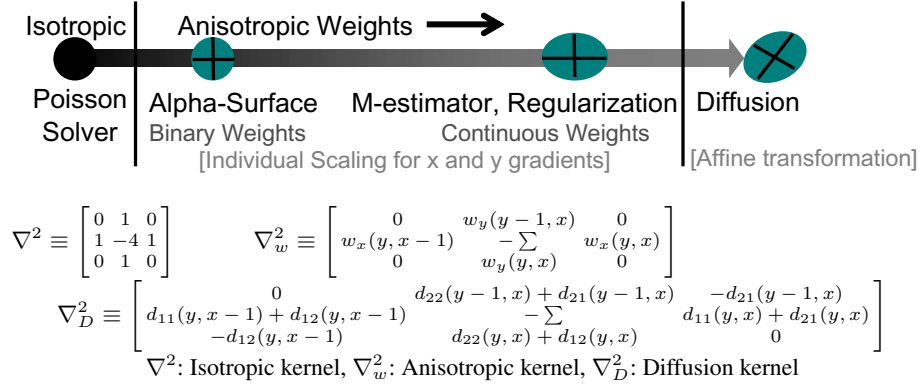
Frankot & Chellappa [8] project the non-integrable gradient field on to a set of integrable slopes using the Fourier basis functions. Several variants of this approach have been proposed by either choosing a different basis function [11] (cosine functions) or using a redundant non-orthogonal set of basis functions (shapelets) [12]. In [7], a direct analytical solution based on solving a Poisson equation was proposed. Petrovic et al. [10] used a loopy belief propagation algorithm to obtain the integrable gradient field from a given non-integrable gradient field assuming Gaussian noise in the gradients. Most of these methods are based on minimizing a least square cost function, try to estimate a smooth surface and do not consider the effect of outliers in the given gradient field. A natural approach to overcome outliers and reduce noise would be to use a robust estimation like RANSAC. However, due to the high dimensionality, applying RANSAC is computationally prohibitive.

Noise reduction in images is a topic commonly addressed in image restoration techniques. Several PDE's based methods such as anisotropic diffusion [13], shock filters and energy based methods [14] (see [15] for detailed analysis and algorithms) have been proposed that try to restore an image while maintaining edges or sharp features. Inspired by the success of these approaches, we show how to incorporate robust estimation, regularization and anisotropic diffusion in the gradient integration problem.

**Contributions:** The contributions of our paper are as follows

- We present a generalized equation for surface reconstruction from non-integrable gradient fields. This unification results in a continuum of solutions based on the degree of anisotropy in assigning weights to the gradients during the integration.
- We show that common approaches such as Poisson solver and Frankot-Chellappa algorithm can be formulated as special cases of our framework at one end of the continuum and correspond to isotropic gradient weights.
- We derive new types of reconstructions using a progression of spatially varying anisotropic weights along the continuum. We propose a solution based on the general affine transformation of the gradients using diffusion tensors near the other end of the continuum and show that it produces better feature preserving reconstructions compared to previous methods.

The subspace of all integrable gradient fields for a  $N \times N$  pixel grid is of dimension  $N^2 - 1$  [16] and it is not possible (and practical) to characterize all the solutions. The solutions we propose constitute a range of *meaningful* solutions that might be close to the desired surface. Although we describe 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 sharp discontinuities, least square approaches may give good solutions



**Fig. 1.** A continuum of solutions can be derived using our framework by changing  $f_i$ 's in (6). At one end is the Poisson solver which gives equal weight to all the gradients, resulting in a spatially invariant isotropic Laplacian kernel  $\nabla^2$ . *Individual scaling* of the gradients using spatially varying weights (binary for  $\alpha$ -surface, continuous for M-estimator and Regularization) results in anisotropic kernel  $\nabla_w^2$  ( $\sum$  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  $\nabla_D^2$ .

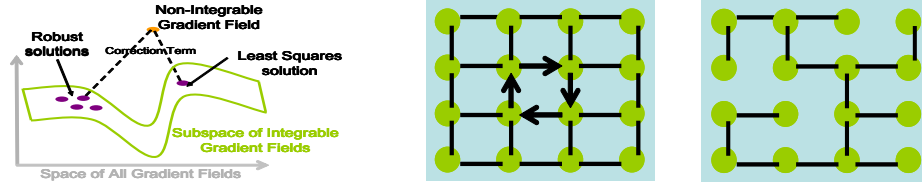
while handling noise. With sharp features in surface, the proposed diffusion and alpha-surface methods produce better feature preserving reconstructions in the presence of noise and outliers.

## 2 Problem Statement

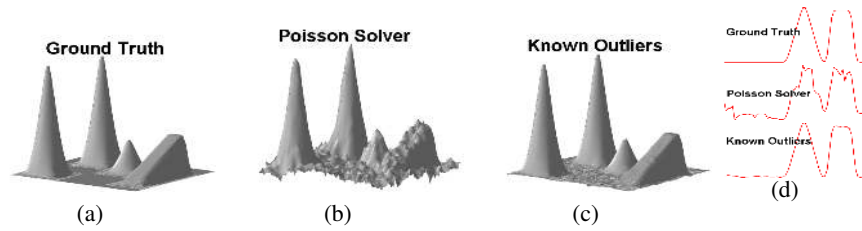
Consider a  $H \times W$  rectangular grid  $(y, x)$  of image pixels. Let  $\{p(y, x), q(y, x)\}$  denote the given non-integrable gradient field over this grid. Define the curl and divergence operators as:  $\text{curl}(p, q) = \frac{\partial p}{\partial y} - \frac{\partial q}{\partial x}$ ,  $\text{div}(p, q) = \frac{\partial p}{\partial x} + \frac{\partial q}{\partial y}$ . 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 function given by [7][1]

$$J(Z) = \int \int ((Z_x - p)^2 + (Z_y - q)^2) dx dy. \quad (1)$$

The Euler-Lagrange equation gives the *Poisson equation*:  $\nabla^2 Z = \text{div}(p, q)$ . We will refer to this method as Poisson solver. One can always write  $\{Z_x, Z_y\} = \{p, q\} + \{\epsilon_x, \epsilon_y\}$ , where  $\{\epsilon_x, \epsilon_y\}$  denote the *correction gradient field* which is added to the given non-integrable field to make it integrable. It was shown in [16] that integrable gradient fields form a subspace of dimension  $HW - 1$  in the  $2HW$ -dimensional space of all gradient fields. By adding the correction gradient field, one can move from a point in the  $2HW$ -dimensional space corresponding to the given non-integrable gradient field to the subspace of valid integrable gradient fields. From (1), Poisson solver minimizes  $J(Z) = \int \int (\epsilon_x^2 + \epsilon_y^2) dx dy$ . Thus, Poisson solver finds that solution which minimizes the norm of the correction gradient field (see Fig. 2).



**Fig. 2. Space of all solutions** (Left) Poisson solver finds the solution corresponding to the minimum norm correction gradient field, but this may not be robust. (Middle) We show that all gradients are not required for integration using a graph analogy. A 2D graph corresponding to a sample  $4 \times 4$  grid. Nodes correspond to the value of the surface at the grid points and gradients correspond to the edges (Right) A spanning tree is the minimal configuration required for gradient integration. Using only those gradients which correspond to the edges in the spanning tree, all node values can be obtained up to a constant of integration



**Fig. 3. Effect of outliers in 2D integration** (a) True surface (b) Gaussian noise ( $\sigma = 0.02g$ ,  $g =$  maximum gradient magnitude) and uniformly distributed outliers were added to the gradients of this surface. Reconstruction using 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

It is well known that a least square solution does not perform well in the presence of outliers. Consider the surface shown in Figure 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 Poisson solver is shown in Figure 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 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. How can we find other meaningful solutions in the space of all solutions? In the next section, we put forward a framework to do so.

### 3 A General Framework

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

$$J = \int \int E(Z, p, q, Z_{x^a y^b}, p_{x^c y^d}, q_{x^c y^d}, \dots) dx dy, \quad (2)$$

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  $J = \int \int E(Z, p, q, Z_x, Z_y) dx dy$ . The Euler-Lagrange equation gives

$$\frac{\partial E}{\partial Z} - \frac{d}{dx} \frac{\partial E}{\partial Z_x} - \frac{d}{dy} \frac{\partial E}{\partial Z_y} = 0 \quad \text{or} \quad \frac{\partial E}{\partial Z} = \text{div} \left( \frac{\partial E}{\partial Z_x}, \frac{\partial E}{\partial Z_y} \right). \quad (3)$$

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

$$\frac{\partial E}{\partial Z_x} = f_1(Z_x, Z_y) - f_3(p, q), \quad \frac{\partial E}{\partial Z_y} = f_2(Z_x, Z_y) - f_4(p, q), \quad (4)$$

where  $f_i : \mathbf{R} \times \mathbf{R} \rightarrow \mathbf{C}$ ,  $i = 1 \dots 4$  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

$$\frac{\partial f_1(Z_x, Z_y)}{\partial Z_y} = \frac{\partial f_2(Z_x, Z_y)}{\partial Z_x}. \quad (5)$$

Substituting (4) into (3) and bringing all  $Z$  terms on one side, we get

$$\text{div}(f_1(Z_x, Z_y), f_2(Z_x, Z_y)) - \frac{\partial E}{\partial Z} = \text{div}(f_3(p, q), f_4(p, q)). \quad (6)$$

We first show that the previous solutions such as Poisson solver and Frankot-Chellappa algorithm (in general, projection onto continuous basis functions) can be derived using (6). We then propose other solutions using the above equation. In all solutions we assume Neumann boundary conditions given by  $\nabla Z \cdot \hat{\mathbf{n}} = 0$ .

**Poisson Solver (Spatially Invariant Isotropic Weights):** The Poisson equation

$\text{div}(Z_x, Z_y) = \text{div}(p, q)$  can be obtained from (6) by substituting  $\frac{\partial E}{\partial Z} = 0$ ,  $f_1(Z_x, Z_y) = Z_x$ ,  $f_2(Z_x, Z_y) = Z_y$ ,  $f_3(p, q) = p$ ,  $f_4(p, q) = q$  (see Table 1). (5) is satisfied as both sides are zero.

*Numerical Solution:* Let  $u = \text{div}(p, q)$ . Using finite differences and vectoring the 2D matrices in lexicographical ordering, the Poisson equation can be discretized to give  $\mathbf{LZ} = \mathbf{u}$ , where  $\mathbf{u} = [u(1, 1), \dots, u(H, W)]^T$  and the matrix  $\mathbf{L}$  is the sparse Laplacian matrix<sup>3</sup> 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  $\nabla^2$ .  $Z$  can be obtained as  $\mathbf{Z} = \mathbf{L}^{-1}\mathbf{u}$ .

**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  $s(x, y)$ <sup>4</sup>. Given  $\{p, q\}$ ,  $Z$  is obtained as [8]

$$Z = \mathcal{F}^{-1} \left( -j \frac{\xi_x \mathcal{F}(p) + \xi_y \mathcal{F}(q)}{\xi_x^2 + \xi_y^2} \right). \quad (7)$$

<sup>3</sup> The Laplacian matrix needs to be modified at the boundary according to the boundary conditions.

<sup>4</sup>  $\mathcal{F}(s(x, y)) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} s(x, y) e^{-j(\xi_x x + \xi_y y)} dx dy$

Let  $\phi(x, y, \xi_x, \xi_y) = e^{j(\xi_x x + \xi_y y)}$ . We have  $\phi_x = j\xi_x \phi$ ,  $\phi_y = j\xi_y \phi$ . Substituting  $\frac{\partial E}{\partial Z} = 0$ ,  $f_1(Z_x, Z_y) = \mathcal{F}(Z_x)\phi$ ,  $f_2(Z_x, Z_y) = \mathcal{F}(Z_y)\phi$ ,  $f_3(p, q) = \mathcal{F}(p)\phi$ ,  $f_4(p, q) = \mathcal{F}(q)\phi$  in (6), we get

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

which is equivalent to (7). The *projection* on the Fourier basis functions is implicit in the above definition of  $f_i$ 's which transforms the domain as weighted basis functions  $\phi$ , the weights being equal to the Fourier transform coefficients. One can generalize this approach to use any set of ortho-normal basis functions  $\phi$ . Kovési's [12] algorithm is in a similar spirit while using a redundant set of non-orthogonal basis functions.

In the next section, we show how the 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 ( $\nabla^2$ ) which gives equal weights to gradients. This results in 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 correction gradient field.

## 4 A Continuum of Solutions

Techniques for robust estimation includes the well-known RANSAC [17] algorithm and M-estimators. We first show that applying RANSAC to gradient integration is computationally prohibitive. To do that 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. 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. This can be seen as follows.

Define a 2D graph over the grid, where the nodes correspond to the value of the surface at each pixel (grid point) and the edges correspond to the gradients (see Fig. 2). To be able to integrate, each 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  $m = HW - 1$ .

### 4.1 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  $\tau$ . 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  $\tau$ .
- Repeat  $T$  times and choose that spanning tree using which maximum number of inliers are obtained.

In [17], it was shown that to ensure with probability  $\gamma$  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  $T = \log(1 - \gamma) / \log(1 - w^m)$  and  $w$  is the probability that a particular data point is an inlier. However,  $T$  becomes extremely large as the size of grid is increased. For example, assuming  $w = 0.95$ , even for a  $16 \times 16$  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.

## 4.2 $\alpha$ -surface: Anisotropic Scaling using Binary Weights

As noted in [17], if there is a problem related rationale for choosing the set of inliers, one should use a deterministic selection process instead of a random one. 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.

Suppose we fix an initial spanning tree, claiming all gradients corresponding to the edges in this spanning tree to be inliers. We define  $\alpha$ -surface as an iterative scheme, where at each iteration, based on the tolerance level  $\alpha$ , all gradients for which the correction term is less than  $\alpha$  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  $\alpha \geq 0$ ,  $\alpha$ -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  $\alpha$  decides between outliers and inliers. If  $\alpha = 0$ , only the gradients corresponding to the initial spanning tree are considered as inliers and are used for integration. As  $\alpha$  is increased, more gradients are used for integration. At a large value of  $\alpha$ , all gradients will be treated as inliers and the solution becomes equivalent to that given by the Poisson solver. **By changing  $\alpha$ , 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,  $\alpha$ -surface is a weighted approach, where the weights are 1 for gradients in  $S$  (used for integration) and 0 otherwise. If we define

$$b_x(x, y) = 1 \quad \text{if } Z_x \in S, \quad 0 \quad \text{o.w.}, \quad b_y(x, y) = 1 \quad \text{if } Z_y \in S, \quad 0 \quad \text{o.w.}, \quad (8)$$

then the error functional  $J$  for each iteration of  $\alpha$ -surface can be written as

$$J = \int \int b_x(Z_x - p)^2 + b_y(Z_y - q)^2 dx dy . \quad (9)$$

The corresponding Euler-Lagrange equation is  $\text{div}(b_x Z_x, b_y Z_y) = \text{div}(b_x p, b_y q)$ . 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.

*Determining initial spanning tree:* An easy way to fix an initial spanning tree is to assign weights to each edge and find the minimum spanning tree (MST). In [18], an approach for curl correction was presented, where first all edges corresponding to non-zero curl were broken. The resulting graph was connected by finding the set of links with minimum total weight by assigning curl values as weights. We have experimented with two types of edge weights: one based on curl values and other based on gradient magnitude. In our experience, assigning gradient magnitude as weights gives better results compared to curl values. For results presented in Sect. 5, we use gradient magnitude as weights.

*Determining  $\alpha$ :* Suppose that the gradients are corrupted by additive IID Gaussian noise  $\mathbf{N}(0, \sigma^2)$ . 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(middle)). The integral along this loop is

$$C_{p,q}(y, x) = p(y + 1, x) - p(y, x) + q(y, x) - q(y, x + 1) . \quad (10)$$

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  $\sigma$  as  $\sigma = \sqrt{(\sigma_C^2 - \mu_C^2)/4}$ , where  $(\mu_C, \sigma_C^2)$  denote the estimated mean and variance of  $C_{p,q}$  using the given gradient field  $\{p, q\}$ . We use  $\alpha = 1.5\sigma$ .

*Numerical Solution:* Let  $u_b = \text{div}(b_x p, b_y q)$ .  $\text{div}(b_x Z_x, b_y Z_y)$  can be written as  $\nabla_b^2 Z$ , where  $\nabla_b^2$  is the *weighted Laplacian kernel* (Fig. 1,  $\nabla_w^2$  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  $\mathbf{Z} = \mathbf{L}_b^{-1} \mathbf{u}_b$ . 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 we show how to generalize the inlier/outlier weighting scheme to approaches based on continuous weights.

### 4.3 Anisotropic Scaling using Continuous Weights

In **M-estimators**, the effect of outliers is reduced by replacing the squared error residual  $\rho(\cdot) = (\cdot)^2$  by another function of residuals. Here  $\rho$  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. M-estimators can be formulated as an iterative re-weighted least squares solution

$$J = \int \int w(\epsilon_x^{k-1})(Z_x - p)^2 + w(\epsilon_y^{k-1})(Z_y - q)^2 dx dy , \quad (11)$$

where 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  $\rho$ . The Euler-Lagrange equation of (11)

gives  $\text{div}(w_x Z_x, w_y Z_y) = \text{div}(w_x p, w_y q)$ . This is similar to  $\alpha$ -surface except that the weights are continuous.  $Z$  can be obtained as  $\mathbf{Z} = \mathbf{L}_w^{-1} \mathbf{u}_w$ .

Ill-posed problems (such as estimating optical flow) are often solved by **regularization**. The Poisson solver can be regularized by modifying the error function as

$$J(Z) = \int \int ((Z_x - p)^2 + (Z_y - q)^2) + \lambda(\phi(Z_x) + \phi(Z_y)) dx dy, \quad (12)$$

where the second term is the regularization term using function  $\phi$ . Common examples include  $\phi(s) = \sqrt{1 + s^2}$  and  $\phi(s) = \log(1 + s^2)$ . The Euler-Lagrange equation of the above error functional gives:  $\text{div}(Z_x, Z_y) + (\lambda/2)\text{div}(\phi'(Z_x), \phi'(Z_y)) = \text{div}(p, q)$ . In terms of (6), this corresponds to  $\frac{\partial E}{\partial Z} = 0$ ,  $f_1(Z_x, Z_y) = Z_x + \frac{\lambda}{2}\phi'(Z_x)$ ,  $f_2(Z_x, Z_y) = Z_y + \frac{\lambda}{2}\phi'(Z_y)$ ,  $f_3(p, q) = p$ ,  $f_4(p, q) = q$  (Table 1). Minimizing the energy as above is difficult because of the above equation being non-linear. Using the principle of half-quadratic minimization (see [14] for details), one can introduce auxiliary variables  $w = (w_x, w_y)$ . Minimizing (12) 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  $Z^k$ :  $\nabla^2 Z^k + \lambda \text{div}(w_x^k Z_x^k, w_y^k Z_y^k) = \text{div}(p, q)$

The equation for solving  $Z^k$  can be rewritten as  $(\nabla^2 + \lambda \nabla_{w^k}^2) Z^k = \text{div}(p, q)$ , where  $\nabla_{w^k}^2$  is the weighted Laplacian kernel (Fig. 1). The solution is given by  $\mathbf{Z}^k = (\mathbf{L} + \lambda \mathbf{L}_{w^k})^{-1} \mathbf{u}$ .

#### 4.4 Affine Transformation of Gradients using Diffusion Tensors

Image restoration from noisy images has been a classical problem in image processing. Anisotropic diffusion [13] and energy minimization methods [14][15] are some of the common approaches for image restoration. Weickert [19] proposed a generalization of divergence based equation for image restoration, given by  $I_t = \text{div}(D \nabla I)$ , where  $D(y, x) = \begin{bmatrix} d_{11}(y, x) & d_{12}(y, x) \\ d_{21}(y, x) & d_{22}(y, x) \end{bmatrix}$  is a  $2 \times 2$  symmetric, positive-definite matrix at each pixel (a field of diffusion tensors). We propose to generalize the Poisson solver using  $D$  as

$$\text{div}(D \begin{bmatrix} Z_x \\ Z_y \end{bmatrix}) = \text{div}(D \begin{bmatrix} p \\ q \end{bmatrix}). \quad (13)$$

The above equation is the Euler-Lagrange equation of the following error functional:  $J(Z) = \int \int d_{11}(Z_x - p)^2 + (d_{12} + d_{21})(Z_x - p)(Z_y - q) + d_{22}(Z_y - q)^2 dx dy$  and can be written as

$$\text{div}(d_{11} Z_x + d_{12} Z_y, d_{21} Z_x + d_{22} Z_y) = \text{div}(d_{11} p + d_{12} q, d_{21} p + d_{22} q). \quad (14)$$

Note that (14) can be obtained from (6) by substituting  $\frac{\partial E}{\partial Z} = 0$ ,  $f_1(Z_x, Z_y) = d_{11} Z_x + d_{12} Z_y$ ,  $f_2(Z_x, Z_y) = d_{21} Z_x + d_{22} Z_y$ ,  $f_3(p, q) = d_{11} p + d_{12} q$ ,  $f_4(p, q) = d_{21} p + d_{22} q$  (Table 1). 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 (5) 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 we loosely call this scheme as Diffusion, there is no notion of time or iteration in this scheme.

Let  $u_D = \text{div}(d_{11}p + d_{12}q, d_{21}p + d_{22}q)$ . (14) can be written as  $\nabla_D^2 Z = u_D$ , where  $\nabla_D^2$  denote the weighted Laplacian kernel based on the diffusion tensor  $D$  (Fig. 1). The solution is given by  $\mathbf{Z} = \mathbf{L}_D^{-1} \mathbf{u}_D$ .

*Obtaining diffusion tensor:* Several schemes for obtaining diffusion tensor such as edge preserving [15](Eq. 3.60) and coherence preserving [19] have been proposed. We use an edge-preserving diffusion tensor obtained as follows. At each pixel, we find a  $2 \times 2$  matrix  $H$  by convolving component wise  $\begin{bmatrix} p^2 & p \times q \\ p \times q & q^2 \end{bmatrix}$  with a Gaussian kernel. Let  $\mu_1 \geq \mu_2$  denote the eigen-values of  $H$ . We obtain new eigen values  $\lambda_1, \lambda_2$  as:  $\lambda_2 = 1, \lambda_1 = 1$  if  $\mu_1 = 0$ ,  $\lambda_1 = \beta + 1 - \exp(-3.315/\mu_1^4)$  if  $\mu_1 > 0$ . Here  $\beta = 0.02$  to ensure positive-definiteness.  $D$  is obtained from the eigen-vectors of  $H$  and the new eigen-values.

In all the above solutions,  $\frac{\partial E}{\partial Z} = 0$ . Our framework could also be used when the  $Z$  values are known at some control points [20][21] by utilizing the  $\frac{\partial E}{\partial Z}$  term.

## 5 Results

We compare<sup>5</sup> Poisson solver, FC algorithm,  $\alpha$ -surface, M-estimator using Huber function, Diffusion and Regularization using  $\phi(s) = \sqrt{1 + s^2}, \lambda = 10$ . Since the divergence of the given gradient field is not modified in the Regularization method, it usually does not perform as well as other approaches. Table 1 gives the summary of  $f_i$ 's and the equation for each algorithm. Figure 4 shows the reconstructed surfaces using various algorithms from the noisy gradient field of the synthetic surface shown in Figure 3. Note that the surface reconstructed using  $\alpha$ -surface and Diffusion are much better than those reconstructed using other approaches. We also present results on calibrated photometric stereo using synthetic and real sequences. 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 2 gives the MSE between the estimated surface and the true surface for various algorithms.

**Vase:** Six images generated using the Vase depth map are shown in Figure 5. We add Gaussian random noise ( $\sigma = 10\%$  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 5.  $\alpha$ -surface, Diffusion and M-estimator gives better shape estimate compared to the rest of algorithms.

**Mozart:** Five images generated using the Mozart depth map are shown in Figure 6. Gaussian random noise ( $\sigma = 5\%$  of maximum intensity) was added to the images. The

<sup>5</sup> Matlab code is available at <http://www.cfar.umd.edu/~aagrawal>

**Table 1.** A continuum of solutions can be obtained by changing  $f_i$ 's in (6), 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 equation. 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 (6), $\frac{\partial E}{\partial Z} = 0$				Equation
	$f_1(Z_x, Z_y)$	$f_2(Z_x, Z_y)$	$f_3(p, q)$	$f_4(p, q)$	
Poisson solver	$Z_x$	$Z_y$	$p$	$q$	$\mathbf{LZ} = \mathbf{u}$
Frank-Chell	$\mathcal{F}(Z_x)\phi$	$\mathcal{F}(Z_y)\phi$	$\mathcal{F}(p)\phi$	$\mathcal{F}(q)\phi$	(7)
$\alpha$ -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 2.** Mean square errors (MSE) for synthetic data sets

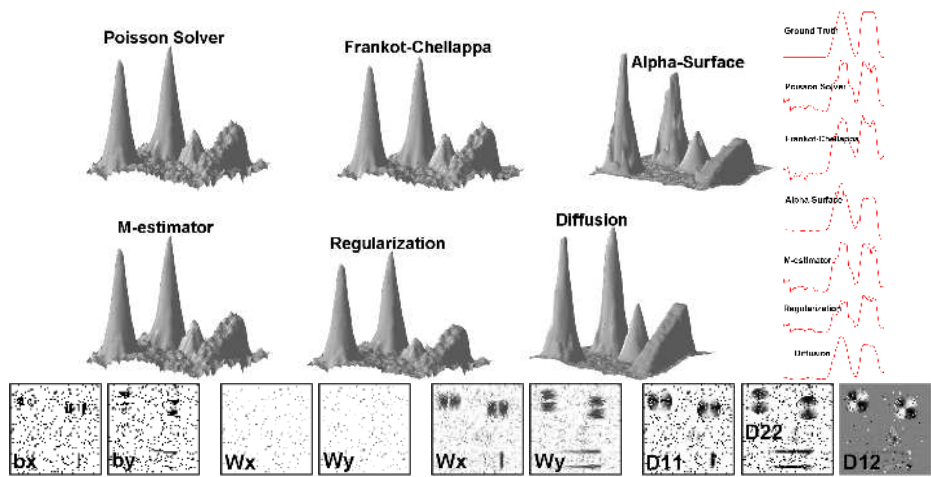
	Poisson-solver	FC	$\alpha$ -surface	M-estimator	Regularization	Diffusion
Ramp-Peaks	10.81	11.20	2.65	9.49	5.35	2.26
Vase	294.46	239.62	22.20	15.14	164.98	2.78
Mozart	2339.24	1316.66	219.72	359.12	806.85	373.72

reconstructed surfaces using various algorithms are also shown in Figure 6. While the discontinuities in the shape are smeared in Poisson solver, FC and Regularization, these are preserved in  $\alpha$ -surface, Diffusion and M-estimator.

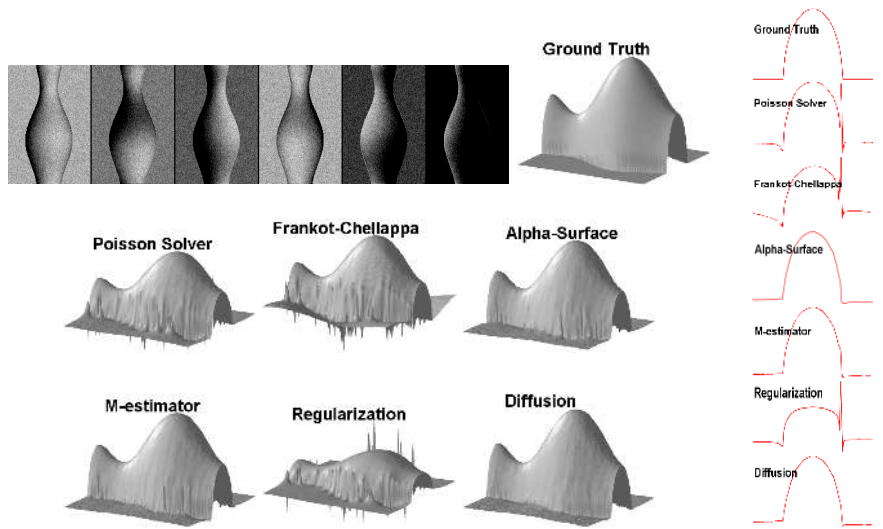
**Flowerpot:** Figure 7 shows results on calibrated photometric stereo using 4 real images of a flowerpot. Notice that least squares solutions (Poisson solver and FC algorithm) are noisy and do not recover all features (such as the top of the flowerpot). Diffusion,  $\alpha$ -surface and M-estimator approaches recovers all salient features while discounting noise.

## 6 Conclusions

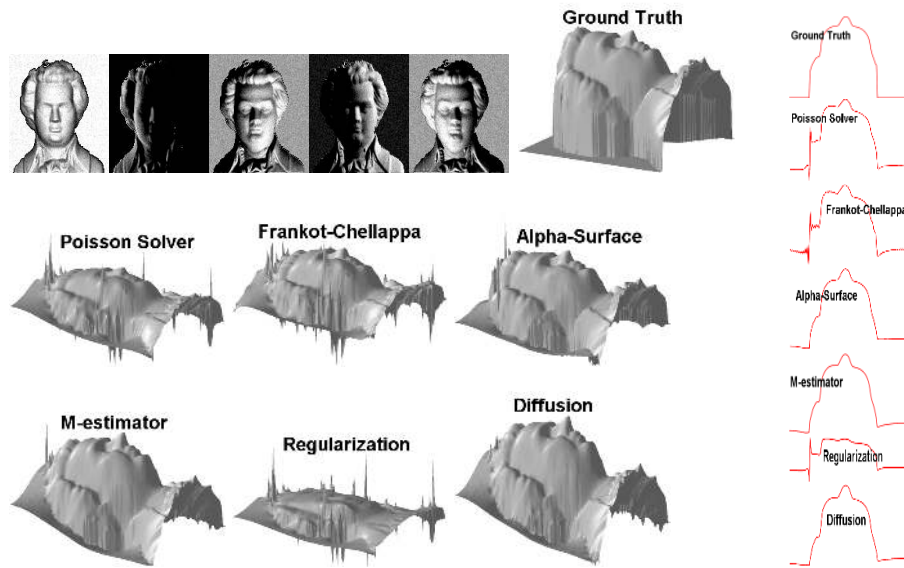
We proposed a general framework for surface reconstruction from gradient fields, based on controlling the anisotropy of weights for gradients during the integration. We showed that previous solutions such as Poisson solver and Frankot-Chellappa algorithm are special cases of our framework. We derived a continuum of solvers:  $\alpha$ -surface (binary weights) where  $\alpha$  allows tradeoff between smoothness and robustness, Regularization and M-estimators (continuous weights) and Diffusion (affine transformation on gradients). Results and comparisons showed that  $\alpha$ -surface and Diffusion method give consistently better feature preserving reconstructions.



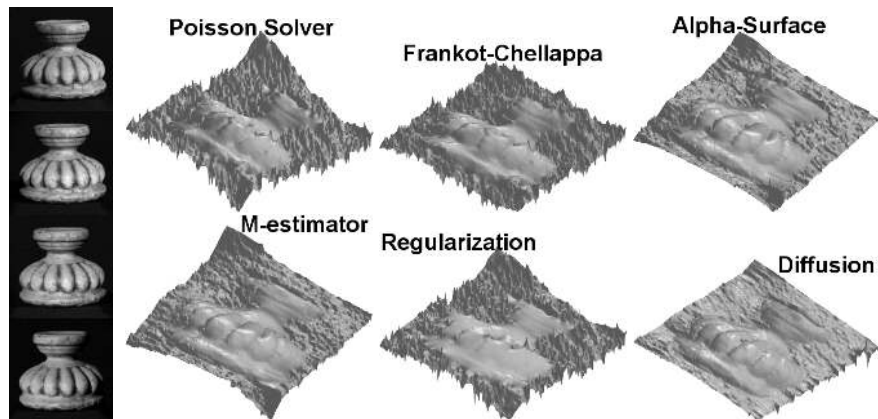
**Fig. 4. Reconstruction in presence of noise and outliers (Ramp-Peaks):** (Top two rows) (Left) Reconstructed surfaces using various algorithms (Right) One-D height plots for a scan line across the middle of grid for various solutions. (Bottom row)  $x$  and  $y$  gradient weights for the last iteration of  $\alpha$ -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  $\alpha$ -surface and Diffusion give much better results compared to other approaches



**Fig. 5. Photometric Stereo on Vase:** (Top row) Noisy input images and 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  $\alpha$ -surface, Diffusion and M-estimator as compared to Poisson solver, FC and Regularization



**Fig. 6. 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  $\alpha$ -surface, Diffusion and M-estimator as compared to other algorithms



**Fig. 7. Photometric Stereo on Flowerpot:** Left column shows 4 real images of a flowerpot. Right columns show the reconstructed surfaces using various algorithms. The reconstructions using Poisson solver and Frankot-Chellappa algorithm are noisy and all features (such as top of flowerpot) are not recovered. Diffusion,  $\alpha$ -surface and M-estimator methods discount noise while recovering all the salient features

## References

1. Horn, B.: Height and gradient from shading. *Int'l J. Computer Vision* **5** (1990) 37–75
2. Horn, B.: Determining lightness from an image. *Comput. Graphics, Image Processing* **3** (1974) 277–299
3. Fattal, R., Lischinski, D., Werman, M.: Gradient domain high dynamic range compression. *ACM Trans. Graph.* **21** (2002) 249–256
4. Agrawal, A., Raskar, R., Nayar, S., Li, Y.: Removing photography artifacts using gradient projection and flash-exposure sampling. *ACM Trans. Graph.* **24** (2005) 828–835
5. Forsyth, D.: Shape from texture and integrability. In: *Proc. Int'l Conf. Computer Vision.* (2001) 447–452
6. Belhumeur, P., Kriegman, D., Yuille, A.: The bas-relief ambiguity. *Int'l J. Computer Vision* **35** (1999) 33–44
7. Simchony, T., Chellappa, R., Shao, M.: Direct analytical methods for solving poisson equations in computer vision problems. *IEEE Trans. Pattern Anal. Machine Intell.* **12** (1990) 435–446
8. Frankot, R.T., Chellappa, R.: A method for enforcing integrability in shape from shading algorithms. *IEEE Trans. Pattern Anal. Machine Intell.* **10** (1988) 439–451
9. Yuille, A., Snow, D.: Shape and albedo from multiple images using integrability. In: *Proc. Conf. Computer Vision and Pattern Recognition.* (1997) 158–164
10. Petrovic, N., Cohen, I., Frey, B., Koetter, R., Huang, T.: Enforcing integrability for surface reconstruction algorithms using belief propagation in graphical models. In: *Proc. Conf. Computer Vision and Pattern Recognition.* Volume 1. (2001) 743–748
11. Georghiades, A.S., Belhumeur, P.N., Kriegman, D.J.: From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Machine Intell.* **23** (2001) 643–660
12. Kovese, P.: Shapelets correlated with surface normals produce surfaces. In: *Proc. Int'l Conf. Computer Vision.* (2005) 994–1001
13. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Machine Intell.* **12** (1990) 629–639
14. Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M.: Deterministic edge-preserving regularization in computed imaging. *IEEE Trans. Image Processing* **6** (1997) 298–311
15. Aubert, G., Kornprobst, P.: *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations.* Volume 147 of Applied Mathematical Sciences. Springer-Verlag (2002)
16. Karacali, B., Snyder, W.: Noise reduction in surface reconstruction from a given gradient field. *Int'l J. Computer Vision* **60** (2004) 25–44
17. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM* **24** (1981) 381–395
18. Agrawal, A., Chellappa, R., Raskar, R.: An algebraic approach to surface reconstruction from gradient fields. In: *Proc. Int'l Conf. Computer Vision.* Volume 1. (2005) 174–181
19. Weickert, J.: *Anisotropic Diffusion in Image Processing.* PhD thesis, University of Kaiserslautern, Germany (1996)
20. Horowitz, I., Kiryati, N.: Depth from gradient fields and control points: Bias correction in photometric stereo. *Image and Vision Computing* **22** (2004) 681–694
21. Kimmel, R., Yavneh, I.: An algebraic multigrid approach to image analysis. *SIAM J. Sci. Comput.* **24** (2003) 1218–1231