

Surface-from-Gradients: An Approach Based on Discrete Geometry Processing

Wuyuan Xie, Yunbo Zhang, Charlie C. L. Wang*, and Ronald C.-K. Chung

Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong

*Corresponding Author; E-mail: cwang@mae.cuhk.edu.hk

Abstract

In this paper, we propose an efficient method to reconstruct surface-from-gradients (SfG). Our method is formulated under the framework of discrete geometry processing. Unlike the existing SfG approaches, we transfer the continuous reconstruction problem into a discrete space and efficiently solve the problem via a sequence of least-square optimization steps. Our discrete formulation brings three advantages: 1) the reconstruction preserves sharp-features, 2) sparse/incomplete set of gradients can be well handled, and 3) domains of computation can have irregular boundaries. Our formulation is direct and easy to implement, and the comparisons with state-of-the-arts show the effectiveness of our method.

1. Introduction

The reconstruction of surfaces from estimated gradients, called *surface-from-gradients* (SfG), is an essential step for the methods such as *shape-from-shading* (SfS) [1] and *photometric stereo* (PS) [2]. SfS and PS methods can estimate a dense and noisy gradient field, which is used to calculate the height field by integration. Mathematically, a gradient field generated from a three-dimensional surface should be integrable – that is the integral along any closed path should be equal to zero and integration results should be independent to the selection of path choice. However, in practices, the gradient fields produced by PS (or SfS) are rarely integrable due to the inevitable noises generated during the estimation process. To integrate a noisy gradient field for obtaining a continuous surface, prior methods (e.g., [3, 4, 5, 6, 7]) enforce the integrability constraints over the entire gradient field. Such enforcement perturbs the input gradients, which also are the indicated normal vectors of surfaces to be reconstructed. As a result, it is effective in curbing noises for the objects with smooth surfaces. But when applying this enforcement on objects with sharp features, it smooths out the sharp features and leads to large distortion.

In this paper, we reformulate the surface estimation problem in a discrete setup so that avoid to adding the integrabil-

ity constraints. Specifically, the surface to be reconstructed is computed on a mesh model M consists of many facets (see Figure 1), where each facet is corresponding to a sample in the gradient field. After converting the gradients at samples to the target normal vectors of the facets, a *discrete geometry processing* (DGP) method is conducted to deform the mesh M to let its facets follow the demanded normal vectors. The deformation of M is computed iteratively. In each iteration, a local shaping step is first performed to determine the position and orientation of each facet according to its target normal and its current shape. Then, a global blending step is applied to glue all the facets (have been broken after the local shaping) back into a connected mesh surface. The blending is formulated under a least-square optimization framework, which can be solved efficiently.

Comparing to the prior research, we directly manipulates the discrete geometry of a piecewise linear surface. To the best of our knowledge, this is the first approach formulating the SfG problem in DGP. Advantages of the DGP-based formulation are three-fold:

- **Sharp-feature Preservation:** As the normal vectors are only enforced inside facets in the local shaping step, sharp-features can be formed along the boundary of facets. Meanwhile, the surface smoothness is preserved by the least-square optimization in the global blending step.
- **Incomplete Data:** The formulate of our DGP-base SfG approach can be applied to incomplete data sets, in which the gradients on some pixels are not known. Examples with up to 55% information missed can be successfully reconstructed.
- **Irregular Domain Boundary:** Benefit from transforming the computing media to a mesh surface, the reconstruction of surfaces with boundaries in general shapes can be easily supported by our approach.

Besides, our direct formulation can be easily implemented – a primary code in MATLAB contains only 95 lines. Our method outperforms conventional and state-of-the-art approaches in experiments on a dataset with 20 real or synthetic models.

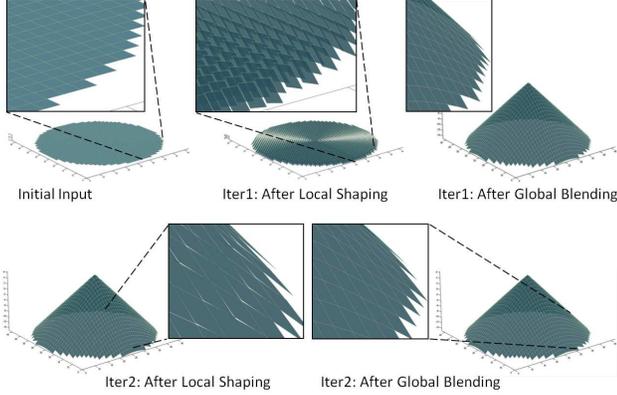


Figure 1. The computation of *surface-from-gradients* (SfG) is formulated as a *discrete geometry processing* (DGP) problem: (left) the mesh surface before processing, (middle) the facets are positioned along the orientation of target normals in the local shaping step, and (right) the facets are glued together into a mesh surface in the global blending step. The final result can be obtained by applying the steps of local shaping and global blending iteratively.

2. Related Work

Surface-from-gradients is a problem studied for many years. At the early stage, it appears as an auxiliary algorithm of SfS and PS in estimating depths from gradients. After being developed for decades, the existing approaches can generally be classified into two main categories: the ones based on enforcing integrability constraints and the methods fitting kernel basis functions. We briefly review these approaches as well as the local/global approaches in discrete geometry processing that inspire our work.

Integrability Enforcement: The first approach of integrability enforcement is proposed by Frankot and Chellappa [3], where they enforce integrability by projecting a dense non-integrable field of gradients onto a set of integrable slopes using the Fourier basis functions. In [8], an integrable surface is defined in terms of an orthonormal set of gradients field, and then the gradient field is partially projected onto the gradient space to obtain a partially integrable field. Kovesi [6] projected the input dense gradients onto a set of wavelet basis functions that satisfy the constraints of integrability. The approach of Goldman et al. [9] aimed at solving the problem of PS with spatially varying BRDFs, where the surface is estimated from the projected gradients that are integrable. The integrability was also enforced by using the zero curl constraints in [10]. After that, they demonstrated a general framework in [7], under which multiple approaches, including Poisson [4], Fourier basis function [3], alpha-surface [7], M-estimator [11], regularization [12], and anisotropic diffusion [13] can be generated by the same objective function. However, a common problem of approaches enforcing integrability is that important sharp-

features may be lost when the integrability constraints are not satisfied on the input gradients.

Kernel Method: The method introduced by Ng et al. in [14, 15] assumes the input set of gradients can be either dense or sparse. Their approach transforms the integration problem into a high-dimensional fitting problem, which can be solved by using certain kernel basis functions. Their approach can deal with the outlier and also claims to have good performance in repairing missing data. One characteristic of their method is that it always outputs continuous surfaces by distorting the input gradients to satisfy the constraints of continuity. As a result, over-flat results could be generated – this is also observed in our experimental tests by using the source code provided by Ng et al. [16]. For a SfG problem with n unknown heights to be determined at the grid nodes, their algorithm needs to solve a system of linear equations with the size of $3n \times 3n$, which is triple the size of linear systems solved in our formulation.

Discrete Geometry Processing: Unlike prior approaches, our work is completely based on a DGP formulation to solve the SfG problem. We convert the input gradients into normal vectors to be satisfied on the facets of a mesh surface M , and then deform M into a shape following these demanded normal vectors. As will be discussed in the following section, a straight-forward formulation for solving this problem is constrained deformation (ref. [17, 18]). However, as the constraints in facet normal are not linear to the positions of mesh vertices, the computation could converge very slowly. Recently, following the strategy of local/global formulation [19, 20], the non-linearly constrained geometry processing problems are solved by iteratively applying the local projection and the global blending step (ref. [21]). This motivates our formulation, in which the orientation of facets is adjusted in a local shaping step and the mesh surface is deformed according to the new orientations in a global least-square optimization. Besides, this approach is further extended to process the cases with sparse normals and/or singular normals.

3. SfG as Discrete Geometry Processing

We consider the surface-from-gradients problem as a mesh deformation problem in discrete geometry processing. The input of SfG is usually a set of 2D points in the image domain, where each point is assigned with an input gradient (p, q) with p and q being the x - and y -components of a gradient. The output is assumed to be a C^0 -continuous surface. To solve the SfG problem, we make a DGP setup as follows.

- For every sample (i, j) , a quadrangular facet $f_{i,j}$ is constructed for it. $f_{i,j}$'s boundary is defined by four vertices $\mathbf{v}_{i,j}$, $\mathbf{v}_{i+1,j}$, $\mathbf{v}_{i+1,j+1}$ and $\mathbf{v}_{i,j+1}$.

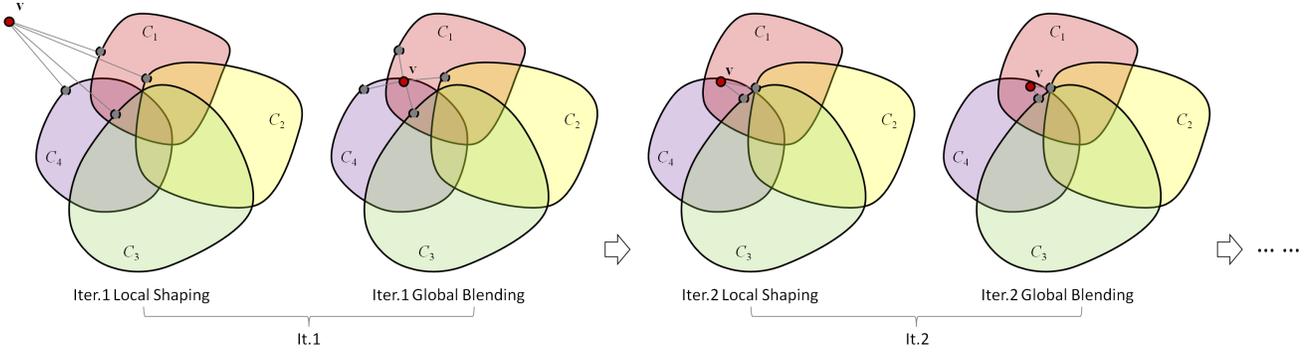


Figure 2. An illustration of the local/global solution for the SfG problem, where each iteration consists of a local shaping step followed by a global blending step. The point, \mathbf{v} , denotes the current position of vertices and C_i s represent the feasible regions determined by the target normals on facets.

- Every vertex $\mathbf{v}_{i,j}$ is initially positioned on the xoy plane with coordinate $((i - \frac{1}{2})h, (j - \frac{1}{2})h, 0)$. h is the user specified pixel width as an input.
- The mesh surface M is now constructed by the set of vertices and quadrangular faces. The geometric shape of a sample (i, j) in the image space is represented by the facet $f_{i,j}$ in the Euclidean space.

Without loss of generality, the input gradient (p, q) specified at a sample can be converted into a normal vector of the corresponding facet as: $\mathbf{n} = (-p, -q, 1)/\sqrt{p^2 + q^2 + 1}$.

3.1. Constrained optimization

Given the target normal vector, $\mathbf{n}_{i,j}$, for each facet $f_{i,j}$, we are going to lift up the mesh surface M by moving every vertex $\mathbf{v}_{i,j}$ away from the xoy plane to a new position $((i - \frac{1}{2})h, (j - \frac{1}{2})h, d_{i,j})$. The x - and y -coordinates of vertices will not be changed and the values of $d_{i,j}$ s are unknown variables to be determined in this setup. A straightforward formulation of this problem is to minimize the shape variation meanwhile fulfilling the requirement on facets' normals as hard constraints. That is,

$$\min_{\{d_{i,j}\}} E(M) \quad s.t. \quad \mathbf{n}(f_{i,j}) = \mathbf{n}_{i,j}$$

where $E(M)$ is a functional measuring the shape variation (and/or smoothness) of M and $\mathbf{n}(\dots)$ returns the normal vector of a facet. As discussed in [22, 23], computation of this optimization converges slowly and may lead to large distortion. More seriously, unlike most deformation setups in geometric modeling, here the initial shape of M does not satisfy the hard constraints – i.e., the initial value is out of feasible region. Directly computing the values of $\{d_{i,j}\}$ by constrained optimization is hard to obtain a surface following the input gradients. Moreover, to enforce $\mathbf{n}(f_{i,j}) = \mathbf{n}_{i,j}$ on facets is too restrictive for letting the normal vectors of resultant surfaces follow the input gradients. Instead, our formulation solves a weak form of the constraints – that is, $\mathbf{n}(f_{i,j})/\|\mathbf{n}_{i,j}\|$.

3.2. Local/global solution

We solve the problem of deforming a mesh surface M according to requested facet normals by a local/global strategy. For a vertex \mathbf{v} , a local shaping step can be taken in its four adjacent facets to adjust their orientations. As a result, four facets with new orientation will give four target positions of \mathbf{v} . As illustrated in Figure 2, if the target orientation of each facet is considered as a feasible region (e.g., C_i in Figure 2), the local shaping step actually project the current position of \mathbf{v} onto its closet points in the feasible regions. Thereafter, a global blending step is applied to determine a compromised position of \mathbf{v} according to the projected four positions, which determines a new shape of mesh surface M . Iteratively applying these two steps can result in a shape of M agree with all the requested normal vectors on facets.

Geometrically, such a local/global approach mimics the phenomenon of first projecting a particle (formed by all the unknown variables $\{d_{i,j}\}$) onto the feasible regions one by one and then blending the projected positions. Repeatedly applying the projection and the blending, the particle will be iteratively dragged to a position in the overlapped area of all feasible regions – that is the solution satisfying all the normal requirements on the facets of M .

Now, we analyze the reason why this local/global approach outperforms other existing methods. The ‘magic’ comes from the local/global decomposition. The local step avoids the problem of finding a good initial guess that is needed by most optimization methods in solving non-convex problems, in which iterations with a ‘bad’ initial guess can be easily stuck at local optimum. In short, the advantage of our approach is to solve a non-convex problem by a local shaping step to estimate good starting points for the following global blending step, which is solving a convex problem.

4. Formulation in DGP

This section details the formulation for solving the SfG problem under the framework of local/global geometry processing. The efficient numerical scheme for determining the positions of vertices in iterations is also discussed. The cases with sparse input gradients and singular normals are also considered.

4.1. Local shaping

In the local shaping step, the vertices of a quadrangular facet $f_{i,j}$ are projected onto the plane with the normal $\mathbf{n}_{i,j}$, where the plane is supposed passing through the *current* center, $\mathbf{c}_{i,j}$, of $f_{i,j}$.

$$\mathbf{c}_{i,j} = (ih, jh, \frac{1}{4}(d_{i,j} + d_{i+1,j} + d_{i+1,j+1} + d_{i,j+1})) \quad (1)$$

The projection of a vertex $\mathbf{v}_{k,l}$ along z -axis onto the plane can be obtained by

$$p_{i,j}(\mathbf{v}_{k,l}) = \mathbf{c}_{i,j}^z - \frac{[\mathbf{n}_{i,j}^x(k-i)h + \mathbf{n}_{i,j}^y(l-j)h]}{\mathbf{n}_{i,j}^z} \quad (2)$$

with $k \in \{i, i+1\}$ and $l \in \{j, j+1\}$. The superscript in $\{x, y, z\}$ indicates the x -, y - and z -components of a vector respectively. The projection gives a new depth component, $d_{k,l}$, for the vertex $\mathbf{v}_{k,l}$ (as illustrated in Figure 3(a)). A vertex $\mathbf{v}_{k,l}$ of M surrounded by four facets will have four projected positions computed by Eq.(2). Simply assigning $\mathbf{v}_{k,l}$ to the average position of these four points does not lead to a good result. A more sophisticated blending method is developed below.

4.2. Global blending

In the global blending step, we wish to deform M to a shape that the positions of the vertices in every facet (e.g., $f_{i,j}$) give the same shape as the projected facet. Specifically, let $\mathbf{z}(f_{i,j})$ denote a column vector of the facet $f_{i,j}$ formed by its vertices's depths as

$$\mathbf{z}(f_{i,j}) = [d_{i,j} \quad d_{i+1,j} \quad d_{i+1,j+1} \quad d_{i,j+1}]^T. \quad (3)$$

The vector of projection, $p_{i,j}(\dots)$, for $f_{i,j}$ is defined as $\mathbf{p}(f_{i,j})$. On an optimal shape of M , the vectors $\mathbf{z}(f_{i,j})$ and $\mathbf{p}(f_{i,j})$ should represent the same shape of a facet. A straightforward formulation is to minimize a functional

$$\Phi(\{d_{k,l}\}) = \sum_{f_{i,j}} \|\mathbf{z}(f_{i,j}) - \mathbf{p}(f_{i,j})\|^2, \quad (4)$$

which enforces $\mathbf{z}(f_{i,j})$ to have the same value as $\mathbf{p}(f_{i,j})$.

By minimizing the above functional can eventually obtain a mesh surface M , the face normals of which are close to the target ones. However, the enforcement of letting

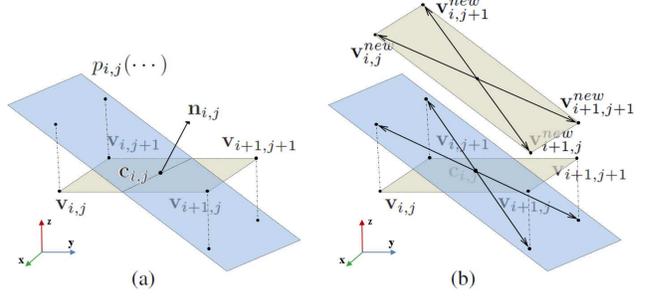


Figure 3. The illustration for (a) the projection of local shaping and (b) the optimal result in global blending based on the relative vectors w.r.t. the centers (i.e., blending with mean-subtraction).

$\mathbf{z}(f_{i,j}) = \mathbf{p}(f_{i,j})$ is too restrictive. In practice, $\mathbf{z}(f_{i,j})$ and $\mathbf{p}(f_{i,j})$ are only expected to represent the same shape instead of coincident. Borrowed the idea of [21], we formulate a new functional by comparing their relative vectors with reference to their own center. This can be realized by subtracting the mean of vertices inside $\mathbf{z}(f_{i,j})$ and $\mathbf{p}(f_{i,j})$. For this purpose, a matrix \mathbf{N} can be defined as

$$\mathbf{N} = \mathbf{I}_{4 \times 4} - \frac{1}{4}\mathbf{1}, \quad (5)$$

where $\mathbf{1}$ is a 4×4 matrix with all elements equal to 1. Note that, $\mathbf{Nz}(f_{i,j})$ returns the relative vectors of $\mathbf{z}(f_{i,j})$ with reference to its own center, and the same rule applies to $\mathbf{Np}(f_{i,j})$. Then, the functional with mean-subtraction is defined as

$$\Phi(\{d_{k,l}\}) = \sum_{f_{i,j}} \|\mathbf{Nz}(f_{i,j}) - \mathbf{Np}(f_{i,j})\|^2. \quad (6)$$

As shown in Figure 3(b), this functional returns minimum when $\mathbf{z}(f_{i,j})$ and $\mathbf{p}(f_{i,j})$ having the same shape and the same orientation. Figure 4 shows a comparison on the speed of convergence, where the same input is tested on the functionals with vs. without mean-subtraction.

4.3. Numerical scheme

Without loss of generality, the mesh surface M is assumed to have n vertices and m quadrangular facets. As M is a two-manifold quadrangular mesh surface, we have $n \simeq m$. The minimum of functional in Eq.(6) can be obtained by solving a linear system with n equations such as $\partial\Phi/\partial d_{k,l} = 0$. When computing in this way, the linear system will need to be constructed and solved repeatedly for many times. A more efficient numerical scheme is developed below.

First of all, the functional in Eq.(6) can be reformulated into the form as

$$\Phi(\{d_{k,l}\}) = \|\mathbf{Ax} - \mathbf{b}\|^2, \quad (7)$$

where \mathbf{A} is a $4m \times n$ matrix derived from $\mathbf{Nz}(f_{i,j})$, \mathbf{b} is a vector with $4m$ components derived from $\mathbf{Np}(f_{i,j})$. The

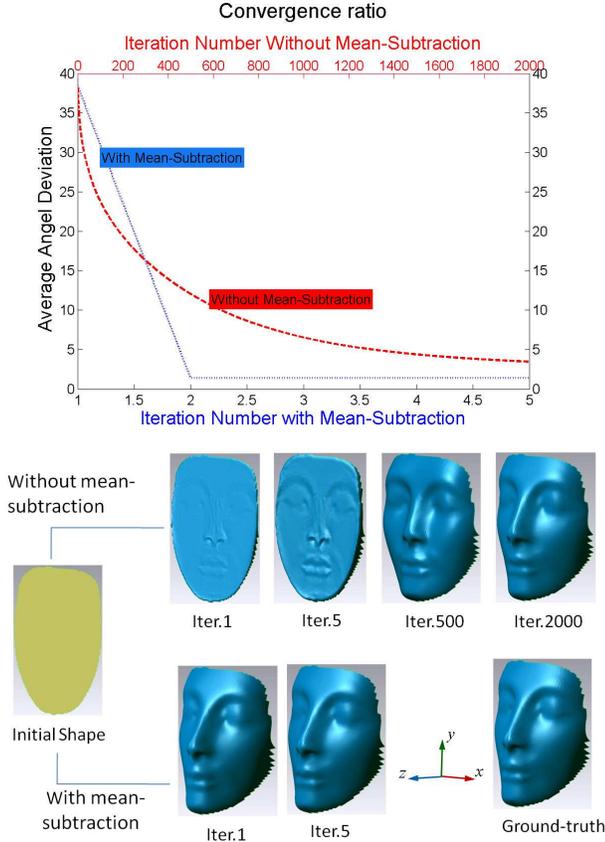


Figure 4. Statistics show that the computation using mean-subtraction converges in 2 iterations on this example while the computation without mean-subtraction takes a few thousand steps.

vector \mathbf{x} contains all the unknown depth values on the vertices of M . This is a standard least-square problem, which can be solved by

$$\mathbf{A}^T \mathbf{A} \mathbf{x} = \mathbf{A}^T \mathbf{b}. \quad (8)$$

Here, \mathbf{A} only depends on \mathbf{N} and the connectivity of vertices on M , which are invariant during the iteration. Therefore, we conduct Cholesky factorization to pre-factorize $\mathbf{A}^T \mathbf{A}$, and only need to substitute the pre-factorized results to determine the values of \mathbf{x} during iterations.

4.4. Extensions

A computing domain Ω with irregular boundary can be easily solved in our framework. We construct facets only for the pixels falling inside Ω . Unlike other approaches based on numerical PDE solver (e.g., [4]), our numerical scheme does not rely on a rectangular domain.

A more difficult problem is the inputs with sparse gradients, which happens when information is lost on some samples. That is, the gradients (i.e., the target normal vectors) are not given on some facets. For these facets, we apply the

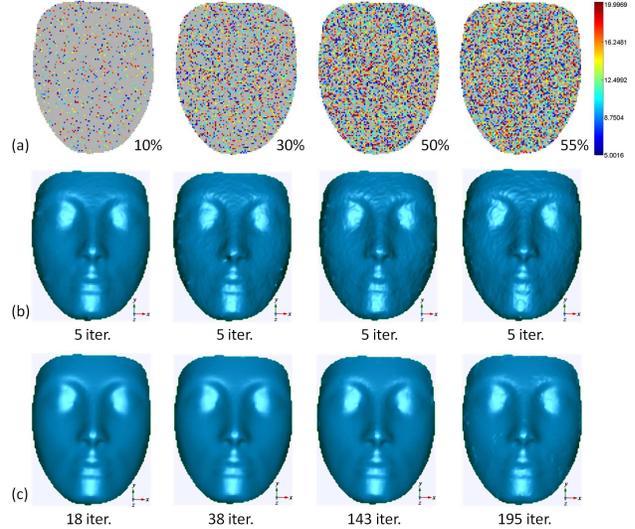


Figure 5. Tests of our approach on inputs with different amount of noises added: (a) Color maps illustrate the distribution of noises and colors denote the amount of perturbation in terms of angle degree. (b) When treating all noisy samples as inliers, the computation of our approach converges in 5 steps. (c) When considering the noisy samples as outliers and removing them from the computation – that leads to an input with sparse samples, our approach can successfully converge to a reconstruction with more higher quality in 18, 38, 143 and 195 steps of iteration respectively.

current normal vectors as $\mathbf{n}_{i,j}$ used in Eq.(2) for the local shaping step. This manipulation tends to keep the shape of a facet as what determined by the previous global blending step. In other words, the shapes of these facets are determined by their neighbors with known target normals. Our algorithm can reconstruct the surface from an incomplete set with up to 55% data lost (see Figure 5).

The local shaping step by projecting points onto a plane with normal $\mathbf{n}_{i,j}$ becomes unstable when the plane is nearly perpendicular to the xy -plane – that can be detected by checking if $\mathbf{n}_{i,j} \cdot (0, 0, 1) \leq \epsilon$. We use $\epsilon = 0.0871557$ in our implementation, which indicates the angle less than 5° . Such normals are defined as outliers and again the corresponding facets’ target normals are set as their current ones. In other words, $\mathbf{z}(f_{i,j})$ determined in the previous step is used as the vector $\mathbf{p}(f_{i,j})$ in the current step.

5. Experimental Results

We have implemented our algorithm in both MATLAB and C++ code and tested its performance on a variety of models. All the results are generated on an Intel i7 CPU with 2.67GHz and 4GB RAM. We also compare our approach with the existing methods in literature. Results show that our approach is efficient and outperforms others in the accuracy of reconstruction. Details are discussed below.

Generally, the approach proposed in this paper is effi-

Table 1. Statistics on Computing Time (ms)

Input Res.	Time of Factorization	Avg. Time for Each Step [†]	
		Local Shaping	Global Blending
128 ²	187	0.620	62.1
256 ²	905	3.42	252
512 ²	4,961	17.5	1,018

[†]The average time is computed from an iteration with 50 steps.

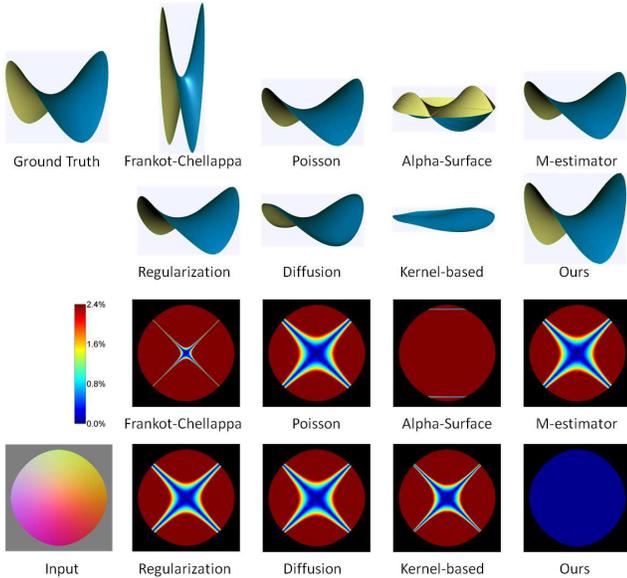


Figure 6. Saddle surfaces reconstructed by eight different methods – the input gradients are free of noise, and the errors have been normalized by the range of depths on all samples.

cient. Statistics of applying our approach on inputs with rectangular domain in different resolutions are listed in Table 1. The statistics are generated on a C++ code using TAUCS library as linear solver, and the average time needed for each step (obtained from 50 steps of iteration) is reported. The most time-consuming step is factorization, which however only needs to be computed once throughout the whole procedure of surface reconstruction. Moreover, for two normal fields having the same resolution and boundaries, the factorization of $\mathbf{A}^T \mathbf{A}$ should be the same (i.e., can be reused). Our DGP-based SfG method usually converges in a few steps on inputs with noises in a relative low degree. Note that, in practice the time needed for each step should be even smaller than what reported in Table 1 as the computing domain with irregular boundary will have much less unknown variables to be determined in each step of iteration.

Our approach is robust to noisy data and sparse data. As shown in Figure 5, we randomly select different amount (15%, 30%, 50% and 55%) of scattered samples in the normal fields of a face model to perturb random angles up to 20°, where the colored dots in Figure 5(a) are these per-

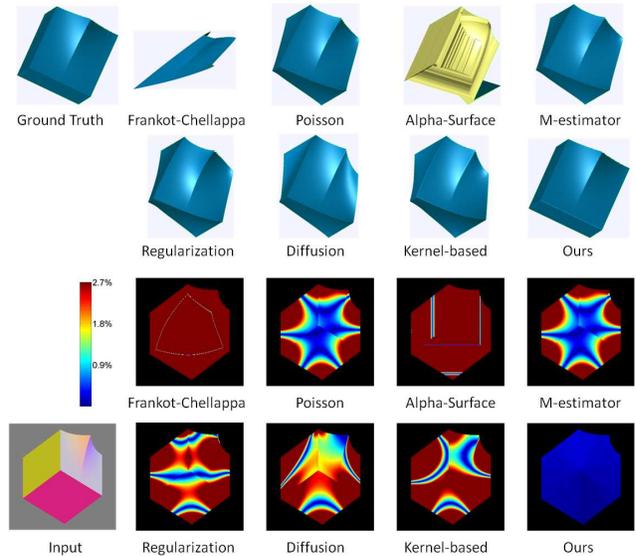


Figure 7. Surfaces with sharp features reconstructed by eight different methods – the input gradients are free of noise, and the errors have been normalized by the range of depths on all samples.

turbed samples and colors denote the amount of perturbation in terms of angle degree. We first treat all samples as inliers; the computations converge in 5 steps on all the noisy inputs. The reconstruction results can be found in Figure 5(b). In the second test, the samples with perturbed normals are considered as outliers and removed from the computation – i.e., the input becomes sparse. Our approach still can reconstruct the surface of face model successfully (see Figure 5(c)). The terminal condition of iteration is set by comparing average angle deviations in the current step and the previous step. When the difference is less than 10^{-3} , we stop the iteration. Generally, more iteration steps are needed for the models with more outliers removed. And the results obtained by removing outliers have better quality than treating all samples as inliers.

Lastly, we compare this approach with seven other methods including Frankot-Chellappa [3], Poisson [4], alpha-surface [7], M-estimator [11], regularization [12], anisotropic diffusion [13] and kernel-based fitting [15, 16]. These eight approaches are evaluated on a set of 20 models downloaded from the shape repository of Aim@Shape. For every model, we first sample them by rendering them via OpenGL and reading back its depth map and the sampled normal map. The depth map serves as the ground-truth for surface reconstruction. All the sampled normal vectors are perturbed with additional Gaussian noises having standard deviation of 5°. Then, tests are conducted on two types of inputs – the noise-free normals and the perturbed normal with Gaussian noises. After the reconstruction, a reconstructed model is moved to let its center coincident with the ground truth’s center to measure the depth-error on the cen-

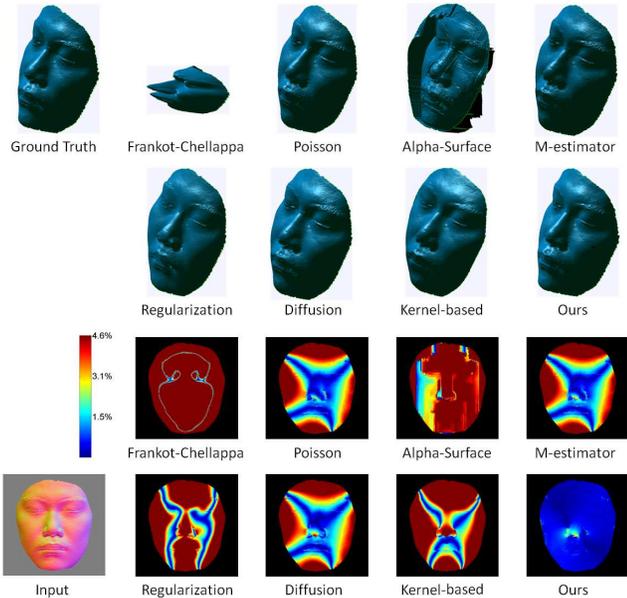


Figure 8. Face models reconstructed by eight different methods – the input gradients are free of noise, and the errors have been normalized by the range of depths on all samples.

ter of every facet. The values of errors are normalized by the range of depths on samples. The distribution of normalized errors are displayed as color maps in Figures 6, 7 and 8, where the inputs are noise-free. Note that, some examples tested here have sharp-features (e.g., Figure 7), which are well recovered by our approach. Moreover, we also compute the average of relative errors on results generated by different approaches and compare them in Figure 9. It is found that our approach outperform others. Note that, results generated by the Frankot-Chellappa method [3] have not added into this comparison as the errors are too large.

A photometric stereo system has also been built to generate gradient-fields from real objects such as 3D fingerprints. Again, reconstructions by different methods have been conducted. As there is no ground-truth, we measure the variations of normals on the reconstructed surface versus the input gradients, where an example is given in Figure 10. Both the average error, E_{avg} , and the maximal error, E_{max} , are measured in terms of angle degrees.

6. Conclusion

We present a new algorithm for solving the SfG problem in this paper. First, our algorithm is based on discrete geometric constraints and completely abandons integrability constraints. We compute a piecewise linear surface that well preserves fine details. Second, we prove that the formulation with mean-subtraction can greatly improve the speed of convergence. Third, we show that our DGP-based SfG algorithm can reconstruct surface from sparse normals where

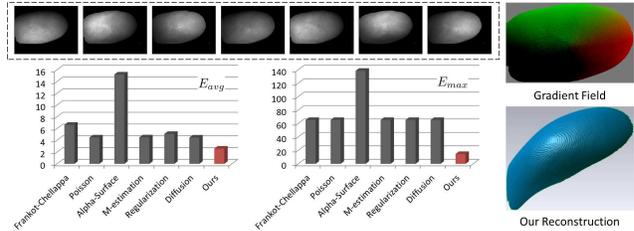


Figure 10. Reconstructed surface of a 3D fingerprint from a gradient field generated by the photometric stereo. Errors are measured in terms of angle variation in degrees. Note that, the code of kernel-based method [16] fails on this example.

the missing data is up to 55%. Besides, when the inputs are perturbed with noisy, our method can reconstruct more accurate results than that generated by other methods.

References

- [1] B.K.P. Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. *Massachusetts Institute of Technology, Cambridge, MA*, 1970. 1
- [2] R.J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):139–144, 1980. 1
- [3] R.T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):439–451, 1988. 1, 2, 6, 7
- [4] T. Simchony, R. Chellappa, and M. Shao. Direct analytical methods for solving poisson equations in computer vision problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(5):435–446, 1990. 1, 2, 5, 6, 8
- [5] 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 *Computer Vision and Pattern Recognition (CVPR), 2001*, volume 1, pages 1–743. IEEE, 2001. 1
- [6] P. Kovesi. Shapelets correlated with surface normals produce surfaces. In *IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 994–1001, 2005. 1, 2
- [7] A. Agrawal, R. Raskar, and R. Chellappa. What is the range of surface reconstructions from a gradient field? In *European Conference Computer Vision (ECCV), 2006*, pages 578–591. Springer, 2006. 1, 2, 6, 8
- [8] B. Karacali and W.E. Snyder. Partial integrability in surface reconstruction from a given gradient field. In *International Conference on Image Processing, 2002*, volume 2, pages II–525. IEEE, 2002. 2
- [9] D.B. Goldman, B. Curless, A. Hertzmann, and S.M. Seitz. Shape and spatially-varying BRDFs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1060–1071, 2010. 2

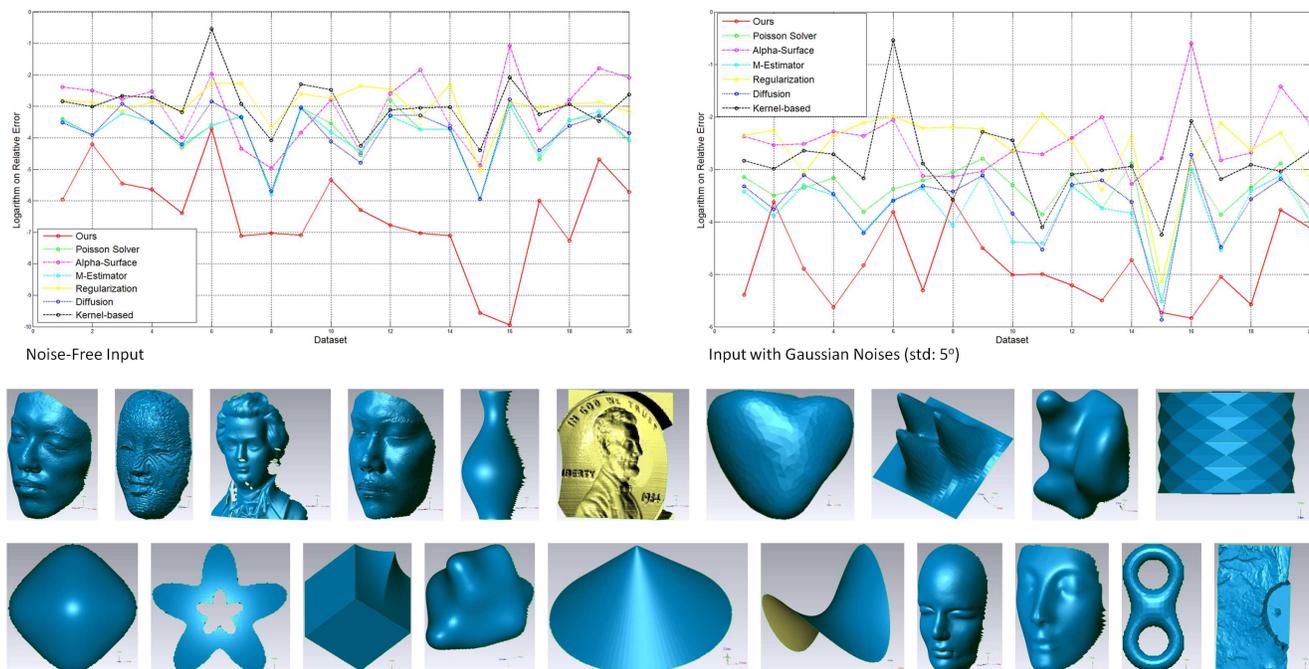


Figure 9. In total, we test 20 models and compare the results generated from eight different algorithms, including ours, Poisson [4], alpha-surface [7], M-estimator [11], regularization [12], anisotropic diffusion [13] and kernel-based fitting [15, 16]. The averages of relative errors are measured on all the results to obtain a quantitative comparison. Both the noise-free input (top-left) and the input with Gaussian noises (top-right) are tested. Our approach outperforms other methods in both cases.

- [10] A. Agrawal, R. Chellappa, and R. Raskar. An algebraic approach to surface reconstruction from gradient fields. In *IEEE International Conference on Computer Vision (ICCV)*, volume 1, pages 174–181, 2005. 2
- [11] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud. Deterministic edge-preserving regularization in computed imaging. *IEEE Transactions on Image Processing*, 6(2):298–311, 1997. 2, 6, 8
- [12] G. Aubert and P. Kornprobst. *Mathematical Problems in Image Processing: Partial Differential Equations and the Calculus of Variations (Applied Mathematical Sciences)*. Springer-Verlag New York, Inc., 2006. 2, 6, 8
- [13] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990. 2, 6, 8
- [14] T.P. Wu, J. Sun, C.K. Tang, and H.Y. Shum. Interactive normal reconstruction from a single image. *ACM Transactions on Graphics (TOG)*, 27(5):119, 2008. 2
- [15] H.S. Ng, T.P. Wu, and C.K. Tang. Surface-from-gradients without discrete integrability enforcement: A gaussian kernel approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(11):2085–2099, 2010. 2, 6, 8
- [16] T.P. Wu. Source code of ‘surface-from-gradients without discrete integrability enforcement: A gaussian kernel approach’. <http://www.cse.ust.hk/pang/>. 2, 6, 7, 8
- [17] Mario B. and Leif K. An intuitive framework for real-time freeform modeling. *ACM Transactions on Graphics*, 23(3):630–634, 2004. 2
- [18] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher. Geometric surface processing via normal maps. *ACM Transactions on Graphics*, 22(4):1012–1033, 2003. 2
- [19] R.W. Sumner and J. Popović. Deformation transfer for triangle meshes. *ACM Transactions on Graphics*, 23(3):399–405, 2004. 2
- [20] R.W. Sumner, M. Zwicker, C. Gotsman, and J. Popović. Mesh-based inverse kinematics. *ACM Transactions on Graphics*, 24(3):488–495, 2005. 2
- [21] S. Bouaziz, M. Deuss, Y. Schwartzburg, T. Weise, and M. Pauly. Shape-up: Shaping discrete geometry with projections. *Computer Graphics Forum*, 31(5):1657–1667, 2012. 2, 4
- [22] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE Transactions on Visualization and Computer Graphics*, 14(1):213–230, 2008. 3
- [23] Y. Liu, H. Pottmann, J. Wallner, Y.L. Yang, and W. Wang. Geometric modeling with conical meshes and developable surfaces. *ACM Transactions on Graphics*, 25(3):681–689, 2006. 3