Shape from Shading: Recognizing the Mountains through a Global View

Qihui Zhu Jianbo Shi
Dept. of Computer and Information Science
University of Pennsylvania, Philadelphia, PA 19104
qihuizhu@seas.upenn.edu jshi@cis.upenn.edu

Abstract

Resolving local ambiguities is an important issue for shape from shading (SFS). Pixel ambiguities of SFS can be eliminated by propagation approaches. However, patch ambiguities still exist. Therefore, we formulate the global disambiguation problem to resolve these ambiguities. Intuitively, it can be interpreted as flipping patches and adjusting heights such that the result surface has no kinks. The problem is intractable because exponentially many possible configurations need to be checked. Alternatively, we solve the integrability testing problem closely related to the original one. It can be viewed as finding a surface which satisfies the global integrability constraint. To encode the constraints, we introduce a graph formulation called configuration graph. Searching the solution on this graph can be reduced to a Max-cut problem and its solution is computable using semidefinite programming (SDP) relaxation. Tests carried out on synthetic and real images show that the global disambiguation works well for complex shapes.

1. Introduction

Resolving local ambiguities is important for accurate 3D shape reconstruction. In shape from shading (SFS) problem, each pixel has a family (cone) of surface normals satisfying the image-irradiance equation [4]. Surface normals are uniquely determined only at points where the surface is frontal to the illumination direction. These points are called *singular points*.

Shapes around singular points can be computed exactly (without using smoothness constraint) using propagation methods [1, 7, 11, 13]. However, local shapes are determined up to three types of ambiguity convex/concave/saddle [2, 10].

Once the label of convex/concave/saddle is determined at a singular point, the shape around it can be computed *without any ambiguity*. In this paper we point out that the image-irradiance equation alone is not enough for SFS: assignment of convex/concave/saddle has a fundamental effect on the overall reconstructed shape. In Figure 1, we show several incorrect labels of singular points can lead to wrong shapes whose re-rendered images look identical to the shading images. The only noticeable differences are tiny white lines due to kinks where local estimations have discrepancies on surface normals. Therefore, resolving local ambiguities boils down to checking kink-free condition:

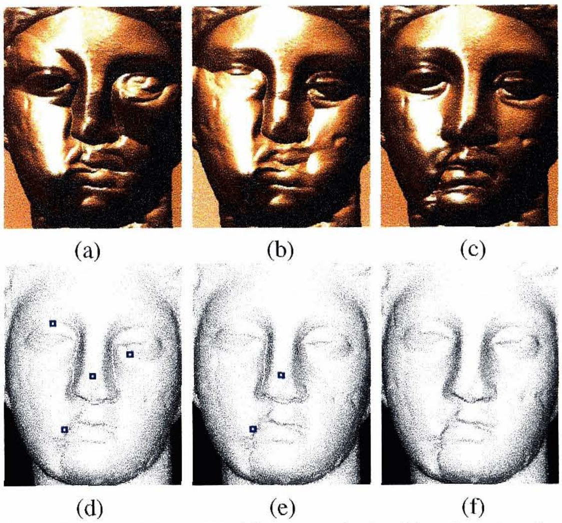


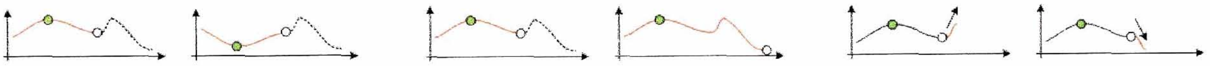
Figure 1. Venus' face. (a) 3D shape obtained by changing four labels of patches with singular points marked by blue squares in (d). (d) shows the re-rendered image, which is almost the same as the input image of (f). (b) and (e) shows the case where we have fewer (two) incorrect patches. (c) is the 3D shape reconstructed by our algorithm.

(P1) Global disambiguation problem. *Assign the convex/concave/saddle labels and heights to singular points such that local patches reconstructed around them form a smooth surface without kinks.*

In order to stitch local patches to form the whole shape, one will have to answer the following two questions:

- **Convexity.** Which singular points are convex? We will construct the shape by travelling monotonously down-

Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)
0-7695-2597-0/06 $20.00 © 2006 IEEE



(a) **Convexity:** convex vs concave (b) **Range:** short vs long (c) **Direction:** upward vs downward

Figure 2. Local choices in problem (P1) and (P2). Singular points are marked in green (shaded).

ward from the peaks.

- **Range.** How far should each local propagation travel? The propagation terminates when it meets the frontiers of propagation from other peaks at the same height. Therefore, ranges of propagations depend on the heights of peaks. The question is how we determine the ranges or the heights such that there are no kinks.

The brute force solution for (P1) is to check all the possibilities by flipping the label and adjusting the height on each singular point. Because combinations of all the labels are *exponentially many*, searching for the correct configuration is hard.

The kink-free constraint is very important in addition to image-irradiance equation. Normally we do not expect kinks on the object surface. We use the term kink-free constraint to avoid confusion with the traditional smoothness constraint [3, 8, 19, 20]. Unlike smoothness constraint, the kink-free constraint in (P1) does not flatten the surface, it only requires the *turning points* (from going up to going down) to be smooth. On the contrary, smoothness constraint regularizes the shape by penalizing the second derivative of surface heights. Hence it forces the surface to be flat. Extra smoothness constraint introduces unnecessary distortions of the shape and often produces over-smooth surfaces [3, 8].

This paper is organized as follows. In Section 2, we outline a computational solution of (P1) using a simplified reformulation. In Section 3, we explain how local shape can be computed. We present the details of our computational solution in Section 4. We demonstrate our results on synthetic data as well as real data in Section 5. Finally we provide discussions in Section 6, followed by conclusion.

2. Problem reformulation

Since searching directly for the solution of (P1) is a formidable task, we simplify the problem as follows. Local propagation from one singular point produce height differences to all the other points. In particular, local propagations give height difference estimations between neighboring singular points as long as heights decrease monotonously from one to the other. Suppose these neighbor pairs of singular points are known, we can check whether the height differences are correct. This becomes the following problem:

(P2) Integrability testing problem. *Given A) the local propagation results around singular points and B) the neighbor pairs of singular points, assign heights on singular points such that the height differences of all the neighbor pairs are consistent with those from the local propagations.*

One alternative way to see (P2) is that we have to choose +/- signs of height differences on the neighbor pairs. The absolute values of height differences are given by local propagations, but we do not know the **directions** (see Figure 2(c)). These choices must be consistent in order to assure height assignments to exist. To be specific, the choices satisfy the *global integrability constraint*: If we traverse some neighbor pairs in a loop, height differences must sum up to zero, meaning that we return to the same height. This is due to the fact that surfaces are assumed to be continuous without sudden jumps in the heights.

Both two questions raised in (P1) are answered by solving (P2). The direction choices determine the convexity. The local patch is convex if the singular point is the higher one in all the neighbor pairs containing it. The height assignments determine the valid ranges of local propagations.

We only consider the case where there is a unique surface satisfying (P1). If there are no noises, solution of (P2) satisfies (P1). Furthermore, if the solution of (P2) is unique, it is exactly the desired surface. When noises present, the integrability testing problem amounts to finding the height assignments least violating the constraints induced by neighbor pairs.

Problem (P2) has a computable solution. We will construct a *configuration graph*whose nodes are singular points and edges are neighbor pairs of singular points. Using the configuration graph formulation we can encode global integrability constraint by computing height on each node and determining direction on each edge. We demonstrate this amounts to a Max-cut problem and the solution can be computed using an SDP relaxation.

3. Local propagation

The local propagation estimates the local shape around a reference point up to convex/concave ambiguity. Singular points are good reference points because their surface normals are fixed in their orientations.

Definition. *Point p is a singular point iff* $\mathbf{n}(p) = \pm\ell$.

Local propagation methods computes the local shape according to image intensities in the following way:

- **Input.** Image intensities $I(\Omega)$ of a region Ω. The position (x, y) of a singular point $p \in \Omega$.

- **Output.** Height differences $D(p, q) = z(p) - z(q)$ between the singular point p and any other point q.

To understand how shape can be estimated locally by propagation, let us first consider the formation of the shading image. Suppose the light source direction is $\ell = (l_1, l_2, l_3)^T$ with $l_1^2 + l_2^2 + l_3^2 = 1$ and the surface normal of a point at p is $\mathbf{n}(p) = (n_1, n_2, n_3)^T$. Given the albedo ρ at p and assuming the surface is Lambertian, the intensity $I(p)$ at p satisfies the image-irradiance equation $I(p) = \rho\,\ell \cdot \mathbf{n}(p)$.

SFS tries to recover the differentiable height field $z(p) : \mathbb{R}^2 \mapsto \mathbb{R}$. Let $z_x = \frac{\partial z}{\partial x}$, $z_y = \frac{\partial z}{\partial y}$. Then z is related to the intensity I by

$$I(p) = \frac{\rho(l_1 z_x + l_2 z_y + l_3)}{\sqrt{z_x^2 + z_y^2 + 1}} \tag{1}$$

Without loss of generality, we can assume that the albedo $\rho \equiv 1$. For the simple case where $\ell = (0, 0, 1)^T$, (1) degenerates to

$$||\nabla z|| = \sqrt{z_x^2 + z_y^2} = \sqrt{\frac{1}{I^2} - 1} \tag{2}$$

Now SFS can be formulated as solving a PDE (2) known as the Eikonal equation [2, 7]. However, there is no simple PDE solution to it. Because the boundary condition is unknown, it does not have a unique solution in general. Therefore, general PDE solutions do not guarantee the desired result.

To estimate local shape, we use a fast and accurate SFS propagation method called fast marching [6, 7, 15]. It reconstructs the surface by a Dijkstra style propagation on the grid. Fast marching works in the monotonous neighborhood around a singular point, which is locally the highest or the lowest. A monotonous neighborhood means when moving further away from the extremal point, heights z decrease or increase monotonously. Rouy and Tourin [14] proved that the result surface of fast marching is a viscosity solution to the Eikonal equation (2) in this neighborhood.

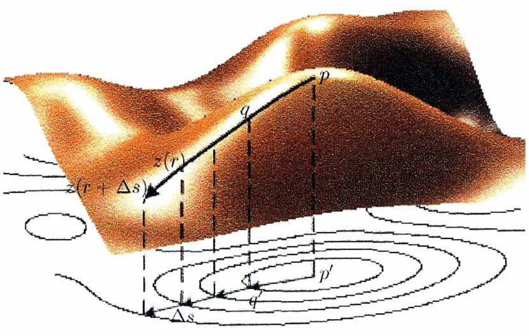
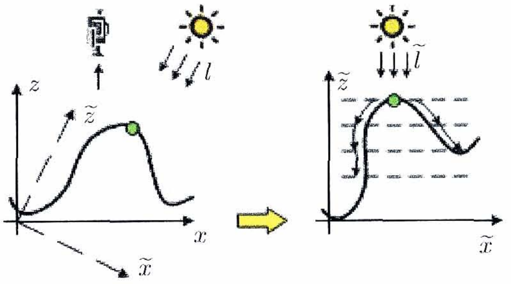


Figure 3. Local shape estimation from singular point p via fast marching: frontier of shortest path produces equal height contours.



(a) (b)

Figure 4. Local propagation with oblique light source. (a) Transformation of the coordinates from camera to illumination. (b) Shape propagation in the new coordinates (illumination) starting from singular points marked in green (shaded).

Conceptually, fast marching computes the length of the shortest path $D(p, q)$ from the local highest point p to every other point q, as an estimation of the height difference $z(p) - z(q)$. The length is computed as the integration of the weights on the points p's, defined as $\sqrt{(1/I(p))^2 - 1}$.

$$D(p, q) = \inf_{l \in L}\{\, d(p, q) = \int_l \sqrt{(1/I(s))^2 - 1}ds\} \tag{3}$$

where L contains all the paths from p to q. Notice ∇z is the fastest descent direction in the height field. Due to the monotonicity, there exist a fastest descent path l^* from p to q. $d(p, q) = z(p) - z(q)$ along this path. For any small segment r to $r + \Delta s$, $z(r) - z(r + \Delta s) \le \Delta s||\nabla z(r)||$. Integration of $||\nabla z(r)||$ along any path is no less than the height difference. Therefore another weighted path can not have a shorter length than the fastest descent path. The length of the weighted shortest path is exactly the height difference.

The nature of fast marching is local propagation. It can be viewed as frontier propagation similar to the *characteristic strip expansion* in [5]. If we trace the propagation frontiers when computing the shortest path, they are in fact the equal height contours of the height field. The shortest path computation extends these contours starting from the singular point. Under the view of frontier propagation, fast marching can be generalized to the case of oblique light source, by performing frontier propagation in the light source coordinates [7, 9] (see Figure 4).

4. Proposed approach

Our approach works as follows. Suppose we can identify the *singular points* from the shading image. These are the brightest points in the image. The local height estimations $D(p, q)$ are computed by fast marching in (3) w.r.t. singular points. Furthermore, if we can determine the subset $\mathcal{P}$ of

0-7695-2597-0/06 $20.00 © 2006 IEEE

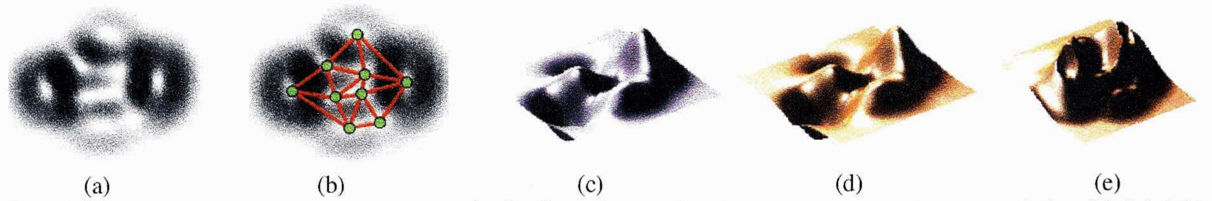


Figure 5. Result of Matlab PEAKS. (a) Input image. (b) *Configuration graph* on singular points: nodes have uncertainties of their heights $\mathbf{h}$ and edges have uncertainties of up and down directions encoded by $\mathbf{d} = \pm 1$. Incorrect configuration $\mathbf{d}$ violating integrability constraint leads to incorrect shape shown in (e). Our result shown in (d) correctly determines the peaks, and the overall shape matches well with the ground truth in (c).

singular points which are *peak* points and their heights, then the heights of all points can be recovered as:

$$z(q) = \max_{p \in \mathcal{P}} \{z(p) - D(p, q)\} \tag{4}$$

We employ a graph to represent the directional choices in (P2). Checking the global integrability constraint amounts to a graph combinatorial search.

4.1. Configuration graph formulation and notations

A *configuration graph* $\mathbf{G} = (\mathbf{V}, \mathbf{E}, \mathbf{W})$ is a graph representing the global configuration of the shape. Figure 5(b) shows one example of the graph. The vertices $\mathbf{V}$ include all the singular points, *i.e.* peaks, valleys and saddle points in the light source direction. These points can be viewed as the representatives of the local patches. $\mathbf{E}$ consists all the neighbor pairs in (P2). Weights $\mathbf{W}(\mathbf{E})$ on these edges are simply the absolute values of height differences between vertices, computed by fast marching. Let us set $n = |\mathbf{V}|$, $m = |\mathbf{E}|$.

Let **height field** $\mathbf{h} = (h_1, h_2, ..., h_n)^T$ be a vector of the heights at all the vertices. Assigning different heights on the vertices can be thought as moving the patches at the vertices vertically and stitching them together.

Edge configurations can be captured formally by $\mathbf{d} = (d_1, d_2, ..., d_m)^T$ with $\mathbf{d_i} = \pm 1 (i = 1, 2, ..., m)$, encoding the directions of height differences [1].

$$d_i = +1 \quad \text{iff} \quad e_i = (v_k, v_l) \text{ and } h_k \geq h_l \tag{5}$$

Values of ± 1 on d_i's give the choices of upward or downward direction in (P2). The optimal choice of the height field $\mathbf{h}$ and edge configuration $\mathbf{d}$ answers the two questions of (P1):

- **Convexity.** Classify the nodes to be peaks or valleys, by checking the signs of d_i's on the incident edges.
- **Range.** Start travelling from the peaks and keep going downward monotonously.

Therefore finding the global consistent shape amounts to finding the height field $\mathbf{h}$ and edge configuration $\mathbf{d}$.

4.2. Constraints on the graph

The height field $\mathbf{h}$ and the edge configuration $\mathbf{d}$ are related to each other through the following quantities.

Define vertex-edge incidence matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ as

$$A_{ij} = \begin{cases} +1 & e_i = (v_k, v_l) \text{ for } j = k \\ -1 & e_i = (v_k, v_l) \text{ for } j = l \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Define the graph weights on the edges $\mathbf{W} \in \mathbb{R}^{m \times m}$ as

$$\mathbf{W} = \text{diag}(w_1, w_2, ..., w_m) \tag{7}$$

where w_i's are the absolute values of height differences $D(p, q)$ between nodes p and q on edge e_i. We place w_i's on the *diagonal* because this is convenient for later discussion.

With the above notions and formulations, we will explore how the global integrability constraint acts on the *configuration graph* $\mathbf{d}$ and $\mathbf{h}$, representing the local patch configurations, are related through $\mathbf{A}$, $\mathbf{W}$. If we made a wrong assignment on one of the edge configuration d_i, contradicting to the height field $\mathbf{h}$, then any loop traversing singular points passing through d_i will not return to the same height. A wrong assignment of $\mathbf{d}$ will violate the integrability constraint. This intuition is captured by the following claim:

Claim 1. *The assignment of height field* $\mathbf{h}$ *and edge configuration* $\mathbf{d}$ *satisfies the integrability constraint, if*

$$\mathbf{Ah} = \mathbf{Wd} \tag{8}$$

Proof. The height differences $\mathbf{f} = (f_1, f_2, ..., f_m)^T$ on all the edges can be computed as

$$\mathbf{f} = \mathbf{Ah} \tag{9}$$

It is not difficult to see, in ideal case, elements of $\mathbf{f}$ are actually the diagonal entries of $\mathbf{W}$, except for the possibilities of +/- signs. So $\mathbf{f}$ can be represented as:

$$\mathbf{f} = \mathbf{Wd} \tag{10}$$

with $\mathbf{d}$ encoding the +/- ambiguity. If a configuration is correct, there must be a height field $\mathbf{h}$ such that (9) and (10) hold simultaneously. This justifies (8). □

[1] We define the vertex order on the edges as $e_i = (v_k, v_l)$ with $k < l$.

Notice that when the heights are shifted by a constant a: $\mathbf{h}^{(2)} = \mathbf{h}^{(1)} + a$, we still have

$$\mathbf{A}\mathbf{h}^{(2)} = \mathbf{A}\mathbf{h}^{(1)} = \mathbf{W}\mathbf{d} \tag{11}$$

So for a fixed $\mathbf{d}$, we have a family of corresponding $\mathbf{h}$.

In order to obtain a unique solution of $\mathbf{h}$, an additional row is added to $\mathbf{A}$. $\mathbf{A}' = \begin{pmatrix}\mathbf{A}\\ \mathbf{b}\end{pmatrix}$ with $\mathbf{b} = (1, 1, ..., 1)$. Accordingly we have $\mathbf{h}' = \begin{pmatrix}\mathbf{h}\\ 0\end{pmatrix}$, $\mathbf{W}' = \begin{pmatrix}\mathbf{W} & 0\\ 0 & 0\end{pmatrix}$ and $\mathbf{d}' = \begin{pmatrix}\mathbf{d}\\ 0\end{pmatrix}$. Then (8) remains in the same form $\mathbf{A}'\mathbf{h}' = \mathbf{W}'\mathbf{d}'$. We replace $\mathbf{A}', \mathbf{h}', \mathbf{W}', \mathbf{d}'$ by $\mathbf{A}, \mathbf{h}, \mathbf{W}, \mathbf{d}$ in later paragraphs.

Our goal is to search over all possible $\mathbf{h}$ and $\mathbf{d}$ such that they satisfy (8). As we will show next, this reduces to a Max-cut problem.

4.3. Max-cut Problem

In the case of noisy image measurements, (8) does not hold strictly. The appropriate $\mathbf{d}$ and $\mathbf{h}$ satisfying the integrability constraint (8) can be computed as

$$(\mathbf{d_{opt}}, \mathbf{h_{opt}}) = \arg\min_{\mathbf{d},\,\mathbf{h}} ||\mathbf{A}\mathbf{h} - \mathbf{W}\mathbf{d}||_2 \tag{12}$$

Claim 2. $\mathbf{d_{opt}}$, $\mathbf{h_{opt}}$ *optimizing (12) can be computed by a Max-cut problem as*

$$\mathbf{d_{opt}} = \arg\max_{\mathbf{d}} \textstyle\sum_{d_i d_j = -1} E_{ij} \tag{13}$$

$$\mathbf{h_{opt}} = \mathbf{A}^{\dagger}\mathbf{W}\mathbf{d_{opt}} \tag{14}$$

with $\mathbf{E} = \mathbf{W}^{\mathbf{T}}(\mathbf{A}\mathbf{A}^{\dagger} - \mathbf{I})^{\mathbf{T}}(\mathbf{A}\mathbf{A}^{\dagger} - \mathbf{I})\mathbf{W}$.

Proof. First it is easy to see $\mathbf{h_{opt}}$ is related to $\mathbf{d_{opt}}$ through $\mathbf{h_{opt}} = \mathbf{A}^{\dagger}\mathbf{W}\mathbf{d_{opt}}$, where $\mathbf{A}^{\dagger}$ is the pseudo inverse of $\mathbf{A}$, *i.e.* $\mathbf{A}^{\dagger} = [\mathbf{A}^{\mathbf{T}}\mathbf{A}]^{-1}\mathbf{A}^{\mathbf{T}}$. Our search task reduces to optimization over $\mathbf{d}$ only, *i.e.* $(\mathbf{d_{opt}}, \mathbf{h_{opt}}) = \arg\min_{\mathbf{d},\mathbf{h}} ||\mathbf{A}\mathbf{h} - \mathbf{W}\mathbf{d}||_2$ is equivalent to

$$\mathbf{d_{opt}} = \arg\min_{\mathbf{d}} ||\mathbf{A}\mathbf{A}^{\dagger}\mathbf{W}\mathbf{d} - \mathbf{W}\mathbf{d}||_2^2 \tag{15}$$

$$= \arg\min_{\mathbf{d}} \mathbf{d}^{\mathbf{T}}\mathbf{E}\mathbf{d} \tag{16}$$

Minimizing $\mathbf{d}^{\mathbf{T}}\mathbf{E}\mathbf{d}$, with $\mathbf{E}$ as a positive semi-definite matrix and $d_i = \pm 1 (i = 1, 2, ..., m)$ is a combinatorial search problem over the 2^m discrete choices of $\mathbf{d}$.

Using the fact $d_i d_j = \pm 1$, we can rewrite it as

$$\mathbf{d}^{\mathbf{T}}\mathbf{E}\mathbf{d} = 2\sum E_{ij} - \sum_{d_i d_j = -1} E_{ij} \tag{17}$$

$$\arg\min_{\mathbf{d}} \mathbf{d}^{\mathbf{T}}\mathbf{E}\mathbf{d} = \arg\max_{\mathbf{d}} \sum_{d_i d_j = -1} E_{ij} \tag{18}$$

If we separate $d_i = 1$ from $d_i = -1$ and regard E_{ij} as the edges connecting two parts, the above equation gives a max-cut of d_i's. □

Finding the solution for integrability test problem (P2) is now computationally reduced to solving a Max-cut. Note that we perform our search on a limited number of singular points instead of all the image pixels. The *configuration graph* actually condenses the information into limited nodes. These are the places where we have to make decisions, while other places are determined by fast marching. The choices of $\mathbf{d}$ and $\mathbf{h}$ have to be made in a global view.

4.4. Numerical approaches

Max-cut is NP-hard [2]. Brute force search is only feasible for small graph size *e.g.* less than 10 nodes. For larger graphs we compute the Max-cut by semi-definite programming (SDP)[3]. We relax the edge configuration $\mathbf{d}$ to real values. Let $\mathbf{X} = \mathbf{d}\mathbf{d}^{\mathbf{T}}$, our problem can be relaxed to

$$\begin{array}{ll} \text{minimize} & \mathbf{d}^{\mathbf{T}}\mathbf{E}\mathbf{d} = tr(\mathbf{E}\mathbf{X}) \\ \text{subject to} & X_{ii} = tr(\mathbf{A_i}\mathbf{X}) = 1, \quad i = 1, 2, ..., m \\ & \mathbf{X} \in S^m_+,\ \mathbf{A_i} = \mathbf{e_i}\mathbf{e_i^T} \end{array}$$

which is a SDP problem [4]. Due to the convexity of the semi-definite cone, SDP could be solved fast and reliably by the state-of-art implementations [16]. In practice, it works very well in optimizing (16).

4.5. Shape integration

Now we have recovered the global configuration encoded by $\mathbf{d}$ and $\mathbf{h}$. Since $d_i = \pm 1$ indicates which one of the neighboring vertices is higher, peaks (convex) are identified by finding the vertex higher than all of its neighbors in the graph. From these peaks and their estimated heights, local propagations are stitched together according to (4). The region $\Omega(p) = \{q : z(q) = \arg\max_{p \in \mathcal{P}}(z(p) - D(p, q))\}$ is called the *influential zone* of p. These are the places where local propagation from p take over the result surface.

If the graph topology is incorrect, the result might still have kinks. Therefore, we need to check whether there are kinks between neighboring patches. Suppose $n_1(s), n_2(s)$ are surface normals estimated from neighboring influential zones Ω_1, Ω_2 respectively. $S(B)$ measures the discrepancy of surface normals propagated from Ω_1 and Ω_2 on the boundary B, which indicates the "sharpness" of the kink.

$$S(B) = [\int_B 1 - n_1(s) \cdot n_2(s) ds] / \int_B ds \tag{19}$$

We check and correct the heights of the patches to ensure that the value of (19) is small, *i.e.* without kinks.

[2] However, there are polynomial algorithms for the Min-cut problem.

[3] An SDP solves this problem: minimize $tr(\mathbf{C}\mathbf{X})$, subject to $tr(\mathbf{A_i}\mathbf{X}) = b_i$, $i = 1, 2, ..., p$, $\mathbf{X} \in S^n_+$. S^n_+ denote the set of positive semi-definite matrices.

[4] $\mathbf{e_i}$ $(i = 1, 2, ..., m)$ are the canonical basis.

4.6. Algorithm overview

In summary, our algorithm contains the following steps:

1. **Singular points detection** Select points with local maximal intensities and above some threshold.
2. **Fast marching** Estimates the height differences $D(p, q)$ from each singular point p to its local neighborhood.
3. **Configuration graph formulation** Construct the graph by connecting neighboring singular points.
4. **Disambiguation** Use SDP to find $\mathbf{d_{opt}}$ optimizing $\mathbf{d^T E d}$ in (16).
5. **Shape integration** Identify peaks from $\mathbf{d_{opt}}$. Adjust the influential zones when necessary. Estimate $\mathbf{h}$ at the peaks by (14). Surface heights are computed by stitching equation (4).

5. Results

We test our approach on synthetic and real images with complex shapes and it outperforms previous energy minimization and local propagation methods.

We first test the algorithm on the Matlab PEAKS image in Figure 5. It consists of 9 singular points, of which 3 are peaks, 3 are valleys and 3 are saddle points. This simple example shows the power of our method to resolve local ambiguities. As seen in Figure 5, we recover correctly the convex/concave/saddle labels as well as the overall 3D shape.

Figure 6 shows results on more complex synthetic images. For the Venus image, the graph has 94 vertices and 213 edges, of which 3 are determined to be global peaks. In the case of the Egyptian queen Nefertiti, the graph has 65 vertices and 148 edges, of which 3 are determined to be global peaks. We see that local propagation is able to produce detailed local 3D structure while integrability constraint is able to determine the relative heights of different parts as well as the propagation direction. For example, the algorithm manages to figure out that propagation should be performed from the lower to the upper part on Nefertiti's headdress, where otherwise the whole shape of the surface will be changed.

Previous methods [1, 9, 17, 19, 20] perform poorly on these examples. Energy minimization approaches suffer from smoothness constraint and local minima: the result surface is flat globally but bumpy locally (Figure 6(e)). Local propagation methods can give reasonable smooth surfaces, but fail to give a correct global shape (Figure 6(f)). For a fair comparison, we have already tuned the parameters for the previous methods [1, 17] and take the best results. Further, we choose the top two results from the six methods surveyed in [19].

Figure 5 shows recovering the correct configuration is critical for propagation methods. If the propagation is performed from incorrect singular points, the result could be very bad. Even if we are lucky to choose the right starting point out of hundreds of candidates, propagation from only one point still does not give correct results. This is due to the fact that the influential zone of a single point usually does not cover the entire image.

Our algorithm is tested on two real images, a stone sculpture and part of the Three Graces. The light source directions are *approximately* estimated as $(0.55, 0.55, 0.62)^T$ and $(0, 0, 1)^T$ respectively. Examples under uncontrolled illumination conditions are extremely difficult. Our single light source, Lambertian assumption is usually violated in practice: incorrect intensities due to diffuse light sources and inter-reflections, multiple self shadows, discontinuities such as cracks and dents, nonconstant albedo due to paints and dirts, just to name a few of them. Our algorithm has done a good job despite these difficulties (Figure 5). The rough shapes are correct and details are well preserved.

6. Discussions

Several authors have pointed out that the local ambiguity problem in SFS. In [18], local ambiguities were resolved by user specifying surface normals at a few key points. Kimmel and Bruckstein [6] observed that the classification of singular points and relations between them are important. However, their method does not work for saddle (shoulder) points, which is a serious limitation. Prados and Faugeras [12] assumed that boundary conditions as well as heights at singular points were given, but these data are generally inaccessible. Their recent work [13] considered the effect of the distance between the light source and the surface and a monotonous scheme was obtained. They actually assumed that points were becoming further away from the camera while they were approaching to the image boundary. This is a very limited assumption.

Comparing to previous energy minimization and local propagation methods our approach directly resolves the ambiguity in SFS. Our method requires no boundary data, user interaction or additional assumptions to solve SFS.

We would like to point out a shortcoming of our solution. The equivalence between problem (P1) (kink-free) and (P2) (integrability) depends on the correct configuration graph topology. Configuration graph is defined on 'neighbor pairs'. Correct neighbor pairs (between singular points) requires knowing *a prior* that the height decreases monotonously from one to the other. We use a heuristic of choosing close-by singular points as neighbor pairs, and it works for most cases. However, there is no guarantee that such heuristic always leads to the correct solution. In fact we have observed bad neighbor pairs causing incorrect solutions (satisfying (P2) but having kinks). One possible

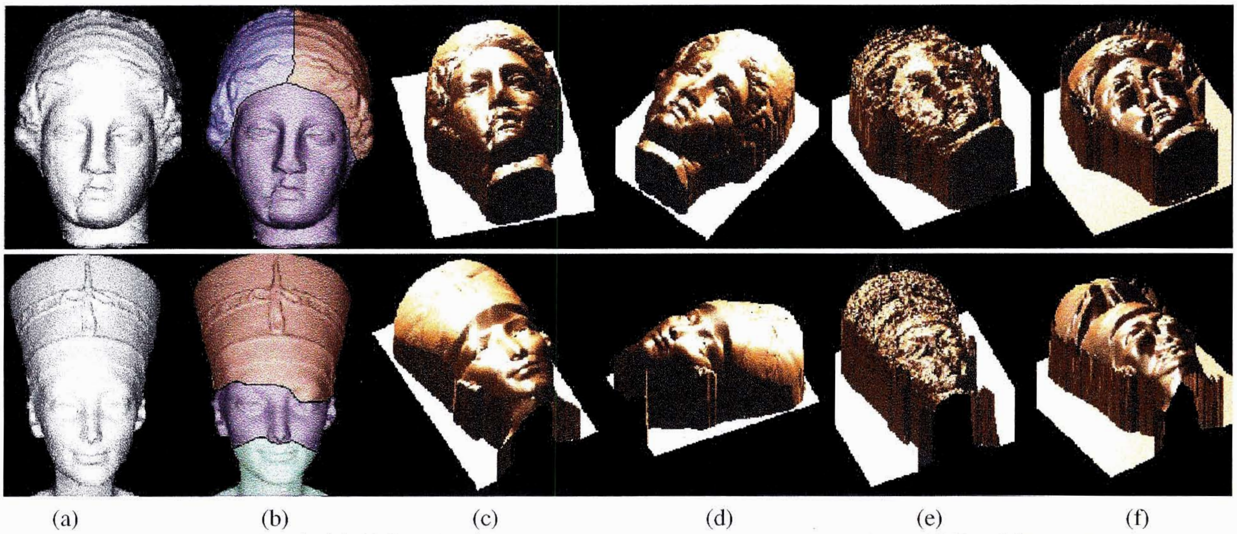


Figure 6. Results of Venus and Nefertiti. (a) Input image. (b) Influential zones of peaks (color coded). (c)-(d) Result by our method under two different views. (e) Result by Tsai and Shah's method [17, 19] with the best parameters. (f) Result by Bichsel and Pentland's method [1, 19] with the best parameters.

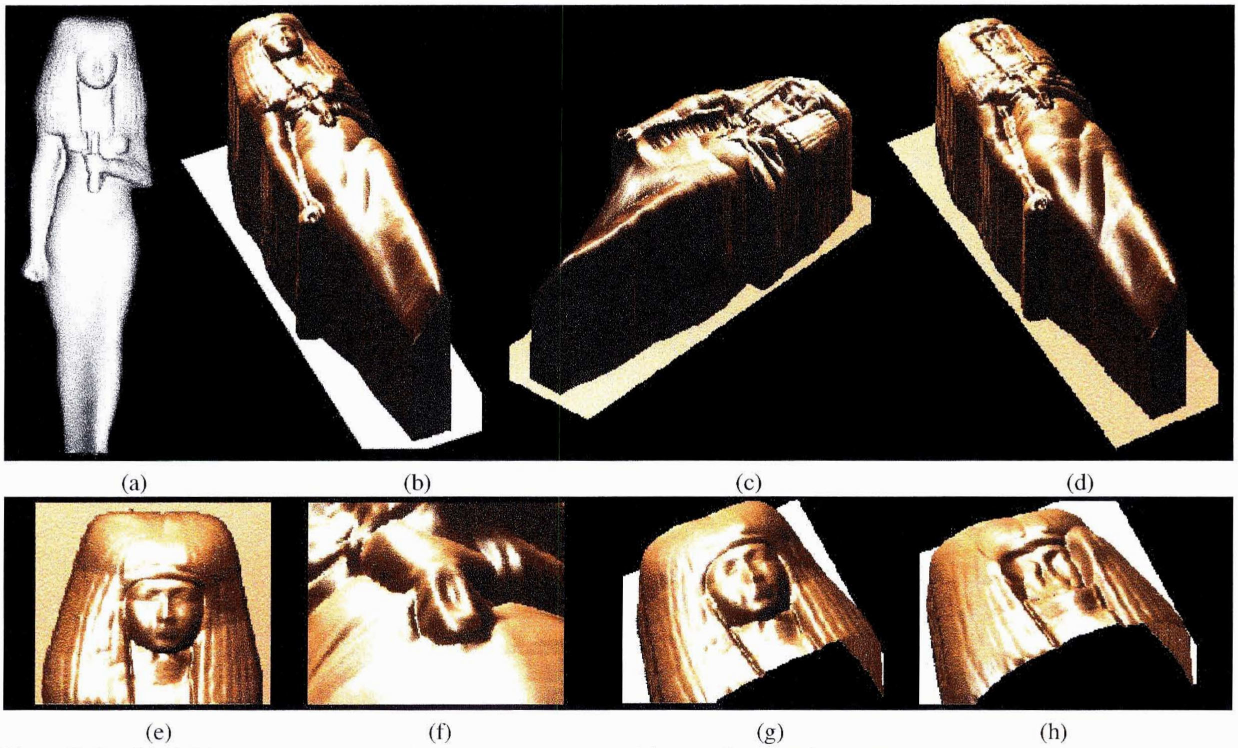


Figure 7. Results of Isis. (a) Input image. (b)(e)(f) Surface computed by our algorithm has a correct global configuration as well as local surface details shown in (e)(f) (her face and left hand). (c) Incorrect labels of singular points leads to incorrect global shape. (d) Propagation from one correct singular point has a very limited range. (h) shows a zoom-in incorrect face reconstruction of (d), compared with our method in (g).

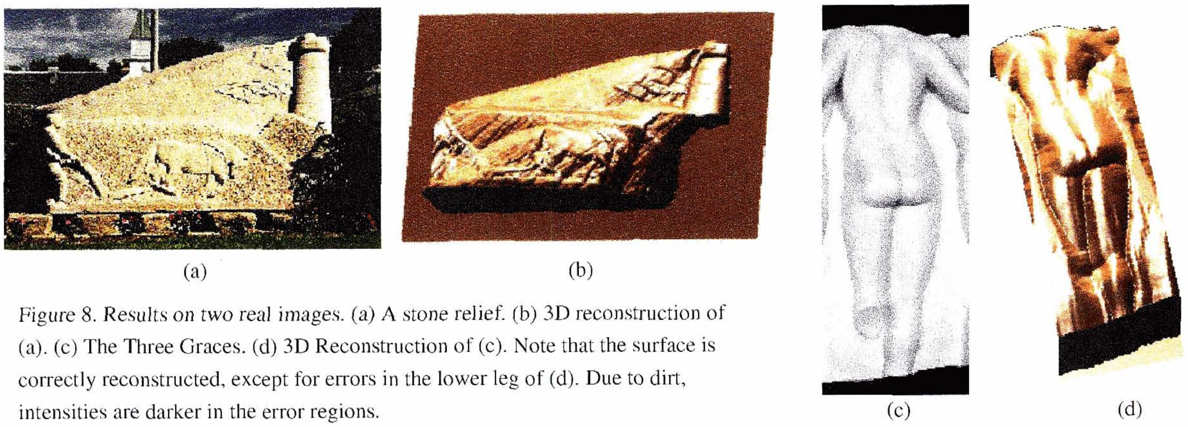


Figure 8. Results on two real images. (a) A stone relief. (b) 3D reconstruction of (a). (c) The Three Graces. (d) 3D Reconstruction of (c). Note that the surface is correctly reconstructed, except for errors in the lower leg of (d). Due to dirt, intensities are darker in the error regions.

solution is to perturb the configuration graph by removing certain neighbor pairs and check for the kink-free constraint upon solving (P2).

7. Conclusion

In this paper, we point out that image-irradiance equation alone is not a sufficient condition for SFS: the shape must also be kink-free. The kink-free constraint should not be confused with smoothness constraint — local shape can be computed exactly up to convex/concave/saddle labels. The question is how we can glue these local shapes together, which amounts to a combinatorial search. In this work, we simplified this search by checking the global integrability constraint between the patches. To solve it, we build the configuration graph and computationally this combinatorial optimization reduces to a Max-cut problem. The experiment results on real and synthetic data demonstrate our method works well on complex shapes.

Acknowledgement This work is supported by National Science Foundation through the grants: NSF-IIS-04-47953(CAREER), NSF-IIS-03-33036(IDLP), NSF-IIS004-31070, and NSF-CNS-04-23891.

References

[1] M. Bichsel and A. P. Pentland. A simple algorithm for shape for shading. In *CVPR*, pages 459–465, 1992.

[2] A. R. Bruss. The eikonal equation: Some results applicable to computer vision. In B. K. P. Horn and M. J. Brooks, editors, *Shape from Shading*, pages 69–87. MIT Press, Cambridge, MA, 1989.

[3] A. Crouzil, X. Descombes, and J.-D. Durou. A multiresolution approach for shape from shading coupling deterministic and stochastic optimization. *PAMI*, 25(11), 2003.

[4] B. K. P. Horn. Image intensity understanding. In *MIT AI Memo*, 1975.

[5] K. Ikeuchi and B. K. P. Horn. Numerical shape from shading and occluding boundaries. *Artificial Intelligence*, 17(1-3):141–184, 1981.

[6] R. Kimmel and A. M. Bruckstein. Global shape from shading. *Computer Vision and Image Understanding*, 64(1):188–189, July 1996.

[7] R. Kimmel and J. A. Sethian. Optimal algorithm for shape from shading and path planning. *Journal of Mathematical Imaging and Vision*, 14(3):237–244, 2001.

[8] Y. Leclerc and A. Bobick. The direct computation of height from shading. In *CVPR*, June 1991. Hawaii.

[9] C. Lee and A. Rosenfeld. Improved methods of estimating shape from shading using the light source coordinate system. *Artificial Intelligence*, 26(2):125–143, 1985.

[10] J. Oliensis. Uniqueness in shape from shading. *IJCV*, 6:75–104, 1991.

[11] J. Oliensis and P. Dupuis. A global algorithm for shape from shading. In *ICCV*, pages 692–701, 1993.

[12] E. Prados and O. D. Faugeras. ”perspective shape from shading”and viscosity solutions. In *ICCV*, pages 826–831, 2003.

[13] E. Prados and O. D. Faugeras. Shape from shading: A well-posed problem? In *CVPR (2)*, pages 870–877, 2005.

[14] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.*, 29(3), 1992.

[15] A. Tankus, N. Sochen, and Y. Yeshurun. Perspective shape-from-shading by fast marching. In *CVPR (1)*, 2004.

[16] K. C. Toh, M. J. Todd, and R. Tutuncu. SDPT3 — a Matlab software package for semidefinite programming. *Optimization Methods and Software*, 11:545–581, 1999.

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

[18] G. Zeng, Y. Matsushita, L. Quan, and H.-Y. Shum. Interactive shape from shading. In *CVPR (1)*, pages 343–350, 2005.

[19] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape from shading: A survey. *PAMI*, 21(8):690–706, 1999.

[20] Q. Zheng and R. Chellappa. Estimation of illuminant direction, albedo, and shape from shading. *PAMI*, 1991.