Polynomial Shape from Shading

Ady Ecker   Allan D. Jepson
University of Toronto

Abstract

We examine the shape from shading problem without boundary conditions as a polynomial system. This view allows, in generic cases, a complete solution for ideal polyhedral objects. For the general case we propose a semidefinite programming relaxation procedure, and an exact line search iterative procedure with a new smoothness term that favors folds at edges. We use this numerical technique to inspect shading ambiguities.

1. Introduction

The shape from shading (SFS) problem [12, 26] is to recover the 3D shape of a surface from a single image, whose intensities are related to angles between surface normals and light source direction. SFS belongs to a wide class of problems in computer vision that involve embedding points in Euclidean space based on angles or distances information. These problems have natural formulations as systems of polynomial equations. Both exact methods, such as Gröbner basis and homotopy continuation, and convex relaxation techniques, have been applied to polynomial systems arising from diverse problems as structure from motion [5, 15, 24], camera calibration [7] and low-dimensional embedding [29, 38]. In this paper we apply similar techniques to the SFS problem, which traditionally was treated mostly as a general nonlinear PDE that is notoriously difficult to optimize. While the polynomial formulation is not new (*e.g.* [25]), only recently theory and software for polynomial systems became widely available.

Throughout we will focus on the standard Lambertian model, *i.e.* orthographic projection, known distant light source, no interreflections, unit albedo, but no other boundary conditions. Denote the unit light source vector by $\mathbf{L} = (a, b, c)$, and the surface normal by $\mathbf{N} = (-p, -q, 1)^T$. The Lambertian intensity at an unshadowed point is

$$I = \frac{\mathbf{L} \cdot \mathbf{N}}{\|\mathbf{N}\|} = \frac{-ap - bq + c}{\sqrt{1 + p^2 + q^2}} \in [0, 1]. \qquad (1)$$

Squaring and rearranging we get the quadratic equation

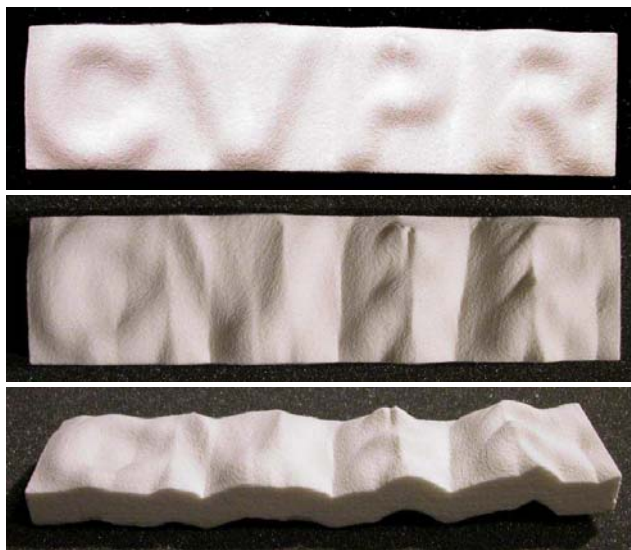$$(1 + p^2 + q^2)I^2 - (-ap - bq + c)^2 = 0. \qquad (2)$$

Figure 1. An exaggerated shading ambiguity illustrated by three images of one real object. The object has a pre-computed shape, designed using techniques described in this paper, which was then realized by 3D-printing and illuminated with a directed light source (plus weak ambient). Bottom: an oblique view of the object. Middle and top: a top view, with the essential difference being that we moved the directed source to a different position. It is implausible that a human viewing the top image would infer the correct surface.

To avoid solutions to $\mathbf{L} \cdot \mathbf{N}/\|\mathbf{N}\| = -I$ we add the constraint

$$-ap - bq + c \geq 0. \qquad (3)$$

The SFS problem is to find a surface satisfying (2) and (3) for each point in the image.

The polynomial form (2) has several advantages over the quotient form (1). First, small polynomial systems can be completely solved. We demonstrate this in section 2, where we show that all solutions to the SFS problem for an ideal, generic polyhedron can be found by homotopy solvers for polynomial systems. Second, exact line searches are possible in the polynomial form, but require expensive bisections in other forms. In section 3 we demonstrate the effectiveness of exact line searches as part of an iterative method for SFS on a grid. Third, semidefinite programming (SDP) relaxations for polynomial systems can produce approximate
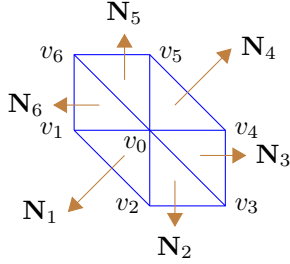
Figure 2. Triangulation around a vertex.

solutions without requiring an initial guess. We derive an SDP relaxation for SFS in section 4. In absence of boundary conditions, the SFS problem is known to be ambiguous [17]. In section 5 we propose a method for generating artificial shading ambiguities as illustrated in figure 1.

2. Exact SFS for polyhedra by homotopy

In this section we examine polyhedral SFS. The problem was first formulated by Horn [11]. Iterative procedures were proposed by Sugihara [34] and Lee and Kuo [19]. Penna [25] dealt with perspective projection. Yang *et al*. [40] highlighted the possibility of multiple solutions. Shimodaira [31] applied the DIRECT (dividing rectangles) solver, which could be the only previous work with a solution guarantee. However, space-partitioning techniques tend to slow considerably as the dimension grows [33], require initial range estimation for the variables, and multiple solutions were not considered.

Since the problem can be formulated as a polynomial system, it is natural to apply homotopy solvers [33]. These solvers start with a structurally similar polynomial system for which all complex solutions are known, and trace the solutions as the system is continuously deformed to the desired polynomial system. Homotopy solvers guarantee "with probability 1" finding all complex solutions without requiring an initial guess, they scale well with problem dimension, and are parallelizable. Modern solvers can be considered reliable, although rare breaking cases exist. See [33] for in-depth discussion and comparison to Gröbner basis and resultants. Previous applications of parameter continuation to SFS include real continuation [9], graduate decrease of smoothness [35] and continuation through scale space [30]. Watson [37] mentions early work related to homotopy and SFS that we were unable to find.

Assume the surface is a triangular mesh, and consider a particular vertex $\mathbf{v}_0$ and its k neighbors $\mathbf{v}_1, \ldots, \mathbf{v}_k$ counterclockwise as in figure 2. To simplify notation we shift the coordinates so that $\mathbf{v}_0 = (0, 0, 0)$. The normal to a triangle $(\mathbf{v}_0, \mathbf{v}_i, \mathbf{v}_{i+1})$ is proportional to $\mathbf{v}_i \times \mathbf{v}_{i+1} = (y_i z_{i+1} - y_{i+1} z_i, x_{i+1} z_i - x_i z_{i+1}, x_i y_{i+1} - x_{i+1} y_i)$. Dividing by the
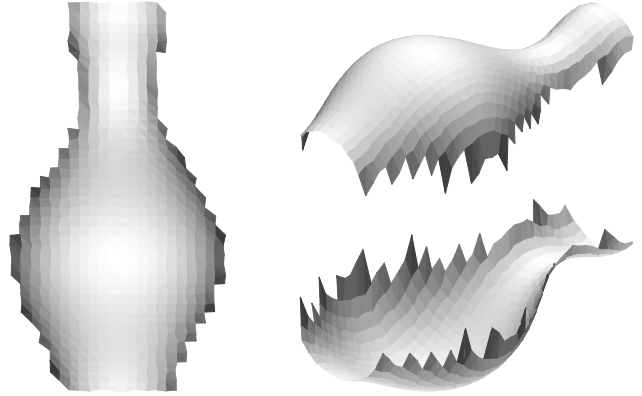

Figure 3. SFS of a polyhedron by homotopy. Left: synthetic Lambertian image with $\mathbf{L} = (0, 0, 1)$ of triangulated vase [26]. Right: the two solutions found by the algorithm.

last coordinate we get

$$-p_i = \frac{y_i z_{i+1} - y_{i+1} z_i}{x_i y_{i+1} - x_{i+1} y_i} \ , \ -q_i = \frac{x_{i+1} z_i - x_i z_{i+1}}{x_i y_{i+1} - x_{i+1} y_i}. \quad (4)$$

Substituting in (2) for every triangle, we get k quadratic equations in $z_1, \ldots, z_k$ (x_i, y_i are known from the image). If the system is generic, there are at most 2^k real solutions, which for small k can be found by a homotopy solver.

After solving for the possible configurations around each (internal) vertex, we prune the solutions. Configurations violating (3) or whose neighboring vertices have no compatible configuration are discarded. Then we need to solve a constraint satisfaction problem (CSP) to identify all consistent global solutions. Luckily, often an assignment of a configuration at a vertex and its neighborhood determines uniquely the configurations of its neighbors and therefore a simple assignment propagation and backtracking algorithm finds all global solutions.

Polyhedral SFS by homotopy is demonstrated in figure 3. The input contains 1143 triangles. Their vertices were randomly jittered to ensure generic systems. We used the homotopy solver HOM4PS-2.0 [20] to successfully find both solutions.

Note that theoretically it is also possible to solve for the light source, by writing the systems of the neighbors of a triangle together and forming a system of quartic equations with the condition $a^2 + b^2 + c^2 = 1$. This is practical only if the number of neighbors is small.

Unfortunately, the method described is sensitive to noise. Even 1% of intensities perturbation can lead to systems whose all solutions are complex. One might form the sum of squares of (2), take the partial derivatives and find all stationary points. This leads to k cubic equations, with up to 3^k solutions. Although solutions can be pruned, there still might be a large number of candidates. Given that for noisy systems it is harder to determine which neighboring configurations are compatible, the result is a much harder CSP.

The large number of inaccurate solutions may explain why humans find SFS of a triangulated mesh difficult [32].

3. Iterative procedure

In this section we present two technical improvements to global iterative SFS methods. These methods, *e.g.* [12, 26, 35], minimize the sum of squares of (2) over a grid. Denote by $\mathbf{z}$ the heights of all grid points arranged as a column vector, and let $\mathbf{p}_{ij} = \mathbf{z}_{i+1,j} - \mathbf{z}_{ij}$, $\mathbf{q}_{ij} = \mathbf{z}_{i,j+1} - \mathbf{z}_{ij}$ be the discrete partial derivatives of the surface. To avoid specifying any boundary conditions, for $M \times N$ image we solve for $\mathbf{z}$ on an extended $(M+1) \times (N+1)$ grid (the $(M+1, N+1)$ pixel is redundant). Then (2) becomes

$$\begin{bmatrix} \mathbf{z}_{i,j}^2, \mathbf{z}_{i+1,j}^2, \mathbf{z}_{i,j+1}^2, \mathbf{z}_{i,j}\mathbf{z}_{i+1,j}, \mathbf{z}_{i,j}\mathbf{z}_{i,j+1}, \\ \mathbf{z}_{i+1,j}\mathbf{z}_{i,j+1}, \mathbf{z}_{i,j}, \mathbf{z}_{i+1,j}, \mathbf{z}_{i,j+1}, 1 \end{bmatrix} \cdot \mathbf{u}_{ij} = 0 \quad (5)$$

$$\mathbf{u}_{ij} = \begin{bmatrix} 2I_{ij}^2, I_{ij}^2 - a^2, I_{ij}^2 - b^2, 2a(a+b) - 2I_{ij}^2, \\ 2b(a+b) - 2I_{ij}^2, -2ab, -2c(a+b), 2ac, 2bc, I_{ij}^2 - c^2 \end{bmatrix}^T. \quad (6)$$

Eq. (5) and (6) can be written in the form

$$r_{ij} = \mathbf{z}^T \mathbf{A}_{ij} \mathbf{z} + \mathbf{e}_{ij}^T \mathbf{z} + h_{ij} = 0, \quad (7)$$

with symmetric $\mathbf{A}_{ij}$. Let $\mathbf{r}(\mathbf{z}) = \begin{bmatrix} r_{11}, \ldots, r_{MN} \end{bmatrix}^T$. The sum of squared errors is $F(\mathbf{z}) = \|\mathbf{r}\|^2$. The Jacobian matrix of $\mathbf{r}$ has the rows

$$\mathbf{J}_{ij} = 2\mathbf{z}^T \mathbf{A}_{ij} + \mathbf{e}_{ij}^T, \quad (8)$$

and the gradient is simply $\nabla F = 2\mathbf{J}^T \mathbf{r}$.

Minimization of a general multivariate quartic is NP-hard [22, 28]. However, a useful property of polynomials is that exact line search takes linear time. Substituting $\mathbf{z} = \mathbf{z}_0 + \alpha \mathbf{d}$ in $F(\mathbf{z})$, where $\mathbf{z}_0$ is the current point and $\mathbf{d}$ is any search direction, we get a univariate quartic in α, whose global minimization involves solving a cubic equation in α. Experience in numerical analysis suggests that exact line search often requires less function evaluations and a smaller number of iterations than inexact line search procedures. In addition, the algorithm is less likely to be trapped in a poor local minimum. Although very natural and easy to implement, we are not aware of previous applications of exact line search to SFS, or embedding problems like [38].

A smoothness term is often added to widen the basin of attraction for convergence and suppress oscillations. The most common regularizer measures thin-plate energy, which integrates the squared second derivatives of the surface. A problem with this regularizer is that it will strive to flatten regions of varying intensities or edges where the surface cannot be flat. In [13] it is proposed to down weight the smoothness term by a measure of intensity change. That smoothness term frees the surface to bend where intensity is varying, but does not enforce folding explicitly. In addition, reducing the smoothness term allows surface oscillations, as will be explained in section 5.

Instead we define a new smoothness term that mitigates these problems. Consider two neighboring pixels with normals $(-p_1, -q_1, 1)$, $(-p_2, -q_2, 1)$ and intensities I_1, I_2. The smallest possible angle θ between these normals satisfies

$$\cos(\theta) = \cos(\arccos(I_1) - \arccos(I_2)) \\ = I_1 \cdot I_2 + \sqrt{1 - I_1^2} \cdot \sqrt{1 - I_2^2}. \quad (9)$$

For nearby pixels on a smooth surface, and for pixels with large intensity difference, it is reasonable to expect that the angle between the normals is close to θ, *i.e.* $\cos(\theta) \approx \frac{p_1 p_2 + q_1 q_2 + 1}{\sqrt{1 + p_1^2 + q_1^2}\sqrt{1 + p_2^2 + q_2^2}}$. Plugging in the Lambertian assumption (1) and multiplying by the denominator we define the smoothness term as another quartic in $\mathbf{z}$

$$S(\mathbf{z}) = \sum \Big((p_1 p_2 + q_1 q_2 + 1) I_1 I_2 \\ - \cos(\theta)(-ap_1 - bq_1 + c)(-ap_2 - bq_2 + c) \Big)^2. \quad (10)$$

Our complete algorithm minimizes $F(\mathbf{z}) + \lambda S(\mathbf{z})$ using conjugate gradient with exact line search, and reduces λ gradually. The method is demonstrated in figures 4 and 5.

In the algorithm above we ignored the non-negativity constraint (3). It is possible to take it into account by minimizing

$$\sum \Big((1 + p^2 + q^2) I^2 - (-ap - bq + c)^2 \operatorname{sign}(-ap - bq + c) \Big)^2. \quad (11)$$

The exact line search requires sorting α according to where the terms switch signs, and searching for the minimum of the quartics in each intermediate interval. The coefficients of these quartics can be accumulated as the line is traversed. This usually makes insignificant difference for images without shadows.

4. SDP relaxation

While iterative techniques with exact line search are effective, often there exist initial guesses that lead to suboptimal local minima. This motivated us to look at the completely different approach of convex relaxations, which guarantee convergence to a global minimum of a related problem. In this section we derive a Lasserre-type SDP relaxation to SFS. The idea is to introduce new variables for products of variables (lifting) so that non-convex terms become linear. Additional positive-semidefinite constraints are added to capture relationships between all variables and tighten the approximation. Since full relaxations are
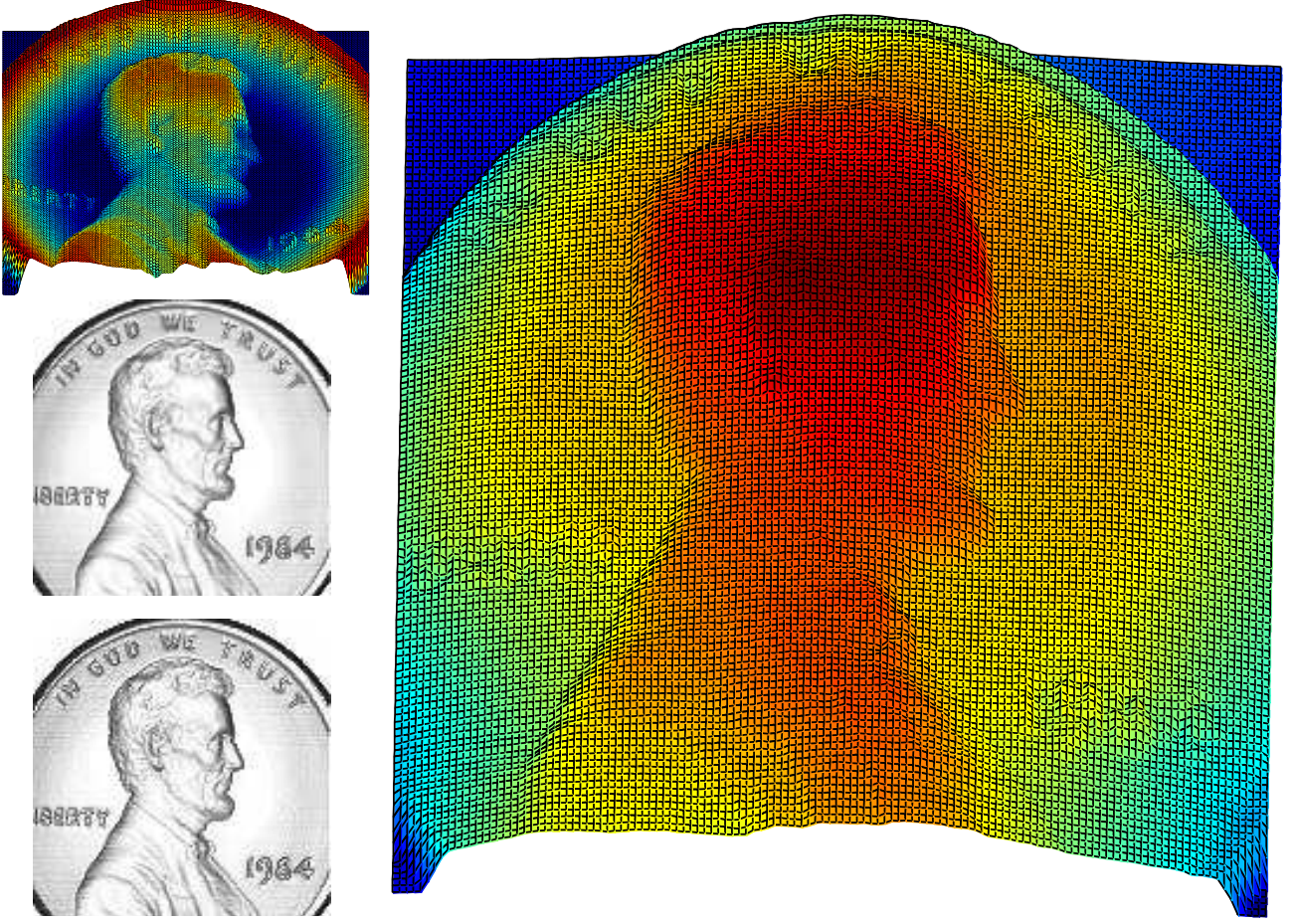
Figure 4. SFS of the synthetic penny [26]. The source surface (upper-left) generates the input image (middle-left) with $\mathbf{L} = (0, 0, 1)$. Our iterative method, initialized from a section of a sphere, produced the surface on the right. The computed surface generates the lower-left image. While the 3D surfaces are different, the RMS of the images difference is only 0.008, and the maximal absolute difference of intensities is 0.117.

very expensive, scalability is achieved via sparse relaxation [23, 36]. The semidefinite constraints are defined on cliques, which in our case are just unit triangles.

For each point (i, j) we use an extended set of variables: $\mathbf{x}_{ij}^{rst}$ corresponds to a monomial $(\mathbf{z}_{ij})^r \cdot (\mathbf{z}_{i+1,j})^s \cdot (\mathbf{z}_{i,j+1})^t$. Here, r, s, t are nonnegative integers, $r + s + t \leq 2d$, and d is called the relaxation order. In this notation $\mathbf{x}_{ij}^{k00}, \mathbf{x}_{i-1,j}^{0k0}$ and $\mathbf{x}_{i,j-1}^{00k}$ refer to the same variable corresponding to $(\mathbf{z}_{i,j})^k$, $k = 1$ refers to the surface height at point (i, j), and $\mathbf{x}_{ij}^{000}$ is the constant 1.

Let $(r_1, s_1, t_1) = (0, 0, 0)$, $(r_2, s_2, t_2), \ldots, (r_D, s_D, t_D)$ be an enumeration of all r, s, t such that $r + s + t \leq d$ and $D = \binom{d+3}{3} = \mathrm{O}(d^3)$. For each image pixel, positive semidefinite constraints are defined on $D \times D$ moment matrices whose elements are variables

$$\left[\mathbf{M}_{ij}\right]_{m,n} = \mathbf{x}_{ij}^{r_m+r_n, s_m+s_n, t_m+t_n}, \quad \left[\mathbf{M}_{ij}\right]_{1,1} = 1. \quad (12)$$

Ideally, $\mathbf{M}_{ij}$ would be equal to the symmetric positive semidefinite rank-one matrix

$$\left[1, \mathbf{x}_{ij}^{r_2 s_2 t_2}, \ldots, \mathbf{x}_{ij}^{r_D s_D t_D}\right]^T \cdot \left[1, \mathbf{x}_{ij}^{r_2 s_2 t_2}, \ldots, \mathbf{x}_{ij}^{r_D s_D t_D}\right]. \quad (13)$$

The SDP relaxation for non-shadowed pixels is

$$\min\left(\sum_{ij} \mathrm{trace}(\mathbf{M}_{ij}) + G \cdot \varepsilon\right) \quad \text{s.t.} \quad (14)$$

$$(a + b)\mathbf{x}_{ij}^{100} - a\mathbf{x}_{ij}^{010} - b\mathbf{x}_{i,j}^{001} + c \geq 0 \quad (15)$$

$$-\varepsilon \leq \left[\mathbf{x}_{ij}^{r+2,s,t}, \mathbf{x}_{ij}^{r,s+2,t}, \mathbf{x}_{ij}^{r,s,t+2},\right.$$
$$\mathbf{x}_{ij}^{r+1,s+1,t}, \mathbf{x}_{ij}^{r+1,s,t+1}, \mathbf{x}_{ij}^{r,s+1,t+1}, \quad (16)$$
$$\left.\mathbf{x}_{ij}^{r+1,s,t}, \mathbf{x}_{ij}^{r,s+1,t}, \mathbf{x}_{ij}^{r,s,t+1}, \mathbf{x}_{ij}^{r,s,t}\right] \cdot \mathbf{u}_{ij} \leq \varepsilon$$

$$\sum_{ij} \mathbf{x}_{ij}^{100} = 0 \quad (17)$$

$$\mathbf{M}_{ij} \succeq 0, \quad \left[\mathbf{M}_{ij}\right]_{1,1} = 1. \quad (18)$$

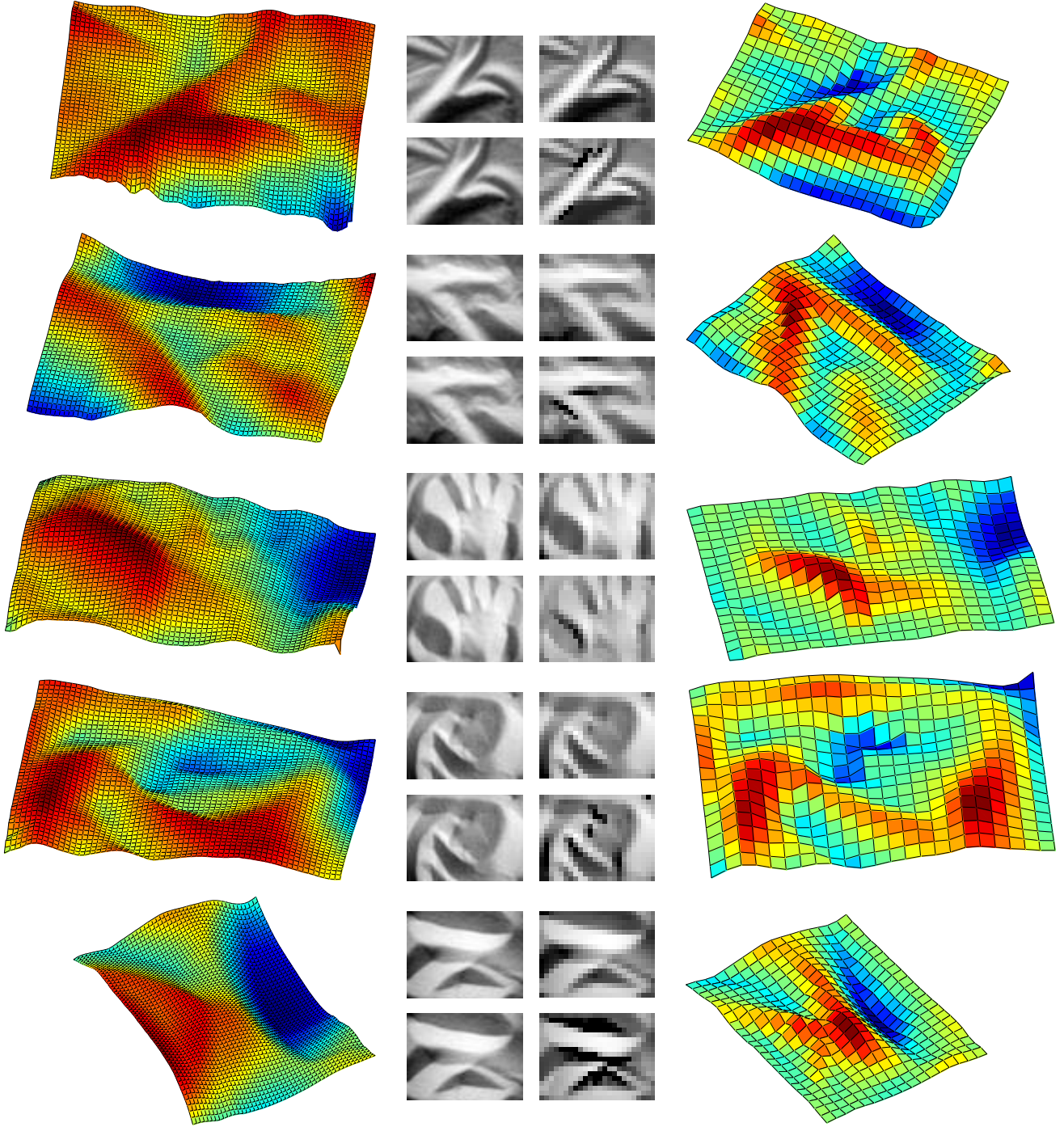In the system above, eq. (15) is analogous to (3).

Figure 5. Surface reconstruction by the iterative (left) and SDP (right) methods. In the middle, the top two images are source images. The bottom images are renderings of the left and right surfaces. At black points in the SDP solutions condition (15) equals zero.

Eq. (16) is derived by multiplying (5) with monomials $(\mathbf{z}_{ij})^r \cdot (\mathbf{z}_{i+1,j})^s \cdot (\mathbf{z}_{i,j+1})^t$ for all nonnegative integers r, s, t such that $r + s + t \leq 2d - 2$. $\mathbf{u}_{ij}$ are vectors of constants (6), and ε is a slack variable. Condition (17) fixes the global depth ambiguity in orthographic projection. Eq. (18) is the positive semidefinite constraint on (13). The objec-

tive (14) minimizes the sum of traces and G times the slack variable ε, where G is the total number of diagonal elements in all moment matrices. Ideally, the matrices $\mathbf{M}_{ij}$ should be rank-one. Trace minimization is a commonly used approximation to rank minimization.

The resulting SDP problem is solved in polynomial time without an initial guess by a SDP solver. Currently we extract the solution simply from $\mathbf{x}_{ij}^{100}$. Note that this doesn't work in case there are multiple solution (in particular $\mathbf{L} = (0, 0, 1)$), because the matrices $\mathbf{M}_{ij}$ could be convex combinations of solutions. Similar solution extraction with a random perturbation is described in [36], and applied to PDEs with boundary conditions [23]. Also note that Lasserre's condition for convergence as $d \to \infty$ in [18] does not hold for our sparse relaxation. However, only small orders d are practical anyways. A solution extraction scheme for dense relaxations is described in [10], but does not extend easily to sparse relaxations with noisy input.

The iterative and SDP procedures are compared in figure 5 on real images of cloth (upper two) and sugar (lower three). The intensities were linearly transformed to the range [0,1]. Since the light source direction is unknown, we sampled 100 light directions using the spherical spiral method [2], ran the iterative algorithm, and kept the best light for each image. Though more expensive, this approach is more robust compared to estimating the light directly from the images or adding the light as a parameter to the optimization. The initial state for the iterative method was $\mathbf{z} = 0$. The SDP results were obtained from low-resolution images (18×24) using the same light sources and relaxation order $d = 2$. The SDP solver we used is DSDP5.8 [4], interfaced with YALMIP [21]. For the iterative method, the RMS errors in reconstructing the input images are all below 0.01. RMS errors for the SDP method are (top to bottom): 0.06, 0.07, 0.09, 0.07 and 0.14.

5. Generating SFS ambiguities

In this section we explore generating SFS ambiguities, *i.e.* substantially different surfaces that have identical Lambertian images. Our main goal is to provide researches a better way to study surface priors. Instead of defining a shape prior and running a SFS algorithm, a good prior should first be able to pick the correct surface from its ambiguous counterparts. We can also envision a future application in surface inspection, where by inspecting the space of ambiguous shapes, possibly with few known anchor points, one can position the light source so that ambiguous deviations are tolerable.

Previously, SFS ambiguities were encountered when SFS algorithms converged to the wrong surface, or in mathematical analysis of simple shapes such as planes and spheres [6, 17]. A recent work by Kemelmacher-Shlizerman *et al.* [16] studied Mooney faces generated by thresholding Lambertian images. They showed that different surfaces can have an identical isophote (iso-intensity contour) and identical binary image. Another related ambiguity is the bas-relief [3, 39], which technically involves small changes in albedo.
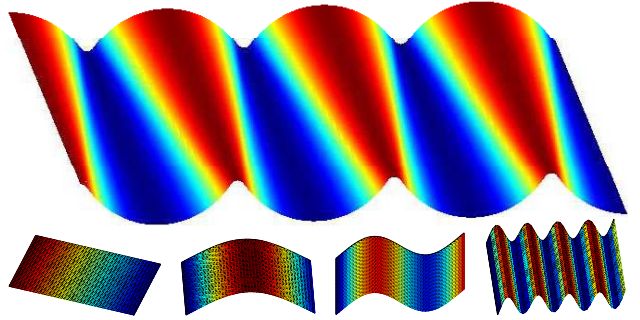

Figure 6. Shading ambiguities of uniform-intensity images. Top: interlaced cone sections with the same opening angle whose axes point in the light direction will generate a uniform intensity image. Bottom: some eigenvectors of $\mathbf{J}$ for the flat surface $\mathbf{z} = 0$ with $L = (\frac{1}{\sqrt{2}}, 0, \frac{1}{\sqrt{2}})$.

SFS ambiguities can arise from micro-perturbations, *e.g.* placing a pyramid reflecting the desired intensity at every pixel [14]. These surfaces are uncommon. On the other extreme, smooth surfaces can also be ambiguous. For instance, a single-intensity image can arise from a plane or a cone whose axis is aligned with the light direction. Moreover, such cones can be reflected and stitched together, as shown in figure 6, creating a wavy surface. In general, a uniform-intensity image comes from a ruled surface [17]. Note that these surfaces can be very smooth. Furthermore, Freeman's generic light source assumption [8] cannot distinguish between a plane and such a cone.

For any surface whose image is clipped to just a patch, the SFS equations represent a PDE without boundary conditions. The lack of these boundary conditions gives rise to ambiguities [17]. We can see these in the discrete case as follows. Consider a solution $\mathbf{z}_0$, i.e. $F(\mathbf{z}_0) = 0$ and $\mathbf{r}(\mathbf{z}_0) = \begin{bmatrix} 0, \dots, 0 \end{bmatrix}^T$. Note that the Jacobian $\mathbf{J}$ in (8) is $MN \times (M+1)(N+1)$. If $\mathbf{J}$ has full column rank, then by the implicit function theorem applied to $\mathbf{r}(\mathbf{z}) = 0$, there is a manifold of solutions in an ε-neighborhood of $\mathbf{z}_0$. While the theorem guarantees solutions only in arbitrarily small neighborhoods, in many cases the ambiguity can be very substantial (see figures 1 and 7).

Next we describe how to generate numerically an ambiguous surface $\mathbf{z}$ from a given surface $\mathbf{z}_0$ and a light source $\mathbf{L}$. The idea is illustrated in figure 8. Starting from a solution $\mathbf{z}_0$ of the SFS equations, we take a large step that would change the image as little as possible. Consider the second order approximation $F(\mathbf{z}_0 + \mathbf{v}) \approx F(\mathbf{z}_0) + \nabla F(\mathbf{z}_0)^T \mathbf{v} + \frac{1}{2} \mathbf{v}^T \nabla^2 F(\mathbf{z}_0) \mathbf{v}$. Note that $F(\mathbf{z}_0) = 0$, $\nabla F(\mathbf{z}_0) = 0$. The Hessian is $\nabla^2 F(\mathbf{z}_0) = 2\mathbf{J}^T \mathbf{J}$. Hence, a desired direction $\mathbf{v}$ is a null vector of $\mathbf{J}$. Taking the step $\mathbf{z} = \mathbf{z}_0 + \alpha \mathbf{v}$, we get away from the solutions manifold. Applying our iterative procedure of section (3) we get to a point $\mathbf{z}_1$ on the solutions manifold. To ensure that we get far from $\mathbf{z}_0$, we project the conjugate
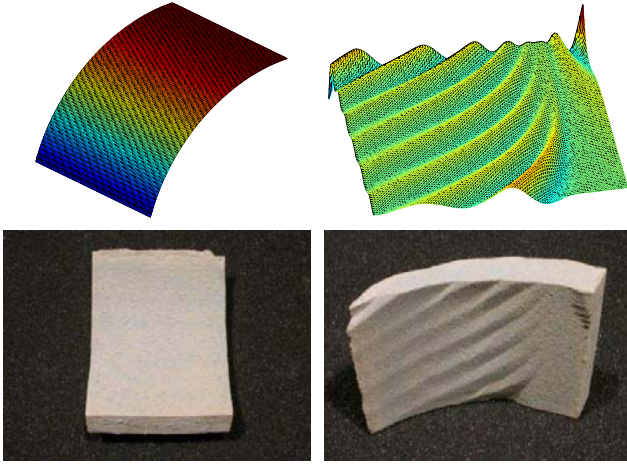
Figure 7. Artificially generated shading ambiguity. Top left: initial cylindrical surface. Top right: the null vector of $\mathbf{J}$. Bottom: 3D-print of the computed surface (right) generating the same image of a cylinder (left).

gradient search directions $\mathbf{d}$ so that $\mathbf{d} \cdot \mathbf{v} = 0$. A step size α is searched so that it is large and yet returning to the manifold is still possible.

It remains to describe how to compute the null vectors of $\mathbf{J}$. Note again that $\mathbf{J}$ is $MN \times (M+1)(N+1)$ and therefore large, sparse, and has at least $M + N + 1$ null vectors. This can be exploited in a divide-and-conquer scheme. We partition the image to four roughly equal blocks $I = \left[\begin{array}{c|c} I_1 & I_2 \\ \hline I_3 & I_4 \end{array} \right]$, and compute bases $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4$ for the null spaces of $\mathbf{J}$ restricted to each block recursively (svd is used for small images). For a $\frac{M}{2} \times \frac{N}{2}$ subimage, we form its restricted Jacobian on a $(\frac{M}{2} + 1) \times (\frac{N}{2} + 1)$ extended grid, making it rank-deficient. Thus, the bases $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3, \mathbf{B}_4$ overlap on the middle row and column of I. We stitch these bases together to form a basis $\mathbf{B}$ for the null space of $\mathbf{J}$ by finding linear combinations of the null vectors that match on the overlapping pixels. Let $\hat{\mathbf{B}}_1, \hat{\mathbf{B}}_2$ be the rows of $\mathbf{B}_1, \mathbf{B}_2$ that correspond to the overlapping pixels, and let $\left[\begin{array}{c} \mathbf{W}_1 \\ \mathbf{W}_2 \end{array} \right]$ be the null space of $\left[\begin{array}{cc} \hat{\mathbf{B}}_1 & -\hat{\mathbf{B}}_2 \end{array} \right]$. Then $\mathbf{B}_1 \mathbf{W}_1$ combined with $\mathbf{B}_2 \mathbf{W}_2$ give a basis for the null space of $\mathbf{J}$ restricted to $\left[I_1 \ I_2 \right]$. We combine $\mathbf{B}_3, \mathbf{B}_4$ and then the upper and lower parts in a similar way. Finally, we orthogonalize the resulting null space matrix $\mathbf{B}$.

It is convenient to sort the null space by smoothness from low to high frequency. We build a matrix $\mathbf{C}$ that measures smoothness by applying the filters $\left[1, -2, 1 \right]$, $\left[1, -2, 1 \right]^T$, $\left[\begin{array}{cc} 1 & -1 \\ -1 & 1 \end{array} \right]$ at every pixel. Let $\mathbf{V}$ be the sorted right-singular vectors of $\mathbf{CB}$. Then the matrix $\mathbf{BV}$ contains the null vectors in sorted order. Some eigenvectors for a plane and a cylinder surfaces are shown in figures 6 and 7.
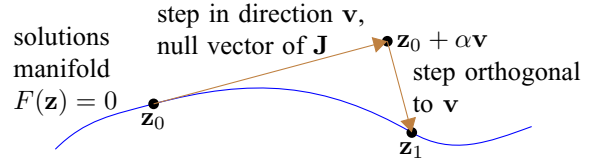

Figure 8. Generating a solution $\mathbf{z}_1$ from a known solution $\mathbf{z}_0$.

6. Conclusions

We applied tools specifically designed for polynomial systems to the SFS problem, both for polyhedral and curved surfaces. Their main advantage is in not requiring boundary conditions. In contrast, many propagation methods (*e.g.* [1, 12, 27]) rely on boundary conditions. Other propagation methods (*e.g.* [41]) rely on the existence of singular points (where $\mathbf{N} = \mathbf{L}$) in the image. While propagation approaches are fast and impressive reconstructions have been demonstrated, these reliances are conceptually unsatisfactory.

The SDP relaxation is a radically different approach to the SFS problem. It is interesting because convergence to a global minimum of the relaxed problem is guaranteed, without depending on the initial guess. Additional depth constraints obtained from other depth cues or user interaction can be added easily to the system. These may also include inequality constraints (*e.g.* (3) or front/behind relationships), which are difficult to incorporate into other methods. However, at present SDP solvers limit the method to low-resolution images and low relaxation orders (note the possibility of variable relaxation orders at different pixels). Better exploitation of sparseness, parallelism, and approximations of the full SDP problem, may allow solving larger problems in the future.

Ambiguities are a serious problem for all iterative SFS methods with no boundary conditions, since the algorithm can start converging to one solution in one part of the image and a conflicting solution in a remote part. For large and noisy images, this can occur even with boundary conditions. Therefore, some prior information is needed to choose between solutions. We proposed a new smoothness term for the iterative method, and used the sum of traces of the moment matrices in the SDP method. These priors are sensible, but obviously not a complete answer. We proposed a numerical method to visualize shading ambiguities, hoping this will contribute to better understanding their extent.

References

[1] A. H. Ahmed and A. A. Farag. A new formulation for shape from shading for non-Lambertian surfaces. In *Proc. CVPR'06*, pages 1817–1824, 2006.

[2] R. Bauer. Distribution of points on a sphere with application to star catalogs. *J. of guidance, control and dynamics*,

23(1):130–137, 2000.

[3] P. N. Belhumeur, D. J. Kriegman, and A. L. Yuille. The bas-relief ambiguity. *IJCV*, 35(1):33–44, 1999.

[4] S. J. Benson and Y. Ye. Algorithm 875: DSDP5: Software for semidefinite programming. *ACM Trans. Math. Software*, 34(3), 2008.

[5] M. Byröd, K. Josephson, and K. Åström. Fast and stable polynomial equation solving and its application to computer vision. *IJCV*, 84(3):237–256, 2009.

[6] J.-D. Durou and D. Piau. Ambiguous shape from shading with critical points. *J. of Mathematical Imaging and Vision*, 12(2):99–108, 2000.

[7] O. Faugeras, Q. Luong, and S. Maybank. Camera self-calibration: Theory and experiments. In *Proc. CVPR'92*, pages 321–334, 1992.

[8] W. T. Freeman. Exploiting the generic viewpoint assumption. *IJCV*, 20(3):243–261, 1996.

[9] D. Gelli and D. Vitulano. Speed up of shape from shading using graduated non convexity. In *Proc. DGCI'03*, pages 504–513, 2003.

[10] D. Henrion and J.-B. Lasserre. Detecting global optimality and extracting solutions in gloptipoly. In *Positive Polynomials in Control*, pages 293–310. Springer, 2005.

[11] B. K. P. Horn. Understanding image intensities. *Artificial Intelligence*, 8(2):201–231, 1977.

[12] B. K. P. Horn and M. J. Brooks, editors. *Shape from Shading*. MIT Press, Cambridge, MA, USA, 1989.

[13] R. Huang and W. Smith. A shape-from-shading framework for satisfying data-closeness and structure-preserving smoothness constraints. In *Proc. BMVC'09*, 2009.

[14] K. Z. I. Barnes. Instability of the eikonal equation and shape from shading. *Math. Model. Numer. Anal.*, 34(1):127–138, 2000.

[15] F. Kahl and D. Henrion. Globally optimal estimates for geometric reconstruction problems. *IJCV*, 74(1):3–15, 2007.

[16] I. Kemelmacher-Shlizerman, R. Basri, and B. Nadler. 3D shape reconstruction of mooney faces. In *Proc. CVPR'08*, pages 1–8, 2008.

[17] R. Kozera. Uniqueness in shape from shading revisited. *JMIV*, 7(2):123–138, 1997.

[18] J. Lasserre. Convergent SDP-relaxations in polynomial optimization with sparsity. *SIAM J. on Optimization*, 17(3):822–843, 2006.

[19] K. M. Lee and C. C. J. Kuo. Shape from shading with a linear triangular element surface model. *PAMI*, 15(8):815–822, 1993.

[20] T. L. Lee, T. Y. Li, and C. H. Tsai. HOM4PS-2.0: A software package for solving polynomial systems by the polyhedral homotopy continuation method. *Computing*, 83(2-3):109–133, 2008.

[21] J. Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proc. CACSD'04*, pages 284–289, 2004.

[22] Z.-Q. Luo and S. Zhang. A semidefinite relaxation scheme for multivariate quartic polynomial optimization with quadratic constraints. *SIAM J. on Optimization*, 20(4):1716–1736, 2010.

[23] J. N. M. Mevissen, M. Kojima and N. Takayama. Solving partial differential equations via sparse sdp relaxations. *Pacific J. of Optimization*, 4(2):213–241, 2008.

[24] F. Moreno-Noguer, M. Salzmann, V. Lepetit, and P. Fua. Capturing 3D stretchable surfaces from single images in closed form. In *Proc. CVPR'09*, 2009.

[25] M. Penna. A shape from shading analysis for a single perspective image of a polyhedron. *PAMI*, 11(6):545–554, 1989.

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

[27] E. Prados and O. Faugeras. A generic and provably convergent shape-from-shading method for orthographic and pin-hole cameras. *IJCV*, 65(1-2):97–125, 2005.

[28] L. Qi, Z. Wan, and Y.-F. Yang. Global minimization of normal quartic polynomials based on global descent directions. *SIAM J. on Optimization*, 15(1):275–302, 2005.

[29] V. Rabaud and S. Belongie. Linear embeddings in non-rigid structure from motion. In *Proc. CVPR'09*, pages 2427–2434, 2009.

[30] I. Rigoutsos. Homotopies: A panacea or just another method? Technical Report UR-CSD-TR-201, University of Rochester, 1986.

[31] H. Shimodaira. A shape-from-shading method of polyhedral objects using prior information. *PAMI*, 28(4):612–624, 2006.

[32] P. Sinha and E. Adelson. Verifying the "consistency" of shading patterns and 3-D structures. In *Proc. IEEE Workshop on Qualitative Vision*, pages 71–80, 1993.

[33] A. J. Sommese and C. W. Wampler. *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*. World Scientific, 2005.

[34] K. Sugihara. *Machine Interpretation of Line Drawings*. MIT Press, 1986.

[35] R. Szeliski. Fast shape from shading. In *Proc. ECCV'90*, pages 359–368, 1990.

[36] H. Waki, S. Kim, M. Kojima, and M. Muramatsu. Sums of squares and semidefinite program relaxations for polynomial optimization problems with structured sparsity. *SIAM J. on Optimization*, 17(1):218–242, 2006.

[37] L. T. Watson. Globally convergent homotopy methods: A tutorial. *App. Math. and Comp.*, 31:369–396, 1989.

[38] K. Q. Weinberger and L. K. Saul. Unsupervised learning of image manifolds by semidefinite programming. *IJCV*, 70(1):77–90, 2006.

[39] T. Weyrich, J. Deng, C. Barnes, S. Rusinkiewicz, and A. Finkelstein. Digital bas-relief from 3D scenes. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 26(3), 2007.

[40] J. Yang, N. Ohnishi, D. Zhang, and N. Sugie. Determining a polyhedral shape using interreflections. In *Proc. CVPR'97*, pages 110–115, 1997.

[41] S. Y. Yuen, Y. Y. Tsui, and C. K. Chow. A fast marching formulation of perspective shape from shading under frontal illumination. *Pattern Recog. Letters*, 28(7):806–824, 2007.