Incorporating Illumination Constraints in Deformable Models for Shape from Shading and Light Direction Estimation

Dimitris Samaras
Computer Science
State University of New York Stony Brook
Stony Brook, NY 11790-4400
samaras@cs.sunysb.edu

Dimitris Metaxas
Computer & Information Sciences
Rutgers University
Piscataway, NJ 08854-8019
dnm@cs.rutgers.edu

Abstract

In this paper we present a method for the integration of nonlinear holonomic constraints in deformable models and its application to the problems of shape and illuminant direction estimation from shading. Experimental results demonstrate that our method performs better than previous Shape from Shading algorithms applied to images of Lambertian objects under known illumination. It is also more general as it can be applied to non-Lambertian surfaces and it does not require knowledge of the illuminant direction. In this paper: (i) We first develop a theory for the numerically robust integration of nonlinear holonomic constraints within a deformable model framework. In this formulation we use Lagrange multipliers and a Baumgarte stabilization approach [2]. (ii) We also describe a fast new method for the computation of constraint based forces, in the case of high numbers of local parameters. (iii) We demonstrate how any type of illumination constraint, from the simple Lambertian model to more complex highly nonlinear models [27, 29] can be incorporated in a deformable model framework. (iv) We extend our method to work when the direction of the light source is not known. We couple our shape estimation method with a method for light estimation, in an iterative process, where improved shape estimation results in improved light estimation and vice versa. (v) We perform a series of experiments on both synthetic and real data. The synthetic data come from a standardized set of images [43]. Our results compare favorably with results of previous SfS algorithms on the same data and our light direction estimation to a previous method by Zheng and Chellapa[44].

Keywords: Physics-based modeling, shape from shading, deformable models, illuminant estimation, diffuse reflectance.

1. Introduction

The integration of visual cues within a physics-based deformable model framework has been attempted recently by several researchers [6, 26, 9] due to its potential for improved shape estimation. In all previous attempts, illumination constraints such as those appearing in the shape from shading problem, have never been considered. This is due to the nonlinear nature of the constraints and the fact that numerically robust methods for their use are required. In this paper, we provide a general methodology for the incorporation of illumination constraints within a deformable model framework and apply it to the coupled problems of object shape from shading (SFS) and light source estimation from images. We address the following five main issues: (i) Integration of nonlinear holonomic constraints in deformable models. (ii) Large systems of constraints on deformable models. (iii) Application to Shape from Shading under any reflectance model. (iv) Coupling of SFS and light source estimation. (v) Experimental verification.

We first develop a theory for the numerically robust integration of nonlinear holonomic constraints within a deformable model framework, regardless of their type. This theory amounts to the use of Lagrange multipliers and a Baumgarte stabilization approach [2] to allow for the robust integration of those constraints. This approach, which is a generalization of the previously developed methodology for linear holonomic constraints [26], allows the incorporation of illumination constraints into deformable models. Furthermore, we show how to handle large systems of constraints on physics-based models, by proposing a fast technique for the computation of constraint forces.

In particular, we demonstrate how any type of illumination constraint, from the simple Lambertian model to more complex highly nonlinear models [27, 29], can be incorporated in our physics-based modeling framework. Instead of extracting the shape parameters directly from these illumination constraints (which is not always possible), we use them to provide the necessary generalized forces that will deform our model and estimate the object's 3D shape. Our methodology obviates the need for commonly used approximations (e.g., linearization) to these equations, or the solution of partial differential equations requiring boundary conditions

[17]. In addition, the use of a deformable model-based approach allows the numerically robust computation of the required derivatives and produces improved experimental results. We show how the method can be used with either orthographic or perspective projection assumptions. We also demonstrate how singular point information and images obtained under perspective projection can be handled in this framework.

We use deformable models or grids with both global and local deformations [26, 25]. During shape estimation, we first fit the model's global parameters given the illumination constraints and then we refine its shape, based on the model's local deformations, using a coarse to fine grid. Use of a deformable model-based approach offers shape flexibility and the additional advantage of the numerically robust computation of the necessary derivatives, producing improved shape estimation results. For the SFS problem, the deformable model formulation enforces the smoothness of the solution, without the need of regularization.

Furthermore, we extend our method to work when the direction of the light source is not known. The quality of a model's fit to shading data strongly depends on knowledge of the lighting conditions. Fitting to an incorrect light will either cause the fitting process to not converge or will introduce additional error in the form of excessive wrinkling. We couple our shape estimation method with a method for light estimation, in an iterative process, where improved light estimation results in improved shape estimation and vice versa. Figure 1 shows two fitted models. In (b), the model is fitted with fixed light position, as estimated by the method of Zheng and Chellapa [44] which is clearly not sufficient for accurate model fitting. In (c), the model is fitted using the method described in this paper; re-estimation of the light position gives a clear improvement, and the error of fit reduces by 11.2 percent.

We have performed a series of experiments with real and synthetic data where we demonstrate the robustness of our method and the improved shape estimation results. The experiments consist of the standard test images used in the thorough comparative survey [43]. Our method outperforms all the SFS methods reviewed in [43], with a median improvement on the average reconstruction error of 45%. We also performed experiments where we estimate both the shape and the light direction. For the light direction part, we compare with the results of

the algorithm of [44].

The paper is organized as follows. Section 2 examines a representative sample of the huge amount of previous work on the topic. Section 3 presents the illumination models used in this work and Section 4 their integration to a deformable model framework. Section 5 discusses the coupled estimation of light direction and shape. Results of the methods described in this paper are presented in Section 6. Future work and conclusions are in Section 7. Appendix A describes generalizations for perspective projection and singular point information.

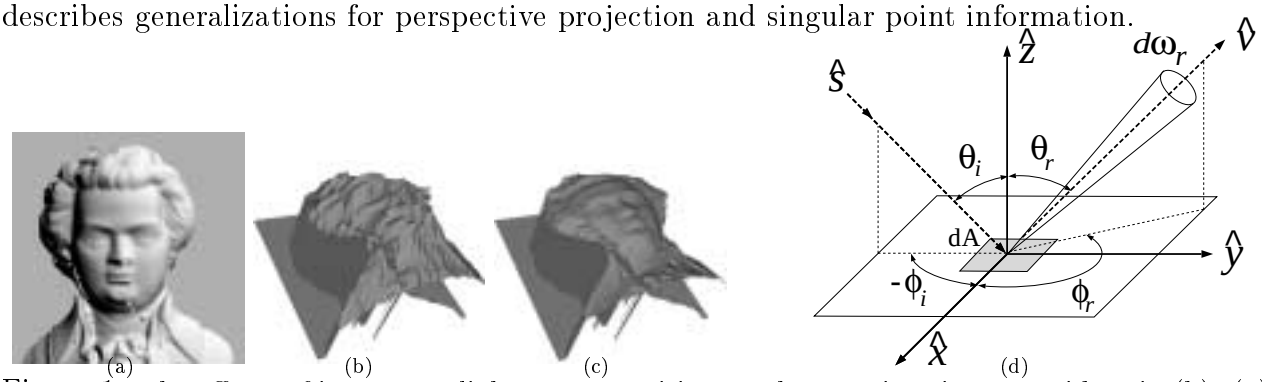


Figure 1: The effects of inaccurate light source position on shape estimation are evident in (b). (a) is the original image; (b) is the fitted model without light position re-estimation; (c) is the fitted model with the light position re-estimation method described in this paper. (d) Geometry used to define radiometric terms

2. Previous work

2.1. Shape from Shading

Most of the earlier work on Shape from Shading (SFS) is included in the book by Horn and Brooks[16]. The first comprehensive comparative study of a number of SFS algorithms was done by [43]. They classify SFS algorithms as either global or local depending on whether they use intensity information across the entire image or only in local neighborhoods.

Most SFS algorithms for Lambertian surfaces follow a regularization approach [17, 18, 38, 23, 21]. Other methods are based on the use of the integrability constraint [8, 15], the intensity gradient constraint [44] and the unit normal constraint. In the above class of approaches, the method of [21] requires good initial depth values, obtained from stereo information, and results in better estimation results compared to shape from shading alone. Later, in [10] an approach was proposed that combined stereo and shading, and could handle smoothly varying albedos.

Another class of algorithms [14, 28, 4] propagate height information from known points in the image. These methods require a priori information and their performance depends on the

accuracy of that information. Finally there are a number of methods that use local information to reconstruct depth. [30] and [22] use a rather restrictive local spherical approximation, while [31] and [39] use linear approximations of the reflectance function.

In this paper we will focus on single image methods. Although photometric stereo approaches [41, 1, 12] give in general better reconstructions than single view methods, there can be no motion in the scene and in the camera position while the images are taken. Other multi-view approaches allow for the integration of stereo and SFS [10], [36].

Most of the work on Shape from Shading has utilized the Lambertian reflectance model. [27] provides a hybrid model combining diffuse and specular reflectance. Later, [29] studied rough diffuse surfaces that do not follow Lambert's law, and, based on statistical modeling of surface geometry, they proposed a much more complex model of diffuse reflection.

As opposed to the previous methods on SFS, we present (initially in [34]) a model-based approach, where shape from shading is incorporated as a nonlinear hard constraint within a deformable model framework. More recently, [42] also used SFS as a hard constraint on the normals to recover normal direction (needle maps) of Lambertian surfaces. Our deformable model framework allows us to use the constraint on the normals to reconstruct plausible 3D shape. In the last few years there have been a number of promising approaches using level sets (a class of non-parametric deformable models) based on a viscosity framework [20, 33]. The general differences between parametric and non-parametric deformable models are discussed in [7]. The most recent and computationally efficient such method [20] which proposes an SFS solution for oblique light sources (although it is not clear how this generalizes to non-Lambertian reflectance), requires knowledge of all local minima, which can be hard to obtain for topologically complex surfaces, such as the ones used in the experiments presented here. This problem could be alleviated by a topology solver [19], but at the time of this research, there were no published examples of a complete implementation which could be compared with the results of our approach. An advantage of deformable model approaches, is that smoothness enforcement is conveniently embedded in the model. The shape flexibility offered by a deformable model improves the quality of shape estimation as is demonstrated by our

experimental results. The existence of a model underlying the reconstruction process allows us to estimate the light source direction without making binding general assumptions of the surface shape [35].

2.2. Illumination estimation

A useful discussion of the ambiguities involved in light source estimation can be found in [3]. A number of researchers have proposed methods for the estimation of the light source direction. [17] proposes an iterative method that updates both the shape and the illuminant direction at every iteration. To avoid local minima, a good initial state is often necessary, and furthermore, the requirement for a light source vector of unit length is not enforced. [22] used a Gaussian sphere model for the surface normal distribution and local spherical patches, but did not take shadowing effects into account. [44] presents the most sophisticated of the image based methods, considering shadowing effects and using a uniform distribution of the tilt and slant angles of surface normals. They still assume local spherical patches and their algorithm suffers on surfaces that deviate significantly from this assumption. [21] derives accurate light source information from surfaces reconstructed using stereo data.

2.3. Deformable Models: Geometry, Kinematics, Dynamics

In this section we review briefly the general formulation of deformable models [26, 25].

Geometrically, the models used in this paper are parameterized surfaces in space whose intrinsic parameters are $\mathbf{u} = (u, v)$, defined on a domain Ω. The positions $\mathbf{x}(\mathbf{u}, t)$ of points on the model relative to an inertial frame of reference Φ in space are given by $\mathbf{x} = \mathbf{c} + \mathbf{R}\mathbf{p}$, where $\mathbf{c}(t)$ is the origin of the model frame, ϕ, and $\mathbf{R}(t)$ is the rotation matrix expressing the orientation of ϕ. $\mathbf{p}(\mathbf{u}, t)$ denotes the positions of points on the model relative to the model frame. We introduce global and local deformations, by further expressing $\mathbf{p}$ as the sum of a reference shape $\mathbf{s}(\mathbf{u}, t)$ and a displacement function $\mathbf{d}(\mathbf{u}, t)$, i.e., $\mathbf{p} = \mathbf{s} + \mathbf{d}$.

For the applications in this paper, we define $\mathbf{s}$ as a geometric primitive parameterized in uv space (see [26, 25] for details and formulas). Local displacements $\mathbf{d}$ are computed based on the use of triangular finite elements [26, 25], which provide a tessellation of the deformable

model's surface. Associated with every finite element node i is a nodal vector variable $\mathbf{q}_{d,i}$. We collect all the nodal variables into a vector of local degrees of freedom $\mathbf{q}_d = (\ldots, \mathbf{q}_{d,i}^\top, \ldots)^\top$, and we compute the local displacement $\mathbf{d}$ based on finite element theory as $\mathbf{d} = \mathbf{S}\mathbf{q}_d$. $\mathbf{S}$ is the shape matrix whose entries are the finite element shape functions.

Our goal when fitting the model to visual data is to recover the vector $\mathbf{q}$ which expresses the model's degrees of freedom. The velocity of points on the model is given by [25],

$$\dot{\mathbf{x}} = \mathbf{L}\dot{\mathbf{q}}. \tag{1}$$

$$\mathbf{q} = (\mathbf{q}_c^\top, \mathbf{q}_R^\top, \mathbf{q}_s^\top, \mathbf{q}_d^\top)^\top \tag{2}$$

are the model parameters with $\mathbf{q}_c = \mathbf{c}$ and $\mathbf{q}_\theta$ the model's rotational degrees of freedom expressed as a quaternion, and $\mathbf{L}$ is a Jacobian matrix. Based on Lagrangian dynamics we make our model dynamic in $\mathbf{q}$, and we arrive at the motion equations:

$$\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{f}_q, \tag{3}$$

where $\mathbf{K}$ is the stiffness matrix, (see [25] for definitions). $\mathbf{f}_q$ are the generalized external forces computed from the 3D forces which in our application will be computed from edges and the brightness constraint.

3. Illumination Models

We will now present the lighting models we will use in SFS and we will reformulate them so that they can be introduced as a constraint in a deformable model framework.

3.1. Lambertian Model

This is the simplest lighting model; it models an ideal diffuser. The amount of light reflected from the surface that reaches the viewer depends only on the orientation of the surface relative to the light source. If we assume a point light source at infinity, and orthographic projection along an axis parallel to the optical axis of the observer, the scene radiance is expressed as

$$I_L = B\frac{\rho}{\pi}\cos\theta = B\rho'\ \hat{\mathbf{s}} \cdot \hat{\mathbf{n}}, \tag{4}$$

where θ is the angle between the surface unit normal vector, $\hat{\mathbf{n}} = \mathbf{n}/\|\mathbf{n}\|$, and the unit light source direction vector $\hat{\mathbf{s}}$. B is the strength of the light source and ρ is the constant albedo of the surface, which represents the fraction of the incident light to the surface that is reflected. $\rho' = \rho/\pi$ is the bidirectional reflection distribution function, which is a constant in this model.

The above image irradiance equation (4) can be rewritten as a brightness constraint:

$$C = \hat{\mathbf{s}} \cdot \mathbf{n} - I'_L \|\mathbf{n}\| = 0, \tag{5}$$

where $I'_L = I_L/(B\rho')$, and the values of I'_L range between 0 and 1. In our application, we want to recover the shape parameters based on the intensity information at m points in the image (corresponding to m points on the model) and therefore we will have an m-component constraint vector $\mathbf{C} = [C_1, C_2, \ldots, C_m]^\top$.

$\mathbf{C}$ is a nonlinear constraint with respect to the model parameters, which in the traditional SFS formulations resulted either in nonlinear first-order differential equations (PDE's) or in attempts to linearize the constraint [31, 23, 39]. PDE's require appropriate boundary conditions [17] that are often not available, whereas linearization introduces additional error. Instead, we incorporate the above brightness constraint as a nonlinear holonomic constraint in the deformable model framework, presented in a following section. This incorporation depends on the computation of the Jacobian $\mathbf{C_q}$ of the constraint vector $\mathbf{C}$, with respect to the model parameters $\mathbf{q}$.

$$\mathbf{C_q} = \hat{\mathbf{S}}\mathbf{N_q} - \mathbf{L_{N_q}}\mathbf{I'_L} - \mathbf{J}_{I'_L}\mathbf{L}\|\mathbf{N}\|. \tag{6}$$

(6) is derived by differentiating, with respect to $\mathbf{q}$, each constraint given by (5). Matrices $\mathbf{N}$, $\hat{\mathbf{S}}$, $\mathbf{I'_L}$ and $\|\mathbf{N}\|$ are formed by collecting the values of $\mathbf{n}$, $\hat{\mathbf{S}}$, I'_L, $\|\mathbf{n}\|$, respectively, for each component of $\mathbf{C}$. $\mathbf{N_q} = \partial\mathbf{N}/\partial\mathbf{q}$, $\mathbf{L_{N_q}} = \partial\|\mathbf{N}\|/\partial\mathbf{q}$.

Assuming orthographic projection, the sampled image values $I'_L(x, y)$ are parameterized by x and y, the coordinates of the sampled model points with positions $\mathbf{x} = (x, y, z)^\top$. We utilize the chain rule to calculate the derivatives of $I'_L(x, y)$, with respect to $\mathbf{q}$.

$$\frac{\partial \mathbf{I'_L}(x, y)}{\partial \mathbf{q}} = \frac{\partial \mathbf{I'_L}(x, y)}{\partial \mathbf{x}} \frac{\partial \mathbf{x}(x, y, z)}{\partial \mathbf{q}}, \tag{7}$$

so, in (6) $\mathbf{J}_{I'_L} = \partial \mathbf{I}'_{\mathbf{L}}(x,y)/\partial \mathbf{x}$ and $\mathbf{L} = \partial \mathbf{x}(x,y,z)/\partial \mathbf{q}$ is the Jacobian of the model points, described in (1). Further in this paper we will extend our approach to perspective projection.

Equation (6) requires the computation of the surface normal $\mathbf{n}$ and its Jacobian $\mathbf{n_q} = \partial \mathbf{N}/\partial \mathbf{q}$. Since we have a parametric representation for our deformable model (with intrinsic parameters (u,v)), $\mathbf{s}(u,v) = [x(u,v), y(u,v), z(u,v)]^\top$, the surface tangents and the surface normal are

$$\frac{\partial \mathbf{s}}{\partial u} = \left[\frac{\partial x}{\partial u}, \frac{\partial y}{\partial u}, \frac{\partial z}{\partial u}\right]^\top, \quad \frac{\partial \mathbf{s}}{\partial v} = \left[\frac{\partial x}{\partial v}, \quad \frac{\partial y}{\partial v}, \frac{\partial z}{\partial v}\right]^\top, \mathbf{n} = \frac{\partial \mathbf{s}}{\partial v} \times \frac{\partial \mathbf{s}}{\partial u}.$$

The fact that our approach is model-based offers greater numerical stability with respect to previous non-model-based methods in the computation of the above derivatives, which are very sensitive to noise in the data. This will lead to improved experimental results as will be demonstrated in the experiment section.

3.2. More Complex Illumination Models

The Lambertian model is simple, but not very accurate for most diffuse surfaces. However it is widely used since it is easy to invert because of its simplicity. Models that try to describe more complex reflection phenomena quickly become mathematically hard to work with. Our approach allows us to use alternative formulations for the brightness constraint, even if they are not invertible with respect to the normal, as long as the normal, or tangents to the surface, appear in the expression that describes the radiance of the surface.

As an example, a highly complex lighting model that can be accommodated in our formulation is the one proposed by Oren and Nayar [29], to model more accurately the reflectance of rough diffuse surfaces, such as sand and plaster. In that model, the diffuse radiance of the surface is dependent on both the viewing and illumination directions. Viewing direction at each point is defined with polar and azimuthal angles ϕ_r and θ_r with respect to the normal and illumination direction with ϕ_i and θ_i, as shown in Figure 1(d). The radiance of the surface is expressed as $L_d = L_1 + L_2$, where the approximations for these components are

$$L_1 = \frac{\rho}{\pi} E_0 \cos\theta_i [C_1(\sigma) \tag{8}$$

$$+\cos(\phi_r-\phi_i)C_2(\alpha;\beta;\phi_r-\phi_i;\sigma)\tan\beta$$
$$+(1-|\cos\phi_r-\phi_i|)C_3(\alpha;\beta;\sigma)\tan(\frac{\alpha+\beta}{2})]$$

$$L_2=\frac{0.17\rho^2\sigma^2}{\pi(\sigma^2+0.13)}E_0\cos\theta_i[1-\cos(\phi_r-\phi_i)(\frac{2\beta}{\pi})^2], \tag{9}$$

where ρ is the albedo, $E_0\cos\theta_i$ is the irradiance, $\alpha=\max[\theta_r,\theta_i]$ and $\beta=\min[\theta_r,\theta_i]$. The coefficients of the model are $C_1=1-0.5\frac{\sigma^2}{\sigma^2+0.33}$, $C_2=0.45\frac{\sigma^2}{\sigma^2+0.09}\sin\alpha$, if $\cos(\phi_r-\phi_i)\geq 0$, and $C_2=0.45\frac{\sigma^2}{\sigma^2+0.09}(\sin\alpha-(\frac{2\beta}{\pi})^3)$ otherwise, and $C_3=0.125\frac{\sigma^2}{\sigma^2+0.09}(\frac{4\alpha\beta}{\pi^2})^2$, where σ is the surface roughness parameter.

Despite the complexity of the model, it can still be differentiated with respect to the surface normal, since the angles that parameterize it can be converted to equivalent dot product expressions of the normal. If $\hat{\mathbf{s}}$ is the light source direction, $\hat{\mathbf{v}}$ the viewing direction and $\hat{\mathbf{n}}$ the surface normal, $\cos\theta_i=\hat{\mathbf{s}}\cdot\hat{\mathbf{n}}$, $\cos\theta_r=\hat{\mathbf{v}}\cdot\hat{\mathbf{n}}$ and $\cos(\phi_r-\phi_i)=(\hat{\mathbf{s}}-\hat{\mathbf{n}}\cos\theta_i)\cdot(\hat{\mathbf{v}}-\hat{\mathbf{n}}\cos\theta_r)$. Thus, if I_{im} is the measured image value, we can form the constraint $C_D=L_d-I_{im}$ and use it in the same way as in the case of a Lambertian model. Two of the examples in the experiment section were fitted using this model.

4. Model-Based Shape Estimation from Shading

In this section we present the mathematical framework and the implementation details for model deformation driven by illumination constraints. Furthermore we introduce a method for fast computation of constraint forces for a large number of local deformations.

4.1. Integration of Lighting Constraints in Deformable Models

Here, we present our method for incorporating the above nonlinear illumination constraints, $\mathbf{C}$, in a deformable model framework. This constraint vector, of the form $\mathbf{C}(\mathbf{q},t)=0$, is nonlinearly related to the model parameters $\mathbf{q}$; constraints of this type are termed *nonlinear holonomic constraints*. Once a constraint is satisfied, its derivative must remain zero, for the constraint to remain satisfied. We incorporate these constraints in a deformable model formulation, using the method of Lagrange multipliers. The case of linear non-holonomic

constraints was treated in [6], and the case of linear holonomic constraints was treated in [25].

Lagrange Multipliers. In order to recover the shape parameters based on the constraint information at m points of our model, we will have an $m \times 1$ constraint vector $\mathbf{C}$, which is incorporated in (3) based on the theory of Lagrange multipliers as follows:

$$\dot{\mathbf{q}} = \mathbf{f}_q - \mathbf{K}_d\mathbf{q} - \mathbf{C}_\mathbf{q}^\top\boldsymbol{\lambda}, \tag{10}$$

where $\mathbf{f}_q$ are point and edge based generalized forces, $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_m]^\top$ are the Lagrange multipliers and $\mathbf{C}_\mathbf{q}$ is the Jacobian matrix of the constraints $\mathbf{C}$ with respect to the shape parameters. We can consider $-\mathbf{C}_\mathbf{q}^\top\boldsymbol{\lambda}$ to be the vector of generalized forces on the model parameters due to the illumination constraint.

Since (10) has fewer equations than unknowns, to obtain the additional necessary equations we differentiate the constraint equation with respect to time $\dot{\mathbf{C}}(\mathbf{q}, t) = 0$ yielding $\mathbf{C}_\mathbf{q}\dot{\mathbf{q}}+\mathbf{C}_t = 0$. In our application $\mathbf{C}$ is not directly dependent on time, therefore $\mathbf{C}_t = 0$, which results in

$$\dot{\mathbf{C}}(\mathbf{q}, t) = \mathbf{C}_\mathbf{q}\dot{\mathbf{q}} = 0. \tag{11}$$

Baumgarte Stabilization The above Lagrange multiplier approach works well in practice when the constraints are almost satisfied initially. However, since we will be fitting a deformable model to the data, these constraints will be far from being satisfied initially, and therefore we will use Baumgarte's [2] stabilization method, taking into account that our constraint is nonlinear. This approach is a generalization of the previously developed methodology for linear holonomic constraints [26]. Based on this method we replace $\dot{\mathbf{C}}(\mathbf{q}, t) = 0$ with the following constraint equation

$$\dot{\mathbf{C}} + \alpha\mathbf{C} = 0, \tag{12}$$

where α is a stabilization factor. Any numerical error that causes the constraint to be violated will be damped out automatically. Although larger values of α can cause greater stabilization, the stabilizing term must not become the dominant term in the differential equation, as that

would introduce numerical stiffness in the equation [40]. In our implementation, we have found $\alpha = 0.5$ worked well in all cases. Based on (11), (12) becomes

$$\mathbf{C_q}\dot{\mathbf{q}} + \alpha\mathbf{C} = 0. \tag{13}$$

Putting it all together: Using (10), (13) becomes

$$\mathbf{C_q}(\mathbf{f} - \mathbf{K}_d\mathbf{q} - \mathbf{C_q^\top}\boldsymbol{\lambda}) + \alpha\mathbf{C} = 0, \tag{14}$$

which we can solve for $\boldsymbol{\lambda}$ and obtain

$$\boldsymbol{\lambda} = (\mathbf{C_q}\mathbf{C_q^\top})^{-1}(\alpha\mathbf{C} + \mathbf{C_q}(\mathbf{f} - \mathbf{K}_d\mathbf{q})). \tag{15}$$

However, we do not explicitly compute the Lagrange multipliers. Instead, we substitute (15) into (10). If we define the vector of all non-constraint (generalized) forces, as $\mathbf{b} = \mathbf{f} - \mathbf{K}_d\mathbf{q}$ then (10) becomes

$$\dot{\mathbf{q}} = \mathbf{b} - \mathbf{C_q^\top}(\mathbf{C_q}\mathbf{C_q^\top})^{-1}(\alpha\mathbf{C} + \mathbf{C_q}\mathbf{b}). \tag{16}$$

The matrix $\mathbf{C_q^+} = \mathbf{C_q^\top}(\mathbf{C_q}\mathbf{C_q^\top})^{-1}$ is the pseudo-inverse of the matrix $\mathbf{C_q}$, and using this notation in the previous equation we get

$$\dot{\mathbf{q}} = \mathbf{b} - \mathbf{C_q^+}(\alpha\mathbf{C} + \mathbf{C_q}\mathbf{b}) = -\mathbf{C_q^+}\alpha\mathbf{C} + (\mathbf{I} - \mathbf{C_q^+}\mathbf{C_q})\mathbf{b}. \tag{17}$$

The first term $-\mathbf{C_q^+}\alpha\mathbf{C}$ in (17) is a model-based least-squares solution to the Baumgarte lighting constraint equations of (12). Remembering that the SVD solution of a linear system spans the "range" subspace of that system [37], we notice that in $(\mathbf{I} - \mathbf{C_q^+}\mathbf{C_q})\mathbf{b}$, the second term of (17), the expression $\mathbf{C_q^+}\mathbf{C_q})\mathbf{b}$ projects the non-constraint forces $\mathbf{b}$ to the space of the constraint forces, thus canceling the part that violates the lighting constraint.

4.2. Implementation Details

In our implementation, we fit dynamically a deformable surface model of the type described in the previous section. In the model fitting procedure we first estimate the model's global degrees of freedom to obtain a first approximation of the underlying surface. In order to

1. Initialize data structures
2. **Repeat** for *global* model parameters
   2.1. Calculate non-constraint (stiffness, external) forces $\mathbf{b}$
   2.2. Calculate constraint matrix $\mathbf{C}$ and its Jacobian $\mathbf{C_q}$
   2.3. Calculate pseudoinverse $\mathbf{C_q^+}$ by SVD
   2.4. Evaluate change in parameters $\dot{\mathbf{q}}$ by Eq. 17
   2.5. Update model parameters
   until *global* model parameters $||\dot{\mathbf{q}}|| < \epsilon$
3. **Repeat** for *local* model parameters
   3.1. Calculate non-constraint (stiffness, external) forces $\mathbf{b}$
   3.2. Calculate constraint matrix $\mathbf{C}$ and its Jacobian $\mathbf{C_q}$
   3.3. Calculate pseudoinverse $\mathbf{C_q^+}$ by method of Sec. 4.3
   3.4. Evaluate change in parameters $\dot{\mathbf{q}}$ by Eq. 17
   3.5. Update model parameters
   until *local* model parameters $||\dot{\mathbf{q}}|| < \epsilon$

Figure 2: Steps of the model parameter estimation algorithm

estimate the details of the model's surface we use local deformations. We discretize our model using triangular finite elements with C^0 continuity, as described in [25]. For efficiency, we use data points that coincide with the barycenter of each element. The generation of the finite element grid is done in a coarse to fine fashion. We start with a few elements and we progressively subdivide them in a regular fashion to get finer meshes until the error of fit does not change. This results in significantly faster convergence. In addition, as our model moves closer to the solution, we lower the model's elasticity by decreasing the relative coefficients, in order to refine surface details.

Fitting with global deformations converges faster than fitting with local ones. While achieving interactive rate performance, global deformations provide only gross shape estimation. To avoid loss of detail, when the error rate stops decreasing fitting continues with local deformations, whose rate of convergence is slower (several minutes) and depends on the desired accuracy of approximation and the complexity of the surface. Furthermore the use of global model parameters can help abstract geometric primitives, which could prove useful for the classification of shading patterns. These applications are beyond the scope of this paper.

The pseudo-inverse of $\mathbf{C_q}$ in (17) is computed through singular value decomposition [32]. Since the number of sampled points is generally greater than the number of global parameters, the system is over-constrained and our solution is equivalent to a least squares error solution.

4.3. Fast Computation of Constraint Forces

In the above method we calculate the pseudoinverse $\mathbf{C_q^+} = \mathbf{C_q^\top}(\mathbf{C_q}\mathbf{C_q^\top})^{-1}$ in Equation (17) using Singular Value Decomposition (SVD). In a high resolution mesh with local deformations only, which has a few thousand nodal parameters, the computational cost of SVD becomes prohibitive, as SVD cannot take advantage of the sparsity of the constraint Jacobian matrix. Approximate SVD techniques which calculate only the most significant singular values, are not useful when there are no global parameters to dominate the shape description. Initially we tried applying the SVD only on a very coarse resolution level, since it was very slow on meshes with more than 200 or so parameters. Then, when subdividing the mesh into finer grids, we would compute the solution to the constraint locally only per element and then add up the contributions. This had the effect of applying a soft constraint instead. Still it worked acceptably well for orthographic images, even though convergence was slow, and test results outperformed other methods as we reported in [34].

Here we propose a new technique for the case of large deformable meshes with only local degrees of freedom. Motivated by the desire to use sparse matrix inversion methods, we notice an alternative formulation of (17) as $\dot{\mathbf{q}} = \mathbf{b} - (\mathbf{C_q^\top}\mathbf{C_q})^{-1}\mathbf{C_q^\top}(\alpha\mathbf{C} + \mathbf{C_q}\mathbf{b})$. $(\mathbf{C_q^\top}\mathbf{C_q})$ is an $n \times n$ matrix where n is the number of nodes in the mesh. Here, without loss of generality, we assume that we can only update the z coordinate of each node in order to keep the matrices smaller. We also assume that every active element has at least one constraint associated with it. Since the system is over-constrained [34], the rank of this matrix is the rank of $\mathbf{C_q}$, which is the number of degrees of freedom in the orientation of the facets of the object. In the case where all the parameters are local nodal variables, possible degeneracies in this matrix can be caused only by nodes in the boundaries of the mesh which belong to only one triangle. Such nodes will correspond to columns of $\mathbf{C_q}$ that have only one non-zero entry. The reason is that the orientation of a triangle with such a node can already be determined by its other two nodes which belong to other triangles also. By excluding these nodes from $\mathbf{C_q}$ we guarantee that the dimensionality of $(\mathbf{C_q^\top}\mathbf{C_q})$ does not exceed the dimensionality of the mesh and thus $(\mathbf{C_q^\top}\mathbf{C_q})^{-1}$ exists, and can be calculated with LU decomposition, faster than SVD.

Furthermore, the matrix is highly sparse and thus sparse matrix routines can speed up computation significantly [13]. If we have a regularly sampled mesh the maximum distance of non-zero entries in the matrix will be on the order of $\sqrt{n}$ around the diagonal of the matrix, making banded matrix routines attractive. We compared the use of banded matrix inversion routines to the iterative biconjugate gradient sparse matrix methods and found them equivalent in terms of speed (that could be possibly improved with further study of appropriate preconditioners) [32]. However, conjugate gradient methods do not compute the exact solution, leaving a small residual. As a result, updates to $\dot{\mathbf{q}}$ contain a small error, which in general increases the number of iterations that are required before the constraints are satisfied. Since our implementation is based on regularly sampled meshes, we opted for the use of banded matrix routines. This is a design choice that could be different in situations where the ordering of the nodes of the mesh is not as regular.

5. Coupling Light and SFS Estimation

One of the requirements of the method presented above is that the light direction is exactly known. This knowledge is rarely supplied in real world situations. Instead we might have a rough estimate (which can often be obtained by image statistics). In this section we will describe how to extend the method described above in order to achieve the simultaneous estimation of the light source direction and the object shape from shading information. This is done by iteratively improving the estimates of the shape and of the light vector.

5.1. Overview

In monocular orthographic images of Lambertian surfaces, there is inherent ambiguity in the configuration of light and surface shape for the generation of images; in the most general case there is an infinite number of combinations of surface shape and albedo on one hand and light position and intensity on the other [3]. However, considering only surfaces of constant and known albedo, it has been shown that for every image point there are two possible configurations for the normal and the light source [17, 3]. More specifically, if the constraint equations $\mathbf{C}$ are satisfied by shape A with normals $\hat{\mathbf{n}}_A$ under light source $\hat{\mathbf{s}}_A$, then they are also

satisfied by a dual shape B with normals $\hat{\mathbf{n}}_B$ under light source $\hat{\mathbf{s}}_B$ with $\hat{\mathbf{n}}_B = 2\hat{\mathbf{v}}(\hat{\mathbf{n}}_A\hat{\mathbf{v}}) - \hat{\mathbf{n}}_A$ and $\hat{\mathbf{s}}_B = 2\hat{\mathbf{v}}(\hat{\mathbf{s}}_A\hat{\mathbf{v}}) - \hat{\mathbf{s}}_A$, where $\hat{\mathbf{v}}$ is the viewer direction. When the viewing direction is perpendicular to the image plane, this relation preserves unit lengths. This is the standard *in-out* ambiguity which can often be resolved by taking shadow information into account. Our goal is to estimate either the pair $(\hat{\mathbf{n}}_A, \hat{\mathbf{s}}_A)$ or the pair $(\hat{\mathbf{n}}_B, \hat{\mathbf{s}}_B)$.

Our method is based on the following observation: The constant albedo assumption and the integrability constraint on the shape of the model surface, (i.e., the requirement for a physically plausible shape), which is implicitly imposed by our deformable model, make the image plausible under only one light vector (and its dual)[3]. Therefore, any fitting process, unless we use the correct light vector (or its dual), will not satisfy the integrability constraint by modifying only the surface normals. This is due to the two sources of error: the incorrect normals and the incorrect light position.

In the beginning of the fitting, while the model surface is still smooth and the model is stiff, mostly gross-scale shape features will emerge. These features (or their duals) will be roughly correct under most light source configurations (or their duals), so they will result into an improved estimate for the correct source. That increases our hope that if we re-estimate the light at this point, we will move closer to the true light vector. We can then fit the model to the new light source, re-estimate the light position and iterate this process until convergence.

An initial formulation of the light and shape estimation problem would be to include the light parameters in the deformable model framework. Although the light source is physically independent from changes in the deformable model's shape parameters, (they do not appear in equations (2) and (3)) they can still be estimated using the parameter update rule of (10) through the constraint Jacobian matrix $\mathbf{C_q}$. We can include the illuminant direction parameters in the parameter vector $\mathbf{q}$ to obtain $\mathbf{q\prime} = (\mathbf{q}^\top, \theta, \phi)^\top = (\mathbf{q}_c^\top, \mathbf{q}_R^\top, \mathbf{q}_s^\top, \mathbf{q}_d^\top, \theta, \phi)^\top$ and proceed in the same manner as in section (4.1). This approach would have the problem of coupling the estimation of global parameters (such as the light parameters) and local parameters. This coupling is usually not very efficient; for example the light parameters need not be updated at every step, together with the shape parameters. It also makes the methods

used for light estimation and for shape estimation dependent on the choices of each other. Instead we opt for a method where light estimation and shape estimation are independent steps, and each one can be implemented in the most appropriate way. Hence, we introduce a process consisting of an initial step and an iterative two step procedure:

1. Obtain initial light estimate
2. **Repeat**
   2.1. Fix light and estimate shape by method of Sec. 4
   2.2. Fix shape and estimate light by Levenberg-Marquart method

   until no more change in light estimate

5.2. Initial light estimate

In the initial step we fit the model under a crude estimate of the light source which we obtain using the image based method described in Zheng and Chellapa [44]. This method estimates the light source based on image statistics, under certain assumptions on the probability distribution of the surface normals. These assumptions are general and do not take into account any already existing knowledge about the model shape. Therefore we cannot use it to take advantage of any improvement in the model shape estimates. However it is useful in the initial step given the absence of any shape information; in most of the images we tested it on, it estimated a light direction vector forming an angle of less than $\pi/4$ from the correct solution. As will be seen in the experiments section this is good enough for our algorithm. Since we want the light source vector to be of unit length, we only need to estimate the slant θ (the angle between the illuminant and the positive z axis) and the tilt ϕ (the angle between the illuminant and the $x-z$ plane). Therefore $\hat{\mathbf{s}} = (\cos\phi\sin\theta, \sin\phi\sin\theta, \cos\theta)$. In Figure 1(d), the slant is angle ϕ_i and the tilt is angle θ_i. For the estimation of ϕ, and $\tilde{x}_L, \tilde{y}_L$ (the x and y components of the local estimate of ϕ) [44] uses the formulas

$$\phi = \arctan\left(\frac{E_{x,y}\left\{\frac{\tilde{y}_L}{\sqrt{\tilde{x}_L^2+\tilde{y}_L^2}}\right\}}{E_{x,y}\left\{\frac{\tilde{x}_L}{\sqrt{\tilde{x}_L^2+\tilde{y}_L^2}}\right\}}\right), \quad \begin{bmatrix} \tilde{x}_L \\ \tilde{y}_L \end{bmatrix} = \mathbf{B}^{+} \begin{bmatrix} \delta I_1 \\ \vdots \\ \delta I_N \end{bmatrix}, \quad \text{with } \mathbf{B} = \begin{bmatrix} \delta x_1 & \delta y_1 \\ \vdots & \vdots \\ \delta x_N & \delta y_N \end{bmatrix}, \quad (18)$$

where $E\{\}$ are ensemble averages over the local estimates. $\mathbf{B}^+$ is the pseudoinverse of $\mathbf{B}$. N is the number of local image derivatives δI. In our implementation $N = 8$.The slant θ can be determined numerically from the monotonically decreasing function

$$f(\theta) = \frac{E_{x,y}\{I\}}{E_{x,y}\{I^2\}} \tag{19}$$

5.3. Two Step Iteration

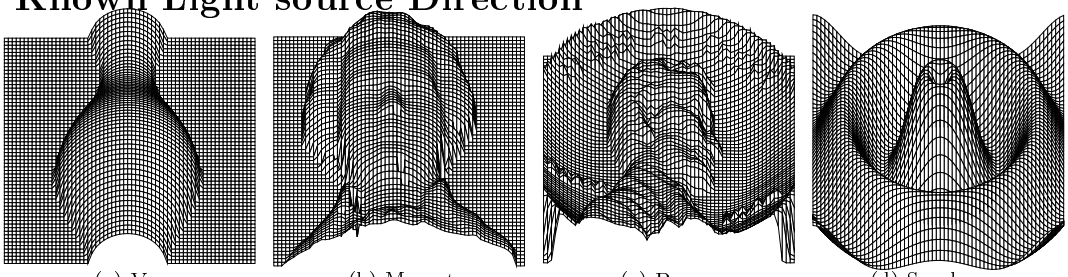
Our method for illumination and shape estimation proceeds in the following way: In the first iteration of the iterative process we fit the model to the image data using the light source direction estimated by the above initial step. Using the method described in the previous section, with high stiffness coefficients ($w_0 = w_1 = 0.05$ in our experiments) we compute an initial fit of the shape. The model parameters are updated using (17). It should be noted that the iterative method does not depend on a particular method for the initial estimate as long as that is not too far from the truth (in our experiments initial estimates less than 45 degrees off the true vector worked well).

Based on the fitted shape, we can now re-estimate the light direction. This time we want to estimate the source direction, taking into account the surface shape. We want the closest estimate to the source direction which, given the current shape would produce an image as close as possible to the original. We define "closeness" in a least squares sense, in the absence of any general knowledge of which parts of the image are more conforming to the illumination model than others. There could be more powerful error metrics for applications where more information is available. We cannot use linear least squares to estimate the three components of the light source independently, because the resulting vector would not be of unit length. For this reason we need the non-linear (ϕ, θ) parameterization. We use the Levenberg-Marquart method [24, 32], a standard non-linear least-squares estimation technique, to update the values of ϕ and θ. Levenberg-Marquart smoothly combines an inverse-Hessian method and a steepest descent method, with the objective of minimizing all the Jacobian of the error with respect to the model's parameters. The method approximates its solution rapidly, but then can wander

around the true minimum. So we stop iterating when the error changes less than a chosen threshold (less than 10^{-3}). In our experiments this typically happens within a few iterations (usually 20 or less). Once we have re-estimated the model's parameters, we repeat the model fitting stage under the new light source. We continue iterating between these two steps until the light stabilizes. This in our examples happens within fifteen to thirty iterations. At that point we decrease (in decrements of 10 percent) the model's stiffness parameters, allowing it to minimize the lighting constraint error.

6. Results

6.1. Known Light source Direction



(a) Vase (b) Mozart (c) Penny (d) Sombrero

Figure 3: Original depth maps used to produce synthetic images

In this section we present results of our shape-estimation method, when the light source direction is known. We present experiments both on synthetic and real images. The test set consists of five synthetic data sets and five real images.

In all the experiments, a time step of 0.01 is the Euler integration step used in the iterative fitting of the model, and the magnitude of the constraint forces acting on the model is weighted by a factor of 15. We discretize the model to 32×32 nodes for the global parameter fitting and the first iterations of the local parameter fitting. As the fitting proceeds, the mesh size is refined to 64×64 and finally to 128×128 (and in the case of the Mozart images up to 192×192). Global parameters and their deformations capture the shape well enough in low resolution grid levels, so we only start using local deformations at higher grid resolutions. Final depth per pixel is generated by interpolation inside the mesh triangle that the pixel projects to.

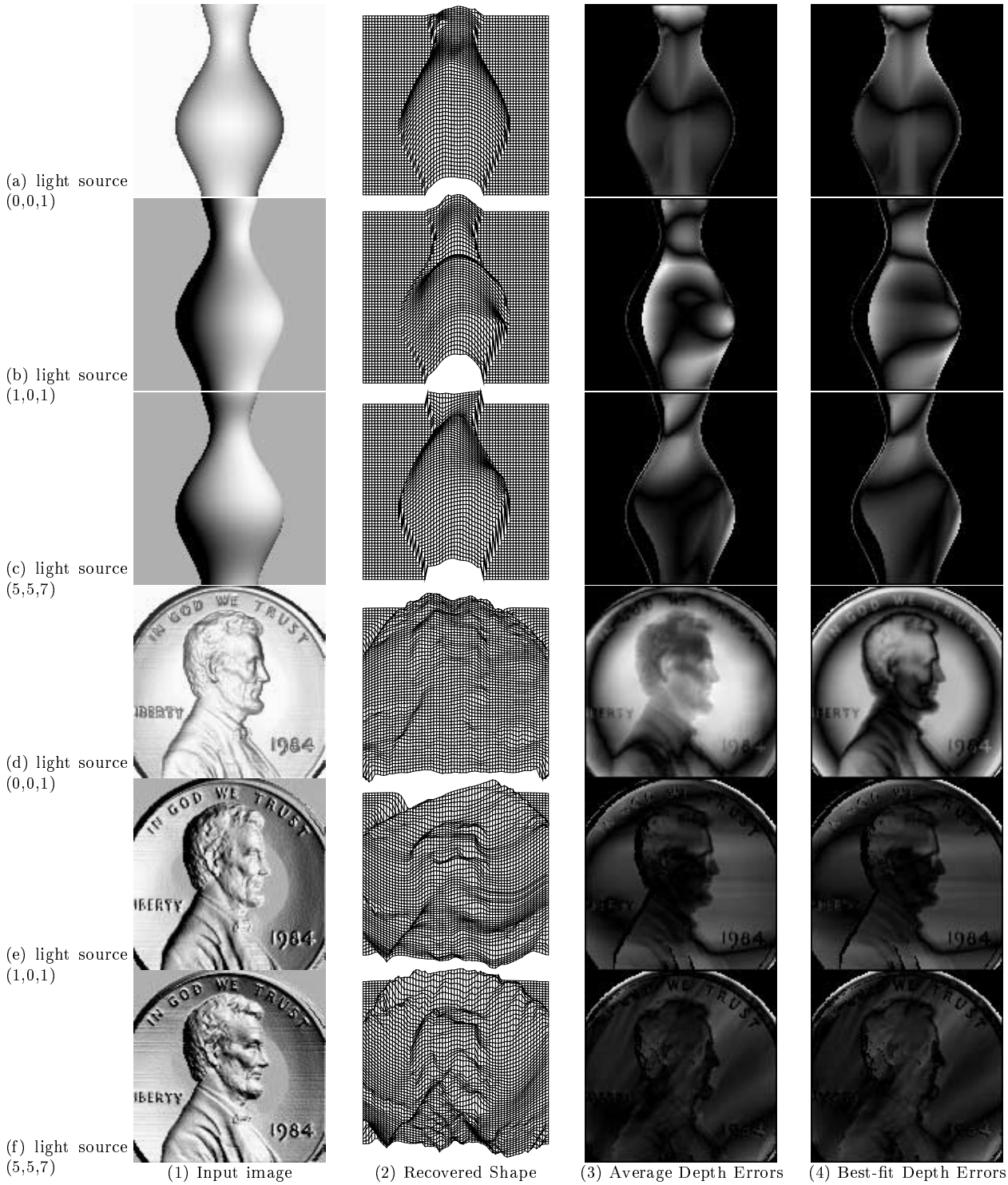


Figure 4: Experiments on 3 synthesized images of Vase and 3 of Penny datasets. The 3-D data were rendered under 3 different light directions: (0,0,1), (1,0,1), (5,5,7). Column(1): input images, Col.(2): recovered surfaces, Col.(3): absolute depth error images, with recovered and true surfaces aligned in the same range of depth, Col.(4): absolute depth error images, with recovered and true surfaces aligned in a "closest fit" manner. Higher intensity means higher error. Error images have all been normalized in the 0-255 range, so only comparisons within the same image are meaningful.

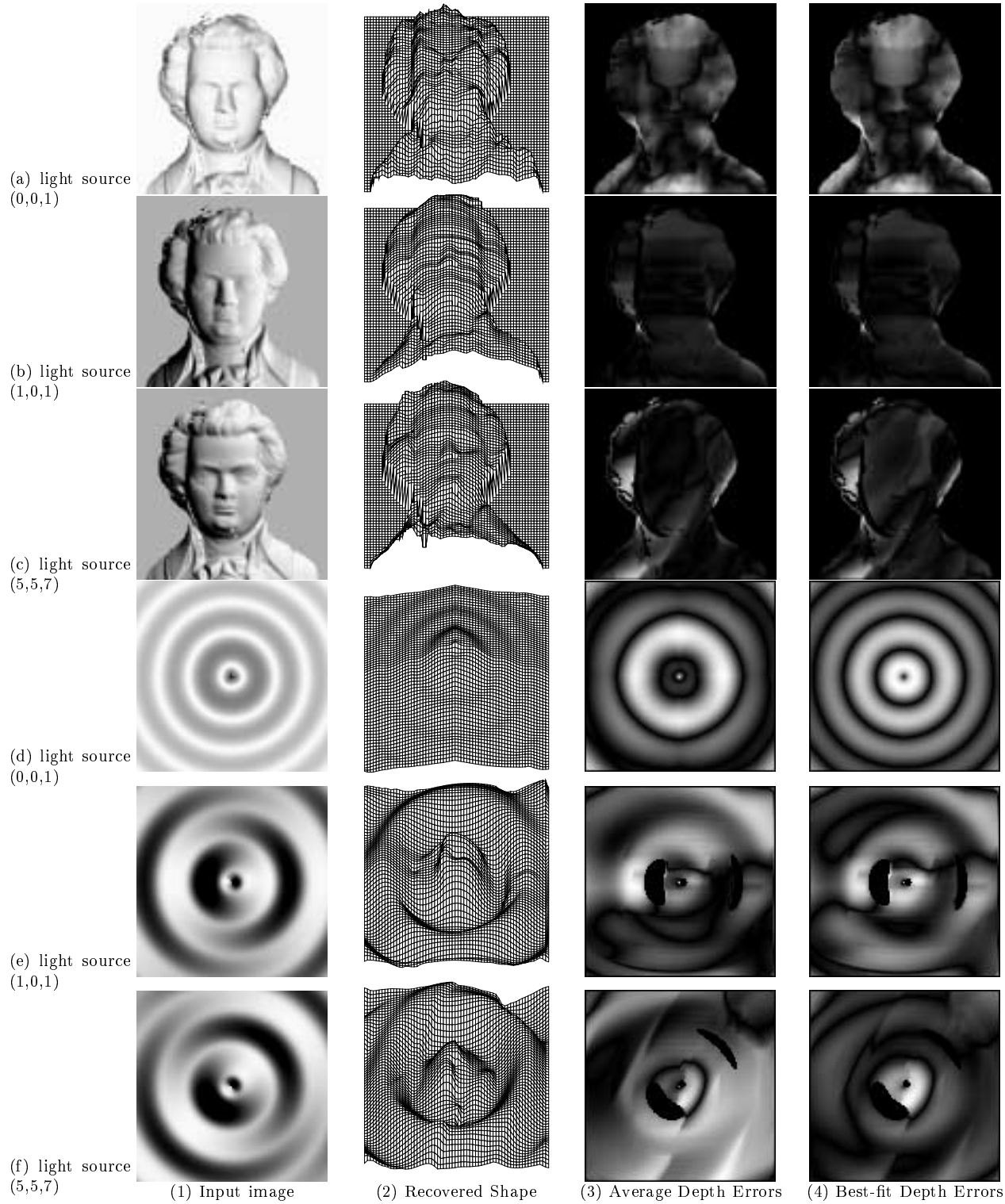


Figure 5: Experiments on 3 synthesized images of Mozart and 3 of Sombrero datasets. The 3-D data were rendered under 3 different light directions: (0,0,1), (1,0,1), (5,5,7). Column(1): input images, Col.(2): recovered surfaces, Col.(3): absolute depth error images, with recovered and true surfaces aligned in the same range of depth, Col.(4): absolute depth error images, with recovered and true surfaces aligned in a "closest fit" manner. Higher intensity means higher error. Error images have all been normalized in the 0-255 range, so only comparisons within the same image are meaningful.

6.1.1. Synthetic images

For the synthetic examples, we have three images for each data set, generated with the following light source vectors: In the first image of each set the direction of the light source is (0,0,1), in the second it is (1,0,1), and in the third it is (5,5,7). We performed experiments on each of the synthetic images. The five data sets that the images were rendered from where used to test the algorithms that are included in the review of SFS algorithms in [43], so that we could compare the performance of our algorithm with some of the standard algorithms in the field. We present detailed results on twelve images generated on four of those datasets. We have not included detailed results on the Sphere dataset, as the sphere is fitted very well with global deformations only. The recovered global shape parameters are within 98 percent of the true parameters. In Figure 3 we have the true height maps used to generate the synthetic test images. To the right of each image in the left columns of Figures 4 and 5 (illuminated under the light vectors described above) are the reconstructed height maps. For uniformity purposes for the results reported in these figures, the initial surface was the same in all the test cases, a flat deformable mesh with 128×128 resolution in most cases and in the case of the Mozart images (which are larger) 192×192. These resolutions describe the whole mesh. In most of the test images, because of the various irregular shapes of the objects, substantial parts of the mesh covered non-illuminated areas were fitting was impossible. Typically about half of the nodes in the mesh were active in SFS fitting. The meshes that we fitted were covering an area somewhat larger than the illuminated area. This was done for uniformity purposes, so that all the experiments were influenced in the same way by boundary conditions.

We performed experiments on different surface initializations as well. The results were comparable in terms of absolute depth error. More specifically, the Penny images were also fitted with a hemispherical grid, with global scaling along the z-axis. The Mozart images were also fitted with a deformable superquadric that was initially fitted around the contours of the object, using edge forces. The latter images were suitable for this approach, as the edges between the object and the background were very clear. We analyzed quantitatively our results using a number of different error measures as follows:

Methods	Vase			Penny			Mozart			Sombrero		
	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3
Best [43]	8.1	7.9	7.5	7.2	4.4	4.7	15.1	7.7	8.8	6.9	5.6	5.4
AVerr-DM	3.0	4.4	3.7	6.4	2.9	2.2	8.4	4.2	4.5	4.6	2.7	4.5
BFerr-DM	2.8	4.1	3.7	4.1	2.9	2.2	8.1	4.2	4.5	4.2	2.7	2.7
% Improved	63	45	50	12	35	54	45	46	49	34	52	17

Table 1: Average absolute Z error for synthetic images. For comparison, "Best [43]" provides for each image, the best mean absolute error of the results obtained by the algorithms surveyed in [43]. Thus, it includes results from several algorithms — no single algorithm performed best on all the images. Results of previous methods should be compared to average error measurements. AVerr-DM gives the mean absolute errors of our method when the recovered and true surfaces are aligned in the same range of depth, BFerr-DM gives the mean absolute errors of our method when the recovered and true surfaces aligned in a "closest fit" manner. The bottom line reports the percentage reduction of the error between the AVerr line and the best from previous methods

Methods	Vase			Penny			Mozart			Sombrero		
	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3
Best [43]	11.1	13.9	12.9	7.2	5.5	7.3	18.2	14.6	15.9	11.9	10.4	9.2
AVerr-DM	2.0	3.3	3.3	3.8	2.1	1.9	6.7	3.5	5.8	3.8	2.0	2.7
BFerr-DM	2.0	2.6	2.7	2.6	2.0	1.8	6.3	3.4	4.5	2.6	1.7	2.0

Table 2: Standard Deviation of the error for synthetic images. "Best [43]" provides for each image, the best standard deviation of the results obtained by the algorithms surveyed in [43]. Thus, it includes results from several algorithms — no single algorithm performed best on all the images. Results of previous methods should be compared to average error measurements. AVerr-DM gives the standard deviation of our method when the recovered and true surfaces are aligned in the same range of depth, BFerr-DM gives the standard deviation of our method when the recovered and true surfaces aligned in a "closest fit" manner.

Mean and standard deviation of depth error Apart from the height maps, following [43], for each image we also report the average absolute error for each image. Since the images are considered under orthographic projection, the units for the heights are the same as the units for the images, i.e. pixels. For each image, we compare our result to the best result on the same test image as reported in [43]. (The algorithms were those described in [17, 44, 21, 23, 4, 22, 31, 39]). Our results are, in almost all cases clearly better, than the results computed using the SFS-based only algorithms reviewed in [43], and in no case worse. In fact they can be compared in terms of accuracy with the algorithm in [21] which is not considered together with the other algorithms, because it is not solely a SFS algorithm, as it relies on good initial values which are provided by stereo. None of the SFS-based only algorithms reviewed in [43] performed best for all images so the best previous results for each image are due to different algorithms. If we compared against each algorithm individually improvements in error reduction would be even more pronounced.

Methods	Vase			Penny			Mozart			Sombrero		
	S1	S2	S3	S1	S2	S3	S1	S2	S3	S1	S2	S3
Best [43]	1.2	0.9	0.9	1.3	1.0	1.1	1.3	0.6	0.7	0.8	0.5	0.7
AVerr-p-DM	0.2	0.5	0.3	0.5	0.4	0.4	0.5	0.3	0.3	0.4	0.2	0.2
AVerr-q-DM	0.3	0.4	0.4	0.5	0.3	0.3	0.5	0.2	0.3	0.4	0.2	0.2
AVerr-$p-q$-DM	0.2	0.5	0.3	0.5	0.4	0.4	0.5	0.3	0.3	0.4	0.2	0.2

Table 3: $p-q$ error measurements. Best [43]" provides for each image, the best $p-q$ error of the results obtained by the algorithms surveyed in [43]. Thus, it includes results from several algorithms — no single algorithm performed best on all the images. Results of previous methods should be compared to average error measurements. AVerr-DM gives the p, q, $p-q$ error of our method when the recovered and true surfaces are aligned in the same range of depth.

Since all the algorithms recover relative height, the range of the recovered height values can vary greatly. In order to compare any two surfaces, we need to normalize them to the same range. The average absolute error measure scales the height values so that the ranges of the two surfaces are equal – we used the same method for this as [43]. This method can be extremely sensitive to noisy outliers – a single point can drastically change the absolute range of height values. For this reason, we evaluate one more error measure, the best error of fit. In this measure we find the scaling and translation along the Z axis of the two surfaces, so that the distance between them will be minimal, in a least squares sense. These results are reported in Table 1. Using the average absolute error method that was used in [43], we notice that the improvement is between 12 and 67 percent in error reduction, depending on the image. The median improvement is 45 percent. 10 out of 12 images of the images had errors reduced by 33 percent or more.

We also provide the standard deviation σ of the absolute depth errors. It is not perfectly clear in [43] whether the numbers they provide for standard deviation are for σ or σ^2. The results are provided in Table 2. In general the standard deviation results agree with the mean error results in the sense that smaller mean errors have smaller standard deviations. We see that with the exception of Vase, head-on (0,0,1) illumination produces significantly higher errors. This is due to the convex/concave ambiguities, either local or global.

Mean gradient error This indicates the error in surface orientation. It is not clear again how this was computed in [43]. In Table 3 we provide measurements for the average error in the p and q directions of the gradient (i.e. the horizontal dz/dx and vertical dz/dy directions) and an average of the two errors is provided. We used the forward approximation in the

calculation of the gradient. The gradient was computed on the final interpolated depth.

Difference images of the absolute depth error The rightmost two columns of Figures 4 and 5 are depth error images. These images provide information on the distribution of the errors and the effect that the underlying surface structure has in the reconstructed results. The intensity of each pixel is relative to the error in depth at that point of the reconstructed surface. The higher the error of a region, the brighter the region will appear. For visibility purposes intensity values are scaled in the 0-255 range. Since the scaling is not the same for all images, it is not possible to compare two different reconstructions in this way. We can only make meaningful comparisons between pixels of the same image.

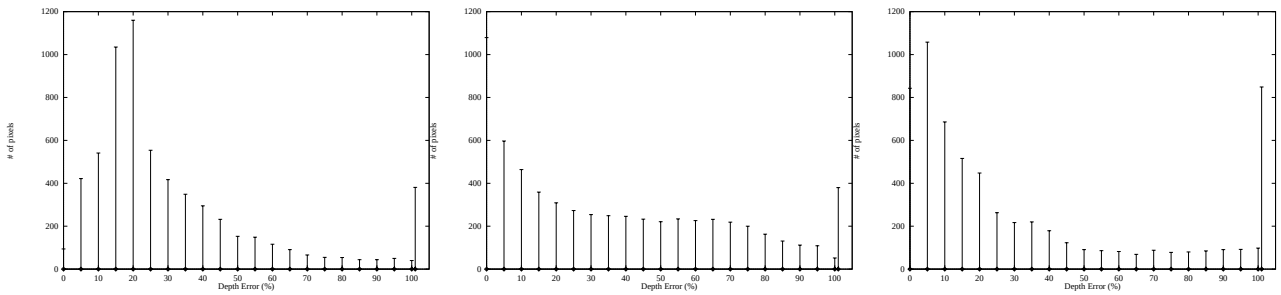
Observing the error images we notice that most of the error is near the boundaries, except for the (0,0,1) images with convex/concave ambiguities. On the side-lit images large self shadowed areas where fitting is not possible violate the integrability of the surface. In the Penny and the Sombrero images the resolution of the fitted model is not high enough for the high frequency details of the images.

Histograms of the percentage depth error In Figures 6, 7 we show the percentages of depth errors with respect to the true depth as distributions of those errors. Each bar of the histograms represents the number of pixels with a relative error within the interval less than or equal to the indicated value. The percentage depth error is computed by:

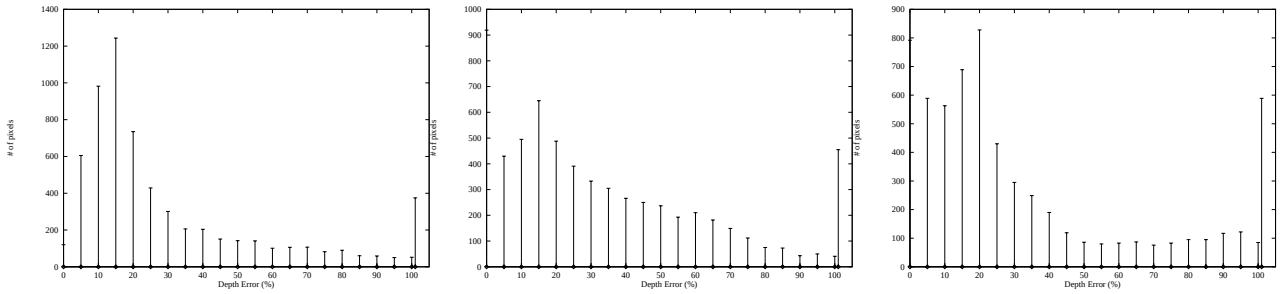
$$\frac{|\text{true depth} - \text{estimated depth}|}{\text{true depth}} \times 100\%$$

Pixels with more than 100% error, typically on problem areas such as boundaries, shadowed areas, or convex concave ambiguities, are plotted as 101% error. Penny and Sombrero images are the ones with a large number of error over 100%, probably due to the discontinuities in the boundaries of Penny and the convex/concave ambiguities and self shadowing of Sombrero.

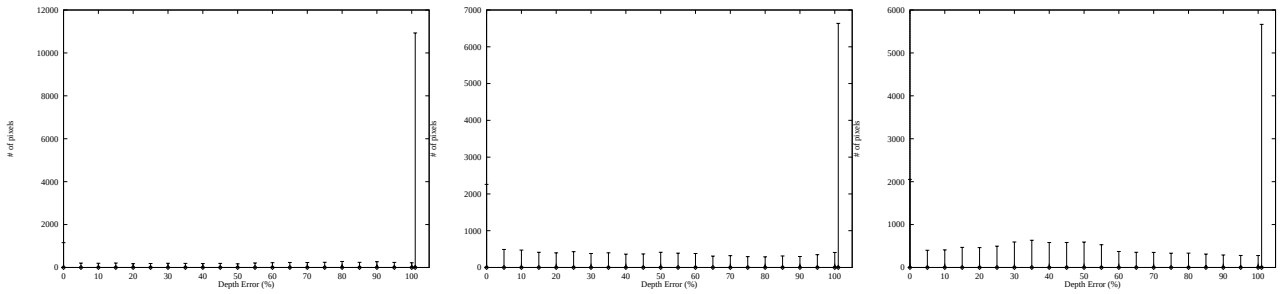
Constraint satisfaction and timing The illumination constraint in Equation (5) which seeks to minimize the difference between the illumination reflected from the model and the intensities measured in the image, is imposed as a hard constraint on the model, meaning



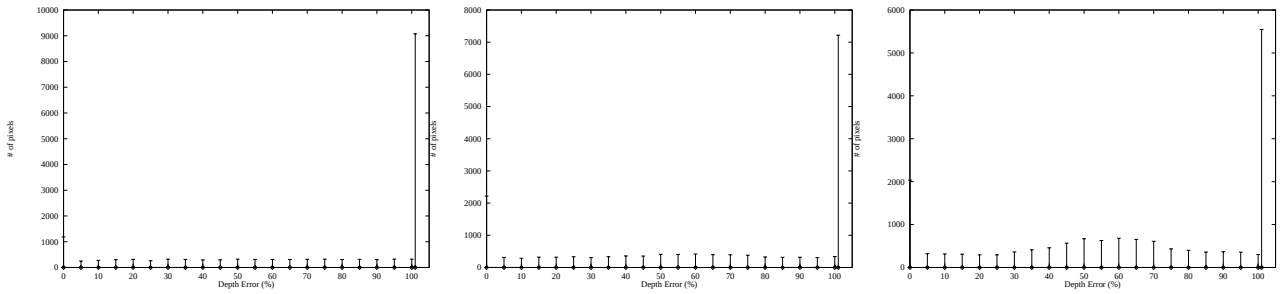
(a1) Vase (0,0,1) "Average" Errors (b1) Vase (1,0,1) "Average" Errors (c1) Vase (5,5,7) "Average" Errors



(a2) Vase (0,0,1) "Best-Fit" Errors (b2) Vase (1,0,1) "Best-Fit" Errors (c2) Vase (5,5,7) "Best-Fit" Errors

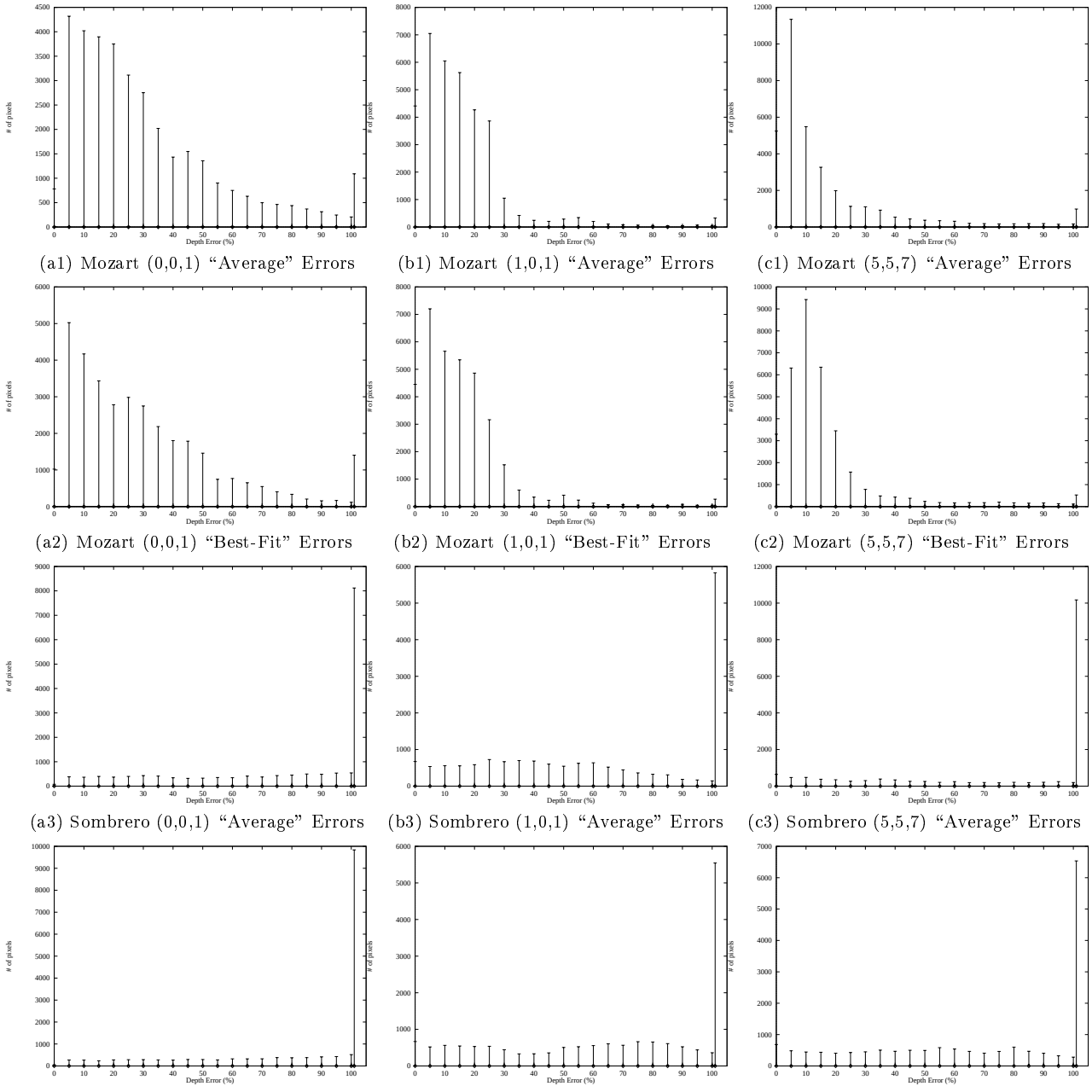


(a3) Penny (0,0,1) "Average" Errors (b3) Penny (1,0,1) "Average" Errors (c3) Penny (5,5,7) "Average" Errors



(a4) Penny (0,0,1) "Best-Fit" Errors (b4) Penny (1,0,1) "Best-Fit" Errors (c4) Penny (5,5,7) "Best-Fit" Errors

Figure 6: Histograms of the percentage depth error for the Vase (Rows 1 and 2) and Penny (Rows 3 and 4) reconstructions, relative to the true depth. Column (a) is for light(0,0,1), Column (b) is for light(1,0,1), Column (c) is for light(5,5,7) Histograms on Rows 1 and 3 are for recovered and true surfaces aligned in the same range of depth, histograms on Rows 2 and 4 are for recovered and true surfaces aligned in a "closest fit" manner. Each bar in the histograms represents the summation of the number of pixels for the depth error within the interval less than or equal to the indicated value. All pixels with more than 100% error are counted as 101% error.



(a1) Mozart (0,0,1) "Average" Errors (b1) Mozart (1,0,1) "Average" Errors (c1) Mozart (5,5,7) "Average" Errors

(a2) Mozart (0,0,1) "Best-Fit" Errors (b2) Mozart (1,0,1) "Best-Fit" Errors (c2) Mozart (5,5,7) "Best-Fit" Errors

(a3) Sombrero (0,0,1) "Average" Errors (b3) Sombrero (1,0,1) "Average" Errors (c3) Sombrero (5,5,7) "Average" Errors

(a4) Sombrero (0,0,1) "Best-Fit" Errors (b4) Sombrero (1,0,1) "Best-Fit" Errors (c4) Sombrero (5,5,7) "Best-Fit" Errors

Figure 7: Histograms of the percentage depth error for the Mozart (Rows 1 and 2) and Sombrero (Rows 3 and 4) reconstructions, relative to the true depth. Column (a) is for light(0,0,1), Column (b) is for light(1,0,1), Column (c) is for light(5,5,7) Histograms on Rows 1 and 3 are for recovered and true surfaces aligned in the same range of depth, histograms on Rows 2 and 4 are for recovered and true surfaces aligned in a "closest fit" manner. Each bar in the histograms represents the summation of the number of pixels for the depth error within the interval less than or equal to the indicated value. All pixels with more than 100% error are counted as 101% error.

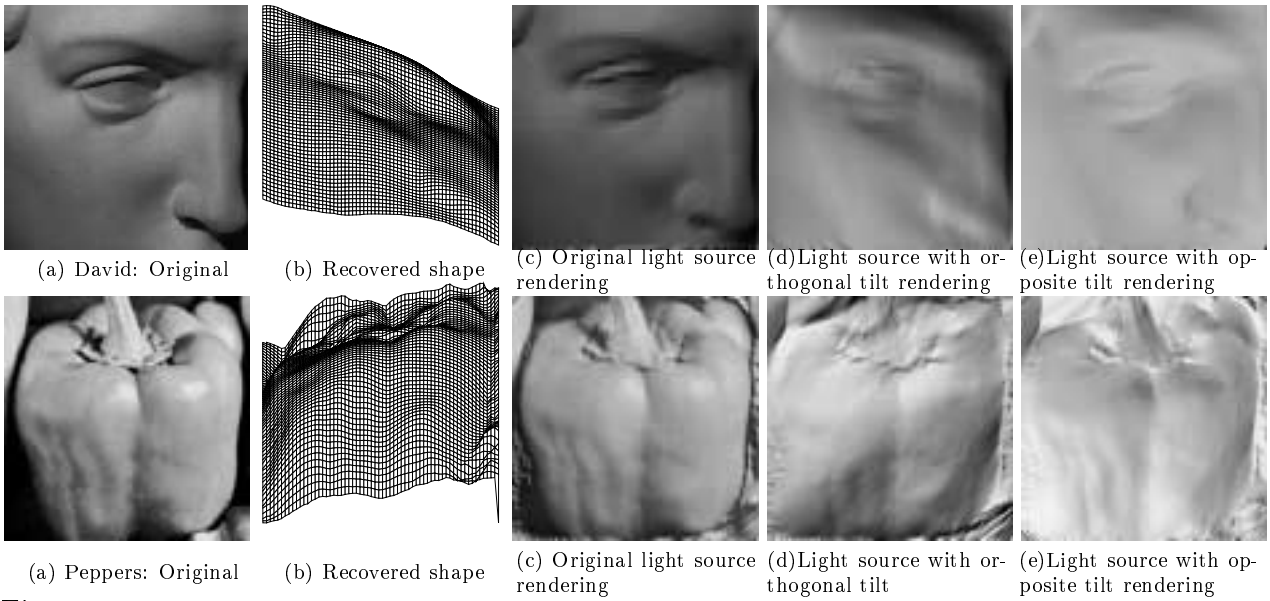
that it has to be entirely satisfied by the fitting process. Thus it is not surprising that in all images the model converged in satisfying the constraint in average within 0.5 to 1.5 8-bit gray levels. As we have seen earlier in the case of not precisely known light source, in cases where the imaging assumption do not hold, in order to satisfy the illumination constraints, excess wrinkling might be introduced in the reconstructed surface.

All the experiments were conducted on an SGI O2 workstation, with an R10000 processor at 175 MHz and 192M of memory. Different images needed significantly different times. The Penny images converged within 1000 iterations whereas the Mozart images needed approximately 10000 to 12000 iterations. The experiments were terminated with a very strict criterion on the average change c on the model parameters, namely $c < 10^{-4} * sc$ where sc is the scale of the model. Measurements of error on results obtained at 10 percent of iterations show an increase in average depth error of less than 15 percent with respect to the final result for most of the images. This suggests that when time is of essence, adequate results can be obtained much faster. By far the most time consuming part of the algorithm, when dealing with large meshes is the matrix inversion described in Section 4.3. For a large mesh with about 8000 active triangles more than 90% of the CPU time is spent on the matrix inversion. An additional 5% is spent on constructing the matrix. For a large mesh with about 8000 active triangles, each iteration needs approximately 2 to 2.5 seconds of CPU time, whereas for smaller meshes with about 800 nodes this times drops below 100 ms. This is why the multiresolution approach is necessary. For really large models, that require a lot of memory to store the data structures the size of the available memory can significantly affect run-times. Indicatively, a low resolution model of Vase of about 200 triangles converged in approximately 30 seconds, a high resolution model of Penny with more than 2000 triangles required approximately 20 minutes and a very high resolution model of Mozart with almost 10000 triangles required more than 4 hours, using a multi-resolution method.

Fast Constraint Force Calculation The effect of the new fast method for constraint force computation described in Section 4.3 were significant both in terms of error reduction and in terms of speed. Compared to the local computation method that we previously used

[34], median drop in average absolute depth errors was 25%. The biggest improvement was for the side illuminated images. In the frontally illuminated images the drop was less than 10%, possibly because SFS in general is not as successful in these images, and smoothing plays a larger role. In terms of speed, improvement was even more dramatic. A mesh with 200 nodes could take more than 10 seconds per iteration with the old method, whereas now this is done in a few milliseconds (approximately 20 ms).

6.1.2. Real Images



(a) David: Original (b) Recovered shape (c) Original light source rendering (d)Light source with orthogonal tilt rendering (e)Light source with opposite tilt rendering

(a) Peppers: Original (b) Recovered shape (c) Original light source rendering (d)Light source with orthogonal tilt (e)Light source with opposite tilt rendering

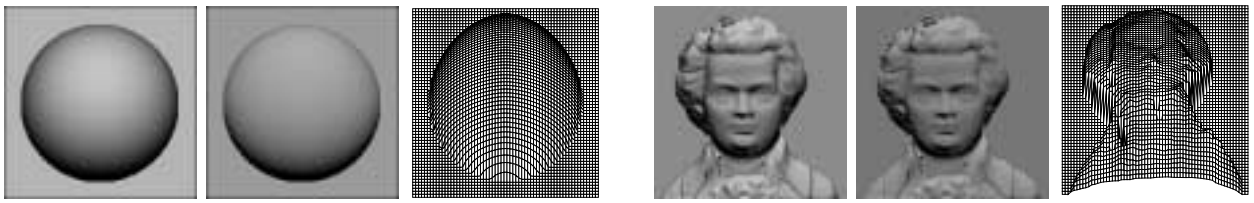
Figure 8: Real images: Reconstruction and 3 renderings under the original light source, light source with orthogonal tilt and light source with opposite tilt

In Figure 8 we have the reconstructed height maps of the two real images. Estimates of the light source were provided with the images. The real images that were tested were: Pepper, with estimated light source (-0.707,0.642,1), David, with estimated light source (-0.707,0.707,1). In Pepper we have albedo discontinuities and self-occlusions, which violate the assumptions of the algorithm. The estimated light sources for those images are the ones given in [43]. We also provide three renderings of each recovered shape; the first under the original light source, the second under a light source with tilt orthogonal to the original and the third under a light source with opposite tilt. The fact that renderings under the light source used for the reconstruction are so close to the original image highlights the advantage of imposing illumination constraints as hard constraints that have to be satisfied exactly.

Visual inspection of our results for synthetic and real images, shows that our method manages to recover surface information and detail in most cases. Discontinuities are handled well, thanks to the elasticity of the model. A number of the inherent problems of SFS, such as convex/concave ambiguities, problems arising from the violation of the assumptions, rippling parallel to the direction of the light [44, 23], appear in some of the results. The C^0 elements that are used for efficiency reasons, tend to favor flat surfaces. The amount of detail recovered depends on the number of elements in the model. When fitting a fine mesh, local detail will appear in the first few iterations. As the fitting progresses, the deformation will deepen and broaden but without significant loss of detail. This is an advantage of the hard constraints approach.

6.2. Non-Lambertian Diffuse Reflectance

Two experiments were performed on synthetic images of a sphere and Mozart, rendered using the lighting model of [29] described in Section 3.2. The parameters for $\sigma = 0.543788$ and $\rho = 0.770950$, are the values that model plaster surfaces [5]. Illumination was at an angle of 22.5 degrees off the viewing axis for the sphere experiment and in the direction (5,5,7) for the Mozart experiment. In both cases the shape was recovered correctly with similar error results as Lambertian reflectance. Reconstruction results are shown in Figure 9.

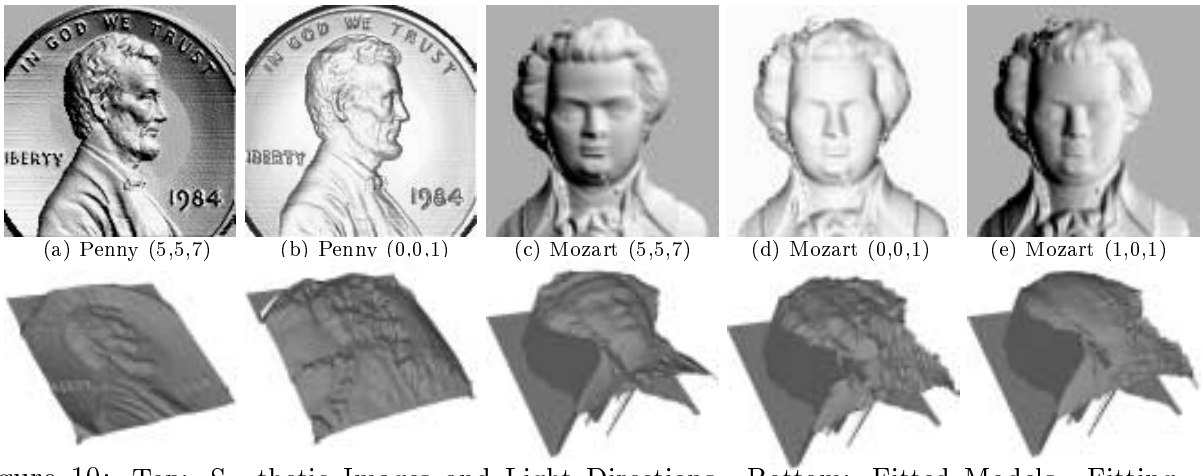


(a) Lambertian reflectance image (b) Diffuse Non-Lambertian [29] (c) recovered shape from (b) (d) Lambertian reflectance image (e) Diffuse Non-Lambertian [29] (f) recovered shape from (e)

Figure 9: (c), (f) are recovered shapes from the images (b), (e) using the Non-Lambertian Diffuse reflectance model described in [29]. (a), (d) are the corresponding images of the same geometry with the same albedo under the Lambertian model.

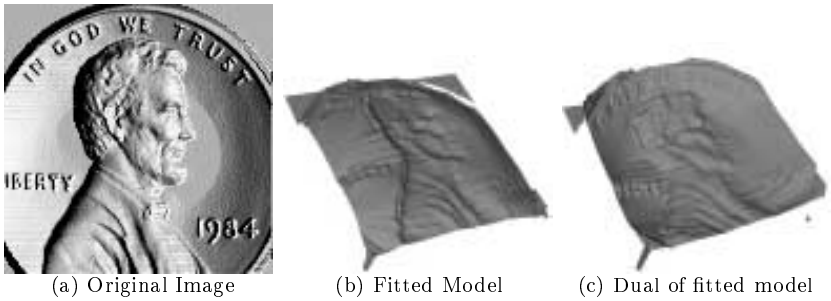
6.3. Unknown Light

We now present experiments of coupled estimation of light direction and SFS on real and synthetic data. We get substantially improved results in light estimation compared to previous methods, and consequently in the shape estimation.



(a) Penny (5,5,7) (b) Penny (0,0,1) (c) Mozart (5,5,7) (d) Mozart (0,0,1) (e) Mozart (1,0,1)

Figure 10: Top: Synthetic Images and Light Directions. Bottom: Fitted Models. Fitting was performed under unknown light i.e. the illumination under which the images were synthesized was not taken into account for the fitting.



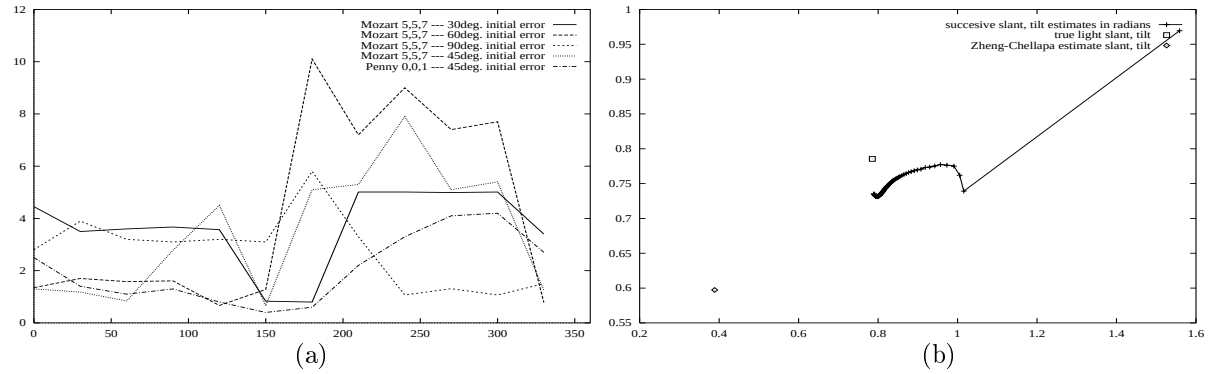
(a) Original Image (b) Fitted Model (c) Dual of fitted model

Figure 11: In the case of the penny lit from [1,0,1], the correct shape (c) is the dual of the fitted model (b).

6.3.1. Synthetic Data

We first present results of experiments on synthetic data sets where we had ground truth for the shape and the generating light source direction. Each data set was illuminated under three different light source directions. For each image we first estimated the light source directions using the method in [44]. This method gave accurate results for the images that were lit head-on (with direction (0,0,1)). In these cases our method obviously gave no improvement (although it did not decrease accuracy — the lighting position remained stable). In all other cases there was a marked improvement over [44].

To test the robustness of our method to an initially poor lighting-direction estimate, we introduced to the images obtained with lighting direction (0,0,1) an initial lighting direction error of 45 degrees. In the Mozart example the light estimate converged to 4.8 degrees from the true direction vector and the in the penny example it converged to 3.1 degrees. The



(a) (b)

Figure 12: (a) Light direction error analysis with respect to initial estimate. We plot the error in the final estimate when the direction of the initial estimate is 30,45,60 and 90 degrees off the true direction. Each initial lighting direction off the true light forms a cone with respect to the true light axis. We sample the cone at intervals of 30 degrees. Four sequences at different angles were generated for the Mozart image with true light direction (5,5,7) and one sequence for Penny illuminated from above (0,0,1). (b) Light estimation intermediate results for Mozart lit from [5,5,7]. Initial estimate at top right was 45 degrees off the true light, denoted by a square. Crosses are the estimates of our algorithm connected by a line in the order that the algorithm successively generates them. The final estimate was 2 degrees off the truth. The circle at the bottom left is the estimate generated by the algorithm in [44] which was 23.7 degrees off the true light.

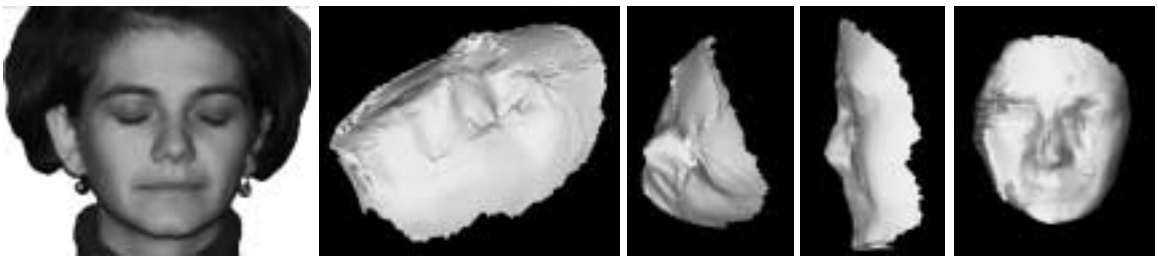


Figure 13: Experiments on real data. Initial image on the left and 4 views of the recovered model

recovered shapes can be seen in Figure 10.

The examples with light source direction (5,5,7) (or $\phi = \theta = \pi/4$) gave the best results. The initial estimate from the [44] method for Mozart was 23.7 degrees off the true vector and our method recovered the illumination direction with a 2 degrees error. The initial estimate for the penny was 83 degrees off. Our model converged to 5.6 degrees of the true light vector.

The examples lit sideways (with $\phi = 0$, direction (1,0,1)) were the ones with the highest errors. The method in [44] gave an estimate for Mozart that was 18.9 degrees off from the correct direction. Our method reduced the error to 9.7 degrees. Although a significant improvement, it still wasn't close enough to avoid visible errors in the reconstruction. As for the penny example, the initial estimate was 83 degrees off from the true light but only 7 from its dual (direction (-1,0,1). As expected, our method converged to the dual with an error of 5

degrees. As can be seen in Figure 11 the fitting process gave us the dual shape, as described earlier. We converted this to the correct fit by taking the complement of the normals with respect to the viewing direction.

In our experiments, we found the Levenberg-Marquart method typically converges very quickly (within three to five iterations) for the estimates of ϕ and θ. We also found our light estimation method typically converges to the accurate estimate quickly, in less than 20 iterations of a low resolution model for the Penny experiments.

We analyzed how the choice of the initial lighting estimate affects the error of the recovered light direction. The initial estimates were chosen in the following way: We selected an initial estimate at 30, 45, 60, 90 degrees off the true vector. Then we rotated each estimate around the true vector and formed a cone. We sampled the cone at intervals of 30 degrees. This gave us 12 different estimates for each fixed angle off the true vector, which we used as initial estimates to our method. We see that for all the experiments where the initial estimate was no more than 30 degrees off the true vector, our algorithm gave a lighting position within 5 degrees from the correct answer (Figure 12(a)). The successive light estimates returned in one of those experiments (initial estimate is 45 degrees off from the truth) are shown in Figure 12(b), as they quickly converge towards the the true light source direction.

We also tested how the initial shape affects the light estimation. The Mozart (5,5,7) image was fitted using a sphere, an ellipsoid and a planar mesh. All three converged to 2.2, 3.5 and 4.6 degrees, respectively. So, based on our experiments, we observe that the method can estimate the light direction, provided that the initial shape has the ability to deform to fit the data. It can take longer to converge if the initial shape is far from the true data.

6.3.2. Real Data

Figure 13 shows the fitting results to a real image of a face. We compare the reconstruction to data obtained by stereo using the method in [11]. The camera pose and the perspective projection parameters were supplied with the stereo data. We compared the recovered light vector, to the light vector that can be estimated from the stereo data, and the two estimates are 2.6 degrees off. The recovered depth is on average 1.34 percent off from the true data. There

are visible errors in the reconstruction of the mouth and the eyes due to the albedo changes and at the tip of the nose due to specularities that violate the Lambertian assumption. These are typical shortcomings when SFS alone is used on real data. However, the light estimation was not adversely affected by the violations of the shading assumptions.

7. Conclusions and Future Work

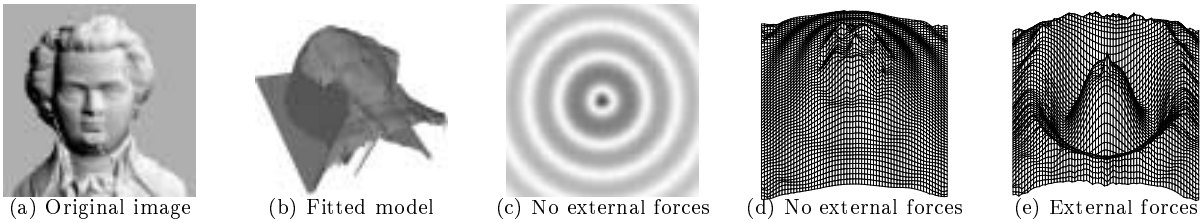
In this paper we presented a method for the integration of nonlinear holonomic constraints in deformable models and its application to the shape and illuminant direction estimation from shading problems. Experimental results demonstrate that our method performs better than previous Shape from Shading algorithms applied to images of Lambertian objects under known illumination. It is also more general as it can be applied to non-Lambertian surfaces and does not require knowledge of the illuminant direction. Using our theory for the numerically robust integration of nonlinear holonomic constraints within a deformable model framework, we demonstrated how any type of illumination constraint, from the Lambertian model to more complex highly nonlinear models [27, 29] can be incorporated in a deformable model framework.

Our method also works when the light source direction is unknown. We coupled our shape estimation method with a method for light estimation, in an iterative process, where improved shape estimation results in improved light estimation and vice versa. This method resulted to light direction estimation for constant albedo Lambertian surfaces within 5 degrees of the true direction, when the initial estimate was within 45 degrees from the truth.

Future work includes integration of this method with other image cues, such as stereo and optical flow. In such cases the question of heterogeneous constraints on the same model will need to be addressed and each type of constraint applied appropriately. We also plan to generalize to more general illumination situations with more than one light sources.

A. Further Generalizations of the Approach

Perspective Projection Perspective projection can be easily handled in the deformable model formulation since the reconstruction takes place in 3D space and so the mathematical treatment does not change. An image point $\mathbf{x_p} = (X, Y, -f) = (-fx/z, -fy/z, -f)$, where f is the focal length of the projection will correspond to model point $\mathbf{x} = (x, y, z) = (-zX/f, -zY/f, z)$. That means that



(a) Original image (b) Fitted model (c) No external forces (d) No external forces (e) External forces

Figure 14: In (b) a model fitted under perspective projection (original image in (a)), with a low-resolution model. In (c) the case of the sombrero lit from the top; there is convex/concave ambiguity, one possible solution seen in (d). Knowledge about how to resolve the ambiguity, (e.g. singular points) can be applied to the model in the form of external forces. Here a few points in the inner lower circle were pulled down. The resulting model is in (e).

as z changes the model point that corresponds to $\mathbf{P}$ changes too. Points that were visible might become occluded, and visibility checking would be required after every iteration. Also Equation (7) needs to be slightly modified, as the sampled image values $I'_L(x,y)$ are parameterized by X and Y, related to the model points by the perspective transformation:

$$\frac{\partial \mathbf{I'_L}(X,Y)}{\partial \mathbf{q}} = \frac{\partial \mathbf{I'_L}(X,Y)}{\partial \mathbf{x}_p(X,Y)} \frac{\partial \mathbf{x}_p(X,Y)}{\partial \mathbf{q}} =, \frac{\partial \mathbf{I'_L}(X,Y)}{\partial \mathbf{x}_p(X,Y)} \frac{\partial \mathbf{x}_p(X,Y)}{\partial \mathbf{x}(x,y,z)} \frac{\partial \mathbf{x}(x,y,z)}{\partial \mathbf{q}} \tag{20}$$

where

$$\frac{\partial \mathbf{x}_p(X,Y)}{\partial \mathbf{x}(x,y,z)} = \begin{bmatrix} f/z & 0 & -fx/z^2 \\ 0 & f/z & -fx/z^2] \end{bmatrix} \tag{21}$$

Furthermore, as the computed depth changes image points might project in different elements and hence constraints might be applied to different elements in subsequent iterations.

In Figure 14(b) we show the results of fitting to a perspective projection image. The focal length was assumed to be 2 and the distance from the focal point was assumed to be 4, so the perspective distortion effects on the image are quite noticeable. The light source direction was (5,5,7). Our light source estimate was 2.5 degrees off. We show the coarse model that we used while still estimating the light source. Once the light stabilizes we increase the model's resolution and fit it more closely to the lighting constraints.

Singular Points Information It has been shown [28], that the known ambiguities in shape from shading estimation can be resolved with the knowledge of singular points in the image. Here we show how we can incorporate such biases in the deformable model formulation. In the sombrero image viewed head-on in figure 14(c), there is a convex-concave ambiguity. Our deformable model fits the convex solution only. If we know that we want the inner lower circle to be lowered down, we can simply apply external forces $\mathbf{f}_e$ in (10) on a few of the points on that circle (in the example presented here just five). Then the whole model will converge to the desired solution, in order to satisfy the illumination constraint. Each shading ambiguity has to be resolved independently through the introduction of the proper forces; no forces were applied to the tip of the sombrero which still points upwards.

This could also be accomplished in a more robust way, by enforcing the knowledge of the position of a certain point as a hard constraint, in the method of [26]. The trade-off would be that the size of the matrix to be inverted in Equation (17) would increase even more.

References

[1] E. Angelopoulou and J. Williams. Photometric surface analysis in a tri-luminal environment. In *ICCV99*, pages 442–447, 1999.

[2] J. Baumgarte. Stabilization of constraints and integrals of motion in dynamical systems. *Computer Methods in Applied Mechanics and Engineering*, 1:1–16, 1972.

[3] P.N. Belhumeur and D.J. Kriegman. What is the set of images of an object under all possible lighting conditions. In *CVPR96*, pages 270–277, 1996.

[4] M. Bichsel and A.P. Pentland. A simple algorithm for shape for shading. In *CVPR92*, pages 459–465, 1992.

[5] K.J. Dana, S.K. Nayar, B. van Ginneken, and J.J. Koenderink. Reflectance and texture of real-world surfaces. In *CVPR97*, pages 151–157, 1997.

[6] D. DeCarlo and D. Metaxas. The integration of optical flow and deformable models: Applications to human face shape and motion estimation. In *CVPR96*, pages 231–238, 1996.

[7] H. Delingette and J. Montagnat. New algorithms for controlling active contours shape and topology. In *ECCV00*, 2000.

[8] R.T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *PAMI*, 10(4):439–451, July 1988.

[9] P. Fua and C. Brechbuhler. Imposing hard constraints on deformable models through optimization in orthogonal subspaces. *CVIU*, 65(2):148–162, February 1997.

[10] P. Fua and Y.G. Leclerc. Object-centered surface reconstruction: Combining multi-image stereo and shading. *IJCV*, 16(1):35–56, September 1995.

[11] P. Fua and C. Miccio. From regular images to animated heads: A least squares approach. In *ECCV98*, pages 188–202, 1998.

[12] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman. Illumination-based image synthesis: Creating novel images of human faces under differing pose and lighting. In *MVIEW99*, 1999.

[13] G.H.Golub and C.F.Van Loan. *Matrix computations*. Johns Hopkins University Press, 3rd edition, 1996.

[14] B.K.P. Horn. *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View*. PhD thesis, MIT, 1970.

[15] B.K.P. Horn. Height and gradient from shading. *IJCV*, 5(1):37–76, August 1990.

[16] B.K.P. Horn and eds Brooks, M.J. *Shape from Shading*. MIT Press, 1989.

[17] B.K.P. Horn and M.J. Brooks. Shape and source from shading. In *International Joint Conference on Artificial Intelligence*, pages 932–936, 1985.

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

[19] R. Kimmel and A. M. Bruckstein. Global shape from shading. *Computer Vision and Image Understanding*, 62(3):360–369, 1995.

[20] R. Kimmel and J. A. Sethian. An optimal time algorithm for shape from shading. In *4th Asian Conference on Computer Vision*, Taipei, Taiwan, January 8-11 2000.

[21] Y.G. Leclerc and A.F. Bobick. The direct computation of height from shading. In *CVPR91*, pages 552–558, 1991.

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

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

[24] D.W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, 11(2):431–441, Jun. 1963.

[25] D. Metaxas. *Physics-Based Deformable Models: Applications to Computer Vision,Graphics and Medical Imaging*. Kluwer-Academic Publishers, November 1996.

[26] D. Metaxas and D. Terzopoulos. Shape and nonrigid motion estimation through physics-based synthesis. *PAMI*, 15(6):580–591, June 1993.

[27] S.K. Nayar, K. Ikeuchi, and T. Kanade. Surface reflection: Physical and geometrical perspectives. *PAMI*, 13(7):611–634, July 1991.

[28] J. Oliensis and P. Dupuis. A global algorithm for shape from shading. In *ICCV93*, pages 692–701, 1993.

[29] M. Oren and S.K. Nayar. Generalization of the lambertian model and implications for machine vision. *IJCV*, 14(3):227–251, April 1995.

[30] A.P. Pentland. Local shading analysis. *PAMI*, 6(2):170–187, March 1984.

[31] A.P. Pentland. Shape information from shading: A theory about human perception. In *ICCV88*, pages 404–413, 1988.

[32] W.H. Press, S.A. Teukolsky, W.T. Vettering, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1992.

[33] E. Rouy and A. Tourin. A viscosity solutions approach to shape from shading. *SIAM Numerical Analysis*, 29(3):867–884, 1992.

[34] D. Samaras and D. Metaxas. Incorporating illumination constraints in deformable models. In *CVPR98*, pages 322–329, 1998.

[35] D. Samaras and D. Metaxas. Coupled lighting direction and shape estimation from single images. In *ICCV99*, pages 868–874, 1999.

[36] D. Samaras, D. Metaxas, P. Fua, and Y. Leclerc. Variable albedo surface reconstruction from stereo and shape from shading. In *CVPR00*, pages I:480–487, 2000.

[37] G. Strang. *Introduction to Applied Mathematics*. Wellesley-Cambridge Press, MA, 1986.

[38] R. Szeliski. Fast shape from shading. *CVGIP*, 53(2):129–153, March 1991.

[39] P.S. Tsai and M. Shah. A fast linear shape from shading. In *CVPR92*, pages 734–736, 1992.

[40] Jens Wittenburg. *Dynamics of systems of rigid bodies*. Teubner, Stuttgart, 1977.

[41] R.J. Woodham. Analysing images of curved surfaces. *AI*, 17:117–140, 1981.

[42] P.L. Worthington and E.R. Hancock. New constraints on data-closeness and needle map consistency for shape-from-shading. *PAMI*, 21(12):1250–1267, December 1999.

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

[44] Q. Zheng and R. Chellappa. Estimation of illuminant direction, albedo, and shape from shading. *PAMI*, 13(7):680–702, July 1991.

Dr. Dimitris Samaras is an Assistant Professor in the Department of Computer Science at the State University of New York at Stony Brook, since September 2000. He received a Diploma in Computer Science and Engineering from the University of Patras in 1992, an M.Sc. in Computer Science from Northeastern University in 1994, and a Ph.D degree from the University of Pennsylvania in 2001. He specializes in deformable model techniques for 3D shape estimation and motion analysis, illumination modeling and estimation and biomedical image analysis. He is a member of ACM and IEEE.

Dr. Dimitris Metaxas is a Professor in the Division of Computer and Information Science and the Department of Bioengineering at Rutgers University. From January 1998 to December 2001 he was an Associate Professor in the Department of Computer and Information Science at the University of Pennsylvania. From September 1992 to December 1997 he was an Assistant Professor in the same Department. He got a Diploma in electrical engineering from the National Technical University of Athens in 1986, a M.Sc. in Computer Science from the Univ. of Maryland in 1988, and a Ph.D. degree in Computer Science from the Univ. of Toronto in 1992. Dr. Metaxas is the Director of the Center for Computational Bioengineering, Imaging and Modeling (CBIM) and while at PENN he was heading the VAST Lab (Vision, Analysis and Simulation Technologies Lab). He specializes in physics-based techniques for the modeling, estimation and synthesis of the shape and motion for complex objects. He is the author of the book "Physics-based deformable models: Applications to computer vision, graphics and medical imaging" published by Kluwer Academic Publishers in Nov. 1996. He is an Associate Editor of GMIP, is on the Editorial Board of Medical Imaging and has been an associate editor for PAMI. Dr. Metaxas has organized several workshops and conferences in his area, has served on the program committees of all major computer vision, medical imaging and computer graphics conferences since 1992 and has over 160 research publications in these areas. He

received a NSF Initiation Award (1993), a NSF Career Award (1996), and a ONR Young Investigator Proposal Award (1997). Dr. Metaxas has been invited nationally and internationally to speak on his research and his research has received several best paper awards and patents. Dr. Metaxas has graduated 13 PhD students and is a member of ACM and IEEE.