Object centered stereo: displacement map estimation using texture and shading

Neil Birkbeck Dana Cobzas Martin Jagersand

Computing Science
University of Alberta
Edmonton, AB T6G2E8, Canada

Abstract

We consider the problem of recovering 3D surface displacements using both shading and multi-view stereo cues. In contrast to traditional disparity or depth map representations, the object centered displacement map representation enables the recovery of complete 3D objects while also ensuring the reconstruction is not biased towards a particular image. Although displacement mapping requires a base surface, this base mesh is easily obtained using traditional computer vision techniques (e.g., shape-from-silhouette or structure-from-motion). Our method exploits shading variation due to object rotation relative to the light source, allowing the recovery of displacements in both textured and textureless regions in a common framework. In particular, shading cues are integrated into a multi-view stereo photo-consistency function through the surface normals that are implied by the displacement map. The analytic gradient of this photo-consistency function is used to drive a multi-resolution conjugate gradient optimization. We demonstrate the geometric quality of the reconstructed displacements on several example objects including a human face.

1. Introduction

The automatic computation of 3D geometric and appearance models from images is one of the most challenging and fundamental problems in computer vision. While a more traditional point-based method provides accurate results for camera geometry, a surface representation is required for modeling and visualization applications. One of the most popular classes of surface-based reconstruction methods are multi-view stereo approaches that represent the surface as a depth or disparity map with respect to one reference image [17] or multiple images [11]. One of the main disadvantage of those methods, referred to as *image-centered* approaches, is that the reconstruction will be biased by the chosen reference image. As a consequence, the results are often aliased due to the limited depth resolution and they do not reconstruct parts of the object that were occluded in the reference image. An *object-centered* model is therefore more suitable for multi-view reconstruction. Examples of object-centered surface representations include voxels [18], level-sets [8], and meshes [10, 7].

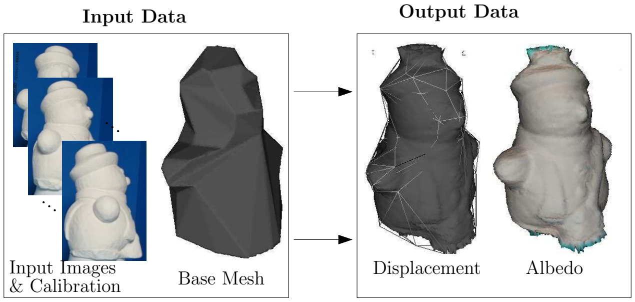

Figure 1: An overview of our method, which recovers the surface displacement and albedo given a set of input images, a base mesh, and calibration data.

We propose a surface reconstruction method that investigates a less explored object-centered representation - a depth field registered with a base mesh (see Fig. 1). Our model is inspired by computer graphics displacement maps [5] that nowadays have efficient HW implementations (e.g., [15]). Compared to a deformable mesh-based representation, our method is more stable as disparities are constrained to move along the normal directions w.r.t. the base mesh. For a purely mesh-based representation, the vertexes are allowed to move freely, so the displacement direction can become unstable. The method can also be regarded as an extension of the stereo-based methods, where depth is calculated with respect to an object-centered base mesh as opposed to the image plane. Therefore reconstruction methods used for stereo can be easily generalized to our representation. One can, for example, make use of the efficient discrete methods like graph cuts or belief propagation that cannot be used with some other object-centered representations (e.g., mesh, level-sets). For our case, as the cost function is based on shading and uses the surface normals, we chose a continuous representation where the normals are implied by the surface.

We formulate the surface reconstruction of Lambertian scenes as an optimization of a photo-consistency function that integrates stereo cues for textured regions with shape from shading cues for texture-less regions. The cost function is calculated over a discretization of the base mesh. Even though the method assumes an a-priori base mesh, this can be easily obtained in practice using shape-from-silhouette or triangulation of structure-from-motion points. Due to the high dimensionality, reconstruction can be difficult and slow, while requiring a substantial amount of image data. To ameliorate these problems, we propose a multi-resolution algorithm.

There exist other approaches that combine stereo for textured regions with shape from shading cues for texture-less regions [10, 14], but, in those works, the two scores are separate terms in the cost function and the combination is achieved either using weights [10] or by partitioning the surface into regions [14]. Like photometric stereo, our method is able to reconstruct the surface of spatially varying or uniform material objects by assuming that the object is moving relative to the light source. A similar constraint was used by Zhang et al. [25] and Weber et al. [22] but with a different surface representation.

To summarize, the main contributions of the paper are:

- We propose a method that reconstructs surfaces discretized as a depth field with respect to a base mesh (displacement map); the representation is suitable for both closed and open surfaces and, unlike traditional stereo, reconstructs whole objects;
- We designed a photo-consistency function suitable for surfaces with textured and uniform Lambertian reflectance by integrating shading cues on implied surface normals;
- We designed a practical setup that provides the necessary light variation, camera and light calibration and requires only commonly available hardware: a light source, a camera, and a glossy white sphere.

2 Related Work

To our knowledge, the work of Vogiatzis et al. [21] is the only source that deals with reconstruction of depth from a base mesh. Most previous approaches reconstruct depth (from stereo) and then fit planes to the resulting 3D points [6]. In contrast, Vogiatzis et al. [21] estimate the displacement for sample points on the base mesh using a belief propagation technique. Their method assumes there is no illumination variation in the images (the scene is fixed w.r.t. illumination) and the cost function is just a variance of the colors that a sample point project to. The method is therefore similar to multi-view stereo reconstruction, requiring scenes with good texture.

A related representation was proposed by Zeng et al. [23] but, in their case, the base geometry is a collection of planes (fitted to a set of 3D points given a-priori) that do not necessarily form a mesh. The depth of the surface patch along the plane normal is reconstructed using a globally-optimal graph cut optimization that ensures that the patch surface goes through any neighboring 3D points. A similar surface patch approach was used to determine voxel consistency in a more recent work by Zheng and his colleagues [24]. Other approaches use parametric models for local depth representation (e.g., planar disks [9], quadratic patches [13]). The patches are then integrated and interpolated to form the final surface.

The mesh-based methods share some similarities to the chosen representation, but they operate by iteratively evolving and refining an initial mesh until it fits the set of images [10, 7, 12, 2]. In these approaches large and complex meshes have to be maintained in order to represent fine detail. In contrast we assume a fixed base mesh and compute surface detail as a height field with respect to it.

The chosen representation is an extension of the traditional disparity map where the depth is registered with respect to a base geometry instead of the reference image plane. Thus it has similarities to the multi-view stereo reconstruction techniques. Like in the global stereo methods (e.g., [20, 11]) we formulated the problem as minimizing a global energy function that takes into account both matching and smoothness costs. But, unlike stereo methods, in our case the discretization and regularization is viewpoint independent. Additionally, our approach is able to reconstruct complete scenes as every point that is visible in at least one image will be reconstructed (in the case of stereo points that are occluded in the reference image are not reconstructed).

3 Problem Definition and Formulation

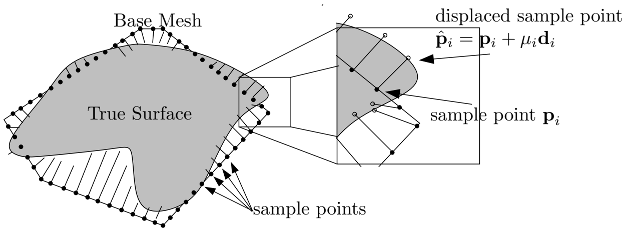
Base Mesh
True Surface
displaced sample point
$\hat{\mathbf{p}}_i = \mathbf{p}_i + \mu_i \mathbf{d}_i$
sample point $\mathbf{p}_i$
sample points

Figure 2: An overview of the representation and notation used in this paper.

We assume that we are given a set of n images, I_i, the corresponding calibration matrices, $\mathbf{P}_i$, the corresponding illumination parameters, L_i, and a base mesh consisting of a set of vertices $V = \{\mathbf{v} \in \Re^3\}$ and a triangulation of these vertices $T = \{(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3) | \mathbf{v}_i \in V\}$. The shape reconstruction problem is then cast as an optimization problem, that recovers a set of displacements μ_j along a direction $\mathbf{d}_j$ from a discrete set of m sample points, $\mathbf{p}_j$, taken on the surface of each triangle. A point on the displaced surface will be denoted as $\hat{\mathbf{p}}_j = \mathbf{p}_j + \mu_j \mathbf{d}_j$, which is a function of the current estimate of the displacement μ_j (see Fig. 2).

The displacements, μ_j, are then related to the image measurements through a photo-consistency function. The photo-consistency function measures the similarity between the reflectance of a point on the surface and the images in which it is observed. The goal is then to find a set of displacements, μ_j, and the reflectance parameters, α_j, of the sample points that minimize the value of the following cost function:

$$F_{data} = \sum_j \frac{1}{|V_j|} \sum_{i \in V_j} f_i(\mu_j, \alpha_j) \tag{1}$$

$$= \sum_j \frac{1}{|V_j|} \sum_{i \in V_j} |R_i(\hat{\mathbf{p}}_j, \alpha_j) - I_i(\Pi(\mathbf{P}_i \hat{\mathbf{p}}_j))|^2 \tag{2}$$

In the above cost function, V_j denotes the set of images that observe displaced point $\hat{\mathbf{p}}_j$, and R_i denotes the *rendering function* that produces a color value for the displaced point observed under the illumination and viewing parameters of camera i. In other words, the cost function expresses the difference between the rendered surface representation and the input images. We used a similar cost function in our mesh-based implementation [2] although in this work we currently take no precautionary measures to filter out specular highlights or address image sampling issues. Again, instead of using the entire set of cameras that observe a sample point, a subset of the visible cameras closest to the median camera are used. This adjustment partially accounts for image sampling problems (e.g., cameras that view the surface at a grazing view).

We assume that the input images and illumination information is sufficient for determining the reflectance parameters α_j. That is, if we fix the surface displacements, there exists a closed form solution for reflectance parameters that minimize Eq. 4. If this is the case, then this minimization process can be done for each α_j independent of the other ones (like in photometric stereo).

In our specific work, we assume that the surface is Lambertian, implying that the surface reflectance parameters at a point are simply the albedo (e.g., α_j is a RGB color). Furthermore, we assume that the illumination for each frame is represented as a directional light source plus an ambient term [1]. We use $\mathbf{l}_i$ to denote the light direction, ℓ_i to denote the light color, and $\mathbf{a}_i$ to denote the ambient color. Under these conditions, the rendering function for image i is simply the Lambertian shading model for the case of a single light source:

$$R_i = \alpha_j \left(\ell_i \frac{\mathbf{n}_j}{|\mathbf{n}_j|} \cdot \mathbf{l}_i + a_i \right) \tag{3}$$

where $\mathbf{n}_j$ is the surface normal at displaced point j. We use the implied surface normal, obtained as a function of the sample points that are within a neighborhood of sample point j, denoted N_j.

Unlike traditional binocular stereo, the use of the implicit surface normal in the above cost function acts a regularizer, but the minimization is still sensitive to noise. As is the case in other multi-view stereo reconstructions, additional regularization is necessary. In this work, we take the approach often used in binocular methods, where the regularization is added as a separate term in the minimization, giving

$$F = F_{data} + \lambda F_{smooth} \tag{4}$$

Currently, we use a simple quadratic regularizer that ensures that neighboring sample points, have similar displacements:

$$F_{smooth} = \sum_{j \in m} \sum_{k \in N_j} (\mu_j - \mu_k)^2 \tag{5}$$

where N_j is a neighborhood around sample point j. Another potential smoothness term would be to use the Euclidean distance between neighboring points, as was done by Vogiatzis et al. [21].

An alternative approach is to keep the regularization implicit (multiplicative regularizer) by considering the weighted minimal surface like in the level set approaches (e.g., Faugeras and Keriven [8]). However, researchers (e.g., Soatto et al. [19]) made the observation that the implicit level set approach might suffer from over-smoothness due to high order derivatives involved into unknowns and also could change the image variability depending on the surface. Therefore, for our system, we used an additive regularizer, giving a more precise and easily interpretable effect.

4 Sampling & Optimization

We now detail the procedures required in minimizing the cost function presented in Section 3. First we discuss how we discretize the base mesh into sample points, and define a specific neighborhood function for the sample points. With these details in place we then present a closed form solution for the reflectance parameters for a set of surface points, followed by details of the optimization procedure.

4.1 Sampling

Clearly, it is desirable to have a set of sample points that are evenly distributed on the base mesh. The resolution of the mesh should be chosen so that the distance between sam-

[1] Actually, we model the light as a point light source, implying that the direction to the light source is dependent on the 3D position of the point.

ple points roughly corresponds to one pixel in image space. One simple solution would be to choose the sample points on a regular grid within each base triangle. However, for visibility and neighborhood purposes it is useful to have a triangulation of the base points. Unfortunately, obtaining a water tight triangulation of these regularly spaced points may become complicated. Moreover, the sampling rate may be different on the edges of the base mesh.

To alleviate these sampling issues we instead use a subdivision approach. In this approach, starting from the base mesh and given a desired sample spacing, any edges in the mesh that are longer than twice the desired sample spacing are split in two. Splitting an edge turns the two triangles that contain the violating edge into four smaller triangles. This operation is performed until all edges in the mesh are less than twice the sample spacing (e.g., splitting an edge would produce two sample points that closer than the desired sample spacing). Simply using this approach may produce sample points with an irregular valence (i.e., an irregular number of neighbors) and may create triangles with poor aspect ratios. To circumvent these problems, we use the topological operators of Lachaud and Montanvert [16]. These operators ensure that edge lengths are within a certain range and that non-neighboring vertices are sufficiently spaced.

The vertices of the subdivided mesh are the sample points $\mathbf{p}_j$, and the neighborhood function N_j is defined as those sample points that share an edge in the subdivided mesh. This approach ensures that the neighborhood function extends across base triangles. Finally, we take the displacement direction $\mathbf{d}_j$ of a sample point to be the corresponding interpolated surface normal at the position $\mathbf{p}_j$ on the base mesh. As our current implementation does not account for self-intersection during the refinement, using the interpolated normal opposed to the base plane normal reduces the occurance of these self-intersections.

4.2 Reflectance Parameters

Recall that we required a closed form solution for the reflectance parameters for any particular choice of displacements. The Lambertian shading function in Eq. 3 is dependent on the implied surface normal, $\mathbf{n}_j$, for surface point j. In this work, we compute the implied surface normal, $\mathbf{n}_j$, as an area weighted average of the triangle normals of the sample point triangulation, which is given below in unnormalized form:

$$\mathbf{n}_j = \sum_{(j,k,i)\in\Delta(\mathbf{p}_j)} (\hat{\mathbf{p}}_k - \hat{\mathbf{p}}_j) \times (\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j) \qquad (6)$$

where $\Delta(\mathbf{p}_j)$ denotes the triangles that contain $\mathbf{p}_j$. Notice that the implied surface normal is a function of the displaced surface points.

Using the above definition of a surface normal for a particular instantiation of surface displacements, we can compute a closed form solution for the α_j. For each particular surface point, we collect all image observations of the point into a system of equations:

$$\begin{bmatrix} \left(\ell_1 \frac{\mathbf{n}_j \cdot \mathbf{l}_1}{|\mathbf{n}_j|} + a_1\right) \\ \left(\ell_2 \frac{\mathbf{n}_j \cdot \mathbf{l}_2}{|\mathbf{n}_j|} + a_2\right) \\ \vdots \\ \left(\ell_m \frac{\mathbf{n}_j \cdot \mathbf{l}_m}{|\mathbf{n}_j|} + a_m\right) \end{bmatrix} \alpha_j = \begin{bmatrix} I_1(\Pi(\mathbf{P}_1 \hat{\mathbf{p}}_j)) \\ I_2(\Pi(\mathbf{P}_2 \hat{\mathbf{p}}_j)) \\ \vdots \\ I_n(\Pi(\mathbf{P}_n \hat{\mathbf{p}}_j)) \end{bmatrix} \qquad (7)$$

$$\mathbf{A}_j \alpha_j = \mathbf{I}_j \qquad (8)$$

As both the left and right hand sides are $n \times 1$ vectors, the least squares solution for the albedo is easily obtained as:

$$\alpha_j = \frac{\mathbf{A}_j^\top \mathbf{I}_j}{\mathbf{A}_j^\top \mathbf{A}_j} = \frac{\mathbf{A}_j \cdot \mathbf{I}_j}{\mathbf{A}_j \cdot \mathbf{A}_j} \qquad (9)$$

4.3 Optimization

One possible method for optimization would be to discretize the displacement values into a set of discrete labels and then use a combinatorial optimization technique to refine the labels (e.g., similar to the method of Vogiatzis et al. [21]). As the surface normals rely on a current estimate of the neighboring displacements, this approach can not directly minimize our cost function (see Section 4.4.1, for more details about a direct discretization of the cost function).

Instead, we perform a continuous optimization of the displacements. As both the memory and computational cost for computing a Hessian of the objective function is prohibitive, we chose the conjugate gradient method. We use the analytic derivatives of the cost function in the optimization. These derivatives are summarized below.

For convenience we define the individual terms of the cost metric for a given sample point in a given view:

$$c_{ij} = \left(\alpha_j \left(\ell_i \frac{\mathbf{n}_j \cdot \mathbf{l}_i}{|\mathbf{n}_j|} + a_i\right) - I_i(\Pi(\mathbf{P}_i \hat{\mathbf{p}}_j))\right)^2$$

The value of the objective function for sample point j in image i (e.g., c_{ij}) is dependent only on its displacement, μ_j, and the displacements of the neighbors $N_j(\mathbf{p}_j)$. Therefore,

$$\tfrac{\partial c_{ij}}{\partial \mu_k} = 0 \quad \text{if } k \neq j \& k \notin N_j(\mathbf{p}_j) \qquad (10)$$

$$\frac{\partial c_{ij}}{\partial \mu_k} = 2\left(\alpha_j \left(\ell_i \frac{\mathbf{n}_j \cdot \mathbf{l}_i}{|\mathbf{n}_j|} + a_i\right) - I_i(\Pi(\mathbf{P}_i \hat{\mathbf{p}}_j))\right)$$
$$\left(\left(\ell_i \frac{\mathbf{n}_j \cdot \mathbf{l}_i}{|\mathbf{n}_j|} + a_i\right) \frac{\partial \alpha_j}{\partial \mu_k} + \alpha_j \ell_i \frac{\partial}{\partial \mu_k} \frac{\mathbf{n}_j \cdot \mathbf{l}_i}{|\mathbf{n}_j|} - \frac{\partial I_i(\mathbf{P}_i \hat{\mathbf{p}}_j)}{\partial \mu_k}\right)$$

The derivative of the least squares computation for the albedo, $\frac{\partial \alpha_j}{\partial \mu_k}$, is straightforward and can be expressed in terms of $\frac{\partial}{\partial \mu_k}(\frac{\mathbf{n}_j \cdot \mathbf{l}_i}{|\mathbf{n}_j|})$. These derivations and those for the $\frac{\partial}{\partial \mu_k} I_i(\Pi(\mathbf{P}_i \hat{\mathbf{p}}_j))$ term can be found in Appendix A.

4.4 Multi-Resolution

The local nature of the conjugate gradient optimization is sensitive to the starting position and is likely to fall into a local minima if this starting position is far from the global minimum. Following other image-based approaches [10], we use a multi-resolution to lessen the dependence on a good starting position. We start the optimization on down-sampled images and a corresponding down-sampled mesh. After convergence the mesh resolution is increased and the conjugate descent optimization is run again. After this step converges the image resolution is increased. These steps are iterated until the true resolution for the input images has been used. We use OpenGL depth buffering to compute visibility of the sample points. This operation can be expensive, so the visibility of sample points is only updated every 100 iterations or when the resolution changes.

4.4.1 Initialization

To further avoid local minima and because our method requires a good initialization for convergence we provide an approximate initialization at the lowest resolution. Specifically, we use a sampling technique that is similar to the other displacement recovery methods [21]. That is, the cost function is evaluated at a discrete set of displacements for each sample point independently.

There is no notion of a current surface in this approach, so we must approximate some measurements. First, we approximate visibility during this stage by using the visibility of the base mesh. Furthermore, each discrete sample of the cost function for each sample point requires a surface normal. As there is no implied surface at this stage, the best we can do is assume that the normal can be arbitrary. Therefore, for each sample point and each discrete depth label we must fit a surface normal that reduces the F_{data} cost. The residual of this fitting is the approximate cost function. The optimization problem is now discrete, with the goal of recovering a set of depth labels that reduce the approximated cost function. This sampled cost function is easily incorporated with the smoothness term and optimized using existing combinatorial optimization techniques, such as graph-cuts or loopy belief propagation. In our case, we adapted the expansion graph-cut proposed by Boykov et al. [3] for the displacement map representation (the original algorithm estimates a disparity map).

To summarize this section, our algorithm takes a low resolution base mesh along with the images and calibration data as input. The downsampled images are created, and the corresponding low resolution sample points are obtained using the subdivision method that was outlined in Section 4.1. The sample displacements on the low resolution mesh are then initialized with the method outlined above (Section 4.4.1). Finally, the multi-resolution optimization is performed. The algorithm outputs the recovered displacements and the albedo at each sample point on the base mesh.

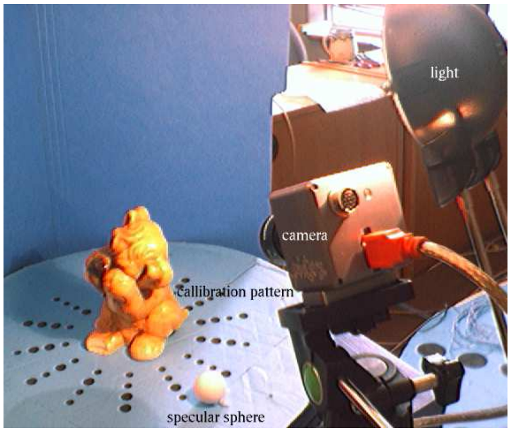

Figure 3: An image of the capture setup used in the experiments. The light is placed nearby the camera. We used a calibration pattern to calibrate the camera and a white specular sphere to calibrate the light position.

5 System

We now discuss the practical elements required in providing the input to our system. Recall that we need a base mesh, image calibration matrices, and a light direction for each frame. Furthermore, the light directions need to vary sufficiently during the capture to ensure that the reflectance parameters can be fit to each displaced surface point.

To provide these requirements we use a turntable based setup, which is illuminated by a regular desk lamp located by the camera (see Fig. 3). Within this setup we use a planar dot-based pattern for the calibration [1]. A blue-screening approach is used to obtain silhouettes that are then used as input to shape-from-silhouette to provide an initial shape. Other computer vision techniques for sparse structure could be substituted for shape-from-silhouette. As shape-from-silhouette may give a fairly dense mesh, we then use the simplification method of Cohen-Steiner et al. to obtain a low polygon base mesh [4].

Notice that the camera is fixed relative to the light positions, implying that the light is actually moving with respect to the turntable coordinate system. To avoid shadows the light source is positioned near the camera, and we capture two full rotations at different heights to ensure that there is sufficient light variation. A glossy white sphere rotates on the turntable and is used to calibrate the the exact position of the light source (i.e., our desk lamp cannot be positioned exactly at the camera center) as well as to estimate the color of the source. Several spheres are not necessary as the rotation of the sphere gives several temporal views. We triangulate the position of the light source using the specular highlight on the sphere. From this point light source we can obtain the direction to the source for any 3D point in the scene to use in the rendering function. Assuming that the sphere is white, after we know the position of the sphere and the position of the light source, each non-specular pixel observation of the sphere gives an equation in the 2 unknowns (ambient light and light source color) for each color channel.

6 Experiments

In a first experiment we demonstrate that the method is accurate on synthetic objects with varying degrees of texture (Figure 4). We have simulated the capture setup using a synthetic object texture mapped with two different textures. In each case the concavities of the object were accurately recovered. We have also computed the distance from the recovered displacements to the ground truth displacements (we used ray tracing to recover the ground truth displacements from a triangulated mesh). A color mapped version of this distance on the ground truth is also portrayed in Figure 4. The reconstruction is quite good with an average geometric error is about 0.0327 units on an object of size 2 units (1.6% accuracy).

The results of the refinement on two real objects are shown in Figure 5 [2]. Notice that in each case the base mesh and the initial displacements (using the method described in Section 4.4.1) are coarse approximations of the true object. After the displacements are refined, the fine scale detail of the 3D geometry becomes apparent. Notice the method performs well on the face sequence, which would be considered hard to capture with traditional methods. The face also exhibits a surface reflectance that deviates from the Lambertian model, but our Lambertian cost function is successful at recovering visually appealing displacements.

7 Conclusions

We have presented a method for reconstructing the displacements from a base mesh using image data. The method couples the surface normals in the cost function, implying that surface shading cues influence the reconstruction. The experiments validated the effectiveness of this approach on objects with uniform and non-uniform Lambertian surface properties. Experimental evidence also suggests that our method is capable of dealing with slight reflectance deviations from the Lambertian model (e.g., the face example).

Currently, one of the main limitations of this approach is the influence that the base mesh has on the final results. For example, the recovered mesh is restricted to have the same topology as the base mesh. Also, it is possible that the true surface cannot be approximated by displacement mapping the base mesh. In future work, we would like to address these problems by refining the base mesh during the optimization.

A Gradient Terms

$$\tfrac{\partial}{\partial \mu_k} I_i(\Pi(\mathbf{P}_i \hat{\mathbf{p}}_j)) = 0 \quad \text{for } k \neq j \qquad (11)$$

$$\mathbf{P}_i = \begin{bmatrix} p^i_{00} & p^i_{01} & p^i_{02} & p^i_{03} \\ p^i_{10} & p^i_{11} & p^i_{12} & p^i_{13} \\ p^i_{20} & p^i_{21} & p^i_{22} & p^i_{23} \end{bmatrix} \qquad (12)$$

$$\begin{aligned} \frac{\partial}{\partial \mu_k} I_i(\Pi(\mathbf{P}_i \hat{\mathbf{p}}_j)) &= \nabla I_i(\Pi(\mathbf{P}_i \hat{\mathbf{p}}_j)) \cdot \nabla \Pi(\mathbf{P}_i \hat{\mathbf{p}}_j) \\ &= \nabla I_i \begin{bmatrix} (p^i_{00}\mathbf{d}_{j,x} + p^i_{01}\mathbf{d}_{j,y} + p^i_{02}\mathbf{d}_{j,z})/w \\ (p^i_{10}\mathbf{d}_{j,x} + p^i_{11}\mathbf{d}_{j,y} + p^i_{12}\mathbf{d}_{j,z})/w \end{bmatrix} \\ &- \nabla I_i \begin{bmatrix} u(p^i_{20}\mathbf{d}_{j,x} + p^i_{21}\mathbf{d}_{j,y} + p^i_{22}\mathbf{d}_{j,z})/w^2 \\ v(p^i_{20}\mathbf{d}_{j,x} + p^i_{21}\mathbf{d}_{j,y} + p^i_{22}\mathbf{d}_{j,z})/w^2 \end{bmatrix} \end{aligned}$$

with $[u, v, w]^\top = \mathbf{P}_i \hat{\mathbf{p}}_j$. In practice a Gaussian blurred version of the image gradient is substituted for ∇I_i.

$$\begin{aligned} \frac{\partial}{\partial \mu_k}\left(\frac{\mathbf{n}_j \cdot \mathbf{l}_i}{|\mathbf{n}_j|}\right) &= \frac{1}{|\mathbf{n}_j|}\frac{\partial \mathbf{n}_j \cdot \mathbf{l}_i}{\partial \mu_k} + \mathbf{n}_j \cdot \mathbf{l}_i \frac{\partial}{\partial \mu_k}\frac{1}{\sqrt{\mathbf{n}_j \cdot \mathbf{n}_j}} \\ &= \frac{(\nabla_{\mu_k}\mathbf{n}_j)\cdot \mathbf{l}_i}{|\mathbf{n}_j|} - \frac{\mathbf{n}_j \cdot \mathbf{l}_i}{|\mathbf{n}_j|^3}((\nabla_{\mu_k}\mathbf{n}_j)\cdot \mathbf{n}_j) \end{aligned}$$

Recall that:

$$\mathbf{n}_j = \sum_{j,k,i \in \Delta(\mathbf{p}_j)} (\hat{\mathbf{p}}_k - \hat{\mathbf{p}}_j) \times (\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j)$$

Letting $\mathbf{e}_1 = (\hat{\mathbf{p}}_k - \hat{\mathbf{p}}_j) = (x_k - x_j, y_k - y_j, z_k - z_j)$ and $\mathbf{e}_2 = (\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_j) = (x_i - x_j, y_i - y_j, z_i - z_j)$,

$$\begin{aligned} \mathbf{e}_1 \times \mathbf{e}_2 &= \begin{bmatrix} \mathbf{e}_{1y}\mathbf{e}_{2z} - \mathbf{e}_{2y}\mathbf{e}_{1z} \\ -\mathbf{e}_{1x}\mathbf{e}_{2z} + \mathbf{e}_{2x}\mathbf{e}_{1z} \\ \mathbf{e}_{1x}\mathbf{e}_{2y} - \mathbf{e}_{2x}\mathbf{e}_{1y} \end{bmatrix} \\ &= \begin{bmatrix} y_k z_i - y_k z_j - y_j z_i - y_i z_k + y_i z_j + y_j z_k \\ -(x_k z_i - x_k z_j - x_j z_i - x_i z_k + x_i z_j + x_j z_k) \\ x_k y_i - x_k y_j - x_j y_i - x_i y_k + x_i y_j + x_j y_k \end{bmatrix} \end{aligned}$$

Therefore

$$\nabla_{\mu_j}\mathbf{n}_j = \sum_{j,k,i \in \Delta(\mathbf{p}_j)} \begin{bmatrix} (y_i - y_k)\mathbf{d}_{j,z} + (z_k - z_i)\mathbf{d}_{j,y} \\ (z_i - z_k)\mathbf{d}_{j,x} + (x_k - x_i)\mathbf{d}_{j,z} \\ (x_i - x_k)\mathbf{d}_{j,y} + (y_k - y_i)\mathbf{d}_{j,x} \end{bmatrix}$$

[2] The human head data set was obtained by placing the calibration pattern upside down on the subject's head while the subject rotated on an office chair.

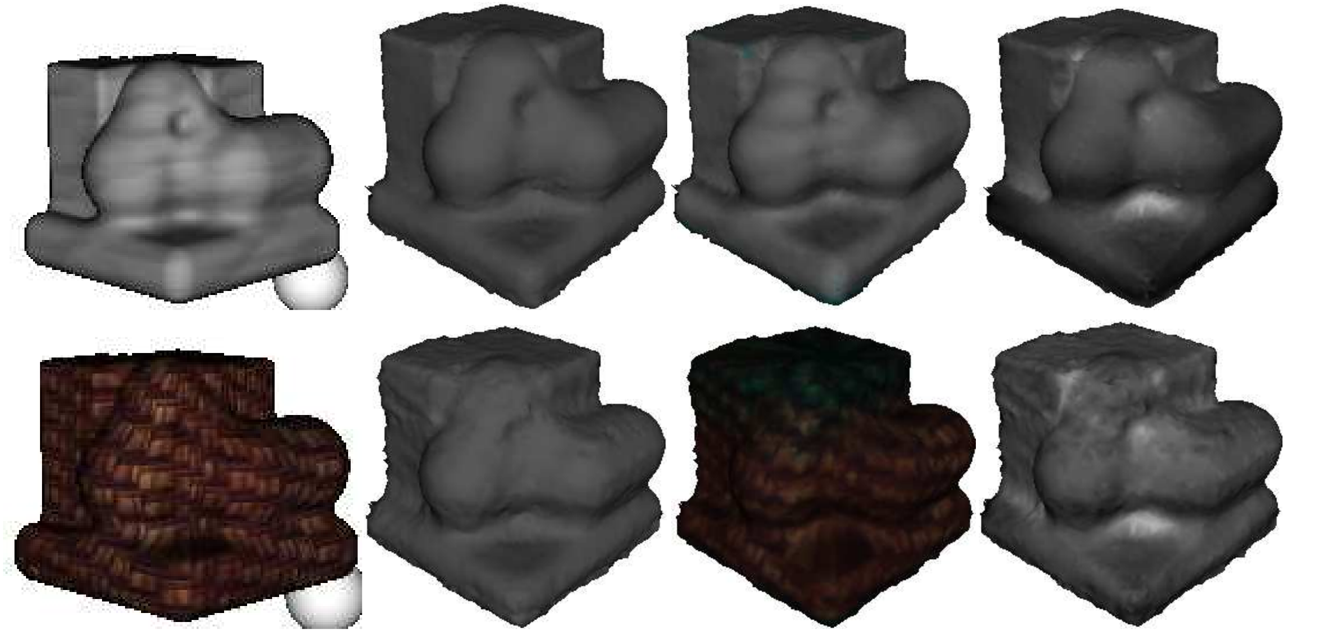

Figure 4: A synthetic object with different textures used in the experiments. From left to right: an input image, the recovered shaded model, the recovered textured model, and distance to ground truth (dark means close, white means far). The average distance of the recovered displacement to the ground truth was 0.0381 and 0.0274 for the untextured and textured sequences respectively (the size of the whole object is around 2 units).

The derivation of $\nabla_{\mu_k}\mathbf{n}_j$ and $\nabla_{\mu_i}\mathbf{n}_j$ are similar.

Finally, using the definition of $\mathbf{A}_j$,$\mathbf{I}_j$ and α_j from Equations 8 and 9, we have

$$\frac{\partial \alpha_j}{\partial \mu_k} = \frac{1}{\mathbf{A}_j \cdot \mathbf{A}_j}\frac{\partial}{\partial \mu_k}(\mathbf{A}_j \cdot \mathbf{I}_j)$$

$$- \frac{\mathbf{A}_j \cdot \mathbf{I}_j}{(\mathbf{A}_j \cdot \mathbf{A}_j)^2}\frac{\partial}{\partial \mu_k}(\mathbf{A}_j \cdot \mathbf{A}_j)$$

$$\frac{\partial}{\partial \mu_k}(\mathbf{A}_j \cdot \mathbf{I}_j) = \frac{\partial}{\partial \mu_k}\sum_i \left(\ell_i \frac{\mathbf{n}_j \cdot \mathbf{l}_i}{|\mathbf{n}_j|} + a_i\right) I_i(\Pi(\mathbf{P}_i \hat{\mathbf{p}}_j))$$

$$= \sum_i I_i(\Pi(\mathbf{P}_i \hat{\mathbf{p}}_j))$$

$$\left(\ell_i \frac{\partial}{\partial \mu_k}\left(\frac{\mathbf{n}_j \cdot \mathbf{l}_i}{|\mathbf{n}_j|}\right) + \left(\ell_i \frac{\mathbf{n}_j \cdot \mathbf{l}_i}{|\mathbf{n}_j|} + a_i\right)\frac{\partial}{\partial \mu_k} I_i(\Pi(\mathbf{P}_i \hat{\mathbf{p}}_j))\right)$$

$$\frac{\partial}{\partial \mu_k}(\mathbf{A}_j \cdot \mathbf{A}_j) = \frac{\partial}{\partial \mu_k}\sum_i \left(\ell_i \frac{\mathbf{n}_j \cdot \mathbf{l}_i}{|\mathbf{n}_j|} + a_i\right)^2$$

$$= \sum_i 2\ell_i \left(\ell_i \frac{\mathbf{n}_j \cdot \mathbf{l}_i}{|\mathbf{n}_j|} + a_i\right)\frac{\partial}{\partial \mu_k}\left(\frac{\mathbf{n}_j \cdot \mathbf{l}_i}{|\mathbf{n}_j|}\right)$$

References

[1] Adam Baumberg, Alex Lyons, and Richard Taylor. 3D S.O.M. - a commercial software solution to 3d scanning. In *Vision, Video, and Graphics (VVG'03)*, pages 41–48, July 2003.

[2] Neil Birkbeck, Dana Cobzas, Peter Sturm, and Martin Jagersand. Variational shape and reflectance estimation under changing light and viewpoints. In *ECCV2006*, 2006.

[3] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[4] David Cohen-Steiner, Pierre Alliez, and Mathieu Desbrun. Variational shape approximation. *ACM Trans. Graph.*, 23(3):905–914, 2004.

[5] Robert L. Cook. Shade trees. In *SIGGRAPH '84: Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 223–231, New York, NY, USA, 1984. ACM Press.

[6] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs. In *SIGGRAPH*, 1996.

[7] Y. Duan, L. Yang, H. Qin, and D. Samaras. Shape reconstruction from 3d and 2d data using pde-based deformable surfaces. In *ECCV*, 2004.

[8] O. Faugeras and R. Keriven. Variational principles, surface evolution, pde's, level set methods and the stereo problem. *IEEE Trans. Image Processing*, 7(3):336–344, 1998.

[9] P. Fua. Reconstructing complex surfaces from multiple stereo views. In *ICCV*, pages 1078–1085, 1995.

[10] P. Fua and Y. Leclerc. Object-centered surface reconstruction: combining multi-image stereo shading. In *Image Understanding Workshop*, pages 1097–1120, 1993.

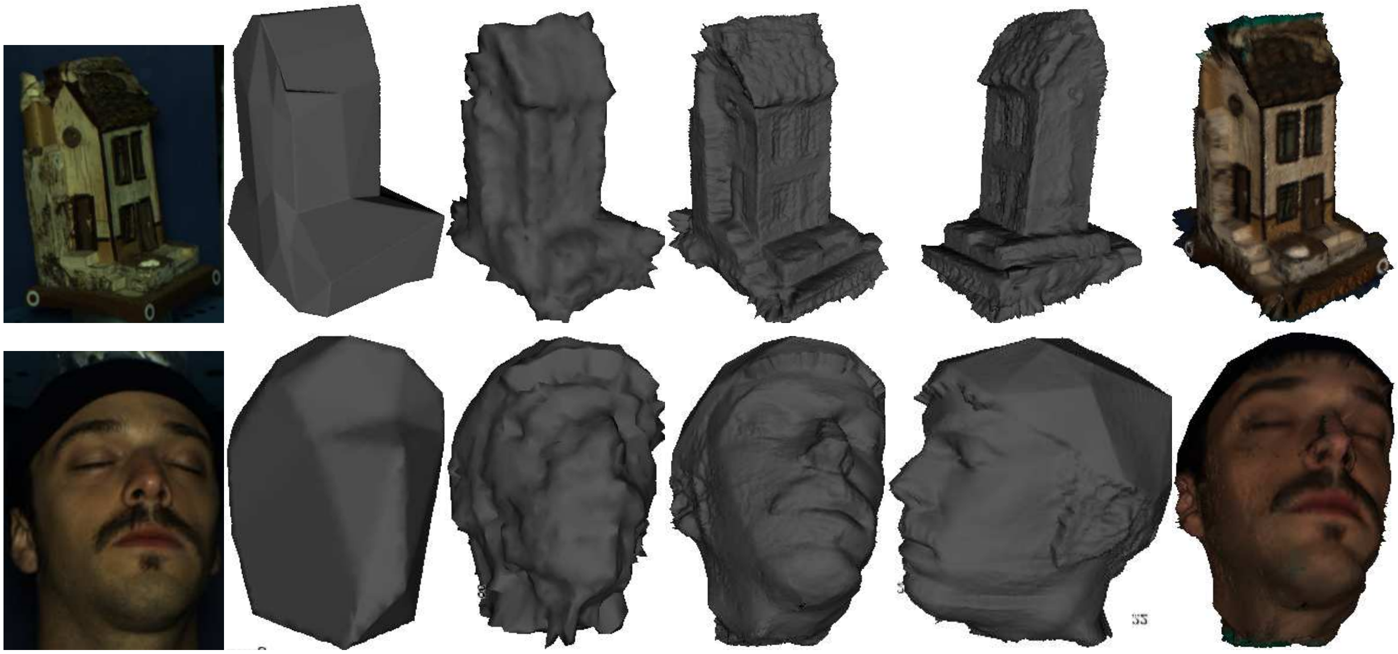

Figure 5: The results of the displacement map estimation for a model house made from wood and bark and a human face. From left to right, an input image, the base mesh, the low-resolution starting point (using discrete optimization), two shaded views of the object, and a textured lit rendering of the recovered object.

[11] Pau Gargallo and Peter Sturm. Bayesian 3d modeling from images using multiple depth maps. In *CVPR*, pages 885–891, 2005.

[12] C. Hernández and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. *Computer Vision and Image Understanding, special issue on 'Model-based and image-based 3D Scene Representation for Interactive Visualization'*, 96(3):367–392, December 2004.

[13] William A. Hoff and Narendra Ahuja. Surfaces from stereo: Integrating feature matching, disparity estimation, and contour detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(2):121–136, 1989.

[14] H. Jin, A. Yezzi, and S. Soatto. Stereoscopic shading: Integrating shape cues in a variational framework. In *CVPR*, pages 169–176, 2000.

[15] Jan Kautz and Hans-Peter Seidel. Hardware accelerated displacement mapping for image based rendering. In *Proccedings of Graphics interface*, pages 61–70, 2001.

[16] J.-O. Lachaud and A. Montanvert. Deformable meshes with autom. topology changes for coarse-to-fine 3D surface extraction. *Medical Im. Anal.*, 3(2):187–207, 1999.

[17] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, 2002.

[18] Gregory G. Slabaugh, W. Bruce Culbertson, Thomas Malzbender, and Ronald W. Schafer. A survey of methods for volumetric scene reconstruction from photographs. In *International Workshop on Volume Graphics*, 2001.

[19] S. Soatto, A. Yezzi, and H. Jin. Tales of shape and radiance in multi-view stereo. In *ICCV*, 2003.

[20] C. Strecha, T. Tuytelaars, and L.J. Van Gool. Dense matching of multiple wide-baseline views. In *ICCV*, pages 1194–1200, 2003.

[21] George Vogiatzis, Philip Torr, Steve Seitz, and Roberto Cipolla. Reconstructing relief surfaces. In *BMVC*, 2004.

[22] Martin Weber, Andre Blake, and Roberto Cipolla. Towards a complete dense geometric and photometric reconstruction under varying pose and illumination. In *BMVC*, 2002.

[23] G. Zeng, S. Paris, L. Quan, and M. Lhuillier. Surface reconstruction by propagating 3d stereo data in multiple 2d images. In *ECCV*, 2004.

[24] Gang Zeng, Sylvain Paris, Long Quan, and Francois Sillion. Progressive surface reconstruction from images using a local prior. In *ICCV*, pages 1230–1237, 2005.

[25] L. Zhang, B. Curless, A. Hertzmann, and S. M. Seitz. Shape and motion under varying illumination: Unifying structure from motion, photometric stereo, and multi-view stereo. In *ICCV*, 2003.