Passive Photometric Stereo from Motion*

Jongwoo Lim[†⋆] Jeffrey Ho[‡] Ming-Hsuan Yang[⋆] David Kriegman[††]

[†] CS Department, University of Illinois, Urbana, IL 61801
[‡] CISE Department, University of Florida, Gainesville, FL 32611
[⋆] Honda Research Institute, 800 California St, Mountain View, CA 94041
[††] CSE Department, University of California, La Jolla, CA 92093

Abstract

We introduce an iterative algorithm for shape reconstruction from multiple images of a moving (Lambertian) object illuminated by distant (and possibly time varying) lighting. Starting with an initial piecewise linear surface, the algorithm iteratively estimates a new surface based on the previous surface estimate and the photometric information available from the input image sequence. During each iteration, standard photometric stereo techniques are applied to estimate the surface normals up to an unknown generalized bas-relief transform, and a new surface is computed by integrating the estimated normals. The algorithm essentially consists of a sequence of matrix factorizations (of intensity values) followed by minimization using gradient descent (integration of the normals). Conceptually, the algorithm admits a clear geometric interpretation, which is used to provide a qualitative analysis of the algorithm's convergence. Implementation-wise, it is straightforward, being based on several established photometric stereo and structure from motion algorithms. We demonstrate experimentally the effectiveness of our algorithm using several videos of hand-held objects moving in front of a fixed light and camera.

1. Introduction

In this paper, we propose a simple and efficient algorithm for shape reconstruction of moving rigid 3D objects from videos. We assume that the camera is orthographic and that the object can be segmented from the background in each frame. The object is also assumed to have Lambertian reflectance with at least a few distinctive feature points that can be tracked throughout the video sequence in order to determine the extrinsic camera parameters. Furthermore, the scene illumination is assumed to be constant, although this condition can be relaxed somewhat. Relative to the fixed illumination source, the object's movement induces illumination changes in the image sequence, and this induced illumination variation is the major cue that we exploit in our shape reconstruction algorithm.

*Support was provided by NSF grants IIS-0308185, EIA-0224431 and CCR 00-86094.

Shape recovery from multiple images has, of course, a long tradition in computer vision. The problem has been studied from various perspectives over the years, and two of which directly related to our work are multi-view stereo (e.g., [5]) and photometric stereo (e.g., [16]). From multi-view stereo, we know how to recover the 3D position of a scene point (point on the object's surface) if the pixel correspondences across different frames are known. In particular, points in textured regions are generally less difficult to handle because their correspondences can be estimated relatively accurately compared to points in textureless regions. For the latter points, pixel correspondences are not easily computable, and in a dynamic environment where illumination changes, many cherished methods for intensity matching, such as those based on the brightness constancy assumption, are usually not effective if they are valid at all.

However, with a fixed viewpoint, photometric stereo recovers the 3D shape using images taken under different lighting. In particular, photometric stereo is able to reconstruct points in textureless regions if images with sufficient illumination variation are provided. The algorithm introduced in [16] recovers the shape by computing the depth values relative to the image plane. The main idea is to compute a rank three factorization of the intensity matrix I, which is an m-by-n matrix, where m is the number of pixels in an image and n is the number of images. Under the Lambertian assumption, I factors as the product of normals and lighting directions: $I = B \cdot L$. The rows of B are the normals and the columns of L are the lighting directions. More work is needed to ensure that the normal vectors represented in B are indeed integrable. Once an integrable normal vector field is obtained, the depth values can be computed by integrating the normals.

Unfortunately, the algorithm by itself cannot be generalized immediately to handle images taken under different views, and the chief obstacle is that in the multi-view setting, it is no longer possible to compute the intensity matrix I directly because we do not know the pixel correspondences across different images. On the other hand, if we know the correspondences between pixels, then given camera parameters, the 3D shape can be recovered without any input from photometric stereo. However, it is then an interesting question to ask if given a rough (inaccurate) cor-

respondences between pixels across frames, whether photometric stereo can improve the estimates of the pixel correspondences. The point, of course, is that given such pixel correspondences, we can compute an intensity matrix using the correspondences, and the entire machinery of photometric stereo then comes alive to produce new correspondences using the recovered depth values.

Answering the question above is the main motivation of this paper. Surprisingly enough, the answer appears to be affirmative. The idea is to start with some initial surface estimate (depth values) S_1, which is assumed to be not too far away from the true surface. Given any hypothesized surface S and the camera parameters for each frame, we can compute the intensity matrix I without any difficulty. Therefore, an integrable normal vector field can be estimated, and a new surface S' can then be computed based on the estimated normals. We can repeat this process to produce a sequence of surfaces $S_1, \cdots, S_n, ..$, and of course, the hope is that the true surface is the limit of this sequence. In the experiments reported below, this is exactly what we have observed. This suggests that a simple and efficient algorithm based on a sequence of matrix factorization followed by integration of normal vector field is able to recover the shape of a moving Lambertian object. This is somewhat of a surprise, and it clearly begs for an explanation. Foremost, is it even realistic to expect the sequence $S_1, \cdots, S_n, ..$ produced by the algorithm actually converge? In the second part of this paper, we will provide a qualitative argument that explains this observed convergence.

This paper is organized as follows. We detail the proposed algorithm in the next section, and experimental results are shown in Section 3. Section 4 contains a detailed comparison between our algorithm and other recently published methods, and Section 5 contains a discussion on the convergence issues related to our algorithm.

2. Reconstruction Algorithm

In this section, we detail our reconstruction algorithm. Let $\{I_1, \cdots, I_j, \cdots, I_F\}$ denote the input sequence of F images. Fix a coordinate system centered at the object, and in the following, we will express all vectorial quantities using this coordinate system. We assume that the object is both rigid and Lambertian; furthermore, the scene illumination is assumed to be modelled by a constant ambient illumination plus a directional source which can vary across frames. The observed intensity of a point x at frame j on the object's surface is given by the following equation:

$$I_j(p) = \alpha + \rho(p) L_j \cdot \vec{N}_p, \tag{1}$$

where $\vec{N}_p$ is the unit normal vector at p, and $\rho(p)$ is the albedo. L_j is the directional light for frame j, and α accounts for the homogeneous ambient illumination.

The proposed algorithm is shown in Figure 1. The algorithm essentially has two parts. The first part, which is both traditional and indispensable, estimates the camera parameters from a few tracked feature points using a standard structure from motion techniques (e.g., [13]). This gives us F orthographic projections P_j, $1 \leq j \leq F$, for each image I_j. Tomasi-Kanade's factorization algorithm also estimates the 3D positions of these feature points, and a *piecewise planar* surface, S_0, is constructed from these 3D points. There are many possible ways to compute such a surface. In our implementation below, the image plane of the first image, I_1, is used as the reference plane. If R is a region in I_1 containing the object, we triangulate R using points on the boundary of R and the tracked feature points. An initial depth map is then computed for every pixel in R by linearly interpolating the known depth values of the tracked feature points. The usage of these feature points in our algorithm differs slightly from some of the previous work [10][12][15]. In these work, the feature points are used mainly to estimate the camera projections and in some cases [10][15], to estimate light source directions. We go one step further by estimating a piecewise linear surface from them. Note that if we have sufficiently many feature points to track, then, the initial surface S_0 is already a good approximation of the true surface.

With the initial surface S_0 computed, the rest of the algorithm is straightforward: just let the machinery of photometric stereo run its own course. Given any surface S_t and its associated depth map $z_t(x, y)$, we can compute the intensity matrix $\mathcal{I}$, by collecting various pixel values across images into an r-by-F matrix (r is the number of pixels in R):

$$\mathcal{I}_{ij} = I_j(P_j(x_i, y_i, z_t(x_i, y_i))). \tag{2}$$

Photometric stereo [16] then provides us with a recipe for producing an integrable (normal) vector field $\vec{N}$ defined on R. The idea is to find an integrable vector field $\vec{N}$ that minimizes the following error function:

$$\sum_{i=1}^{F} \sum_{(x,y)\in R} \Big(I_j(P_j(x, y, z_t(x, y))) - \rho L_t \cdot N_p - \alpha \Big)^2 \tag{3}$$

$\vec{N}$ can be solved in a least-square sense and because it is assumed to be integrable, we can integrate it to obtain a new surface S_{t+1} and its associated depth map $z_{t+1}(x, y)$. Note that our viewpoint here is slightly different from that of [16]. In photometric stereo, one assumes that images were taken under a single fixed view and the factorization of the intensity matrix (with more processing) yields a normal vector field of the underlying surface. In our multi-view setting, because of incorrect pixel correspondences, there is, in general, *no* surface that can account for the intensity matrix. However, we can still try to find an *integrable* vector field $\vec{N}$ that minimizes the error function above. And a surface

S_{t+1} is defined to be one such that $\vec{N}$ is its (unit) normal vector field. In Section 5, we will re-interpret Equation 3 by formulating an analogous expression on the surface S.

Once the normals $\vec{N} = (N_x, N_y, N_z)$ have been estimated, the depth map $z(x, y)$ is computed by minimizing the following objective function:

$$\mathcal{E}(z(x,y)) = \sum_{x,y} \left(\frac{\partial z(x,y)}{\partial x} + \frac{N_x}{N_z} \right)^2 + \left(\frac{\partial z(x,y)}{\partial y} + \frac{N_y}{N_z} \right)^2 \quad (4)$$

Because of the Generalized Bas-Relief (GBR) ambiguity [1], which can be traced back to the least square problem in Equation 3, the depth values of the tracked feature points will be, in general, in poor agreement with the values estimated using structure from motion technique. The last step in computing S_{t+1} is to correct the depth values by a GBR transform that brings the surface closest to the tracked feature points.

This process can be repeated indefinitely, and we get a sequence of surfaces $S_0, S_1, \cdots, S_t, \cdots$. At this moment, there is no obvious reason to believe that it would converge to anything reasonable. In the next section, we will demonstrate that the algorithm does indeed converge to the "correct" surfaces most of the time, and in Section 5, we will provide a qualitative argument explaining this convergence.

3. Implementation and Experimental Results

In this section, we demonstrate some experimental results of the proposed algorithm. The algorithm is implemented in C++. The experiments are run on a laptop with 1.6GHz CPU and 512MB RAM. The feature point tracking and camera calibration steps take a couple of minutes and each iteration of the surface evolution takes about one to two minutes (depending on the number of iterations in minimizing Equation 4) for most experiments.

First, we present the reconstruction result of a paper cup. The video sequence is recorded with a Firewire CCD camera with a long focal length (for our orthographic projection assumption). The cup is moving in front of the camera in a dark room with a distant point light source. Feature points are manually chosen from the output of a feature point detector, and the feature point tracker tracks them throughout the sequence (Figure 2.a). Due to memory limitations, we do not use all frames in the video sequence. Instead, we subsampled the sequence every three frames (box and figurine sequences are sampled every five frames). The resulting estimated normal field was good enough to give a convincing result.

The initial depth map (Figure 2.b) is computed from the 3D positions of the tracked feature points. It is not close to the true surface; however, it provides rough estimates

---

Given a collection of F images (frames), I_t, indexed by $t = 1, \cdots, F$, and m scene points indexed by $p = 1, \cdots, m$ with $\mathbf{x}_{t,p} = [x_{t,p}, y_{t,p}]^t$ denoting the position of the scene point p in I_t. The algorithm produces a depth map $z(x, y)$ with respect to the image plane of the initial frame I_1.

1. **Estimate Camera Parameters**
   Using Tomasi-Kanade factorization algorithm, we can recover (up to some unknown rotation) the camera projection matrix P_t for each frame t and the 3D positions $[x_p, y_p, z_p]^t$ of the m scene points. They are represented by their respective depth values z_p with respect to the image plane of I_1.
2. **Construct an Initial Piecewise surface S_0**
   Use the 3D positions of the m scene points to compute an initial piecewise planar surface. Let R be the region in I_1 containing the object, and r denote the number of pixels in R. We compute a Delaunay triangulation of R using the projections of the m-scene points on I_1 and some points on the boundary of R. All m-scene points are assumed to be projected onto the interior of R and each triangle in the triangulation is assumed to contain at least one scene point as its vertex. We linearly interpolate the depth values across each triangle using the depth values of the vertices. If the vertex is on the boundary of R, its depth value is interpolated using its projection onto the edge spanned by vertices that are one of the m scene points. The result is a piecewise planar surface S_0.
3. **Iterate Until Converge**
   For $t = 0, \cdots$, a surface S_t and its associated depth map $z_t(x, y)$:
   (a) Compute Intensity Matrix $\mathcal{I}$: an r-by-F matrix such that
   $$\mathcal{I}_{ij} = I_j(P_j(x_i, y_i, z_t(x_i, y_i)))$$
   (b) Matrix factorization: Perform a rank three SVD on $\mathcal{I}$ to obtain $\mathcal{N}'$, an r-by-3 normal matrix and $\mathcal{L}$, a 3-by-F lighting matrix. Determine (up to an unknown GBR transform) a new $\mathcal{N}$ from $\mathcal{N}'$ that is integrable.
   (c) Integrating Normals: Determine a new depth map $\bar{z}_{t+1}$ by minimizing Equation 4. The depth map $\bar{z}_{t+1}$ defines a new surface $\bar{S}_{t+1}$.
   (d) GBR Correction: Use the known positions of the m scene points to determine the unknown GBR transform. Find a GBR transform that brings the surface $\bar{S}_{t+1}$ closest to these m scene points. S_{t+1} and z_{t+1} are then the GBR-corrected surface and depth map, respectively.

Figure 1: Shape Reconstruction Algorithm.

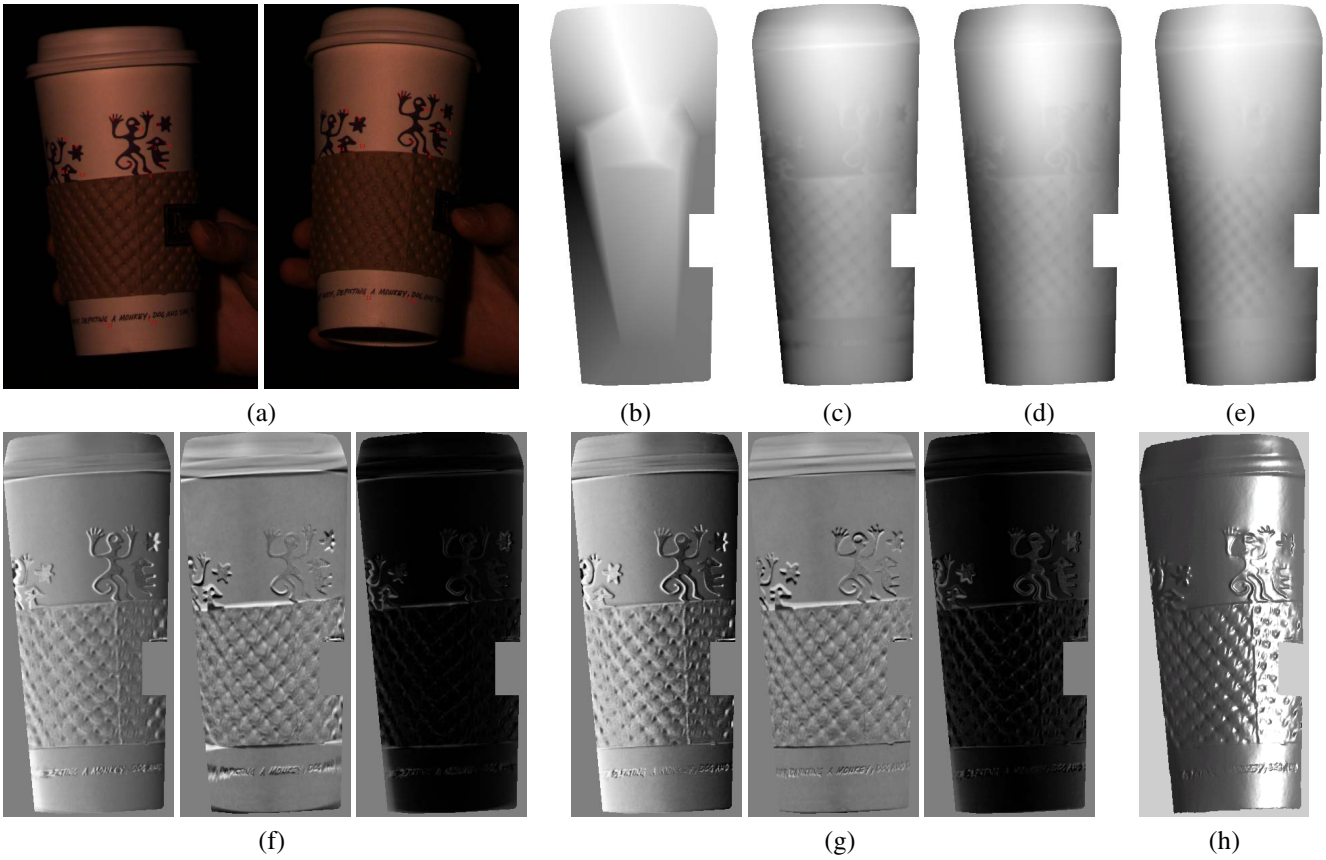
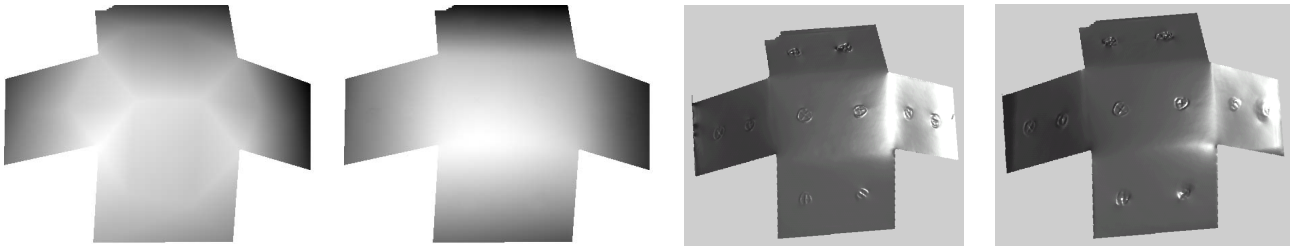


Figure 2: The cup sequence : (a) Fifteen feature points are detected and tracked throughout the sequence. (b) The initial piecewise planar surface is built from the 3D positions of the tracked feature points. A part of the holder is excluded since the region is too dark. (c) The surface constructed by integrating the normal vector field shown in (f). (d) The surface after two iterations. (e) The final estimated surface. (f) The normal vector field (N_x, N_y, N_z) computed based on the initial piecewise planar surface. For a more detailed explanation, see the text. (g) The normal vector field computed from the estimated surface shown in (c). It presents very accurate normals, and most of the problematic regions in the previous iteration have disappeared. (h) The final reconstructed surface rendered with Phong reflectance model.

for back-warping frames to the reference frame. Figure 2.f shows the estimated normals from the photometric stereo algorithm with warped images using the initial depth map. One may notice that there are some noisy normals in the textured region in the left boundary of the cup, and also the y-components of the normals along the cup lid and the lower edge of the holder change their directions due to the incorrect initial surface estimate. After integrating the normal field and correcting the GBR ambiguity, the depth map (Figure 2.c) looks much better, but there exist regions where the normals are noisy or incorrect. However the estimated surface is much closer to the true surface, and the back-warped images based on this surface produce more accurate normals than the previous ones (Figure 2.g). Integration of this normal field gives the next surface (Figure 2.d), and it is already very close to our final result (Figure 2.e and 2.h: after 4 iterations).

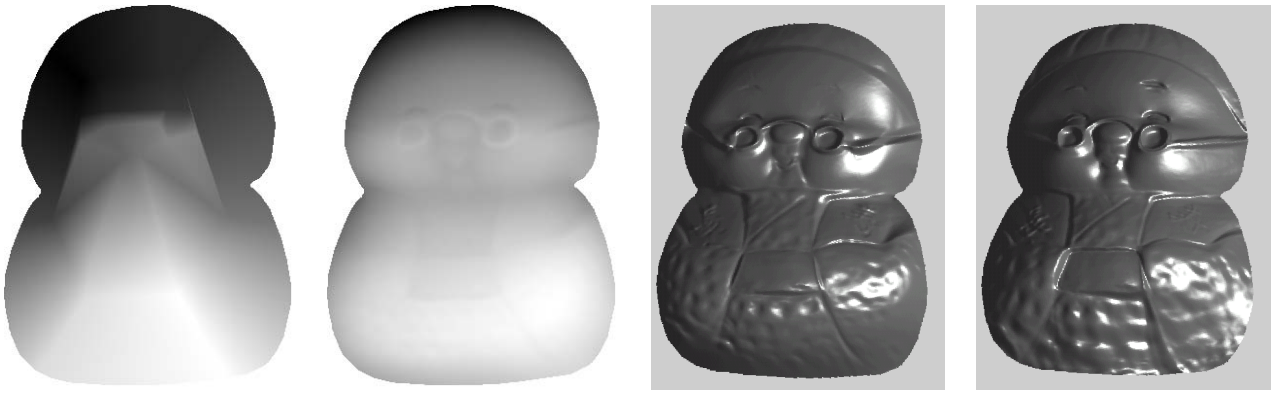
We have also applied our algorithm to the sequences presented in [15] (Figure 5). The two sequences are publicly available at the authors' website. Figure 3 shows the result of the box sequence. Thanks to the initialization, the initial surface (Figure 3.a) is already close to the true surface except that the folds between the faces are missing. Our algorithm gives a nice reconstruction result, in which each face is smooth and the folds between faces are visible and sharp. Figure 3.c and 3.d show the evolution of the surface produced by the proposed algorithm.

The figurine sequence is more interesting since the object has many fine structures on its surface, such as facial parts, eyeglasses, bumps and creases. Since the surface is complex and the textureless region is large, the initial surface does not give a good approximation of the true surface. After a few iterations, most of the details on the surface have been recovered. Notice that after the first iteration, the glasses frame is not correctly reconstructed, and the bumps in the lower part of the figurine are not clear. In the final sur-



(a) (b) (c) (d)

Figure 3: The box sequence : (a) The initial piecewise planar surface. (b) The final depth map. (c) The rendering of the surface after one iteration. There are sharp spikes along the left edge, and the folds between faces are not very clear. (d) The rendering of the final surface. The spikes are removed, and the folds become more clear.



(a) (b) (c) (d)

Figure 4: The figurine sequence : (a) The initial piecewise planar surface. (b) The final depth map. Prominent features in the facial area such as glasses and nose are clearly reconstructed. (c) The rendering of the surface after one iteration. The glasses frame is bent due to the incorrect initial depth estimates, and the right end of the vest is blurred. (d) The rendering of the final surface.

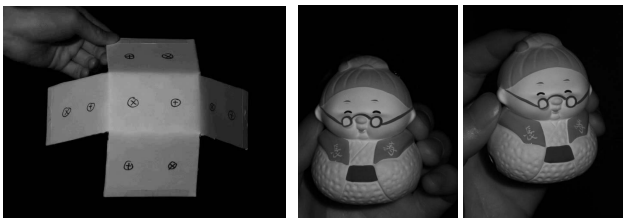


Figure 5: The original images from the box and figurine sequences. [15]

face produced after six iterations, these inaccuracies have all disappeared, and the reconstruction result looks convincing (Figure 4).

4. Comparisons with Previous Work

Our work is partially inspired by the the recent works of [12] and [15]. In this section, we will concentrate on comparing our work with these two methods. More detailed surveys of the related literature can be found in these two papers.

In [15], the problem of shape reconstruction from video sequences is addressed by reformulating the optical flow algorithm in order to better handle illumination change. Our algorithm shares two similarities with their algorithm. First, shape recovery in both algorithms is based on integration of normal vector fields. Second, both algorithms are iterative. However, the similarities are somewhat superficial, and fundamentally, the two algorithms are completely opposite of each other. The way the normals are estimated in [15] is a local process in the sense that each individual normal vector is estimated separately in a linear system that includes also the estimate of optical flow. Because of locality, it seems unlikely that the optical flows and normals can be directly estimated from full-resolution images. Therefore, in their algorithm, the iterative step is a coarse-to-fine refinement using an image pyramid, where the optical flow and the normal vector field are estimated first at a lower resolution and then successively at higher resolutions. In our algorithm, normals are estimated globally in a single matrix factorization, and we work directly with full-resolution images. It can be argued that a global estimate, instead of local estimates, is more robust against noise. Partially because of this, there is no need for a coarse-to-fine refinement in our

algorithm. Although our iterative step can also be considered as a refinement step, the refinement is over the entire surface at full resolution.

The reconstruction algorithm proposed in [12] is based on stereo matchings. The algorithm assumes knowledge of the relative motion between the object and the illumination source. The idea is that this knowledge can be exploited to define a correspondence measure that is insensitive to illumination change. The shape is then recovered by minimizing an energy function defined on some graph using these correspondence measures. Because normals are not estimated, textureless planar regions can not be resolved by this algorithm. In an environment with fixed lighting *and* camera, relative motion between the object and the illumination source can be computed from the relative motion between the object and the camera. However, in the more general setting when both the camera and object (and possibly the illumination source) are moving, it is not clear to us how the relative motion between the object and the illumination source can be estimated directly from images. Presumably, one can estimate the light source directions using a few feature points as in [10][15] and this estimate could be used to determine the relative motion between the object and light source. However, in general, there is an unresolved GBR amguity in the lighting directions [1], and it is unclear how this ambiguity can be resolved in the algorithm proposed in[12].

5. Convergence Analysis

In this section, we give a qualitative argument showing that the convergence of our algorithm demonstrated in the previous section is not fortuitous. The detailed quantitative analysis of the convergence question is beyond the scope of this paper;

To this end, we will first formulate the reconstruction problem following the usual variational approach [3][8] in multi-view stereo. As before, let $\{I_1, \cdots, I_F\}$ denote the input collection of F images. Let $\{P_1, \cdots, P_F\}$ denote the F orthogonal projections of points in $\mathbb{R}^3$ to the images $\{I_1, \cdots, I_F\}$, respectively. Each pair of (I_i, P_i) defines a function $\mathbf{Im}_i : \mathbb{R}^3 \to \mathbb{R}$: $\mathbf{Im}_i(x) = I_i(P_i(x))$. In the usual variational approach, the reconstructed surface S should be a (local) minimum of the following functional:

$$\begin{aligned} \mathcal{E}(S) &= \int_S \sum_{i=1}^{F} (\mathbf{Im}_i(x) - L_i \cdot \vec{N})^2 dA_S \\ &= \int_S \Phi(X, N) dA_S \end{aligned} \quad (5)$$

where $\vec{N}$ is the unit normal vector field of S, and L_i is the lighting direction at frame i. We use $\Phi(X, N)$ to denote the integrand in the integral above with X denoting the spatial variables in $\mathbb{R}^3$ and N denoting the surface normal (see

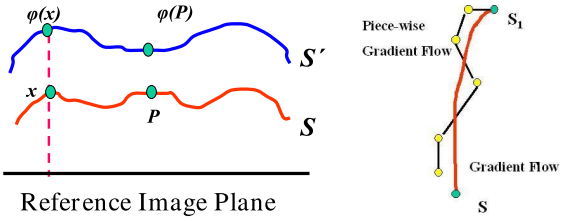


Figure 6: (a) Definition of φ. We use the image plane of the first image to define correspondences between points on S and S'. (b) Comparison of Flows: The smooth red curve denotes the smooth gradient flow according to Equation 7. The "piecewise" flow followed by our algorithm is denoted by the black line segments.

[3]). To simplify the discussion, we have assumed the following: 1) we know the lighting direction at each frame, 2) the albedos are constant (with value 1). These simplifications have no effect on the convergence question. We remind the reader that although we are following the usual formulation by working directly with the surface $S \subset \mathbb{R}^3$, our algorithm actually operates only on the part of the surface S that can be parametrized by the image plane.

Given the functional $\mathcal{E}$ above, the usual PDE approach to solve the reconstruction problem is to start with an initial surface S, and compute a surface evolution S_t, $t \geq 0$, with $S_0 = S$, that is the gradient flow of $\mathcal{E}$. For $\mathcal{E}$ given in Equation 5, its gradient is [3]

$$\begin{aligned} \nabla\mathcal{E} &= (\Phi_X N - 2H(\Phi - \Phi_N N) + \\ &\quad \mathbf{Tr}((\Phi_{XN})_{TS} + dN \circ (\Phi_{NN})_{TS})\vec{N} \end{aligned} \quad (6)$$

where H denotes the mean curvature of S, dN the derivative of the Gauss map, and **Tr** denotes the trace of the linear transform. TS denotes the tangent spaces of S. The gradient flow for $\nabla\mathcal{E}$ is simply the following PDE that is usually solved using the level-set method:

$$\begin{aligned} \frac{\partial S_t}{\partial t} &= -\nabla\mathcal{E} = -(\Phi_X N - 2H(\Phi - \Phi_N N) \\ &\quad + \mathbf{Tr}((\Phi_{XN})_{TS} + dN \circ (\Phi_{NN})_{TS})\vec{N} \end{aligned} \quad (7)$$

We will take as a faith that the gradient flow above will converge to a critical point of $\mathcal{E}$ given a (sufficiently nice) initial surface S. With this, our argument for convergence proceeds in the following two steps:

1. Given a surface S, we interpret Equation 3 (with $\rho = 1$ and $\alpha = 0$) geometrically in terms of some new functional $\mathcal{E}'_S$. In fact, $\mathcal{E}'_S$ will be a modified version of $\mathcal{E}$ defined in Equation 5, and $\mathcal{E}'_S$ will be defined only for surfaces "close" to S. In the following, we will omit the subscript S and denote $\mathcal{E}'_S$ by $\mathcal{E}'$.

2. The gradient vector $\nabla\mathcal{E}'$ at S can then be considered as a (small) perturbation of $\nabla\mathcal{E}$ at S.

Note that we claim only that $\nabla\mathcal{E}'$ is a small perturbation of $\nabla\mathcal{E}$ at S (and hence also in a small neighborhood around S). We make no claim that globally (at every surface S) $\nabla\mathcal{E}'$ is a small perturbation of $\nabla\mathcal{E}$. Schematically, our argument is depicted in Figure 6(Right). Let S_t be the solution to the PDE in Equation 7. The red curve in the Figure denotes the gradient flow S_t from an initial surface S_0 to a critical point S_c of $\mathcal{E}$. What we want to explain (qualitatively) is that from S_0, our algorithm is heading towards the same critical point S_c using a different path. For our algorithm, starting at S_0, it tries to find a surface S_1 (close to S_0) that minimizes the functional $\mathcal{E}'$. This is basically the re-interpretation of Equation 3 in 1) above. The point is that this minimum can be computed using a linear method (and without PDE) when phrased in the form of Equation 3. However, because of 2) above, we know that for a short time, flowing down along $-\nabla\mathcal{E}'$ (that's how we get S_1), is not going to get too far away from flowing down along $-\nabla\mathcal{E}$ because $-\nabla\mathcal{E}'$ is a (small) perturbation of $\nabla\mathcal{E}$. In the figure, this is indicated by the jagged path. Instead of flowing down smoothly to S_c, at each S_i, we move down along a small perturbation of the gradient for a short time to get S_{i+1}. Because of the small perturbation at each step, we argue that at the end, our algorithm will still flow down to the same critical point S_c as in the usual gradient flow of $\mathcal{E}$.

We can interpret Equation 3 as follows: given a surface S, we try to find another surface S' close to S, such that

1. There is a one-to-one correspondence φ between S and S'. For each point $x \in S$, $\varphi(x) \in S'$. In Equation 3, φ is given by the (x, y) on the image plane.

2. The surface S' minimizes the following functional:

$$\mathcal{E}'(S') = \int_S \sum_{i=1}^{F} (\mathbf{Im}_i(x) - L_i \cdot \vec{N}_{S'}(\varphi(x)))^2 dA_S \quad (*)$$

where the integration is on S not S', and $N_{S'}(\phi(x))$ denotes the normal vector of S' at the point $\varphi(x)$. Let $\Phi'(X, N)$ denote the integrand above. Note that the spatial part (X) of $\Phi'(X, N)$ is defined over a neighborhood $\mathcal{U}$ of S using φ. See Figure 6(Left).

Note that the integral above is what has been computed in Equation 3[1]. In both places, each normal vector of the new surface S' is determined by the image intensity values of corresponding point on the old surface S. In Equation 3, the normal vector field computed through SVD is in general not integrable. Instead, we estimate a near-by integrable vector field, and this allows us to interpret the resulting surface S' as the one that minimizes $(*)$ above. The comparison between our approach and that of [7] is quite interesting. The problem studied in [7] is very similar to ours, and in principle, their solution surface is a critical point of the functional $\mathcal{E}$ in Equation 5. Because of negative curvature flow which can cause numerical instability, a straightforward level-set implementation for solving the PDE in equation 6 is not feasible. So [7] modifies $\mathcal{E}$ by including an auxiliary (unit) vector field V. Although the energy functional proposed in [7] is a little complicated, basically, the vector field V is "determined" by the photometric data (the image intensities and lighting) while the surface normals is a "near-by" vector field of V. This is clearly very similar to our discussion above. The difference, however, is that their modified energy function again leads to a PDE solution of the problem. Our (locally) modified problem (Equation (*) above) leads to a simple linear least-square solution.

What's left now is to compute the gradient of $\mathcal{E}'$ at S:

$$\begin{aligned}\nabla\mathcal{E}' &= (\Phi'_X N + 2H\Phi'_N N + \\ &\quad \mathbf{Tr}((\Phi'_{XN})_{TS} + dN \circ (\Phi'_{NN})_{TS}))\vec{N} \quad (8)\end{aligned}$$

Since $\mathcal{E}'(S')$ involves only the integral over S (not S'), the corresponding term $2H\Phi'$ in Equation 6, which comes from the area variation, is not present in the gradient of $\mathcal{E}'$. Note also that, at S, $\Phi'_N = \Phi_N$ as well as $\Phi'_{NN} = \Phi_{NN}$. Therefore, at S, the difference $\nabla\mathcal{E}' - \nabla\mathcal{E}$ is

$$(2H\Phi + (\Phi'_X - \Phi_X)N + \mathbf{Tr}((\Phi'_{XN} - \Phi_{XN})_{TS}))\vec{N}. \quad (9)$$

Our task now is to show that the magnitude of this term is relatively small compared to $\nabla\mathcal{E}$. Since our algorithm starts with a piecewise planar surface S_1. Therefore, at the initial surface S_0, $-2H\Phi = 0$. In general, when we are close to the true surface, the curvature H will of course no longer be zero. However, the term Φ will be small and therefore, $-2H\Phi$ is also small as well provided that the curvature H is bounded, which is usually the case. The term $\Phi'_X - \Phi_X$ (as well as $(\Phi'_{XN} - \Phi_{XN})_{TS}$) is related to the image gradients and the relative motion between the camera and the object. Using orthorgraphic camera model, Φ'_X and Φ_X can be easily computed. In our case, the motion has a rough symmetry with respect to the initial relative position between the camera and the object. That is, the object is rotated first to the right and then to the left and so on. This can be used to argue that there are some cancellations among the terms that make up Φ_X, and the difference $\Phi'_X - \Phi_X$ is usually small. To empirically verify these claims, we compute both gradient vectors $\nabla\mathcal{E}$ and $\nabla\mathcal{E}'$ for the surfaces produced during six iterations in the solution of the figurine sequence (Figure 4). The results are listed in Table 1. Our aim here is to justify that $\nabla\mathcal{E}'$ can be considered as a small perturbation of $\nabla\mathcal{E}$. Therefore, we show the "angle" between the two

[1]Strictly speaking, this would require a weighted sum of squares in Equation 3, with weights given by the area elements.

Table 1: Comparison between $\nabla\mathcal{E}$ and $\nabla\mathcal{E}'$ for the figurine sequence. $\angle(\nabla\mathcal{E}', \nabla\mathcal{E})$ is reported in degrees below.

Surface S	$\angle(\nabla\mathcal{E}', \nabla\mathcal{E})$	$\frac{\|\nabla\mathcal{E}'-\nabla\mathcal{E}\|_S}{\|\nabla\mathcal{E}\|_S}$	$\mathcal{E}(S)$
S_0	12.7°	0.225	225.98
S_1	8.9°	0.152	187.78
S_2	8.1°	0.141	186.05
S_3	4.9°	0.087	170.25
S_4	4.9°	0.088	171.95
S_5	4.0°	0.07	164.84

gradient vectors and the relative magnitude of their difference are both small. The angle $\angle(\nabla\mathcal{E}', \nabla\mathcal{E})$ between $\nabla\mathcal{E}'$ and $\nabla\mathcal{E}$ is defined as

$$\angle(\nabla\mathcal{E}', \nabla\mathcal{E}) = \cos^{-1}\left(\frac{< \nabla\mathcal{E}, \nabla\mathcal{E}' >_S}{\|\nabla\mathcal{E}\|_S \|\nabla\mathcal{E}'\|_S}\right). \qquad (10)$$

In the above, the inner product $< \nabla\mathcal{E}, \nabla\mathcal{E}' >_S$ between $\nabla\mathcal{E}$ and $\nabla\mathcal{E}'$ is defined as

$$< \nabla\mathcal{E}, \nabla\mathcal{E}' >_S = \int_S < \nabla\mathcal{E}, \nabla\mathcal{E}' >_{\mathbb{R}^3} dA, \qquad (11)$$

and $\|\nabla\mathcal{E}\|_S^2 = < \nabla\mathcal{E}, \nabla\mathcal{E} >_S$. The results in Table 1 show that it is indeed reasonable to regard $\nabla\mathcal{E}'$ as a small perturbation of $\nabla\mathcal{E}$. Note also that the value $\mathcal{E}(S)$ always decreases after each iteration.

6. Conclusions and Future Work

We have presented a method for shape reconstruction from multiple images of a moving object. Our algorithm is iterative: it starts with a rough piecewise planar estimate of the true surface, and then successively refines the estimate using photometric stereo techniques. The algorithm is simple both conceptually and implementation-wise. Results of our experiments have demonstrated the feasibility of our algorithm, and a qualitative explanation of our algorithm's convergence is also introduced.

The first item on our list of future work is a more detailed numerical analysis of the convergence issues that were raised in the previous section. A detailed study of this question, both numerically and mathematically, could potentially not only improve our algorithm but also unearth previously unknown fundamentals of 3D reconstruction and shading. One serious limitation of our algorithm is its dependence on some reference image for integrating normals. This dependency makes it awkward to generalize our algorithm to a 360-degree reconstruction of an object. What is needed here is a scheme to integrate normals directly on surfaces in $\mathbb{R}^3$ instead of on the image plane.

Finally, the reconstructions in Figs. **??** to 4 show a common problem of larger reconstruction error near image intensity discontinuities (step edges), and this arises due to inaccuracies in alignment that may accumulate during the iterative process. On the other hand, it is at just these locations where conventional structure-from-motion techniques excel. What is needed is a technique which can merge multi-view geometric constraints with the result of photometric stereo

References

[1] P. Belhumeur, D. Kriegman and A. Yuille, "The bas-relief ambiquity," *IJCV*, 35(1):33-44, 1999.

[2] A. Broadhurst, T. Drummond and R. Cipolla, "A Probabilistic framework for space carving." *Proceedings of ICCV*, pp. 388-393, 2001.

[3] O. Faugeras and R. Keriven, "Complete dense stereovision using level set methods," *Proceedings of ECCV*, pp. 379-393, 1998.

[4] A. Georghiades, D. Kriegman and P. Belhumeur, "From few to many: Generative models for recognition under variable pose and illumination", *IEEE PAMI* , 23(6), pp.643-660, 2001.

[5] R. Hartley and A. Zisserman, "Multiple view geometry in computer vision," Cambridge University Press, 2003.

[6] J. Isidoro and S. Sclaroff, "Stochastic refinement of the visual hull to satisfy photometric and silhouette consistency constraints," *Proceedings of ICCV*, pp. 1335-1342, 2003.

[7] H. Jin, D. Cremers, A. Yezzi and S. Soatto, "Shedding light on stereoscopic segmentation," *Proceedings of CVPR*, pp. 36-42, 2004.

[8] H. Jin, S. Soatto, A. Yezzi, " Stereoscopic shading: Integrating multi-frame shape cues in a variational framework," *Proceedings of CVPR*, pp. 169-177, 2000.

[9] K. Kutulakos and S. Seitz, "A theory of shape by space carving," *IJCV*, 38(3):199-218, 2000.

[10] A. Maki, M. Watanabe and C. Wiles, "Geotensity: Combining motion and lighting for 3D surface reconstruction." *IJCV*, 48(2):75-90, 2002.

[11] J. Shi and C. Tomasi, "Good feature to track," *Proceedings of CVPR*, pp. 593-600, 1994.

[12] D. Simakov and R. Basri, "Dense shape reconstruction of a moving object under arbitrary unknown lighting," *Proceedings of ICCV*, pp. 1202-1209, 2003.

[13] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography: A factorization method," *IJCV*, 9(2):137-154, 1992.

[14] L. Torresani and C. Bregler, "Space-time tracking," *Proceedings of ECCV*, pp. 801-815, 2002.

[15] L. Zhang, B. Curless, A. Hertzmann and S. Seitz, "Shape and motion under varying illumination: Unifying structure from motion, photometric stereo, and multi-view stereo," *Proceedings of ICCV*, pp. 618-625, 2003.

[16] A. Yuille and D. Snow, "Shape and albedo from multiple images using integrability," *Proceedings of CVPR*, pp.158-164, 1997.