Shape from Varying Illumination and Viewpoint

Neel Joshi David J. Kriegman
University of California, San Diego

Abstract

We address the problem of reconstructing the 3-D shape of a Lambertian surface from multiple images acquired as an object rotates under distant and possibly varying illumination. Using camera projection matrices estimated from point correspondences across views, the algorithm computes a dense correspondence map by minimizing a multi-ocular photometric constraint. Once correspondence across views is established, photometric stereo is applied to estimate a surface normal field and 3-D surface. Conceptually, the algorithm merges multi-view stereo and photometric stereo and uses aspects of both methods to recover shape. The method is straightforward to implement and relies on established principles from the two stereo methods. We empirically validate the method on images of a number of objects and show that it outperforms previous methods.

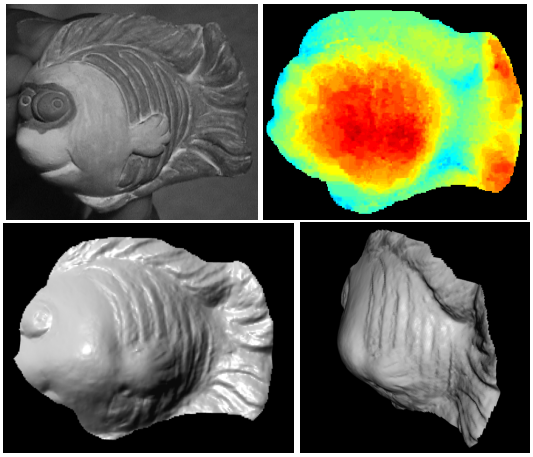

Figure 1. Shape from varying illumination and viewpoint. One of eight views of a figurine rotating it in front of a camera and light (top left). The intermediate dense depth map recovered by our method – red is closer and blue further from the camera or for a gray-scale printout white is closer and black further (top right) and renderings of the final surface (bottom row).

1. Introduction

Recovering the 3-D shape of objects from images dates back to the early days of computer vision. Over the years, researchers have developed systems using stereo, structured light, and photometric stereo to scan objects from ancient statues to human faces. These methods have produced some stunning results; however, they can be cumbersome, requiring unwieldy setups, extensive calibration procedures, and at times strict control of the environment, leaving 3-D scanning abilities out of the reach of a common person. In this paper, we explore a lightweight setup that acquires high-quality, detailed object structure using several images of an object rotating under distant, unknown, and possibly varying illumination. With this simple setup, a hand-held camera becomes a powerful high-quality 3-D scanner.

When an object rotates in front of a camera under distant and varying illumination the appearance of the object changes both geometrically and photometrically. These changes provide clues to the shape of an object; however, due to their simultaneous variation they can not be exploited by traditional methods. Since the lighting conditions change between images, standard stereo methods will fail as they assume brightness constancy for finding correspondence . Similarly, due to viewpoint change, standard photometric stereo cannot be performed since pixel correspondences are unknown across images. While this joint variation of view and lighting may appear to complicate the matter of shape acquisition, we show that it in fact enables high-quality shape reconstruction.

This primary contribution of this work is recovering shape under varying viewpoint and lighting by combining multi-view and photometric stereo to derive a multi-ocular photometric matching cost. This cost is minimized using a graphcut method to find dense correspondence between images. Once correspondence is found, photometric stereo recovers normals, and the depth map from correspondence and the normals are fused to recover high-quality 3-D shape. Relative to previous related work, we produce higher-quality reconstructions using a non-iterative method with fewer images. We present several results including comparisons on data used by two recent related papers [12, 6] and show that our results are qualitatively more accurate.

The rest of this paper proceeds as follows: in the next section, we will discuss the previous work in this area. In Section 3, we present our algorithms. We present results in Section 4 followed by a discussion and our conclusions.

2. Previous Work

Shape recovery using stereo and photometric clues has received increasing attention in recent years. Several researchers have explored extending stereo matching to han-

dle changes in shading or illumination due to object rotation. Maki *et al*. [7] use a linear subspace constraint as we do and require several known correspondences to estimate light source directions up to an arbitrary invertible 3×3 transform, but they do not recover surface normals. Our method, in contrast, requires a few known correspondences to recover camera parameters and then implicitly solves for the unknown lighting directions when recovering normals relative to a dense depth map. Simakov *et al*. [10] also merge multi-view stereo and photometric constraints by assuming that the relative motion between the object and the illumination source is known. While this is recoverable in certain situations, in the more general case when camera, object, and illumination can be moving, it is not clear how to recover the relative motion between the object and the illumination. Furthermore, their process can only recover normals up to an ambiguity along a plane. Our method makes no assumptions about light and camera motion and recovers albedo and normals unambiguously.

Our work is most similar to the work by Zhang *et al*. [12] and Lim *et al*. [6]. Both of these methods perform reconstruction using a similar setup and recover pixel correspondences and photometric normals. Zhang *et al*. present an iterative method based on an extension of optical flow, which requires numerous images from a dense video sequence. Our work, in contrast, extends a multi-view stereo algorithm and only requires a small number (e.g., eight) sparsely distributed views of the object. Lim *et al*. [6] start with very sparse initial estimate of the shape computed from the 3-D locations for a sparse set of features. This shape is then refined using iterative reconstruction and re-warping to match the photometric information. While their iterative procedure generally improves the geometry, it does not always converge on the correct structure and high-frequency structure can actually degrade in quality over iterations. In contrast with both of the methods, our technique is not iterative; it computes dense shape and normals in one pass and uses both together to recover a high-quality surface.

3. Reconstruction Algorithm

The goal of our work is to recover a depth map and set of corresponding normals for an input set of images acquired with different views under different illumination directions. We compute a depth map by minimizing a window-based, multi-ocular photometric cost using a graphcut method. The key insight behind the error function is to use a rank-constrained approximation of the observed pixels. Specifically, for a Lambertian object, a matrix comprised of observations for corresponding points acquired under different lighting conditions is rank three. We use this fact to develop a depth-parameterized error function, such that only when a depth hypothesis is correct is the error minimal between the original observations and their rank three approximation. For an incorrect depth, i.e., an incorrect correspondence across views, a rank three approximation should not suffice as the matrix of observations would contain image intensities for unrelated points with different normals on the object surface, as illustrated in Figure 2.

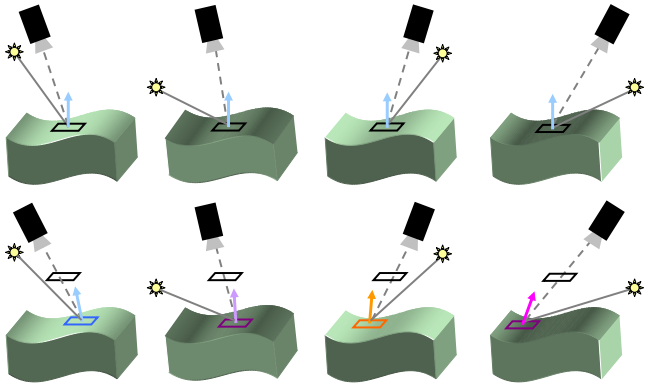

Figure 2. Multi-ocular photometric constraint for dense correspondence computation. The two rows show an object with the same camera and light motion, and the black square represents a patch at a 3D location. When the depth hypothesis is correct (black square, top row), the observed intensities across $n > 3$ views are well-approximated by a rank three approximation as a single normal and n illumination directions describe the appearance of the surface patch. For an incorrect depth hypothesis (black square, bottom row), the observed intensities are not well modeled by rank three approximation as the hypothesis leads to incorrect correspondence where there is no single normal and set of n illumination directions to describe the appearance.

Once we compute the depth map, correspondence is established for all points, and the images from different views are aligned to recover normals using photometric stereo. Our algorithm then jointly uses the depth map and normals to recover a final surface. The algorithm is summarized in Figure 3.

3.1. Motivation

We will now illustrate the motivation for our algorithm. Let I denote an image of a Lambertian object illuminated by a distant point light. For a point ρ, the observed intensity $i(\rho)$ is given by:

$$i(\rho) = \alpha(\rho)\vec{l} \cdot \vec{n}_\rho, \tag{1}$$

where $\vec{n}_\rho$ is the unknown unit surface normal for the point ρ, $\alpha(\rho)$ is the albedo, and $\vec{l}$ is the unknown scaled vector representing the illumination direction and intensity.

Given a fixed world-space coordinate system, correspondence between world-space points and image coordinates is established given depth, $z(x, y)$, corresponding to the object's surface and the camera-projection matrix P. For a specific point $\rho = (x, y, z(x, y))$, the observed intensity is:

$$i(\rho) = I(P(x, y, z(x, y))). \tag{2}$$

Let $\{I_1, ..., I_j, ..., I_n\}$ denote a sequence of n images of an object where each image is acquired from a different view illuminated by a distant point light with a potentially different direction and/or intensity. If the depth map is known, correspondence is established for all points. Observations across all n views for all points $\{\rho_1 ... \rho_k ... \rho_m\}$ can be assembled in to an observation matrix $\mathcal{I}$:

$$\mathcal{I}_{kj} = I_j(P_j(x_k, y_k, z(x_k, y_k))). \quad (3)$$

Un-calibrated photometric stereo shows us that we can then solve for normals and illumination directions up to an ambiguity by factorizing $\mathcal{I}$ using SVD [4]. Due to the Lambertian image formation model $\mathcal{I}$ should be a rank three matrix. Thus if the number of observations is greater than three, the best rank three approximation is used:

$$USV^T = \mathcal{I}, \tilde{N} = \hat{U}\hat{S}^{\frac{1}{2}}, \tilde{L} = \hat{S}^{\frac{1}{2}}\hat{V}^T, \quad (4)$$

where $\hat{U}$, $\hat{S}$, $\hat{V}$ corresponds to taking appropriate columns and rows of U, S, and V for the three largest singular values in S. $\tilde{N}$ and $\tilde{L}$ represent a set of pseudo-normals and pseudo-lights that differ from their true values by an arbitrary invertible 3×3 transform A:

$$\tilde{N} = NA, \tilde{L} = A^{-1}L. \quad (5)$$

Thus, the central challenges in our method are to solve for the unknown depth map, which establishes correspondence, and to solve for the unknown transform A, to recover normal directions unambiguously. For now we will assume projection matrices are given, and we will specifically address how to compute them in Section 3.5.

3.2. Solving for the Dense Depth Map

We solve for the depth map by minimizing a multi-ocular photometric cost function. For a particular depth hypothesis, the cost function measures, on a per-pixel basis, how well a rank three approximation models observations for corresponding patches across n views, where $n > 3$. If the depth hypothesis is correct, the error should be close to zero, if incorrect, the error should be high as a rank three approximation should not suffice as the observations would be for unrelated points on the object surface.

Specifically, let $\Omega_j(p)$ represent a neighborhood of pixels around a pixel coordinate p in an image I_j. For a world-space point $\rho = (x, y, z)$, the matrix of corresponding observations O_p is:

$$O_\rho = [\vec{\Omega}_1(P_1(\rho))...\vec{\Omega}_j(P_j(\rho))...\vec{\Omega}_n(P_n(\rho))], \quad (6)$$

where $\vec{\Omega}_\rho$ is the column vectorized neighborhood of pixels. Let $\hat{O}_\rho$ represent a rank three approximation of O_p, then the approximation error is given by:

$$E_\rho = |O_\rho - \hat{O}_\rho|. \quad (7)$$

1. **Estimate Camera Projection Matrices**
   - Using the Tomasi-Kanade factorization algorithm, recover (up to some unknown rotation) the camera projection matrix P_j for each frame I_j.
   - Pick one frame I_r to be the reference view and set the world-space coordinate system coincident with this view by accordingly transforming all projection matrices relative to P_r.
2. **Compute Dense Depth Map**
   Find the dense depth map by minimizing the multi-view photometric constraint using graphcuts.
3. **Compute Normal Field**
   Compute the pseudo-normals by aligning images according to the depth map and performing photometric stereo using SVD. Remove the linear ambiguity by computing a transform to match the pseudo-normals to normals estimated by differentiating the depth map.
4. **Compute Final Surface**
   Recover the final surface using the dense depth map and normal field.

Figure 3. Our shape reconstruction algorithm.

Each row of E contains the error across views for different pixels in the neighborhood around the point in question. Thus let r be the index for the row that contains the errors for the corresponding pixels for point ρ. The data cost for point ρ is then $\epsilon(\rho) = ||E(r, 1..j..n)||^2$.

We use a graphcut framework to find a labeling for each point that minimizes this cost subject to a smoothness constraint. Graphcuts minimizes a cost function defined on a 2D grid over a set of fixed labels. We minimize a cost function C of a depth labeling Z. The cost function is the sum of a data cost C_d over all 2D grid points σ and a smoothness penalty C_s over pairs of neighboring points σ and ϕ:

$$C(Z) = \Sigma_\sigma C_d(\sigma, Z(\sigma)) + \Sigma_{\sigma,\phi} C_s(\sigma, \phi, Z(\sigma), Z(\phi)), \quad (8)$$

where $(\sigma, \phi) \in \{x_{[min,max]}, y_{[min,max]}\}$ and $(Z(\sigma), Z(\phi)) \in \{z_{[min,max]}\}$). The data cost is then:

$$C_d(\sigma, Z(\sigma)) = \epsilon((\sigma, Z(\sigma))). \quad (9)$$

The smoothness term is a truncated $L1$ cost:

$$C_s(\sigma, q, Z(\sigma), Z(\phi)) = min(\beta * |Z(\sigma) - Z(\phi)|, \gamma). \quad (10)$$

As shown above, the cost function is defined on a 2D grid over a range of depths; the 2D grid is simply the pixel grid of our orthographic camera. We have empirically found 200

or 400 depth values, $\beta = 1$, 2, or 4, and $\gamma = 100$ or 1000 to work well for our scenes. After minimizing Equation 8, using the graphcut method of Boykov *et al*. [2], we obtain a depth map z.

3.3. Recovering Normals Unambiguously

After we solve for the depth map, Equation 3 is factored using SVD to obtain a set of pseudo-normals $\tilde{N}$ and pseudo-lights $\tilde{L}$ as shown in Equation 4. They are defined up to an arbitrary linear transform A. We solve for A by finding the transform that maps the normals so that they best match the direction of normals estimated by differentiating the depth map z. Specifically, we minimize this error function over all m points:

$$J(A) = \Sigma_{k=1}^{m} \left|\left| N^D(\sigma_k) - \frac{\tilde{N}(\sigma_k)A}{||\tilde{N}(\sigma_k)A||} \right|\right|^2, \quad (11)$$

where $N^D(\sigma_k)$ represents the unit normal obtained by differentiating the depth map and $\tilde{N}(\sigma_k)$ is the pseudo-normal vector for point σ_k. We minimize this non-linear function using Levenberg-Marquardt. The true scaled normals (albedo multiplied unit-normals) are then $N = \tilde{N}A$.

3.4. Surface Reconstruction

After recovering the depth map z and normal field N, the next step is to recover the final surface. Integrating the normal field alone can create a surface corrupted by low-frequencies bias due to error accumulation during integration. Thus it is desirable to use both the depth values and normals to solve for a final surface. Nehab *et al*. [8] combine depth from a range scanner and photometric normals by computing a surface whose low-frequency components are from the depths and the high-frequencies are from the normals. Their algorithm finds a surface by minimizing a position error, which drives the surface towards matching the dense depth map values, and a normal error that constrains the final surface to have tangents orthogonal to the input normals. Their original algorithm produces nice results, and we have extended it to provide an additional benefit. Using the original normal constraint of Nehab *et al*., we have found that when there is high-frequency noise in the normal (e.g., due to camera noise, small image misalignments, etc.) the surface will have visible high-frequency errors. This occurs because the original constraint only restricts the surface's first derivatives to match the normals, but it imposes no constraint on surface smoothness. Thus our extension is a smoothness term that minimizes the magnitude of the second derivatives of the surface; we found such a constraint very valuable for obtaining convincing and pleasing final surfaces.

We will now summarize Nehab *et al*.'s algorithm and our modifications to solve for the final surface. The algorithm first corrects normals by computing rotations to transfer the high-frequency detail from the potentially low-frequency biased photometric normals. Let N^P be unit-normals obtained from normalizing N as recovered using the method in the previous section, and let N^D be normals obtained from differentiating the recovered depth map. If $G(N^P)$ and $G(N^D)$ are low-pass filtered normals, the corrected normals field is:

$$N^C(\sigma_k) = R_k G(N^D(\sigma_k)), \quad (12)$$

for a point σ_k, where R_k is a rotation matrix which moves the normal in $G(N^P(\sigma_k))$ to the corresponding normal in $N^P(\sigma_k)$ There is an independent rotation matrix for each normal. For further details on the normal correction, we refer the reader to Nehab *et al*.'s work [8].

Once bias is corrected in the normal, the final surface is recovered by a regularized minimization of a position error, normal constraint, and smoothness penalty. Specifically, we minimize an error function:

$$J(S) = E^P + E^N + E^S. \quad (13)$$

The position error is the sum of squared distances between the final depth values and our recovered dense depth map:

$$E^P = \lambda_1 \Sigma_{k=1}^{m} [S_k - z_k]^2. \quad (14)$$

λ_1 controls the relative weighting of the position constraint versus the normal constraint. For the sake of readability, we have used a subscript k to as shorthand for values defined for a point σ_k. The normal error constrains the tangents of the final surface to be orthogonal to the input normals:

$$E^N = (1 - \lambda_1) \left[\Sigma_{k=1}^{m} [T_k^x \cdot N_k^C]^2 + \Sigma_{k=1}^{m} [T_k^y \cdot N_k^C]^2 \right], \quad (15)$$

where N^C are the corrected normals as described in the previous paragraph. T_x and T_y represent the tangent vectors:

$$T_k^x = \left[-1 \; 0 \; \frac{\partial S_k}{\partial x} \right] \text{ and } T_k^y = \left[0 \; -1 \; \frac{\partial S_k}{\partial y} \right], \quad (16)$$

for our surface parameterized on 2D grid points using an orthographic camera. Given a $N_k^C = [N_k^x N_k^y N_k^z]$, the normal constraint in Equation 15 is simply:

$$N_k^x = N_k^z \frac{\partial S_k}{\partial x} \text{ and } N_k^y = N_k^z \frac{\partial S_k}{\partial y}. \quad (17)$$

The smoothness constraint penalizes high second-derivatives by penalizing the Laplacian of the surface:

$$E^S = \lambda_2 \Sigma_{k=1}^{m} \left[(\nabla^2 S)(\sigma_k) \right]. \quad (18)$$

λ_2 is a regularization parameter to control the amount of smoothing. We have empirically determined the ranges of $\lambda_1 = [0.01 \; 0.1]$ and $\lambda_2 = [0.5 \; 0.8]$ to work well.

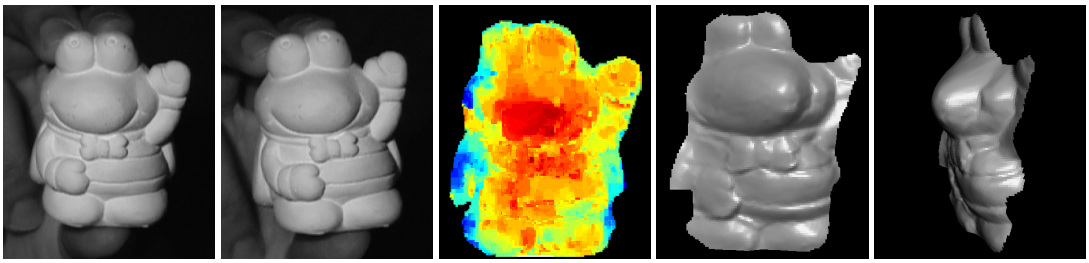

Figure 4. Frog Figurine. Two of eight views of a figurine rotating in front of a camera and light (first and second image). Each image is from a different camera viewpoint where the illumination direction and view are unknown and different for each image. The dense depth map from step 2 of our method (third image) and our final recovered surface (fourth and fifth image).

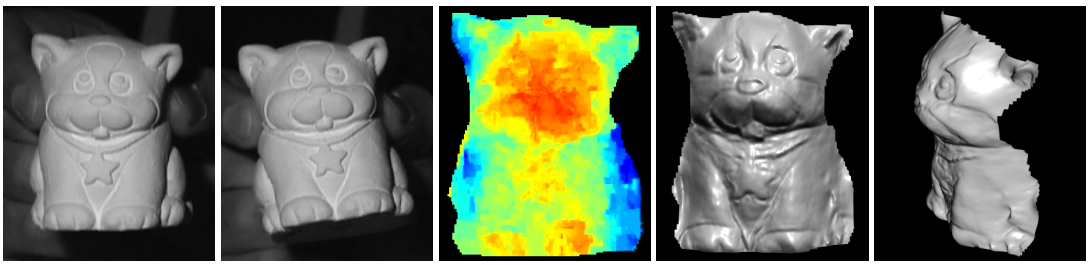

Figure 5. Cat Figurine. Two of eight views of a figurine rotating in front of a camera and light (first and second images). The dense depth map from step 2 of our method (third image) and our final recovered surface (fourth and fifth image).

Each constraint is linear in S and there are four constraints per point in total; therefore, the minimization can be formulated as a large, sparse over-constrained system, solvable by linear least squares:

$$\begin{bmatrix} \lambda_1 \mathcal{I} \\ (1-\lambda_1)\mathcal{N} \cdot \mathcal{T}^x \\ (1-\lambda_1)\mathcal{N} \cdot \mathcal{T}^y \\ \lambda_2 \nabla^2 \end{bmatrix} [\hat{S}] = \begin{bmatrix} \lambda_1 \hat{z} \\ (1-\lambda_1)\hat{N}^x \\ (1-\lambda_1)\hat{N}^y \\ 0 \end{bmatrix}. \quad (19)$$

$\mathcal{I}$ is an identity matrix and $\mathcal{N} \cdot \mathcal{T}^x$ and $\mathcal{N} \cdot \mathcal{T}^y$ are matrices that, when multiplied by the unknown vector $\hat{S}$, evaluate the right sides of the constraints in Equation 17. The "hat" operator indicates column-wise vectorization of matrices. We solve this system using a Conjugated Gradient method for solving sparse linear least squares problems [9].

3.5. Recovering Camera Projection Matrices

From hand-clicked or tracked feature point correspondences across n views, we estimate camera parameters using standard structure from motion techniques [11]. This gives orthographic projection matrices P_j for each image I_j. The projection matrices are recovered up to an unknown arbitrary rotation. As is commonly done, we simply assume that the world-space coordinate system is coincident with that of a chosen reference frame and accordingly transform the projection matrices.

4. Results

In this section, we present several experimental results using our proposed algorithm. For each result, we run our algorithms using only eight images with varying illumination and view, and in each result figure, we show one or two frames of the eight frames. While we only use eight images, we found the most convenient way to record data was to film a video sequence, so that we could track feature points automatically and then select eight images from the sequence. The reader is encouraged to view the original longer input clips online at http://vision.ucsd.edu/kriegman-grp/research/shape_illum_view/. We pick the first image as the reference view and thus assume the world-space coordinates to be aligned with this view, as described in Section 3.5. For the reference view, we draw a mask to segment the object from the background. For camera calibration, we automatically track corners from an automatic corner detector and only track points inside the mask.

In Figure 1, we show a reconstruction for a painted plaster fish figurine. Figure 4 and Figure 5 show results for two constant albedo objects: a frog and cat figurine, respectively. Our method can recover structure for objects with and without texture since the multi-view photometric cost leverages both albedo and shading clues. All three of the datasets in these first figures were recorded using a consumer hand-held DV camera. We zoomed the camera and filmed from a distance to simulate an orthographic camera, and removed gamma correction before processing the images. Each figure displays a dense depth map and the resulting surface estimated using both the depth map and normals. The recovered surfaces are convincing.

In Figure 6, we show a result for a figurine from the paper by Zhang *et al*. [12]. Figure 7 shows a result for the box sequence from the same paper. The input sequences

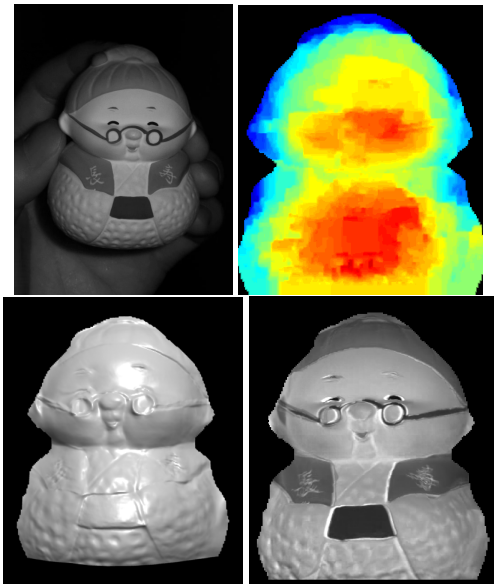

Figure 6. Results for the figurine dataset. One of eight views used from a sequence of a figurine rotating it in front of a camera and light (top left). The intermediate dense depth map (top right), final surface (bottom left), and recovered albedo map (bottom right).

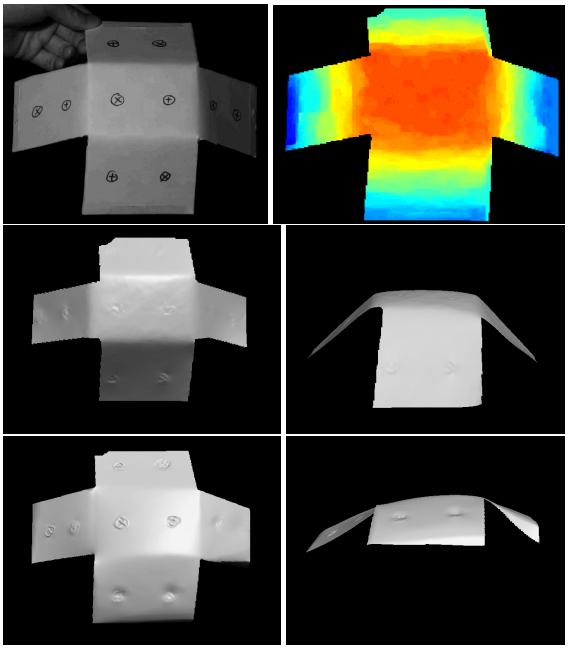

Figure 7. Results for the box dataset. One of eight views of a box acquired by rotating it in front of a camera and light (top left). The intermediate depth map (top right), our final surface (middle row), and the result from Lim *et al*. [6] (bottom row).

are publicly available on the authors' website. Figures 7, 8, and 9 show comparisons between our results for these two datasets to results from Zhang *et al*. [12] and Lim *et al*. [6]. For the Zhang *et al*. results, we used images available on the authors' website, while the Lim *et al*. results are renderings of the meshes provided by the authors for comparison in this paper. The results from Zhang *et al*. are quite low-frequency and do not show the detail of our results or those of Lim *et al*., e.g, they are missing the diagonal crease below the chin for the figurine (Figure 8). Lim *et al*.'s result for the figurine has inverted eyebrows where they are sunken into the surface. There is also a sharp protruding corner of the belly of the object when viewed from the side. In our result, the eyebrows are improved and the belly has a shape more consistent with the input data. As shown in Figure 7, Lim *et al*.'s result for the box shows relatively large low-frequency errors, while our result has a shape that appears more consistent with the input data. Our results do contain some remaining artifacts due slight image misalignment, as we discuss in the next section. The reader is encouraged to view videos of renderings of the final surfaces online at http://vision.ucsd.edu/kriegman-grp/research/shape_illum_view/.

5. Discussion and Future Work

While the results of our algorithm are better than prior methods on comparative image sequences, the method does suffer from some limitations. One limitation is an inherent fronto-parallel assumption in our dense depth map computation. We use small square patches of support for computing the rank three approximation cost per-pixel. This assumes that the surface is locally smooth and planar within the patch surrounding a pixel. In practice, as the final surface only partly uses the depth map data, the fronto-parallel assumption does not introduce significant errors in the final result. Nevertheless, this assumption can cause errors when finding depth for points in significantly slanted parts of the surface. In some of our results, there are changes in the surface induced by errant normals generated from misaligned of the input images at high-contrast edges – the breakdown of the front-parallel assumption could be partly responsible. This assumption is common in the stereo literature, and some methods for avoiding it have been addressed, such as by computing cost using slanted planes [5]. Additionally, we could perform our method iteratively where the method is performed a second time using the final surface from the first pass to deform and adapt windows during matching in a second pass to account for non-frontal tangents. We are interested in using such methods to correct the minor errors in our results.

Another limitation in our work is that we do not account for self-occlusions, shadowing, or inter-reflections. These effects can cause errors in photometric stereo as they break the simple Lambertian shading assumption that we use in this work. Researchers have investigated methods for performing photometric stereo in the presence of some of these effects [1, 3]; a good potential direction for future work is to use these methods to derive a multi-view photometric cost function that is robust to shadows and occlusions.

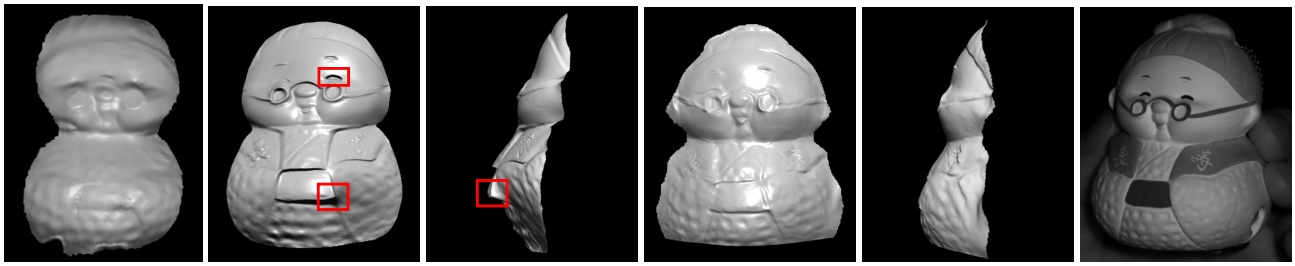

Figure 8. Comparison with previous work for the figurine dataset. Result from Zhang *et al*. [12] (first image), Lim *et al*. [6] (second and third image), our result (fourth and fifth image), and a side view from the input data (sixth image). The result from Zhang *et al*. is low-frequency and has less detail, such as the diagonal crease below the chin. The results from Lim *et al*. show two inaccuracies: the sunken eyebrows in the second image and the sharp protruding corner of the belly in the third image (areas indicated on the images with red boxes). In our result, the eyebrows are improved and the belly has a shape that appears more consistent with the input data.

6. Conclusions

We showed how to reconstruct the shape of an object from multiple images acquired as an object rotates under distant and possibly varying illumination. Our method conceptually merges multi-view stereo and photometric stereo and is straightforward to implement. By solving for a final surface using both the recovered dense depth map and normal field, the recovered surface does not have error due to low-frequency bias that can occur when integrating normals alone, and the unknown constant of integration is removed. We empirically validated the proposed method on a number of objects and showed that it is a considerable improvement over previous methods.

7. Acknowledgements

We would like to thank the anonymous reviewers for their comments and Jongwoo Lim for providing us with meshes for the results that we used in our comparisons. The first author was supported by NSF grant DGE-0333451.

References

[1] S. Barsky and M. Petrou. The 4-source photometric stereo technique for three-dimensional surfaces in the presence of highlights and shadows. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1239–1252, 2003.

[2] Y. Boykov, O. Veksler, and R. Zabih. Efficient approximate energy minimization via graph cuts. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 20(12):1222–1239, November 2001.

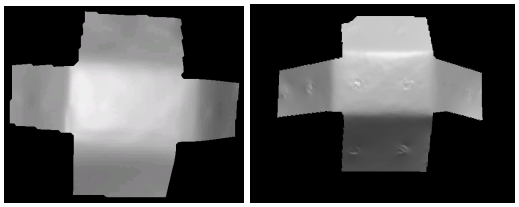

Figure 9. Comparison with Zhang *et al*. [12] for the box dataset. Result from Zhang *et al*. [12] (left) and our result (right).

[3] M. K. Chandraker, F. Kahl, and D. J. Kriegman. Reflections on the generalized bas-relief ambiguity. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 788–795, Washington, DC, USA, 2005. IEEE Computer Society.

[4] H. Hayakawa. Photometric stereo under a light-source with arbitrary motion. 11(11):3079–3089, November 1994.

[5] G. Li and S. W. Zucker. Stereo for slanted surfaces: First order disparities and normal consistency. In *EMMCVPR*, pages 617–632, 2005.

[6] J. Lim, J. Ho, M.-H. Yang, and D. Kriegman. Passive photometric stereo from motion. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1635–1642, Washington, DC, USA, 2005. IEEE Computer Society.

[7] A. Maki, M. Watanabe, and C. Wiles. Geotensity: Combining motion and lighting for 3d surface reconstruction. *Int. J. Comput. Vision*, 48(2):75–90, 2002.

[8] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi. Efficiently combining positions and normals for precise 3D geometry. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 24(3), Aug. 2005.

[9] C. C. Paige and M. A. Saunders. Lsqr: An algorithm for sparse linear equations and sparse least squares. *ACM Trans. Math. Softw.*, 8(1):43–71, 1982.

[10] D. Simakov, D. Frolova, and R. Basri. Dense shape reconstruction of a moving object under arbitrary, unknown lighting. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 1202, Washington, DC, USA, 2003. IEEE Computer Society.

[11] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *Int. J. Comput. Vision*, 9(2):137–154, 1992.

[12] 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 *The 9th IEEE International Conference on Computer Vision*, pages 618–625, Oct. 2003.