This ICCV2013 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

# Multiview Photometric Stereo using Planar Mesh Parameterization

Jaesik Park[1] * Sudipta N. Sinha[2] Yasuyuki Matsushita[3] Yu-Wing Tai[1] In So Kweon[1]
[1]Korea Advanced Institute of Science and Technology, South Korea
[2]Microsoft Research Redmond, USA [3]Microsoft Research Asia, China

## Abstract

*We propose a method for accurate 3D shape reconstruction using uncalibrated multiview photometric stereo. A coarse mesh reconstructed using multiview stereo is first parameterized using a planar mesh parameterization technique. Subsequently, multiview photometric stereo is performed in the 2D parameter domain of the mesh, where all geometric and photometric cues from multiple images can be treated uniformly. Unlike traditional methods, there is no need for merging view-dependent surface normal maps. Our key contribution is a new photometric stereo based mesh refinement technique that can efficiently reconstruct meshes with extremely fine geometric details by directly estimating a displacement texture map in the 2D parameter domain. We demonstrate that intricate surface geometry can be reconstructed using several challenging datasets containing surfaces with specular reflections, multiple albedos and complex topologies.*

## 1. Introduction

Recovering an accurate 3D shape from images is an important and challenging problem in computer vision. With recent progress in structure from motion (SfM) [12] and multiview stereo (MVS) [18], it is nowadays possible to reconstruct 3D models for many challenging scenes. These geometric methods recover 3D shape by estimating pixel correspondences in multiple views. Hence, they can suffer in accuracy when surfaces are weakly textured or cameras have wide baselines. On the other hand, photometric methods, such as shape-from-shading [9] and photometric stereo [25], use shading cues to estimate per-pixel surface normal maps but do not directly provide depth estimates. These two types of approaches have complementary strengths and have been combined in prior work [28, 15, 6, 27, 26] in the literature.

*Part of this work was done while the first author was visiting Microsoft Research Asia as a research intern. This work was also supported in part by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MEST) (No.2010-0028680) and by the Ministry of Trade, Industry and Energy, Korea, under the Human Resources Development Program supervised by the NIPA (H1502-13-1001). Yu-Wing Tai was supported by NRF-2011-0013349.

In this paper, we present a new multiview photometric stereo method that efficiently combines geometric and photometric cues. Our method takes as input, a set of images captured from multiple viewpoints illuminated by different light sources. It starts by recovering a coarse 3D mesh using existing state of the art SfM and MVS techniques. The idea of transforming this mesh into a parameterized 2D space using a distortion minimizing piecewise continuous 3D-to-2D mapping lies at the core of our method. Unlike prior methods that use explicit 3D representations [6, 15], we use a planar parameterization of the mesh [19] and cast the mesh refinement problem into one of estimating a *displacement map* texture in the 2D parameter domain. We show that both photometric stereo based surface normal estimation and mesh refinement can be efficiently and accurately performed in the parameterized 2D space.

Our proposed technique has two advantages. First, images from multiple viewpoints can be naturally handled when performing multiview photometric stereo in the 2D parameter domain, because all the images can be registered without introducing large pixel distortions. As surface normals can be directly estimated in this space using multiple registered images captured under varying lighting, it avoids the needs to first estimate per-view normal maps and then merge normal maps obtained from multiple viewpoints. Instead, images from multiple views can be jointly handled in our representation. Second, we can efficiently recover an extremely detailed 3D mesh exploiting the full resolution available in the input images. The level of geometric detail in our representation can be easily controlled by specifying the appropriate resolution of the estimated displacement map and the optimization is more efficient than direct 3D methods that must resort to subdividing the mesh and refining the vertex positions.

We describe how the proposed technique is used within a complete 3D reconstruction pipeline. We have evaluated our method on challenging sequences involving objects that have intricate 3D shapes or have multiple albedos, reflective surfaces or complex topologies. We also perform a quantitative evaluation which demonstrates the advantage of our mesh refinement technique over existing methods [6, 15].

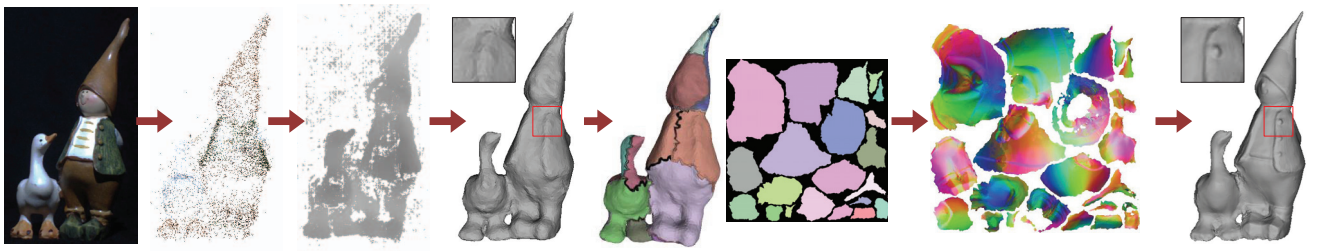



Figure 1. Overview of our method. Structure-from-motion is used to calibrate the cameras and multiview stereo is used to recover a coarse mesh. After parameterizing the mesh, multiview photometric stereo and mesh refinement are performed in the 2D parameter domain.

| | Base Mesh | Calibration | 3D Shape Representation | Optimize by | Normals Estimated in |
|---|---|---|---|---|---|
| Nehab *et al*. [15] | 3D Scanner | Manual | Regular 3D Mesh | Mesh Refinement | Individual Image |
| Hernandez *et al*. [6] | Visual Hull | Auto | Regular 3D Mesh | Mesh Refinement | Face Normals of Mesh |
| Our method | SfM+MVS | Auto | Mesh+Displacement Map | Displacement Map | 2D Parameterized Space |

Table 1. A comparison of our method and prior techniques [6, 15]. We use a different 3D shape representation, which is key to the high accuracy and efficiency of the normal map estimation and mesh refinement steps in our method.

## 2. Related Work

The idea of fusing geometric and photometric cues for high-quality 3D scanning is gaining attention due to their complementary strengths; multi-view geometric approaches are quite robust, but the recovered shapes can be coarse whereas photometric approaches can recover fine details by estimating surface normals. These methods can be broadly classified into 2D depth map refinement approaches and the ones that perform 3D mesh refinement.

Nehab *et al*. [15] propose an efficient method for 2D depth map refinement by adjusting depth values using orthogonality between depth gradients and surface orientations. Zhang *et al*. [29] extend their method to better preserve depth discontinuities. Okatani and Deguchi [16] propose a probabilistic framework for shape refinement using the first-order derivative of surface normals. While these methods are effective and can be used for recovering a full 3D mesh, they require additional processing for registering and merging multiple view-dependent depth maps.

For 3D mesh refinement, Rushmier and Bernardini [17] adjust local normal directions obtained using photometric stereo. Nehab *et al*. [15] state that their 2D depth refinement method can be extended to handle a 3D mesh. Lensch *et al*. [11] introduce a generalized method for modeling non-Lambertian surfaces by wavelet-based BRDFs and use it for mesh refinement. Hernandez *et al*. [6] iteratively refine mesh polygons by minimizing a quadratic energy function. Wu *et al*. [26] use the spherical harmonics representation to estimate global illumination, and refine a preliminary mesh using photometric stereo by minimizing $\ell_1$ penalties. In an extended approach [27], geometric details are added using shape-from-shading under natural lightings. Vlasic *et al*. [23] first integrate per-view normal maps into partial meshes, then deforms them using thin-plate offsets to improve the alignment while preserving geometric details.

These 3D mesh refinement methods generally use a high-resolution mesh in order to enclose high frequency details obtained by photometric methods; however, determining the appropriate mesh resolution is non-trivial due to the view-dependent variation of effective resolutions. In contrast, our method allows the mesh resolution to be derived directly from the normal map resolution and avoids the problem of undersampling mesh vertices. In addition, our 2D parameterization approach performs mesh refinement efficiently, where only 1D vertex displacements are optimized rather than directly working in the 3D coordinates. Table 1 summarizes how our approach related to the two closely related methods proposed by Nehab *et al*. [15] and Hernandez *et al*. [6].

## 3. Proposed Method

In this section, we describe the key elements of the proposed method. For now, let us assume that all the cameras are calibrated and the initial base mesh is available. The methods for calibration and obtaining the initial base mesh are later explained in Sec. 4. After describing the mesh parameterization scheme, we explain how surface normal estimation and mesh refinement is performed in the 2D parameter domain. Figure 1 shows an overview of our approach.

### 3.1. Mesh Parameterization

In our method, first the triangulated base mesh denoted by $\mathcal{M}$, is mapped to a planar parameterized space using a piecewise continuous function $f : \mathbb{R}^3 \rightarrow \mathbb{R}^2$, which is referred to as *mesh parameterization* [19] (see Fig. 1). While this process is not limited to a particular mesh parameterization method, in this paper, we use the Iso-charts method proposed by Zhou *et al*. [30], which minimizes non-uniform distortions of the original mesh by finding optimal cuts that partition the mesh into segments. Each connected segment

**Algorithm 1**: Image Warping

**Input**: Image $\mathcal{I}$, camera projection matrix $\mathcal{P}$, mesh $\mathcal{M}$ and its face visibility
**Output**: Warped image $\mathcal{I}'$
**for** *each pixel* $\mathbf{u} \in \mathcal{U}$ **do**
  Find triangle $t \in \mathcal{U}$ that contains $\mathbf{u}$
  Find barycentric coefficients, $\mathbf{w}_t$ for $\mathbf{u}$ in $t$
  Find face $f \in \mathcal{M}$ that maps to $t$ and its vertices $\{\mathbf{x}_t\}$
  **if** $f$ *is visible* **then**
    $\mathbf{x}' \leftarrow$ Barycentric-interpolation($\{\mathbf{x}_t\}$, $\mathbf{w}_t$)
    $\mathcal{I}'(\mathbf{u}) \leftarrow \mathcal{I}(\mathcal{P}\mathbf{x}')$

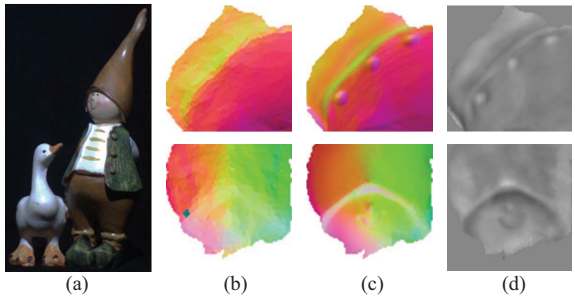
is mapped by its own mapping function to a single *chart* in the parameter domain. We denote the 2D parameter domain as $\mathcal{U}$, which contains an arbitrary arrangement of the charts. Using Iso-charts, we obtain a one-to-one mapping $f_{\mathcal{M}}$ from a 2D point $\mathbf{u} = [u, v]^{\mathsf{T}}$ in $\mathcal{U}$ to a 3D point $\mathbf{x}$ on the mesh $\mathcal{M}$. For maximally utilizing photometric stereo estimates, the resolution of $\mathcal{U}$ is set proportional (0.8 times smaller) to the input image resolution.

### 3.2. Image Warping

Using the camera calibration and inverse mapping $f_{\mathcal{M}}^{-1}$, we warp input images $\mathcal{I}$ to images $\mathcal{I}'$ in the $\mathcal{U}$ coordinates. The images are warped using the standard inverse mapping technique, *i.e.*, we begin with a pixel $\mathbf{u}$ in the $\mathcal{U}$ coordinates and determine its corresponding pixel location in the input image $\mathcal{I}$ via the inverse mapping function $f_{\mathcal{M}}^{-1}$. Since the forward mapping function $f_{\mathcal{M}}$ is discrete, we use a piece-wise linear interpolation to approximate $f_{\mathcal{M}}^{-1}$. Specifically, our method finds the projected mesh face that encloses pixel $\mathbf{u}$ in the $\mathcal{U}$ coordinates, determines 3D position $\mathbf{x}'$ that corresponds to pixel $\mathbf{u}$ using barycentric interpolation. Finally, the intensity of pixel $\mathbf{u}$ in $\mathcal{I}'$ is determined by mapping the pixel in image $\mathcal{I}$ via the 3D scene point $\mathbf{x}'$. This procedure is summarized in Algorithm 1. We use kd-trees [14] to accelerate the search for the 2D triangle. The warping function is computed once for each viewpoint and that warp is applied to multiple images captured from that viewpoint and illuminated by different light sources. During image warping, only visible mesh faces are considered and z-buffering is used to find which faces are visible to the camera.

### 3.3. Surface Normal Estimation

One of the key benefits of our distortion minimizing mesh parameterization scheme [30] is that pixels in images from multiple viewpoint and different lighting are well aligned in the 2D parameter domain of the base mesh without significant errors caused by viewpoint variations. Unlike single-view photometric stereo, in our case, we have more observations from different nearby viewpoints that are reasonably well aligned using the base mesh geometry.



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

Figure 2. An example of surface normal map and displacement map estimation. (a) Input image. (b) Initial normal map obtained from the base mesh, in $\mathcal{U}$. (c) Disambiguated normals from photometric stereo in $\mathcal{U}$. Here, unit 3D vectors have been linearly mapped to RGB. (d) The estimated displacement map.

Therefore, the parameterization allows images from multiple viewpoints to be used effectively for multiview photometric stereo.

In this section, we introduce our method for estimating surface normals given warped images $\mathcal{I}'$. To handle a large amount of observations efficiently, we use the Lambertian reflectance model at this stage. By denoting the image intensities in the form of an observation matrix $\mathbf{I} \in \mathbb{R}^{p \times q}$, where $p$ is the number of valid pixels in $\mathcal{U}$, and $q$ is the number of all the images that are taken from varying view points under varying lightings, the Lambertian image formation model can be expressed in matrix form, as $\mathbf{I} = \mathbf{NL}$. Here, $\mathbf{N} \in \mathbb{R}^{p \times 3}$ is an albedo-scaled surface normal matrix, and $\mathbf{L} \in \mathbb{R}^{3 \times q}$ represents a lighting matrix. Unlike the single-view photometric stereo case, $\mathbf{I}$ has many missing elements as most 3D points are not visible from all the viewpoints. Therefore, we compute surface normals $\mathbf{N}$ using subsets of the observations which form dense block matrices in $\mathbf{I}$. In general, finding dense block matrices $\mathbf{I}_S$ from the matrix $\mathbf{I}$ is a NP-hard problem. However, since the columns of $\mathbf{I}$ are arranged in the image capture sequence in our case, valid intensity observations at $\mathbf{u}$ tend to be in adjacent columns of $\mathbf{I}$. The problem of finding the sub-matrices then reduces to finding maximum cliques in an interval graph. We use the method proposed by [3] for finding multiple, overlapping dense block matrices in $\mathbf{I}$.

Next, given an observation matrix $\mathbf{I}_S$, we apply the uncalibrated photometric stereo method of Hayakawa [4] to each $\mathbf{I}_S$. Each observation matrix $\mathbf{I}_S$ can be approximated and decomposed into a product of two rank-3 matrices as

$$\mathbf{I}_S \approx \mathbf{U}_3 \mathbf{\Sigma}_3 \mathbf{V}_3^{\mathsf{T}} = \rho(\mathbf{N}_S \mathbf{A}^{-1})(\mathbf{A}\mathbf{L}_S), \qquad (1)$$

where $\mathbf{N}_S = \mathbf{U}_3 \mathbf{\Sigma}_3^{\frac{1}{2}}$, $\mathbf{L}_S = \mathbf{\Sigma}_3^{\frac{1}{2}} \mathbf{V}_3^{\mathsf{T}}$, and $\mathbf{A}$ is a non-singular $3 \times 3$ matrix that represents a linear shape-light ambiguity that exists in uncalibrated photometric stereo. $\mathbf{\Sigma}_3$ is a diagonal matrix with three singular values, and $\mathbf{U}_3$ and $\mathbf{V}_3$ are orthonormal matrices with only first three columns and rows, respectively. To automatically resolve the linear

ambiguity $\mathbf{A}$, we use the mesh normals $\mathbf{N}_f \in \mathbb{R}^{p\times 3}$ obtained from the base mesh, which is coarse yet reasonably close to the correct surface normal. Specifically, we regard $\mathbf{N}_S\mathbf{A}^{-1} \approx \mathbf{N}_f = \mathbf{U}_3\mathbf{\Sigma}_3^{\frac{1}{2}}\mathbf{A}^{-1}$. Using the pseudo-inverse of $\mathbf{N}_f$, we solve for $\mathbf{A}$ and obtain the surface normal estimate $\hat{\mathbf{N}}_S$ as

$$\left\{\begin{array}{rcl} \mathbf{A} & \leftarrow & (\mathbf{N}_f^{\mathsf{T}}\mathbf{N}_f)^{-1}\mathbf{N}_f^{\mathsf{T}}\mathbf{U}_3\mathbf{\Sigma}_3^{\frac{1}{2}}, \\ \hat{\mathbf{N}}_S & = & \mathbf{U}_3\mathbf{\Sigma}_3^{\frac{1}{2}}\mathbf{A}^{-1}, \end{array}\right. \tag{2}$$

where $\hat{\mathbf{N}}_S$ is a disambiguated surface normal matrix for subset $S$. To combine duplicate solutions from distinct submatrices $\mathbf{I}_S$, we apply weighted sum to consolidate the normal estimate as

$$\mathbf{n}_p(\mathbf{u}) = \frac{1}{M}\sum_{S\in\mathbb{S}(\mathbf{u})}(\mathbf{n}_f(\mathbf{u})^{\mathsf{T}}\mathbf{n}_S(\mathbf{u}))\mathbf{n}_S(\mathbf{u}), \tag{3}$$

where $\mathbb{S}(\mathbf{u})$ denotes a set of sub-matrix indices that include $\mathbf{u}$. $\mathbf{n}_f^{\mathsf{T}}\mathbf{n}_S(=w_S)$ is a weighting factor, which is the cosine of the angle between the face normal and estimated normal vectors, and $M = \Sigma w_S$ normalizes the weighted sum of $\mathbf{n}_S$. This weighting is used for reducing the effect of outliers. Figure 2 shows an example of the computed surface normal maps.

### 3.4. Geometry Refinement

The major advantage of working in the 2D parameter domain is that 3D mesh refinement can be performed by estimating a 2D displacement map of the base mesh $\mathcal{M}$. The geometry refinement problem can now be formulated as finding the optimal displacement $d \in \mathbb{R}$ per pixel $\mathbf{u}$ as

$$\mathbf{x}^*(\mathbf{u}) = \mathbf{x}(\mathbf{u}) + d(\mathbf{u})\mathbf{n}_f(\mathbf{u}), \tag{4}$$

where $\mathbf{n}_f$ is a unit face normal of the triangle in $\mathcal{M}$ to which $\mathbf{x}$ is mapped, and $\mathbf{x}^*$ is the refined 3D position. Notice that the refinement is defined in the 2D domain using $\mathbf{u}$ as indices. Now, given photometric normals $\mathbf{n}_p$ obtained via photometric stereo and the initial position $\mathbf{x} \in \mathcal{M}$, we estimate the displacement $\hat{d}$ by minimizing the following energy function:

$$\hat{d} = \underset{d}{\operatorname{argmin}} \sum_{\mathbf{u}\in\mathcal{U}} \left(\mathbf{n}_p^{\mathsf{T}}\frac{\partial \mathbf{x}^*}{\partial \mathbf{u}}\right)^2 + \lambda \sum_{\mathbf{u}\in\mathcal{U}} d^2(\mathbf{u}). \tag{5}$$

The first term of Eq. (5) is a data term that encourages the surface gradient at $\mathbf{x}^*$ to be orthogonal to the orientation of photometric normal $\mathbf{n}_p$. This term is related to the one proposed by Nehab *et al.* [15]. However, we estimate only a single displacement for each 3D point, optimizing a single scalar instead of three coordinates thereby reducing mesh refinement to estimating the optimal displacement map. We use a cross-shape operator for computing partial derivatives, *i.e.*, $[-1, 0, 1]$ for $\frac{\partial}{\partial u}$, and $[1, 0, -1]^T$ for $\frac{\partial}{\partial v}$. To define partial derivatives at pixels on the boundary of two charts, we use their respective inverse mappings to look up neighboring 3D points on the mesh. This operation is important as it prevents seams from occurring on the chart boundaries by encouraging points across seams to have similar displacement values. The second term of Eq. (5) is a regularization term that discourages large displacements.

The problem of Eq. (5) can be formulated as a sparse linear system that can be efficiently solved using an off-the-shelf sparse linear solver. In our implementation, we empirically choose $\lambda = 0.3$. An example of an estimated displacement map is shown in Fig. 2. In our method, the level of geometric detail is controlled by the resolution of $\mathcal{U}$ regardless of the resolution of the base mesh. For example, a base mesh with as few as $2K$ vertices with a $512\times512$ displacement map can generate $262K$ effective vertices. Since our approach directly estimates a displacement map on a coarse mesh, our 3D models can be efficiently stored and rendered efficiently on modern graphics hardware that supports displacement mapping [21].

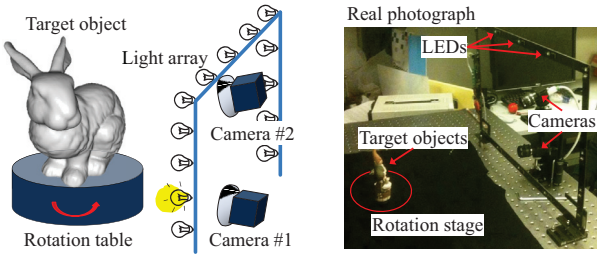



Figure 3. Imaging setup: rotation stage, light array and two cameras. For a particular rotation angle, several images are captured under varying lightings.

## 4. Reconstruction Pipeline

This section describes our reconstruction pipeline – the imaging setup and the SfM and MVS pre-processing stages.

### 4.1. Imaging Setup

Our acquisition system consists of a rotation stage, LED array, and two cameras as illustrated in Fig. 3. We assume that the camera response functions are known. All images of the target object are captured automatically using a remotely controlled rotation stage with synchronized cameras and LEDs. A typical acquisition captures 312 images (24 viewpoints, 15 degrees apart illuminated by 13 different LEDs) and takes about three minutes.

### 4.2. Camera Calibration and Multiview Stereo

We calibrate the camera intrinsics a priori and assume that they remain constant during the acquisition. The extrinsic parameters are estimated using a generic SfM

| | Bunny | | | Gargoyle | | | Happy-Buddha | | |
|---|---|---|---|---|---|---|---|---|---|
| Mesh Perturbation | Level 1 | Level 2 | Level 3 | Level 1 | Level 2 | Level 3 | Level 1 | Level 2 | Level 3 |
| Nehab *et al*. [15] | 2.87, 99.7 | 2.93, 99.7 | 6.31, 93.6 | 4.72, 99.9 | 4.56, 99.9 | 4.58, 99.9 | 4.01, **99.8** | 3.95, 99.8 | 4.06, 99.8 |
| Hernandez *et al*. [6] | 1.66, 99.7 | 2.30, 99.7 | 5.51, 92.4 | **3.03**, 100.0 | **3.40**, **100.0** | 4.15, 100.0 | **2.92**, 99.8 | **3.42**, **99.9** | **3.04**, **99.9** |
| Ours | **1.50**, **100.0** | **1.94**, **100.0** | **2.67**, **100.0** | 3.43, **100.0** | 3.47, 100.0 | **3.70**, **100.0** | 3.50, 99.5 | 3.65, 99.5 | 3.50, 99.5 |
| Mesh Resolution | 70K | 50K | 25K | 70K | 50K | 25K | 70K | 50K | 25K |
| Nehab *et al*. [15] | 3.56, 99.9 | 5.44, 94.4 | 7.54, 67.8 | 6.02, 98.5 | 8.38, 83.6 | 11.56, 49.3 | 4.82, 99.2 | 6.51, 91.2 | 8.37, 64.8 |
| Hernandez *et al*. [6] | **1.11**, 99.8 | 1.43, 96.7 | 1.67, 76.0 | 3.64, 96.7 | 4.14, 89.8 | 4.88, 64.5 | **2.76**, 98.3 | **3.43**, 93.4 | 4.29, 72.1 |
| Ours | 1.39, **100.0** | **1.40**, **100.0** | **1.41**, **100.0** | **3.33**, **100.0** | **3.37**, **100.0** | **3.45**, **99.9** | 3.45, **99.6** | 3.48, **99.5** | **3.49**, **99.5** |

Table 2. Comparison using synthetic dataset. In this experiment, each method refines degraded meshes, and the results are evaluated in comparison with the ground truth. Each cell of the table shows *accuracy* ($\times 10^{-3}$) and *completeness* (%) for two experiments, *mesh perturbation* and *mesh resolution* (see text for more details).

pipeline [20], which we found to be quite accurate. However, methods tailored to turn-tables [2] can also be used.

**Stereo matching.** Using the visibility of the SfM point cloud, we estimate a depth range for each viewpoint and then perform plane-sweep stereo matching for each viewpoint using two other images captured from adjacent viewpoints under identical lighting. Using normalized cross correlation as the matching cost and semi-global matching based cost aggregation [7], we first estimate a dense depth map with discrete depth estimates. Sub-pixel refinement is then performed on these depth maps using a standard local parabolic refinement of the aggregated matching costs. We compute per-pixel confidence associated with the depth map using the ratio of the minimum and the second smallest costs to measure distinctiveness and prune depth estimates at pixels with very low confidence. See Fig. 1 for an example of a refined depth map.

**Mesh extraction.** The filtered depth maps are fused using an energy minimization framework based on volumetric graph-cuts [24]. The step computes an implicit 3D shape of a closed object by labeling voxels on a uniform 3D grid with binary labels – *occupied*, or *empty*. This optimization is formulated using a discrete binary Markov Random Field using unary and pairwise terms on a 6-connected voxel grid with a typical resolution of $100^3$. The unary potentials are computed using free space occupancy of the 3D points in the depth map [5], where the contributions from depth maps are weighted by their confidences. The pairwise potentials are derived from the sub-voxel positions of these 3D points. As our acquisition setup allows simple foreground silhouette extraction, we also include a silhouette-based unary term in the energy – voxels that are projected outside the silhouette are given a high penalty for taking the label *occupied*. The optimal binary labeling can be exactly computed in an efficient manner using graph cuts [1]. Finally, from the labeled grid, we recover a triangulated mesh $\mathcal{M}$ using marching cubes [13]. We prefer MVS in computing our base mesh over a visual-hull based approach [6], since MVS yields more accurate mesh in our experience, especially for objects with large concavities or complex topologies.

# 5. Results

We first quantitatively evaluate our method using synthetic datasets and compare our method with existing state-of-the-art approaches [15, 6] focusing on the performance of our mesh refinement algorithm. In this evaluation, we used synthetically rendered images and the original mesh as the preliminary mesh. In a second set of experiments, we show 3D reconstruction results on various real-world objects where the level of detail in our reconstructed models is of the order of a few millimeters.

## 5.1. Experiments on Synthetic Data

Using the *Bunny*, *Gargoyle*, and *Happy-Buddha* models, we render $712 \times 712$ images under 8 different light directions and 16 distinct viewpoints using the Lambertian shading model. All the methods in the evaluation have access to the true camera and light calibration parameters. The number of triangles for these models varies from $70K$ to $1M$. For consistency, these 3D models are scaled by setting the radius of their tightest bounding spheres to a unit. To simulate errors and irregularities of real data, these meshes are corrupted by adding noise, and vertices are sub-sampled to produce meshes with smaller triangle counts.

**Mesh perturbation test.** In this test, we perturb the original mesh by adding random vertex displacements as noise and then use the Taubin operator [22] to apply mesh smoothing. The perturbation is performed at three levels, where level 3 has the highest error.

**Mesh resolution test.** In this test, we vary the number of faces of the base mesh to analyze the effect of mesh resolutions. Using a mesh simplification technique [8], we generate meshes with $25K$, $50K$, and $70K$ vertices.

**Evaluation metrics.** Given the ground truth mesh $\mathcal{G}$, we measure the accuracy of the refined mesh $\mathcal{R}$ by computing the *accuracy* and *completeness* metrics that are used in the Middlebury multiview stereo benchmark [18]. These are based on asymmetric distances $\text{dist}_{\mathcal{R}\to\mathcal{G}}$ and $\text{dist}_{\mathcal{G}\to\mathcal{R}}$, where $\text{dist}_{A\to B}$ represents the minimum distance from vertices of $A$ to vertices of $B$. *Accuracy* refers to the distance $d \in \text{dist}_{\mathcal{R}\to\mathcal{G}}$ such that $x\%$ of the points are within distance

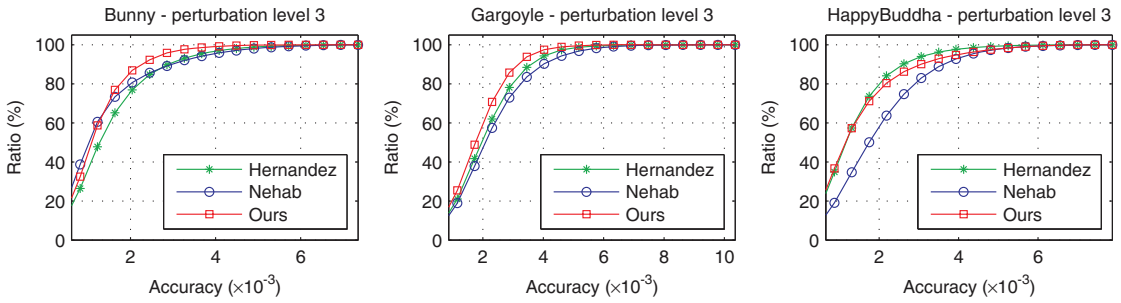



Figure 4. Cumulative error distributions of *Accuracy* for three synthetic dataset; *Bunny*, *Gargoyle*, and *Happy-Buddha*. The graph corresponds to mesh perturbation experiment in Table 2 when perturbation level is 3. Except for *Happy-Buddha*, our curve locates above the other curves. This means larger proportion of per-vertex errors are smaller than others.

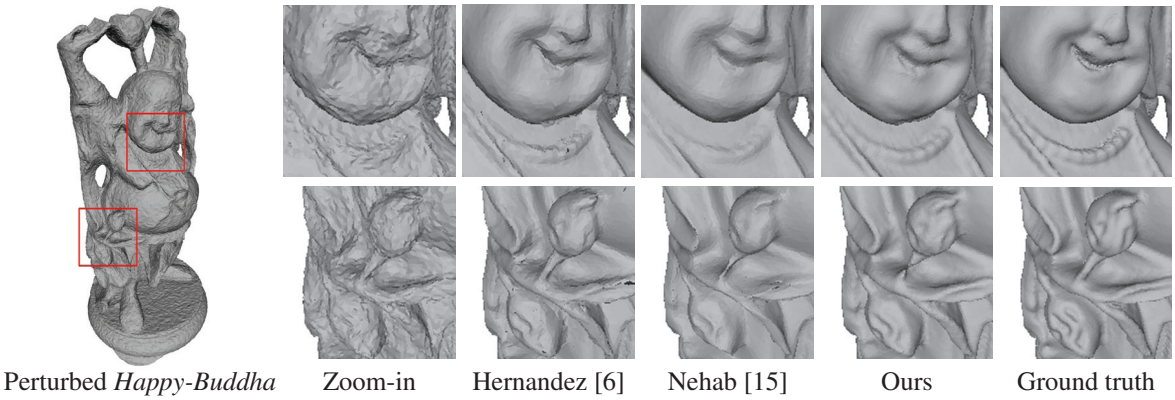



Figure 5. A perturbed *Happy-Buddha* model and refinement results by three methods. The result corresponds to mesh perturbation experiment in Table 2 where perturbation level is 3.

$d$ to $\mathcal{G}$. *Completeness* refers to the proportion of vertices, where $\text{dist}_{\mathcal{G}\to\mathcal{R}}$ is less than threshold $\text{dist}_{th}$. In our experiments, we set $x = 90$ and $\text{dist}_{th} = 0.01$.

**Results.** Table 2 shows quantitative results on the synthetic data. For the *mesh perturbation* experiment, our method consistently performs better than Nehab *et al*. [15] because our method naturally avoids mesh flipping and overlapping triangles. In this test, the accuracy and completeness of our results are comparable to those of Hernandez *et al*. [6]. As our approach estimates a displacement map whose resolution is derived from the original image resolution, our method recovers fine geometric details regardless of the resolution of the base mesh (see *mesh resolution* in Table 2).

Figure 4 shows the cumulative error distributions for the *Accuracy* metric. The percentage of vertices within an accuracy threshold is plotted for different thresholds. The plot shows that our method is consistently the most accurate, except for the *Happy-Buddha* model where our method is comparable to [6]. The refined meshes for *Happy-Buddha* are shown in Fig. 5. Our method faithfully reconstructs fine details such as the necklace and flower on the model.

**Computation time.** We compare the computational cost of our mesh refinement method with that of Nehab *et al*. [15]. The result is shown in Fig. 6 where its clear that our method is computationally more efficient than theirs when both methods are configured to produce results with comparable accuracy. On average, our proposed method (excluding acquisition and pre-processing) as described in Sec. 3 takes less than a minute to run. The timings are measured on a system equipped with an Intel i7 quad-core 3.06GHz CPU and 8GB memory.

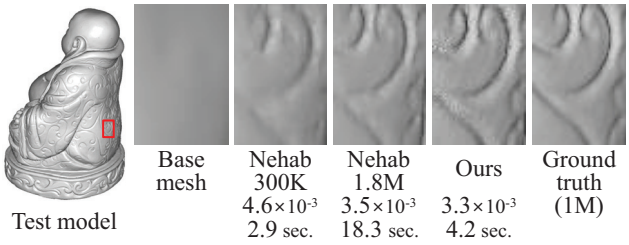



Figure 6. Computational cost and accuracy of mesh refinement. Base mesh resolutions, reconstruction accuracies, and computation times are shown under sub-figures. Our method does not require tuning the mesh resolution because it is automatically determined by the input image resolution.

## 5.2. Experiments on Real Data

Figure 7 shows the result of five real scenes whose images are taken using the imaging setup described in Sec. 4.1. The first two objects, BUDDHA and AGRIPPA, have mostly uniform albedos. However, BUDDHA statue is made of copper and has many specular reflections. Even though our normal estimation method assumes Lambertian reflectances, the normal aggregation process of Eq. (3) effectively handles outliers arising from non-Lambertian reflectance.

The other three objects, DOLL-1, DOLL-2, and

TEAPOT, show more interesting topologies and multiple albedos. In DOLL-1, we can observe the detailed shape of buttons on the jacket of the right doll as well as facial expression of the dolls, which cannot be seen in the original base mesh. The English characters in the middle region of DOLL-2 are clearly visible in the final mesh. The geometric details on the TEAPOT model are faithfully reconstructed. Note that these embossed patterns are only a few millimeters deep. On the other hand, an artifact can be seen below the left doll's skirt in DOLL-1 as indicated by red rectangles in Fig. 7. Since no valid normal could be estimated from any of the viewpoints, our method is unable to refine the coarse mesh in this region.

## 6. Discussions

Our 3D reconstruction approach enables the acquisition of high-fidelity 3D models where a mesh parameterization scheme is used to fuse photometric and geometric cues. Although our automatic pipeline demonstrates high accuracy, there are currently a few limitations. First, we have used a linear photometric stereo approach for efficiency reasons, but the accuracy of our system can be potentially boosted using recent advances in robust photometric stereo [10]. Dark and textureless surfaces are currently difficult to handle in our method due to the lack of reliable photometric or geometric cues. In the future, we plan to explore a joint optimization approach that simultaneously estimates surface normals and scene depth for greater accuracy and robustness. Recovering surface reflectance as well as accurate 3D shape is another interesting direction for future work.

## References


[1] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. *IEEE Trans. Pattern Anal. Mach. Intell.*, 26(9):1124–1137, 2004.

[2] A. W. Fitzgibbon, G. Cross, and A. Zisserman. Automatic 3D model construction for turn-table sequences. In *3D Structure from Multiple Images of Large-Scale Environments*, pages 155–170. 1998.

[3] U. I. Gupta, D. T. Lee, and J. Y.-T. Leung. Efficient algorithms for interval graphs and circular-arc graphs. *Networks*, 12(4):459–467, 1982.

[4] H. Hayakawa. Photometric stereo under a light source with arbitrary motion. *Journal of the Optical Society of America*, 11(11):3079–3089, 1994.

[5] C. Hernández, G. Vogiatzis, and R. Cipolla. Probabilistic visibility for multi-view stereo. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2007.

[6] C. Hernández, G. Vogiatzis, and R. Cipolla. Multi-view photometric stereo. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(3):548–554, 2008.

[7] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(2):328–341, 2008.

[8] H. Hoppe. New quadric metric for simplifying meshes with appearance attributes. In *Proc. of IEEE Visualization Conference*, 1999.

[9] B. K. P. Horn. Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. *PhD thesis, MIT*, 1970.

[10] S. Ikehata, D. Wipf, Y. Matsushita, and K. Aizawa. Robust photometric stereo using sparse regression. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 318–325, 2012.

[11] H. P. A. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H. peter Seidel. Image-based reconstruction of spatial appearance and geometric detail. *ACM Trans. on Graph.*, 22:234–257, 2003.

[12] H. C. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. *Nature*, 193:133–135, 1981.

[13] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proc. of SIGGRAPH*, pages 163–169, 1987.

[14] D. M. Mount and S. Arya. ANN: A library for approximate nearest neighbor searching. *http://www.cs.umd.edu/˜mount/ANN/*, 2010.

[15] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi. Efficiently combining positions and normals for precise 3D geometry. *ACM Trans. on Graph.*, 24(3), 2005.

[16] T. Okatani and K. Deguchi. Optimal integration of photometric and geometric surface measurements using inaccurate reflectance/illumination knowledge. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 254–261, 2012.

[17] H. Rushmeier and F. Bernardini. Computing consistent normals and colors from photometric data. In *3*.

[18] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2006.

[19] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Found. Trends. Comput. Graph. Vis.*, 2(2):105–171, 2006.

[20] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3d. In *Proc. of SIGGRAPH*, pages 835–846. ACM Press, 2006.

[21] L. Szirmay-Kalos and T. Umenhoffer. Displacement mapping on the GPU - State of the Art. *Computer Graphics Forum*, 27(1), 2008.

[22] G. Taubin. A signal processing approach to fair surface design. In *Proc. of SIGGRAPH*, pages 351–358, 1995.

[23] D. Vlasic, P. Peers, I. Baran, P. Debevec, J. Popović, S. Rusinkiewicz, and W. Matusik. Dynamic shape capture using multi-view photometric stereo. *ACM Trans. on Graph.*, 28(5), 2009.

[24] G. Vogiatzis, C. Hernández Esteban, P. H. S. Torr, and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(12), Dec. 2007.

[25] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1), 1980.

[26] C. Wu, Y. Liu, Q. Dai, and B. Wilburn. Fusing multiview and photometric stereo for 3d reconstruction under uncalibrated illumination. *IEEE Trans. on Visualization and Computer Graphics*, 17(8):1082–1095, 2011.

[27] C. Wu, B. Wilburn, Y. Matsushita, and C. Theobalt. High-quality shape from multi-view stereo and shading under general illumination. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2011.

[28] 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 *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 618–625, 2003.

[29] Q. Zhang, M. Ye, R. Yang, Y. Matsushita, B. Wilburn, and H. Yu. Edge-preserving photometric stereo via depth fusion. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, 2012.

[30] K. Zhou, J. Snyder, B. Guo, and H.-Y. Shum. Iso-charts: Stretch-driven mesh parameterization using spectral analysis. In *Eurographics Symposium on Geometry Processing*, pages 45–54, 2004.

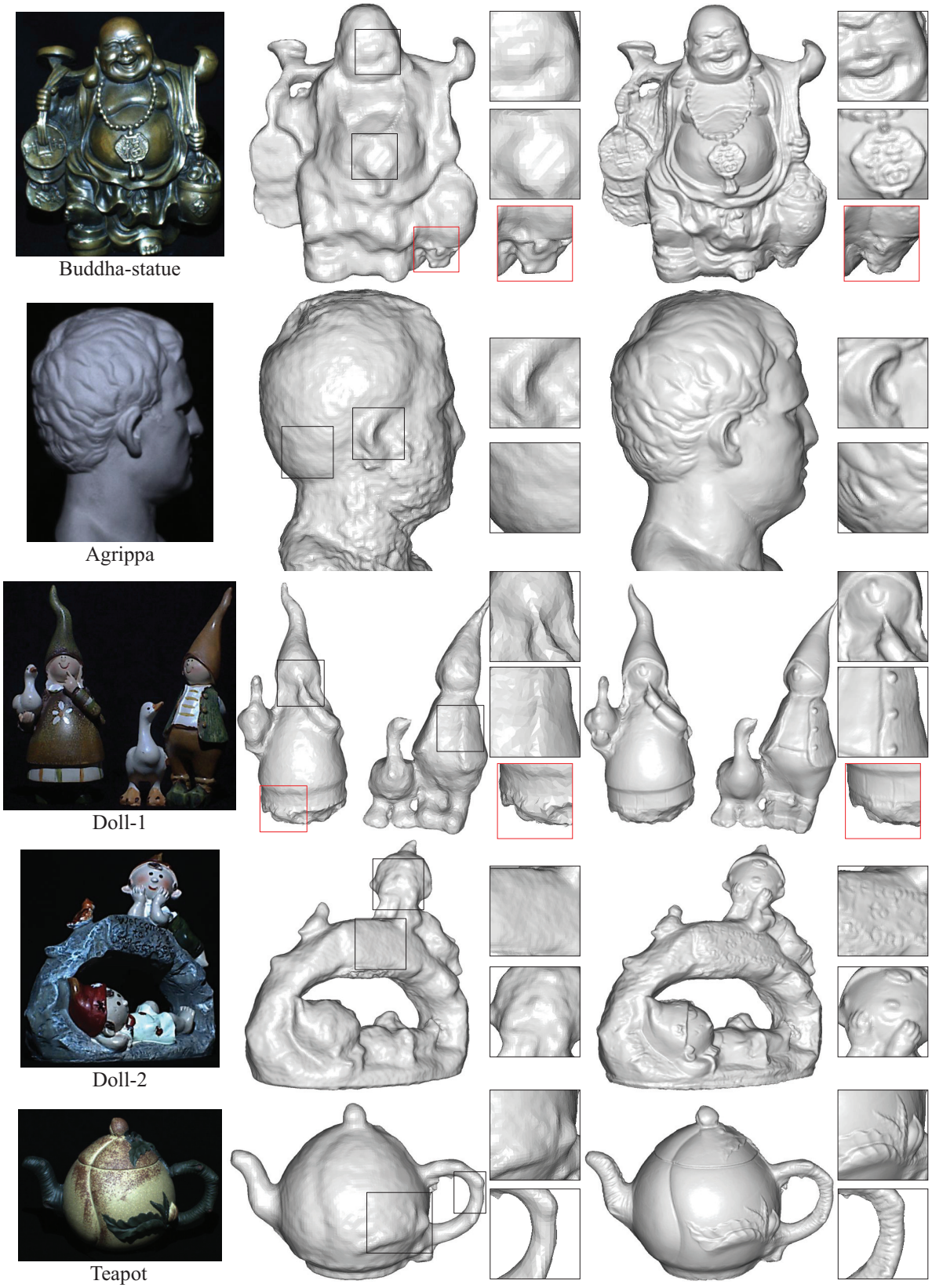



Figure 7. Reconstruction results by our method for BUDDHA-STATUE, AGRIPPA, DOLL-1, DOLL-2, and TEAPOT scenes. Each row shows one of input images, the base mesh from MVS, and the refined mesh. The corresponding surface normal and displacement maps are shown in the supplementary material. Clear failure cases are highlighted by red rectangles; these occur at textureless dark regions.