Combining Depth Fusion and Photometric Stereo for Fine-Detailed 3D Models

Erik Bylow[1][0000−0002−6665−1637], Robert Maier[2][0000−0003−4428−1089], Fredrik Kahl[3][0000−0001−9835−3020], and Carl Olsson[1,3][0000−0003−3545−7695]

[1] Lund University erikb@maths.lth.se
[2] Technical University of Munich robert.maier@in.tum.de
[3] Chalmers University of Technology {fredik.kahl, caols}@chalmers.se

Abstract. In recent years, great progress has been made on the problem of 3D scene reconstruction using depth sensors. On a large scale, these reconstructions look impressive, but often many fine details are lacking due to limitations in the sensor resolution. In this paper we combine two well-known principles for recovery of 3D models, namely fusion of depth images with photometric stereo to enhance the details of the reconstructions. We derive a simple and transparent objective functional that takes both the observed intensity images and depth information into account. The experimental results show that many details are captured that are not present in the input depth images. Moreover, we provide a quantitative evaluation that confirms the improvement of the resulting 3D reconstruction using a 3D printed model.

1 Introduction

Three-dimensional object reconstruction is a classical problem in computer vision. It is still a highly active research area, and we have witnessed steady progress on recovering reliable and accurate representations of scene geometry. There is a wide range of applications where fine-detailed 3D reconstructions play a central role, including visualization, 3D printing, refurbishment and e-commerce.

Several different methods exist for recovering 3D scene geometry. Classical algorithms include Structure from Motion [1, 2] which yields sparse point clouds and multiple-view stereo [3, 4] which generates dense reconstructions. Since the advent of the Microsoft Kinect, a lot of effort has been put into developing methods that can create dense models directly from the depth images. KinectFusion [5] and extensions like [6, 7] can reliably compute high-quality 3D models. However, due to limitations in the resolution of the depth sensor, fine details are often missed in the reconstruction.

To obtain highly detailed 3D models, a common approach is to use photometric stereo [8, 9], which can capture fine details under the assumption of a Lambertian surface [10, Chapter 2]. This technique originates from Shape-from-Shading [11] where surface normals are estimated from a single image. Shape-from-Shading is often considered to be an ill-posed problem. In contrast, photometric stereo uses several images with varying illumination of the same scene, which makes the problem of recovering surface normals well-posed with known lighting. Although some works that utilize multiple views exist, e.g. [12], many methods require that the images are captured from the same view point [9].

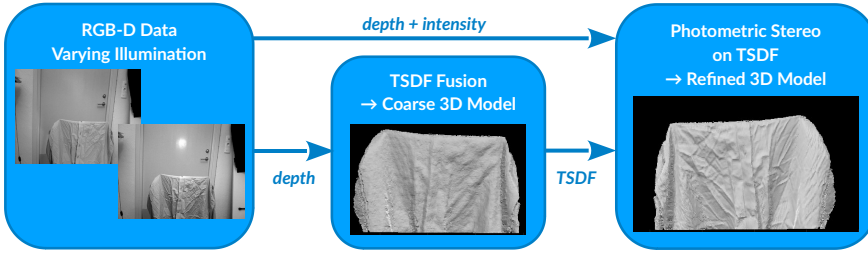


Fig. 1: Out method first fuses all depth images into a single coarse 3D reconstruction without details. This fused 3D model and the intensity images with varying illumination are used to compute a refined 3D model with fine-scale details.

The advantages with our formulation are manifold. Many papers that combine depth images and shading only refine a single depth image [13–17]. How to fuse refined depth images from multiple views is not a trivial task. In contrast, we derive an energy functional on the surface using a Truncated Signed Distance Function (TSDF) [18]. This has the advantage that we can combine measurements from different views and refine the whole model at once. Another benefit is that the implicit representation makes it easy to estimate normals since these are directly obtained from the gradient of the TSDF.

Our main contribution is that we derive an objective functional using a TSDF [18] as a parametrization of the surface. This functional takes both intensity and depth information into account together with a varying light source and allows us to handle data captured from different viewpoints. Figure 1 shows a schematic overview of our method. We experimentally demonstrate on real imagery that this results in a system that can recover finer details than current state-of-the-art depth fusion methods. Both quantitative and qualitative evaluations are provided.

1.1 Related Work

Since the launch of the Kinect, several papers have tried to incorporate techniques like Shape-from-Shading [11] into the reconstruction process to enhance the quality of the depth images. For example, [13] improves the entire depth image by estimating the shading and reflectance from a single image. In [14–17], Shape-from-Shading techniques are applied in conjunction with RGB-D images. These approaches typically employ strong priors such as piece-wise constant albedo to constrain the optimization problem. The depth image is often used as a means to resolve the Bas-Relief ambiguity [19]. The idea is to separate albedo and shading to catch fine surface details that are not visible in the depth image. In [20] a fixed depth sensor captures multiple images under varying illumination. Super-resolution and photometric stereo are then combined in a variational framework to get a more detailed depth image. All papers above have in common that they try to obtain a high resolution depth image. In contrast, we work directly over the entire surface and take all views simultaneously into account.

The most closely related work to ours are [21] are [22], where the 3D model is encoded in a TSDF surface representation [18] and refined using Shape-from-Shading techniques. In [21] the fused color information and the fused depth information in the

voxel grid are incorporated in a shading based energy functional. The resulting functional is optimized to get an improved reconstruction. Just recently, [22] extended and improved [21] by approximating multiple static light sources to better model the illumination. They additionally used the input intensity images instead of the fused color to measure the difference between rendered intensity gradient and observed intensity gradient. Both [21] and [22] regularize the surface with a Laplacian and the albedo by measuring chromaticity differences between neighboring voxels. Furthermore, [22] also regularizes the light sources with a Laplacian.

In this paper we investigate how one can benefit from the TSDF representation and the observed color and depth images by using ideas from photometric stereo. The main difference between [21, 22] and our work is that we allow the light source to vary between the input RGB-D frames. The theoretical motivation to move both the light source as well as the camera is that varying the light source generates richer data, in contrast to keeping the light source fixed as in [21, 22].

Furthermore, in addition to the intensity error measure, our energy only has two additional terms: an error penalty that measures deviations from the observed depth maps and an albedo regularizer that penalizes large albedo changes between neighboring voxels. In contrast, both [21] and [22] require smoothness priors on the surface. To the best of our knowledge, we are the first to combine photometric stereo with the TSDF parametrization of the surface. Our results show that by illuminating the object from different directions one can get both a smoother (where appropriate) and a more detailed reconstruction without any explicit smoothness prior. Our results are also compared to [22] and evaluated on a 3D printed model with ground truth data.

2 The Lambertian Surface Model

In this work we assume that the objects we are observing are diffuse. Under the Lambertian reflectance model, the image intensity at a projected point is given by

$$\mathcal{I}(\pi(\mathbf{x})) = \rho(\mathbf{x})\mathbf{s}^T\mathbf{n}(\mathbf{x}), \tag{1}$$

where $\mathcal{I}$ is the observed grayscale image, $\pi(\mathbf{x})$ is the projection of the 3D point $\mathbf{x}$, $\rho(\mathbf{x})$ and $\mathbf{n}(\mathbf{x})$ are the per-voxel surface albedo and normal at $\mathbf{x}$ respectively, and $\mathbf{s}$ is the lighting direction. We assume a rigid scene, hence only the position of the light source and the camera are changing between consecutive input frames. Consequently, by illuminating the scene from varying directions, the observed intensity will be different as visualized in Figure 2. We show that optimizing the illumination, the albedo and the surface normals to generate image projections that agree with the observed image results in a more detailed reconstruction.

To represent the surface and its albedo we use a TSDF [18], i.e. a voxel grid where each voxel V consists of 8 corners. A corner is denoted with v and contains the two values d_v and ρ_v representing the estimated distance from v to the surface and the albedo at v respectively. We use tri-linear interpolation between the voxel corners to compute distance and albedo estimates within a voxel. We let $g_V : \mathbb{R}^3 \times \mathbb{R}^8 \rightarrow \mathbb{R}$ be an interpolation function that takes a point $\mathbf{x}$ within the voxel and the 8 corner values,

Fig. 2: Two images captured from almost identical viewpoints, but with different illumination. The wrinkles appear differently in the two images, with more prominent details on the right. This effect is caused by the varying light source and not by the surface.

either $\boldsymbol{\rho}_V = (\rho_{v_1}, ..., \rho_{v_8})$ or $\mathbf{d}_V = (d_{v_1}, ..., d_{v_8})$, where $v_i \in V$, and computes the distance and albedo estimates $g_V(\mathbf{x}, \boldsymbol{\rho}_V)$ and $g_V(\mathbf{x}, \mathbf{d}_V)$ at $\mathbf{x}$.

The Lambertian model in Equation (1) also requires surface normals in order to estimate image intensities. By normalizing the gradient we get the expression for the normal at a surface point $\mathbf{x} \in V$ by

$$\mathbf{n}(\mathbf{x}, \mathbf{d}_V) = \frac{\nabla g_V(\mathbf{x}, \mathbf{d}_V)}{\|\nabla g_V(\mathbf{x}, \mathbf{d}_V)\|}, \tag{2}$$

where ∇ is a spatial gradient with respect to $\mathbf{x}$.

It was shown in [9] that general lighting conditions can often be better estimated than Equation (1) by employing low-order spherical harmonics. Their formulation essentially replaces $\mathbf{n}$ and $\mathbf{s}$ in Equation (1) with 9-dimensional vectors $\tilde{\mathbf{n}}$ and $\tilde{\mathbf{s}}$, where the elements of $\tilde{\mathbf{n}}$ are quadratic expressions in the elements of $\mathbf{n}$. To compute $\tilde{\mathbf{n}}$ we thus first compute $\mathbf{n}$ via Equation (2) and then use the quadratic functions from [9].

3 Objective Functional

In this section we derive our objective functional that consists of the three terms presented in the following sections.

3.1 Intensity Error Term

We first consider a term that takes the agreement between rendered intensity and the observed image into account. Let us now assume that we have captured a sequence of depth and intensity images $\mathcal{D}$ and $\mathcal{I}$, respectively. We denote the depth image at time step k by $\mathcal{D}^k$ and the corresponding gray-scale intensity image by $\mathcal{I}^k$.

We assume that we have a set $\mathcal{S}$ of surface points that we project into the images and a set of voxels $\mathcal{V}$ that contain the surface. In Section 3.2 we describe how these points are extracted from the TSDF. Projecting a surface point $\mathbf{x} \in \mathcal{S}$ contained in voxel V on image k, we can extract the observed intensity at $\mathcal{I}^k(\pi(\mathbf{x}))$. Through the Lambertian reflectance model, we should have

$$\mathcal{I}^k(\pi(\mathbf{x})) \approx \rho(\mathbf{x}, \boldsymbol{\rho}_V)\tilde{\mathbf{n}}(\mathbf{x}, \mathbf{d}_V)^T \tilde{\mathbf{s}}^k. \tag{3}$$

Our first term penalizes deviations from this assumption using

$$E_{\text{Lambert}}(\mathbf{d}, \boldsymbol{\rho}, \tilde{\mathbf{s}}^1, \ldots, \tilde{\mathbf{s}}^K) = \sum_{k=1}^{K} \sum_{V \in \mathcal{V}^k} \sum_{\mathbf{x} \in V \cap \mathcal{S}} (\mathcal{I}^k(\pi(\mathbf{x})) - \rho(\mathbf{x}, \boldsymbol{\rho}_V) \tilde{\mathbf{n}}(\mathbf{x}, \mathbf{d}_V)^T \tilde{\mathbf{s}}^k)^2, \tag{4}$$

where $\mathcal{V}^k$ is the set of voxels containing the surface observed in frame k, $\mathbf{d}_V$ and $\boldsymbol{\rho}_V$ are the distances and the albedo in the voxel corners of V and $\mathbf{x}$ is a detected surface point in voxel V. The set $\mathcal{V}^k$ is constructed by projecting all voxels in $\mathcal{V}$ into the depth image $\mathcal{D}^k$ and keeping those that are close to the observed surface. Note that each view has its own light source $\tilde{\mathbf{s}}^k$, allowing the lighting conditions to change between views. This error term permits the normals, albedo and the light source to change so that the observed intensities coincide with the rendered intensity. By varying the light source, the same surface point will have different intensities in the images; using that, we seek to improve the three-dimensional shape of the surface.

3.2 Sampling Surface Points

In order to evaluate Equation (4), we need to compute a set of surface points. Recall that the surface is located in the zero crossing of the TSDF. Any voxel V that is intersected by the surface has both positive and negative values among the distances $\mathbf{d}_V = (d_{v_1}, ..., d_{v_8})$ stored in the voxel corners $(\mathbf{x}_{v_1}, \ldots, \mathbf{x}_{v_8})$. By randomly generating N points $\{\hat{\mathbf{x}}_n\}_{n=1}^N$ in the voxel and computing their interpolations $\hat{d}_n = g_V(\hat{\mathbf{x}}_n, \mathbf{d}_V)$, we get a set of points with signed distances $\{\hat{d}_n\}_{n=1}^N$ to the surface. If $\hat{d}_n$ is positive, we match it to one of the corner points with negative distances and vice versa. This gives a pair of points $(\mathbf{x}_{v_i}, \hat{\mathbf{x}}_n)$, where the surface lies somewhere on the line segment defined by these points. To find a surface point, we simply traverse the segment until we are sufficiently close to the zero crossing.

3.3 Depth Error Term

The Lambertian term (4) is by itself not sufficient for estimating the normals, albedo and light sources uniquely, due to the well known Generalized Bas-Relief ambiguity [19]. Additionally, while it gives good local normal estimates, computing the surface from local orientation estimates alone is a sensitive process. When normals are integrated, error buildup can cause large depth deviations if the surface depth is not sufficiently constrained. To ensure uniqueness and constrain the overall surface depth, we include information from the depth images.

We define the depth error term as

$$E_{\text{depth}}(\mathbf{d}) = \sum_{k=1}^{K} \sum_{v \in \mathcal{V}^k} (D^k(\mathbf{x}_v) - d_v)^2, \tag{5}$$

where d_v is the currently stored estimated distance to the surface for voxel corners v in $\mathcal{V}^k$ as in [23]. $D^k(\mathbf{x}_v)$ is the estimated distance between observed surface and the voxel in frame k, computed as in [23].

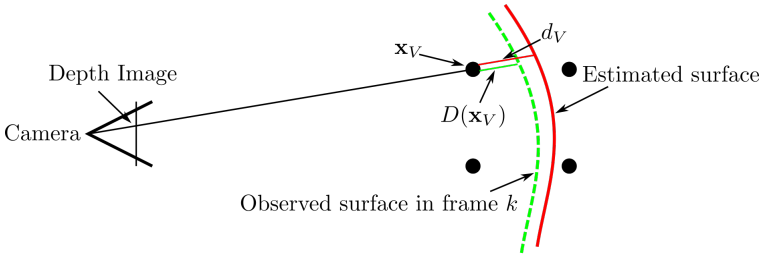


Fig. 3: To penalize large deviations from the observed depth images, we penalize the observed distance to the object surface. This is done for all frames and resolves the Generalized Bas-Relief ambiguity.

This error term penalizes solutions where the estimated distance is far from the observed depth image, which constrains the surface to resolve the Generalized Bas-Relief ambiguity. We illustrate the underlying idea behind this data term in Figure 3.

3.4 Albedo Regularization

It is common to put some form of prior on the albedo, see [22] and [24], and we do the same to favor solutions where neighboring voxel corners have similar albedo. This helps to disambiguate between variations in the albedo and the surface geometry. Considered a voxel V with corners $(v_1, \ldots, v_8)$, we penalize the albedo differences between all its neighbouring corners. By summing over all voxels, we get:

$$E_{\text{albedo}}(\boldsymbol{\rho}) = \sum_{V \in \mathcal{V}} \sum_{v_i \neq v_j \in V} (\rho_{v_i} - \rho_{v_j})^2. \tag{6}$$

Note that the corners typically occur among several voxels.

4 Optimization

With the three different error terms defined in Equations (4), (5) and (6), we assemble our final objective functional as follows:

$$E(\mathbf{d}, \boldsymbol{\rho}, \tilde{\mathbf{s}}^1, \ldots, \tilde{\mathbf{s}}^K) = E_{\text{Lambert}}(\mathbf{d}, \boldsymbol{\rho}, \tilde{\mathbf{s}}^1, \ldots, \tilde{\mathbf{s}}^K) + \lambda E_{\text{depth}}(\mathbf{d}) + \mu E_{\text{albedo}}(\boldsymbol{\rho}), \tag{7}$$

where λ and μ are positive weights for the individual cost terms.

To optimize over the variables we perform alternating optimization:

1. Extract voxels $\mathcal{V}$ that contain the surface and create the sets $\mathcal{V}^1, \ldots, \mathcal{V}^K$ containing the voxels in $\mathcal{V}$ visible in frame k. Then find surface points in each voxel V.
2. Optimize the light sources $\tilde{\mathbf{s}}^1, \ldots, \tilde{\mathbf{s}}^K$:

$$(\tilde{\mathbf{s}}^1_{n+1}, \ldots, \tilde{\mathbf{s}}^K_{n+1}) = \underset{\tilde{\mathbf{s}}^1, \ldots, \tilde{\mathbf{s}}^K}{\arg\min} \sum_{k=1}^{K} \| A^l_{\tilde{s}_n} \tilde{\mathbf{s}}^k - \mathbf{b}^l_{\tilde{s}_n} \|^2. \tag{8}$$

3. Optimize the albedo values $\boldsymbol{\rho}$:

$$\boldsymbol{\delta}_{\boldsymbol{\rho}}^{*} = \arg\min_{\boldsymbol{\delta}_{\boldsymbol{\rho}}} \|A_{\boldsymbol{\rho}_n}^{l}\boldsymbol{\delta}_{\boldsymbol{\rho}} - \mathbf{b}_{\boldsymbol{\rho}_n}^{l}\|^2 + \mu\|A_{\boldsymbol{\rho}_n}^{\rho}\boldsymbol{\delta}_{\boldsymbol{\rho}} - \mathbf{b}_{\boldsymbol{\rho}_n}^{\rho}\|^2 + \gamma_{\boldsymbol{\rho}}\|\boldsymbol{\delta}_{\boldsymbol{\rho}}\|^2 \quad (9)$$

$$\boldsymbol{\rho}_{n+1} = \boldsymbol{\rho}_n + \boldsymbol{\delta}_{\boldsymbol{\rho}}^{*}. \quad (10)$$

4. Optimize the distance values $\mathbf{d}$:

$$\boldsymbol{\delta}_{\mathbf{d}}^{*} = \arg\min_{\boldsymbol{\delta}^{\mathbf{d}}} \|A_{\mathbf{d}_n}^{l}\boldsymbol{\delta}_{\mathbf{d}} - \mathbf{b}_{\mathbf{d}_n}^{l}\|^2 + \lambda\|A_{\mathbf{d}_n}^{d}\boldsymbol{\delta}_{\mathbf{d}} - \mathbf{b}_{\mathbf{d}_n}^{d}\|^2 + \gamma_{\mathbf{d}}\|\boldsymbol{\delta}_{\mathbf{d}}\|^2 \quad (11)$$

$$\mathbf{d}_{n+1} = \mathbf{d}_n + \boldsymbol{\delta}_{\mathbf{d}}^{*}. \quad (12)$$

5. Update the TSDF with $\mathbf{d}_{n+1}$ and $\boldsymbol{\rho}_{n+1}$.

In Equation (8) we optimize directly over the light sources, given the current estimates $\boldsymbol{\rho}_n$ and $\mathbf{d}_n$, which is a linear least-squares problem. To optimize over $\boldsymbol{\rho}_n$ in Equation (9), we linearize $\rho(\mathbf{x}, \boldsymbol{\rho}_V)$ for each voxel V and obtain the matrices $A_{\boldsymbol{\rho}_n}^{l}$ and $\mathbf{b}_{\boldsymbol{\rho}_n}^{l}$. We also put a damping on the step size of $\boldsymbol{\delta}_{\boldsymbol{\rho}}$ to prevent too rapid changes in the albedo. Similarly, for Equation (11) we linearize $\tilde{\mathbf{n}}(\mathbf{x}_V, \mathbf{d}_V)$ with respect to $\mathbf{d}_V$ analytically and put a damping on the step size. This is crucial, since a too big step can alter the surface severely and ruin the optimization. Furthermore, we are doing local refinements of the surface, so we have a prior that the step size should be small. In fact, the voxel cube has a fixed side-length α, so the distance between the surface and a voxel in that cube cannot be greater than $\sqrt{3}\alpha$. Hence in each iteration we do not want the distance to change more than a fraction of the cube's side-length. Note that due to the use of tri-linear interpolation, all derivatives with respect to $\mathbf{d}_V$ and $\boldsymbol{\rho}_V$ can be computed analytically.

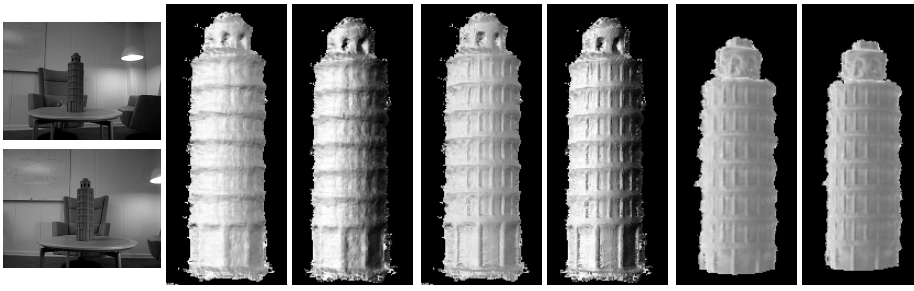
Surface Initialization via Depth Fusion. We essentially follow [5, 18, 23] to initialize the TSDF. Similarly, each voxel corner is assigned an initial albedo estimate ρ derived from the captured gray-scale images. This is used to estimate the appearance of the surface. The light sources are initialized randomly.

5 Experimental Results

To evaluate our method we perform a number of experiments. Note that the approaches [21, 22] do not vary the illumination between images, while other works either fix the light source or do not vary the camera position. The datasets from [21, 22] with fixed light sources are consequently not directly applicable for our approach. Instead, we collect our own data where we record the scene using a depth sensor and illuminate the object with a detached lamp from different directions. For the recordings, we put the camera on a tripod, illuminate the object with a lamp and capture one image at a time.

We also recorded some datasets with fixed light in order to enable a comparison with the approach in [22]. The recordings were taken from approximately the same distance and view point as our sequences. All the results from the algorithm in [22] in our paper were produced by the original implementation.

Furthermore, a sequence of a 3D printed object was acquired. The ground truth for the object is known, hence we can register the obtained reconstructions to the ground truth model and get a quantitative evaluation as well. All sequences were captured using an Asus Xtion Pro Live Sensor with a resolution of 640×480 for both color and depth.



(a) Input (b) Fusion (c) Ours (d) [22]

Fig. 4: (a) Two used input images from the *Tower* sequence. (b) Two shading images from the initial reconstruction using only the depth images. (c) Two shading images from our refined model. (d) Two resulting shadings from [22].

5.1 Qualitative Results

In this section we show results from several experiments on real data and compare our method with [22]. The first result in Figure 4 is from the *Tower* sequence where we also show some of the input images. The models shown in all experiments are visualized by synthetically rendering the shading image from one of the estimated camera poses with optimized light source.

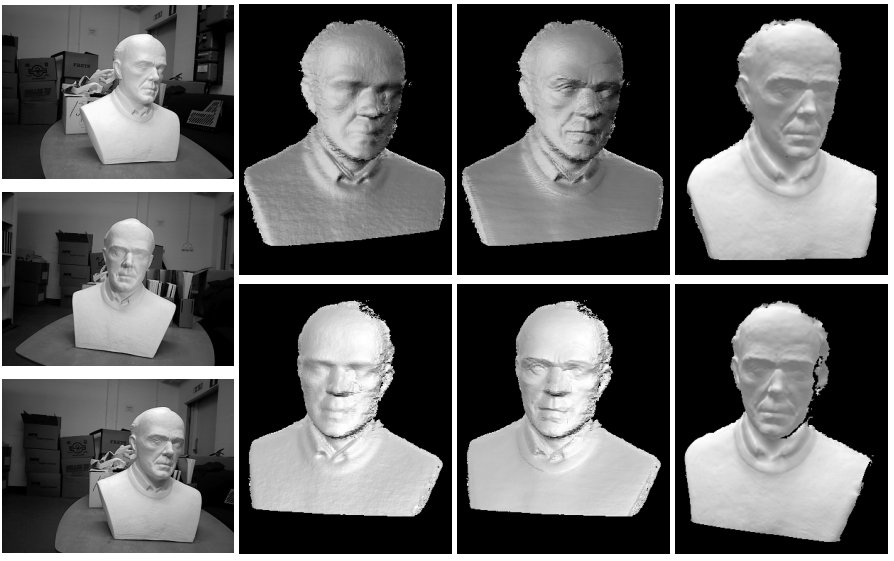
There is a clear difference between the shading from just fusing the depth and the refined shading using our proposed method, with significantly enhanced details. The presented result of [22] exhibits comparable quality. For this experiment we set $\lambda = 0.001$ and $\mu = 8.5$ and we used 15 images in both our sequence and the sequence for the algorithm in [22].

In the second experiment we scan a statue (Figure 5) and a shirt with a lot of wrinkles (Figure 2). The results are displayed in Figure 5 (b)-(d) and Figure 6 (b)-(d). The rendered shading images exhibit a significant refinement of the surface normals. In Figure 5, the eyes and mouth of the statue are much more detailed and fine-scaled compared to the initial solution. The wrinkles in Figure 6 are much sharper and more prominent in the refined shading image (d) compared to the initial fused shading image (c). In the top row of Figure 6 the initial albedo (a) can be seen together with the optimized albedo (b). In the initial solution, most shading effects are present in the albedo estimation, in contrast to the optimized solution where most details are in the shading image and the estimated albedo is roughly constant.

The *Statue* experiment consists of 19 images and we set $\lambda = 0.05$ and $\mu = 12.5$, the *Shirt* sequence contains 9 images with the same set of parameters.

5.2 3D Renderings

For a qualitative evaluation, we also render the obtained surfaces with Phong-shading using Menderer [25] in order to visualize the reconstructed 3D model differences between our proposed method and the algorithm from [22]. The results of the *Statue*



(a) Input images (b) Fusion (c) Ours (d) [22]

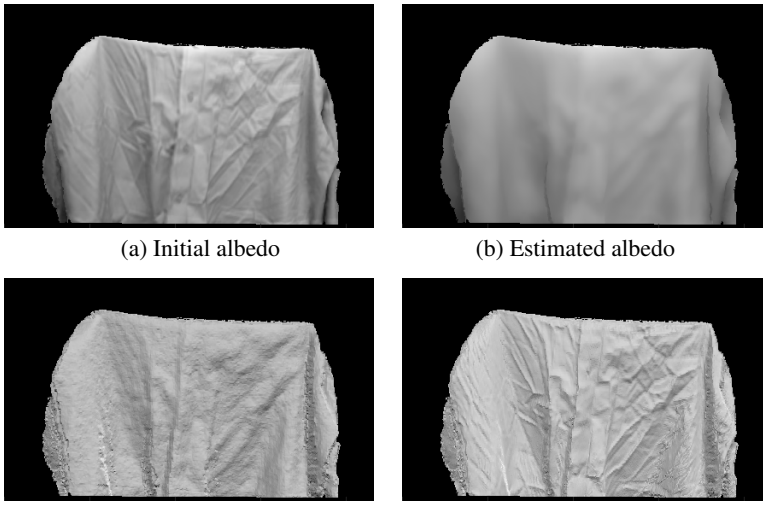
Fig. 5: (a) Input images from the *Statue* sequence. Shading of initial TSDF (a) in comparison with our method (c) and with [22] (d). The optimized shadings are considerably improved, which is particularly visible at the eyes, mouth and sharper edges of the shirt.

sequence in Figure 7 suggest that our method better preserves details on the refined 3D model. Comparing Figures 7a and 7b, it is clear that the details on the shirt are better preserved in Figure 7a. For a better visualization, close-ups are provided in Figure 7. We believe that the surface regularization in [22] is a reason to why these details are smoothed out. Please note that the holes in Figure 7a are rendering artifacts due to inverted normals.

Implementation details. All code is implemented in Matlab. The generally most runtime expensive part is the extraction of 3D points. For the *Shirt* experiment, which is the most time consuming dataset, our method required about 16 seconds per frame. Computing the matrices for optimization takes about 6 seconds per image on a desktop computer with 6 Intel i7-4930K cores and 48 GB of RAM. All experiments typically converged after about 50 iterations.

5.3 Quantitative Evaluation

In the following, we provide quantitative results on the *Tower* dataset, where we have the ground truth 3D model for the 3D printed object. We generated two point clouds, one from the initial model P_{init} and one from the optimized model P_{opt}. Moreover, we also rendered a point cloud from the ground truth data P_{gt}. To measure the quantitative difference between the reconstructions, we selected a part of the ground truth model with valid existing correspondences on the reconstructed model. Then we registered



(a) Initial albedo (b) Estimated albedo

(c) Fusion (d) Ours

Fig. 6: *Shirt* sequence: (a) Input albedo to our framework. (b) Estimated albedo. (c) Shading from initial solution. (d) Shading from optimized model.

	Initial Model	Optimized Model
RMSE (m)	0.00128	0.00105

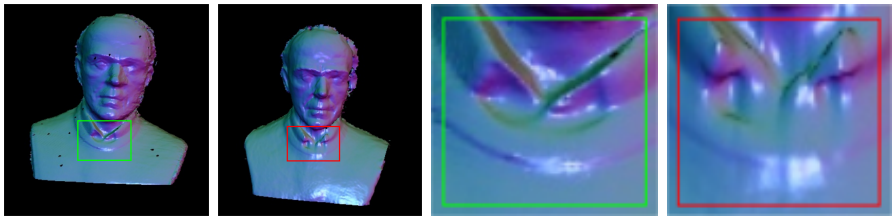
Table 1: RMSE in meters for the initial and the optimized models compared to ground truth. The RMSE for the optimized model is approximately 18% lower.

P_{init} and P_{gt}, matched each point in P_{gt} to a point in P_{init} and computed the Root Mean Square Error (RMSE). Similarly, we got the RMSE for P_{opt}.

The results are given in Table 1. The error is about 18% lower for the optimized model, which is a significant improvement. Hence, we obtain millimeter accuracy in the reconstruction. To visualize the quantitative difference, Figure 8 provides a contour plot of the ground truth, the initial and the optimized models. Looking at the optimized green line and comparing that to the ground truth blue line, it is evident that we manage to get a more exact estimation of the surface. This demonstrates the advantages of (i) optimizing over rendered intensity and (ii) observing the surface from different views to reduce the impact of occlusions.

6 Conclusion and Future Work

In this paper we have successfully combined ideas from photometric stereo and volumetric depth fusion in order to refine a 3D model represented as a TSDF. The derived objective functional favors solutions where the distance values in the TSDF and the albedo make the rendered images consistent with the observations. By illuminating



(a) Ours (b) [22] (c) Ours (d) [22]

Fig. 7: Qualitative comparison of our method with [22] on the *Statue* dataset. The shirt in the Phong-shaded rendering of our method (a) is sharper and better preserved compared to [22] (b). This is particularly visible in the close-ups shown in (c)-(d).

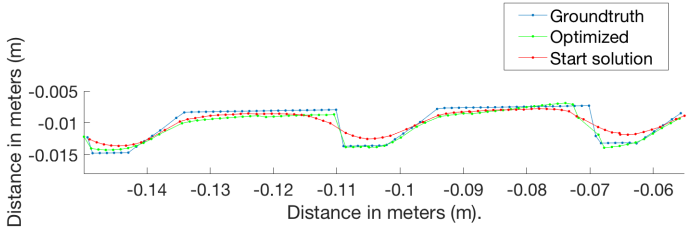


Fig. 8: Contour of the initial surface (red), the optimized surface (green) and the true surface (blue) in a slice through the reconstructed *Tower*. Note that the optimized model better captures the shape of the true surface.

the object from different directions, it is possible to significantly improve the recovery of fine-scale details in the 3D models. The experimental evaluation demonstrates both quantitatively and qualitatively that we obtain accurate reconstruction results of high quality. Potential future work could be how to disambiguate shading from albedo without the L_2–norm.

7 Acknowledgements

This work was funded by The Swedish Research Council (grant no. 2018-05375), the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation

References

1. Thormählen, T., Broszio, H., Weissenfeld, A.: Keyframe selection for camera motion and structure estimation from multiple views. In: ECCV. (2004)
2. Enqvist, O., Kahl, F., Olsson, C.: Non-sequential structure from motion. In: ICCV Workshops. (Nov 2011) 264–271

3. Newcombe, R.A., Lovegrove, S., Davison, A.J.: DTAM: dense tracking and mapping in real-time. In: ICCV. (2011) 2320–2327
4. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Trans. Graph. **28**(3) (July 2009) 24:1–24:11
5. Newcombe, R., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A., Kohli, P., Shotton, J., Hodges, S., Fitzgibbon, A.: Kinectfusion: Real-time dense surface mapping and tracking. In: ISMAR. (2011) 127–136
6. Whelan, T., Leutenegger, S., Salas-Moreno, R.F., Glocker, B., Davison, A.J.: Elasticfusion: Dense slam without a pose graph. In: RSS, Rome, Italy (2015) 1697–1716
7. Steinbruecker, F., Kerl, C., Sturm, J., Cremers, D.: Large-scale multi-resolution surface reconstruction from rgb-d sequences. In: ICCV, Sydney, Australia (2013)
8. Woodham, R.J.: Photometric method for determining surface orientation from multiple images. Optical Engineering **19**(1) (1980)
9. Basri, R., Jacobs, D., Kemelmacher, I.: Photometric stereo with general, unknown lighting. IJCV **72**(3) (May 2007) 239–257
10. Szeliski, R.: Computer Vision: Algorithms and Applications. Springer (2010)
11. Horn, B.K.: Shape from shading: A method for obtaining the shape of a smooth opaque object from one view. Technical report, Massachusetts Institute of Technology, Cambridge, MA, USA (1970)
12. Esteban, C.H., Vogiatzis, G., Cipolla, R.: Multiview photometric stereo. T-PAMI **30**(3) (March 2008) 548–554
13. Barron, J.T., Malik, J.: Intrinsic scene properties from a single rgb-d image. In: CVPR. CVPR '13 (2013) 17–24
14. Han, Y., Lee, J.Y., Kweon, I.S.: High quality shape from a single rgb-d image under uncalibrated natural illumination. In: ICCV. (Dec 2013) 1617–1624
15. Yu, L.F., Yeung, S.K., Tai, Y.W., Lin, S.: Shading-based shape refinement of rgb-d images. In: CVPR. (June 2013) 1415–1422
16. Kim, K., Torii, A., Okutomi, M.: Joint estimation of depth, reflectance and illumination for depth refinement. In: ICCV Workshops, IEEE Computer Society (2015) 199–207
17. Daniel Maurer, Yong Chul Ju, M.B., Bruhn, A.: Combining shape from shading and stereo: A variational approach for the joint estimation of depth, illumination and albedo. In: BMVC. (2016) 76.1–76.14
18. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: Conference on Computer Graphics and Interactive Techniques, New York, USA (1996) 303–312
19. Belhumeur, P., Kriegman, D., Yuille, A.: The bas-relief ambiguity. Int. Journal on Computer Vision **35**(1) (1999) 33–44
20. Peng, S., Haefner, B., Quèau, Y., Cremers, D.: Depth super-resolution meets uncalibrated photometric stereo. In: ICCV Workshops. (2017)
21. Zollhöfer, M., Dai, A., Innmann, M., Wu, C., Stamminger, M., Theobalt, C., Nießner, M.: Shading-based refinement on volumetric signed distance functions. ACM Trans. Graph. **34**(4) (July 2015) 96:1–96:14
22. Maier, R., Kim, K., Cremers, D., Kautz, J., Nießner, M.: Intrinsic3d: High-quality 3D reconstruction by joint appearance and geometry optimization with spatially-varying lighting. In: ICCV, Venice, Italy (October 2017)
23. Bylow, E., Sturm, J., Kerl, C., Kahl, F., Cremers, D.: Real-time camera tracking and 3d reconstruction using signed distance functions. In: RSS. Volume 9., Berlin, Germany (2013)
24. Quèau, Y., Lauze, F., Durou, J.D.: Solving the uncalibrated photometric stereo problem using total variation. In Kuijper, A., Bredies, K., Pock, T., Bischof, H., eds.: SSVM. (2013)
25. Maier, R.: Menderer - batch mesh renderer. https://github.com/robmaier/menderer (2019)