Multi-View Neural Surface Reconstruction with Structured Light

Chunyu Li
chunyuli@preferred.jp
Taisuke Hashimoto
hashimotot@preferred.jp
Eiichi Matsumoto
matsumoto@preferred.jp
Hiroharu Kato
hkato@preferred.jp

Preferred Networks, Inc.
3F Otemachi-building,
1-6-1 Otemachi, Chiyoda-Ku,
Tokyo, Japan

Abstract

Three-dimensional (3D) object reconstruction based on differentiable rendering (DR) is an active research topic in computer vision. DR-based methods minimize the difference between the rendered and target images by optimizing both the shape and appearance and realizing a high visual reproductivity. However, most approaches perform poorly for textureless objects because of the geometrical ambiguity, which means that multiple shapes can have the same rendered result in such objects. To overcome this problem, we introduce active sensing with structured light (SL) into multi-view 3D object reconstruction based on DR to learn the unknown geometry and appearance of arbitrary scenes and camera poses. More specifically, our framework leverages the correspondences between pixels in different views calculated by structured light as an additional constraint in the DR-based optimization of implicit surface, color representations, and camera poses. Because camera poses can be optimized simultaneously, our method realizes high reconstruction accuracy in the textureless region and reduces efforts for camera pose calibration, which is required for conventional SL-based methods. Experiment results on both synthetic and real data demonstrate that our system outperforms conventional DR- and SL-based methods in a high-quality surface reconstruction, particularly for challenging objects with textureless or shiny surfaces.

1 Introduction

Importing 3D objects from the real world into virtual worlds is an essential technology in digital arts, extended reality (XR) applications, cultural heritage protection [20, 23], paleontology [34], visual inspection [14], and 3D printing [4]. Conventional methods applied to achieve this automatically include photogrammetry, which integrates multiple 2D images from different views.

Differentiable rendering (DR) [15, 33] is an emerging tool for multi-view stereo. In contrast to traditional methods that rely on matching features from different views [5, 7,

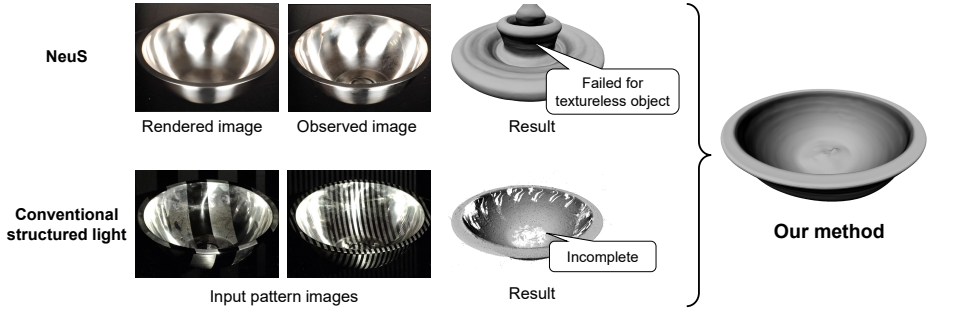


Figure 1: 3D reconstruction results on a textureless and shiny object (metallic bowl). NeuS [29], which only uses photometric information to supervise a learning process, failed to reconstruct the textureless and concave parts, even if the rendered images were close to the observed images. However, conventional structured-light (SL) technology, which is competent in a textureless surface, provides incomplete results owing to the highlights in captured pattern images for shiny object. By contrast, our method can achieve high-quality 3D reconstruction by combining SL technology and DR-based multi-view stereo.

8, 28], in DR-based methods, 3D geometries are directly optimized using gradient descent, such that rendered images are close to the observed images. A high-fidelity reconstruction can be achieved [22, 31] when combined with 3D representations using neural networks (*neural fields* [30]). However, DR-based methods are subject to an inherent problem in terms of the geometrical ambiguity of the observations. In other words, the observed images can be explained by several different geometries, although which among them is more accurate cannot be determined. This ambiguity is high on concave or textureless surfaces, such as the result produced by NeuS, as shown in Fig. 1.

Active vision alleviates this problem. A representative method is the use structured light (SL) [6, 13, 18, 19, 21, 26], in which multiple texture patterns are projected onto objects using a projector. Although the SL system requires an additional device (the projector), it can measure the depth (3D point clouds) with high quality even for textureless objects owing to an active projection. However, standard SL systems have certain disadvantages. First, the camera and projector must be rigidly fixed (mounted on a rig) and precisely calibrated. second, the SL system provides a 3D reconstruction of poor quality to recover optically complex scenes such as shiny objects, because the appearance of highlights and inter-reflections lead to incomplete 3D point cloud result. Third, SL systems are sensitive to occlusions, and by extension to holes, because they rely on an optical disparity.

In this work, we propose to introduce active sensing with SL into DR-based multi-view stereo. We follow the implicit differentiable renderer [29, 31, 32] to represent the surface as a zero-level set of a signed distance field (SDF) and the scene appearance as a color field. To train these networks, a sequence of SL patterns is projected onto the object surface and captured by two cameras while the object is arbitrarily rotated to obtain multiple observations of SL patterns around the object (see Fig. 2). The correspondence between the two camera views extracted from the captured images with projected patterns is used to supervise the training of the object surface. In addition, images without a SL pattern for each camera view are captured for photometric supervision, which has been widely used in previous DR frameworks. Both the surface and appearance of the object are optimized by minimizing the loss between observations and rendered images. On the one hand, compared to previous DR frameworks [29, 31, 32] which only use photometric data to supervise the training, the

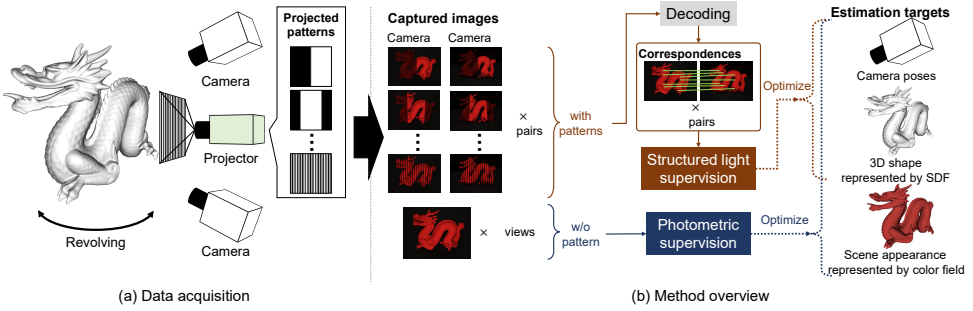


Figure 2: We use the projector to project a sequence of gray code patterns. To scan the whole shape of the object, we rotate it repeatedly until the whole shape of the object is scanned. We obtain N pairs of image sequences with projector patterns (used in SL supervision) and M multi-view images without projector pattern (used in photometric supervision).

SL supervision can provide important cues to reduce the geometrical ambiguity. On the other hand, shiny objects that cannot be handled by SL technology can be approximately represented using the color field. Thus, their shapes can be optimized through photometric supervision. In addition, self-occlusion, which cannot be handled by SL technology, can be solved through the global optimization of the surface represented by an implicit SDF using photometric supervision. Furthermore, by establishing a geometric relationship between the neural implicit surface and camera poses, the camera extrinsic parameters are also refined during optimization; therefore, our method does not require strict calibration in contrast to conventional SL methods [6, 13, 21, 26]. To the best of our knowledge, our work is the first to combine SL and differentiable rendering.

We experimentally validated the effectiveness of the proposed method under both synthetic and real-world scenarios. The results demonstrate that our method can reconstruct many types of challenging targets that are textureless, self-occluded, or shiny even with rough camera information. They also show that the proposed approach outperforms other state-of-the-art neural scene representation methods as well as conventional SL methods.

2 Method

We aim to reconstruct the 3D shape of an object from multi-view structured-light (SL) pattern images with rough camera poses with known intrinsic parameters. Additionally, the proposed method does not require mask supervision. Inspired by IDR [31], NeuS [29], and VolSDF [32], we adopt a neural implicit SDF and use a zero-level set to represent the surface of the object. To obtain a high-quality 3D model result, we introduce SL pattern consistency to supervise the neural network training. Our approach combines SL and DR-based multi-view stereo to ensure self-calibration and high-quality 3D reconstruction of objects.

2.1 Data acquisition and pattern decoding

Data acquisition: Fig. 2 (a) illustrates the data-acquisition procedure of proposed method. A projector was used to project a sequence of SL patterns, and two cameras (a and b) were used as imaging devices to capture images with projector patterns (used in SL supervision) and an image without the pattern (used in photometric supervision). To scan the entire shape of the object, we changed the viewpoint of the cameras and projector by rotating the object, as shown in Fig. 2 (a). Finally, we obtain N pairs of images sequences with projector

patterns (used in SL supervision) and M multi-view images without projector pattern (used in photometric supervision).

Images with patterns were further used to perform SL supervision. In our experiment, we generated patterns by encoding each pixel coordinate $\boldsymbol{q}$ of the projector to a binary gray code [9, 13]. In contrast to other patterns [12, 24, 25] that use different levels of intensity or different color to produce unique coding, gray code pattern only uses binary values (white and black), thus it is less sensitive to the surface characteristics. Our system generates $\log_2 O$ bits of gray code to assign an independent value to each pixel, where O is the number of projector rows (or columns). To improve the decoding accuracy, similar to conventional SL methods [13], the gray code patterns insert the original and its inverse to identify code words, a white image, and black illumination to identify shadow regions. Therefore, a sequence of gray code patterns consists of 46 frames for a projector with the resolution of 1920×1080 (including $2 \times \lceil \log_2 1920 \rceil = 22$ patterns representing the columns, $2 \times \lceil \log_2 1080 \rceil = 22$ patterns representing the rows, and one pair of white and black images).

Pattern decoding and noise reduction: After capturing the images we follow the decoding algorithm in [13] to decode each pixel $\boldsymbol{p}$ in the images into their corresponding decimal number $\boldsymbol{q}$ representing the column and the row of the projector. However, some noise will be captured due to the inter-reflection of the projected pattern, especially for shiny or concave surfaces. Therefore we determine whether a decoded pixel is affected by inter-reflection using the epipolar line. As the camera poses are unknown in our experiment, we calculate a rough fundamental matrix between the camera and projector from the noisy corresponding points, and estimate the epipolar lines using this fundamental matrix. Then, we eliminate correspondences whose camera pixels are not on the epipolar line. More details about noise reduction are in the supplementary material.

Finally, for each camera pair, we can map the pixels that share the same corresponding projector pixel $\boldsymbol{q}$ to output a list $\mathcal{A}$ of $\{\boldsymbol{p}_a, \boldsymbol{p}_b\}$, where $\boldsymbol{p}_a$ and $\boldsymbol{p}_b$ denote the pixel coordinate in camera a and b, respectively. By repeating the process for all N camera pairs, we can obtain N corresponding lists.

2.2 Geometry, appearance and camera pose representations

In our network, the surface $S(\theta)$ is modeled explicitly as the zero-level set of an SDF, which is represented by an MLP $f(\boldsymbol{x};\theta) : \mathbb{R}^3 \to \mathbb{R}$ with learnable parameters θ. The network f takes a query location $\boldsymbol{x} \in \mathbb{R}^3$ as input and outputs a signed distance from the location to the closest surface point (a positive distance for $\boldsymbol{x}$ outside and a negative distance for $\boldsymbol{x}$ inside). $S(\theta)$ can be represented as

$$S(\theta) = \left\{ \boldsymbol{x} \in \mathbb{R}^3 \mid f(\boldsymbol{x};\theta) = 0 \right\}.$$

We denote the camera poses (extrinsic parameters), including the camera positions and rotations, using the parameter τ, which are also optimized during network training. We assume the intrinsic parameters of the cameras are known. Given a pixel in a camera view, let $R(\tau)$ denote the ray through this pixel, and we obtain

$$R(\tau) = \{\boldsymbol{o} + t\boldsymbol{v} \mid t \geq 0\}, \tag{1}$$

where $\boldsymbol{o} = \boldsymbol{o}(\tau) \in \mathbb{R}^3$ is the center of the camera, $\boldsymbol{v} = \boldsymbol{v}(\tau) \in \mathbb{R}^3$ is the unit direction vector of the ray, and t is the distance from point $\boldsymbol{x}$ to camera center $\boldsymbol{o}$. Here, we suppose that the surface exists only in an interval $[t_{\mathrm{L}}, t_{\mathrm{R}}]$ such that $t \in [t_{\mathrm{L}}, t_{\mathrm{R}}]$.

The scene appearance is represented by a color field using another MLP $c(\boldsymbol{x}, \boldsymbol{v}; \phi) : \mathbb{R}^3 \times \mathbb{R}^3 \to \mathbb{R}^3$ with learnable parameters ϕ. This MLP c encodes the RGB color associated with the query location $\boldsymbol{x} \in \mathbb{R}^3$ and the viewing direction $\boldsymbol{v} \in \mathbb{R}^3$.

2.3 SL supervision

For high-quality 3D shape reconstruction, our main idea is to exploit the dense and accurate correspondences extracted by SL patterns as constraints during the optimization of the 3D shapes and camera poses.

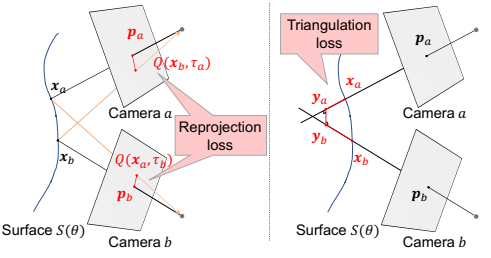


Figure 3: Reprojection loss $\mathcal{L}_{\mathrm{SR}}(\theta, \tau)$ (left) and triangulation loss $\mathcal{L}_{\mathrm{ST}}(\theta, \tau)$ (right).

Given the extracted correspondences between each camera pair, we calculate the intersections between the surface and the ray passing through each pixel, as shown in Fig. 3. Following previous works [22, 31], we first use a ray marching algorithm to find the intersection point, and construct differentiable intersection which has a correct value and first-order derivative with respect to θ and τ. Let $\boldsymbol{x} = \boldsymbol{o} + t\boldsymbol{v}$ denote the intersection point of the ray $R(\tau)$. For the current network parameters θ_0 and camera parameters τ_0, we denote $\boldsymbol{o}_0 = \boldsymbol{o}(\tau_0)$, $t_0 = t(\theta_0, \tau_0)$, $\boldsymbol{v}_0 = \boldsymbol{v}(\tau_0)$, and $\boldsymbol{x}_0 = \boldsymbol{o}_0 + t_0\boldsymbol{v}_0$. We take the implicit differentiation of equation $f(\boldsymbol{x}; \theta) \equiv 0$, and the surface intersection is expressed as a function of θ and τ:

$$\boldsymbol{x}(\theta, \tau) = \boldsymbol{o} + t_0\boldsymbol{v} - \frac{f(\boldsymbol{o} + t_0\boldsymbol{v}; \theta)}{\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_0; \theta_0) \cdot \boldsymbol{v}_0}\boldsymbol{v}, \tag{2}$$

where $\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_0; \theta_0)$ is constant.

We consider two types of pattern consistency: reprojection loss and triangulation loss. From our experiment (see supplementary material), we find that using both loss functions results in better reconstruction results than using only one of them.

Reprojection loss: The reprojection loss ensures that a surface point $\boldsymbol{x}$ on a ray is projected near pixel $\boldsymbol{p}$ from another view that corresponds to the ray, as shown in Fig. 3 left column. Concretely, it models the error between the reprojected pixels from the surface intersections and the corresponding pixels. The loss function is described as

$$\mathcal{L}_{\mathrm{SR}}(\theta, \tau) = \sum_{n}^{N} \sum_{i \in \mathcal{A}} \left(\| Q(\boldsymbol{x}_a^{n,i}, \tau_b^n) - \boldsymbol{p}_b^{n,i} \| + \| Q(\boldsymbol{x}_b^{n,i}, \tau_a^n) - \boldsymbol{p}_a^{n,i} \| \right), \tag{3}$$

where $Q(\boldsymbol{x}, \tau)$ is the reprojection of surface point $\boldsymbol{x}$ on the image with camera parameters τ.

Triangulation loss: The triangulation loss enforces consistency between the estimated 3D shape and 3D point cloud, which is directly calculated by triangulation from the extracted correspondences between each camera pair. As shown in the right column of Fig. 3, suppose we obtain a correspondence between camera pair a and b by decoding the SL pattern. We can separately calculate the intersections $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$ between the surface and the two camera rays via ray tracing as described above. In addition, we can obtain the intersection between these two camera rays by triangulation. However, considering that these two rays may not intersect because of the camera pose error, we calculate the closest point $\boldsymbol{y}_a$ to the ray $R_b(\tau)$ on the ray $R_a(\tau)$ and the closest point $\boldsymbol{y}_b$ to the ray $R_a(\tau)$ on the ray $R_b(\tau)$. More details on

the calculation of $\boldsymbol{y}_a$ and $\boldsymbol{y}_b$ are provided in the supplementary material. Finally, to ensure that these four points ($\boldsymbol{x}_a^{n,i}$, $\boldsymbol{x}_b^{n,i}$, $\boldsymbol{y}_a^{n,i}$ and $\boldsymbol{y}_b^{n,i}$) are located in the same position, we calculate the triangulation loss which evaluates the distance between these four points, as given below.

$$\mathcal{L}_{\mathrm{ST}}(\theta,\tau)=\sum_{n}^{N}\sum_{i\in\mathcal{A}}\left(\|\boldsymbol{x}_a^{n,i}-\boldsymbol{y}_a^{n,i}\|+\|\boldsymbol{y}_a^{n,i}-\boldsymbol{y}_b^{n,i}\|+\|\boldsymbol{y}_b^{n,i}-\boldsymbol{x}_b^{n,i}\|\right). \tag{4}$$

2.4 Photometric supervision

The SL supervision in Section 2.3 can correctly recover the surface geometry. However, as the correspondences extracted using SL patterns are usually noisy and incomplete for some special materials, such as shiny surfaces, we propose to consider rendered image consistency during network training.

Because we acquire multi-view observations by rotating an object with fixed camera positions, as described in Section 2.1, the background for each observation is constant and can be easily obtained in advance. Therefore, our SDF f and color field c only represent the shape and appearance of the foreground (i.e., the object). The rendered images are obtained by mixing the rendered foreground images from the neural networks and known background images. For foreground rendering, we use the same rendering method as NeuS [29], which adopts a volume-rendering scheme. Specifically, the output color for pixel k in the rendered image is calculated by accumulating the weighted colors along the ray. Because the MLP would only be queried at a discrete set of locations, the viewing ray is sampled by partitioning $[t_{\mathrm{L}},t_{\mathrm{R}}]$ into n evenly-spaced bins, and Eq. (1) can be rewritten as follows.

$$R(\tau)=\left\{\boldsymbol{o}+t_j\boldsymbol{v}\mid j=1,...,n,t_j<t_{j+1}\right\}. \tag{5}$$

The foreground rendering formula for this ray can be defined as

$$C_{\mathrm{fore}}(\theta,\phi,\tau)=\sum_{j=1}^{n}w_j(\theta)c(\boldsymbol{o}+t_j\boldsymbol{v},\boldsymbol{v};\phi), \tag{6}$$

where $w_j(\theta)$ is the weight of the sampled point $\boldsymbol{o}+t_j\boldsymbol{v}$, which is a function of the distance to the surface $S(\theta)$. Thus, the weight is the connection between the output colors and the implicit SDF f. For details regarding the weight function, please refer to [29]. The rendered color is then calculated by mixing the estimated foreground $C_{\mathrm{fore}}(\theta,\phi,\tau)$ and the known background C_{back}.

$$C(\theta,\phi,\tau)=C_{\mathrm{fore}}(\theta,\phi,\tau)+C_{\mathrm{back}}\left(1-\sum_{j=1}^{n}w_j(\theta)\right). \tag{7}$$

Even though our method assumes known background images, in our setup obtaining background images is easier than calculating accurate object masks. The object masks have to be generated from input images by manual annotation or some automatic foreground detection methods. However automatic methods are sometimes inaccurate, so eventually the manual annotation might be required. And the performance of conventional methods highly relies on the accuracy of the object masks. Thus the mask generation is always a costly process for conventional methods. On the other hand, our method uses a turntable to change the direction of the object, thus the relative positions between cameras and the background are fixed (i.e. the background is constant in all viewpoints for one camera). Therefore, we

just capture one background image (without object) for each camera before (or after) object capture.

Finally, given the rendered color $C^k = C^k(\theta, \phi, \tau)$ of pixel k and the input image color I^k of pixel k, the render loss is calculated as the L1 distance.

$$\mathcal{L}_{\mathrm{R}}(\theta, \phi, \tau) = \sum_{k \in \mathcal{V}_I} |I^k - C^k|, \tag{8}$$

where $\mathcal{V}_I$ denotes the set of all image pixels in all camera views without patterns.

For shiny surfaces, the 3D point clouds extracted from SL patterns are usually incomplete, as mentioned above. Thus we consider the photometric supervision by minimizing the error between the observed images and the rendered images to guarantee geometry accuracy in the missing areas of SL supervision. However, the view-dependent (specular) reflection model of shiny surfaces using the traditional point cloud or mesh is high-complexity. By contrast, the SDF and color field are able to efficiently represent the view-dependent scene. Thus our optimization framework introducing the DR represented by SDF is reasonable and effective to make up the shortcomings of SL methods.

2.5 Training

Similar to previous works [29, 31], we regularize our SDF network with an Eikonal loss function [11] that restricts the expectation of the gradient magnitude to 1, as given below.

$$\mathcal{L}_{\mathrm{E}}(\theta) = \mathbb{E}_{\boldsymbol{x}} \left(\|\nabla_{\boldsymbol{x}} f(\boldsymbol{x}; \theta)\| - 1 \right)^2. \tag{9}$$

The final loss is expressed as a weighted sum of all the losses listed above.

$$\mathcal{L}(\theta, \phi, \tau) = \mathcal{L}_{\mathrm{R}}(\theta, \phi, \tau) + \lambda_{\mathrm{SR}} \mathcal{L}_{\mathrm{SR}}(\theta, \tau) + \lambda_{\mathrm{ST}} \mathcal{L}_{\mathrm{ST}}(\theta, \tau) + \lambda_{\mathrm{E}} \mathcal{L}_{\mathrm{E}}(\theta). \tag{10}$$

3 Experiments

3.1 Experimental setting

Datasets: We experimentally validated the effectiveness of the proposed method for real-world and synthetic scenes, with a wide variety of materials, appearances and geometries, including challenging cases for reconstruction algorithms, such as textureless and glossy surfaces. Because there is no existing multi-view structured light dataset which is directly applicable to our setup, all the datasets used in our experiment were produced by the authors.

For experiments on real-world scenes, we used a Mitsubishi LVP-FD630 projector and two Sony α6600 digital cameras in our projector-camera system. The sequence of structured light patterns was encoded with a resolution of 1920×1080 and captured by the cameras using a video format with a resolution of 3840×2160. We used a turntable to rotate the object, and each scene was captured from $N = 12$ pairs of camera viewpoints with structured-light patterns, and $M = 270$ single images without structured-light patterns. The initial camera poses were measured using AprilTag 16h5 Markers [3] fixed on a turntable. Details of the strategy for the initial camera pose estimation are provided in the supplementary material.

For experiments on synthetic scenes, we used a Dragon model obtained from the Stanford 3D Scanning Repository [10] and a Bowl model downloaded from the Internet [1]. To demonstrate the proposed method on the challenging targets, we rendered all the models with shiny materials, such as plastic (Fig. 4, left), ceramic (Fig. 4, right) and marble (Fig. 5); for each synthetic scene, we generated $N = 20$ pairs of camera viewpoints with structured-light

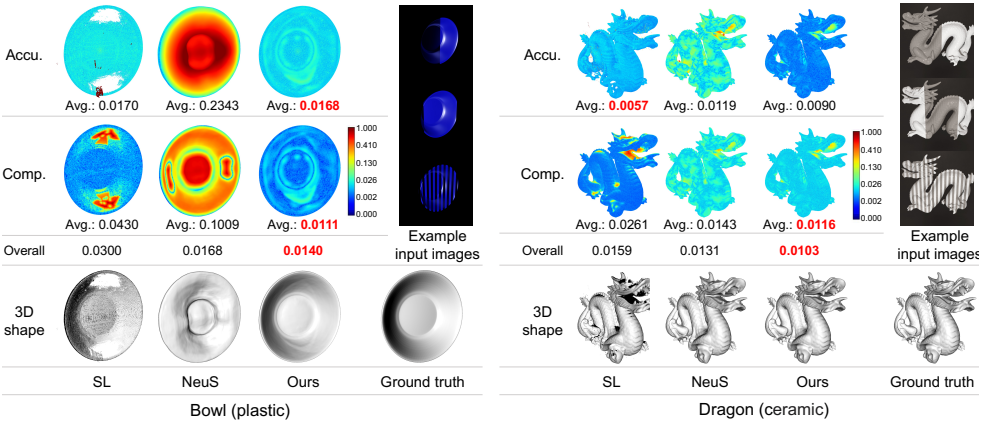


Figure 4: Example input images, 3D reconstruction results, and their accuracy and completeness errors on two synthetic scenes with *fixed ground truth* camera poses.

patterns and $M = 40$ single images without structured-light patterns. The evaluation of the other models is included in the supplementary material.

Evaluation metrics: To evaluate the 3D shape quality, we used the accuracy (Accu.) and completeness (Comp.) as two metrics [2, 17]. The accuracy is the distance from each estimated 3D point to its nearest ground-truth 3D point. The completeness is the distance from each ground-truth 3D point to its nearest estimated 3D point. We define the overall score, the average of mean accuracy and mean completeness, as the reconstruction quality.

Implementation details: For the MLPs of the implicit SDF f and color field c, we followed the architectures used in IDR [31] and NeuS [29]. We implemented the proposed method in PyTorch and trained our model using the Adam optimizer [16]. The learning rates were first linearly warmed up from 0 to maximums (5.0×10^{-4} for MLP training and 1.0×10^{-5} for camera pose training) in the first 5k iterations, and then controlled by the cosine decay schedule to the minimum learning rates (2.5×10^{-5} for MLP training and 5.0×10^{-7} for camera pose training). The loss weights in Eq. (10) were empirically set as $\lambda_{\mathrm{SR}} = 1.0 \times 10^{-4}$, $\lambda_{\mathrm{ST}} = 0.1$, and $\lambda_{\mathrm{E}} = 0.1$. We sampled 512 rays per batch and trained our model for 100k iterations for 7 h with a single NVIDIA V100 Tensor Core GPU.

3.2 Effectiveness of DR and SL combination

The core of our proposed system is to combine the differentiable rendering (DR)- and structured-light (SL)-based methods to obtain the benefits of both. In this subsection, we describe our evaluation of the proposed approach. We used our method to generate 3D reconstructions in two different setups: (1) fixed ground-truth cameras and (2) trainable cameras with noisy initializations obtained using an SfM approach [27]. We compared our method with a DR-based method called NeuS [29] and an existing SL method [13]. In the SL method [13], we first reconstructed the 3D point cloud for each camera pair based on triangulation using the extracted corresponding points. We tried both the ground truth and noisy camera poses for the reconstruction. Subsequently, the 3D point cloud parts were aligned using the input camera pose. For a fair comparison, we show the results of the existing SL method [13] after the noise reduction which is described in Section 2.1, even though the original approach does not include this process.

Fig. 4 shows the reconstructed 3D shapes and their quantitative evaluations of the syn-

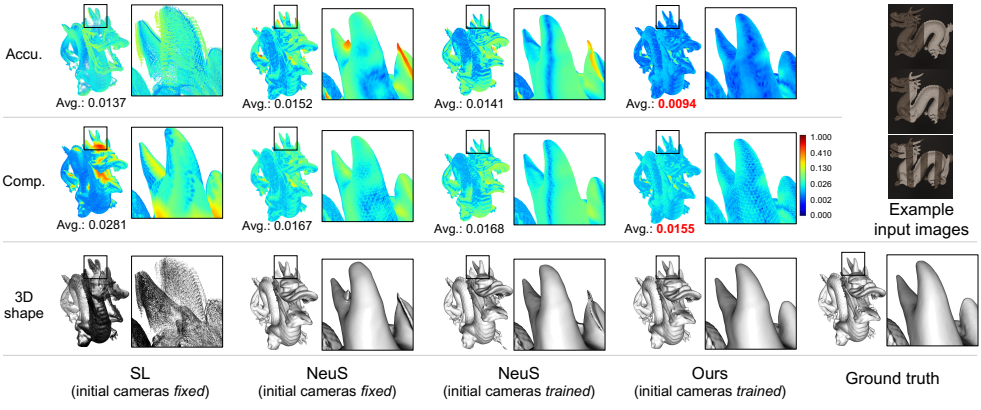


Figure 5: Example input images, 3D reconstruction results, and their accuracy and completeness errors on the synthetic Dragon model (marble) with *noisy* camera poses.

thetic dataset using fixed ground-truth camera poses. For quantitative evaluations, the calculation of accuracy and completeness is conducted using the same density of 3D point clouds. The completeness and accuracy errors for each 3D point were colorized, and the average errors are shown below the error maps. Compared to NeuS, which only uses photometric supervision based on a passive illumination, our method reconstructed more accurate results because structured light can reduce the geometric ambiguity of textureless surfaces. In the SL method, the inter-reflection of the inner surface of the Bowl model results in holes, whereas the proposed method provides a complete and accurate result using photometric supervision based on differentiable rendering. Although the SL method provides better accuracy on the Dragon model (after the noise reduction), the result of the 3D point cloud is incomplete owing to the occlusion (e.g. inside of the mouse of Dragon). Regarding the average of accuracy and completeness results, our method was able to provide better results than the SL method.

The results obtained using the initial noisy camera poses shown in Fig. 5 demonstrate the effectiveness of our global optimization of the camera poses. Although NeuS does not optimize the camera poses, we attempted to incorporate camera training into their original method for comparison. It can be observed from the results that our method outperformed all baselines by training the camera poses using structured light supervision. The quality of the 3D reconstruction degraded for the NeuS and SL methods, which require high-accuracy camera calibration. In addition, our method performed better than NeuS with camera training, indicating that structured light supervision contributed to the improvement of accuracy during training for both 3D shape and camera poses. Table 1 shows a comparison of camera directions (Dire.) and positions (Posi.) between initial values and optimized values (Opt.). Note the considerable improvement in optimized camera accuracy over initial values.

Table 1: Camera poses accuracy on the Dragon model (marble).

	Initial	Opt.
Dire.(deg)	0.070	**0.049**
Posi.(m)	0.075	**0.011**

3.3 Evaluation on real-world dataset

So far, evaluations have been conducted on a synthetic dataset because the ground truth geometries for a quantitative evaluation are available. Here, we present the results on a real-world dataset with noisy camera parameters. Fig. 6 shows the reconstruction results for (a) a metallic ashtray, (b) a ceramic bowl, (c) a fan and (d) a hanger, compared with the NeuS and SL methods. Both targets of the ashtray and bowl were textureless objects, which led to

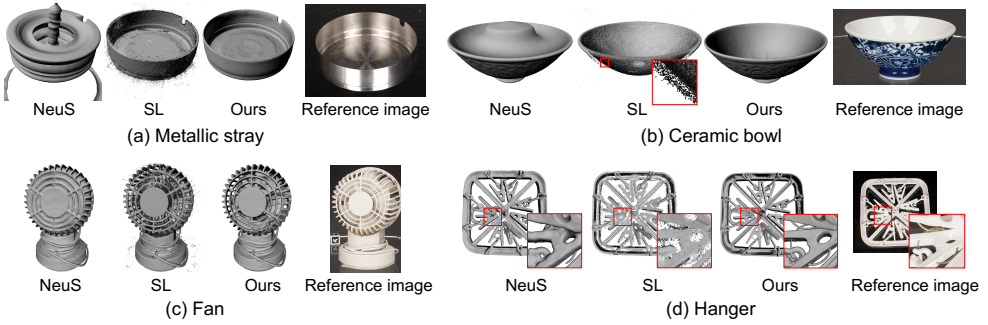


Figure 6: 3D reconstruction results on the real dataset.

a geometric ambiguity for NeuS. Therefore, we can confirm that NeuS fails to reconstruct the concave parts. In the close-up of the bowl, we can see that the SL method produces a double-layer surface. This occurred because the point clouds reconstructed from different camera pairs could not be tightly aligned based on noisy camera poses. In addition, it may be observed that the shiny surface of the ash tray led to many error points around the surface for the SL method. By contrast, our proposed approach, which benefits from both DR- and SL-based methods, provided the most accurate 3D reconstruction results.

Ablation study and **limitations** of proposed method are described in supplementary.

4 Conclusion

We proposed to supervise multi-view neural surface reconstruction by active sensing using structured light. Although existing neural reconstitution methods suffer from textureless surfaces, point clouds and multi-view correspondences obtained by structured-light provide sparse but more accurate supervision in such cases. On the other hand, structured-light systems are unsuitable for reflective surfaces and occlusions. These weaknesses are alleviated by the dense photometric supervision based on differentiable rendering. In experiments conducted on both synthetic and real-world datasets, we demonstrated that this combination significantly improves the performance of reconstructing challenging objects, and our method outperforms state-of-the-art neural surface reconstruction methods and conventional structured-light-based methods. We also found that the end-to-end self-calibration of camera poses enabled by our proposed loss functions is crucial for a high-quality reconstruction.

References

[1] TurboSquid. https://www.turbosquid.com/3d-models/free-simple-bowl-3d-model/893871.

[2] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *IJCV*, 120(2): 153–168, 2016.

[3] Richardson Andrew, Strom Johannes, and Olson Edwin. AprilCal: Assisted and repeatable camera calibration. In *IROS*, 2013.

[4] John Biehler and Bill Fane. *3D Printing with Autodesk: Create and Print 3D Objects with 123D, AutoCAD and Inventor*. Que Publishing, 2014.

[5] Neill DF Campbell, George Vogiatzis, Carlos Hernández, and Roberto Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *ECCV*, 2008.

[6] Dalit Caspi, Nahum Kiryati, and Joseph Shamir. Range imaging with adaptive color structured light. *TPAMI*, 20(5):470–480, 1998.

[7] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *TPAMI*, 32(8):1362–1376, 2009.

[8] Silvano Galliani, Katrin Lasinger, and Konrad Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *ICCV*, 2015.

[9] Jason Geng. Structured-light 3D surface imaging: A tutorial. *Advances in Optics and Photonics*, 3(2):128–160, 2011.

[10] Turk Greg and Levoy Marc. The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3Dscanrep/.

[11] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *ICML*, 2020.

[12] Carrie L Heike, Kristen Upson, Erik Stuhaug, and Seth M Weinberg. 3D digital stereophotogrammetry: a practical guide to facial image acquisition. *Head & face medicine*, 6(1):1–11, 2010.

[13] Kyriakos Herakleous and Charalambos Poullis. 3DUNDERWORLD-SLS: An open-source structured-light scanning system for rapid geometry acquisition. *arXiv*, 2014.

[14] Sanao Huang, Ke Xu, Ming Li, and Mingren Wu. Improved visual inspection through 3d image reconstruction of defects based on the photometric stereo technique. *Sensors*, 19(22):4970, 2019.

[15] Hiroharu Kato, Deniz Beker, Mihai Morariu, Takahiro Ando, Toru Matsuoka, Wadim Kehl, and Adrien Gaidon. Differentiable rendering: A survey. *arXiv*, 2020.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[17] Andreas Ley, Ronny Hänsch, and Olaf Hellwich. Syb3r: A realistic synthetic benchmark for 3D reconstruction from images. In *ECCV*, 2016.

[18] Chunyu Li, Akihiko Torii, and Masatoshi Okutomi. Robust, precise, and calibration-free shape acquisition with an off-the-shelf camera and projector. *Proc. of IEEE Conf. on Consumer Electronics (ICCE)*, 2018.

[19] Chunyu Li, Yusuke Monno, Hironori Hidaka, and Masatoshi Okutomi. Pro-Cam SSfM: Projector-camera system for structure and spectral reflectance from motion. In *ICCV*, 2019.

[20] Renju Li, Tao Luo, and Hongbin Zha. 3d digitization and its applications in cultural heritage. In *Proc. Digital Heritage*, 2010.

[21] Hieu Nguyen, Dung Nguyen, Zhaoyang Wang, Hien Kieu, and Minh Le. Real-time, high-accuracy 3D imaging and shape measurement. *Applied Optics*, 54(1):A9–A17, 2015.

[22] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *CVPR*, 2020.

[23] Laura Niven, Teresa E Steele, Hannes Finke, Tim Gernat, and Jean-Jacques Hublin. Virtual skeletons: using a structured light scanner to create a 3D faunal comparative collection. *Journal of Archaeological Science*, 36(9):2018–2023, 2009.

[24] Pierre Payeur and Danick Desjardins. Structured light stereoscopic imaging with dynamic pseudo-random patterns. In *ICIAR*, pages 687–696, 2009.

[25] Tomislav Pribanić, Saša Mrvoš, and Joaquim Salvi. Efficient multiple phase shift patterns for dense 3d acquisition in structured light scanning. *Image and Vision Computing*, 28(8):1255–1266, 2010.

[26] Joaquim Salvi, Sergio Fernandez, Tomislav Pribanic, and Xavier Llado. A state of the art in structured light patterns for surface profilometry. *Pattern Recognition*, 43(8): 2666–2680, 2010.

[27] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.

[28] Engin Tola, Christoph Strecha, and Pascal Fua. Efficient large-scale multi-view stereo for ultra high-resolution image sets. *Machine Vision and Applications*, 23(5):903–920, 2012.

[29] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, 2021.

[30] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *arXiv*, 2021.

[31] Lior Yariv, Yoni Kasten, Dror Moran, Meirav Galun, Matan Atzmon, Ronen Basri, and Yaron Lipman. Multiview neural surface reconstruction by disentangling geometry and appearance. In *NeurIPS*, 2020.

[32] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. *NeurIPS*, 2021.

[33] Shuang Zhao, Wenzel Jakob, and Tzu-Mao Li. Physics-based differentiable rendering: from theory to implementation. In *ACM SIGGRAPH Courses*. 2020.

[34] Michael J Ziegler, Victor J Perez, Jeanette Pirlo, Rachel E Narducci, Sean M Moran, Molly C Selba, Alexander K Hastings, Claudia Vargas-Vergara, Pavlo D Antonenko, and Bruce J MacFadden. Applications of 3D paleontological data at the Florida Museum of natural history. *Frontiers in Earth Science*, 2020.

Supplementary Material to Multi-View Neural Surface Reconstruction with Structured Light

Chunyu Li
chunyuli@preferred.jp
Taisuke Hashimoto
hashimotot@preferred.jp
Eiichi Matsumoto
matsumoto@preferred.jp
Hiroharu Kato
hkato@preferred.jp

Preferred Networks, Inc.
3F Otemachi-building,
1-6-1 Otemachi, Chiyoda-Ku,
Tokyo, Japan

1 Details on noise reduction

As described in Section 2.1 of the main paper, we reduce the misdetection of the structured-light pattern caused by inter-reflection by calculating the epipolar line between the projecter and camera pair. To be specific, as shown in Fig. 1, the light projected from the projector pixel $\boldsymbol{q}$ can reach the camera in one of two general ways: (1) by direct surface reflection, captured by a camera pixel $\boldsymbol{p}$ on the epipolar line (black path), which is the desirable path of the light for pattern decoding, or (2) by inter-reflection, captured by a camera pixel $\boldsymbol{p}'$ that is not on the epipolar line (orange path). Therefore, we can determine whether a decoded pixel is affected by inter-reflection using the epipolar line. As the camera poses are unknown in our experiment, we calculate a rough fundamental matrix between the camera and projector from the noisy corresponding points using Ransac algorithm, and estimate the epipolar lines using this fundamental matrix. Then, we eliminate correspondences whose camera pixels are not on the epipolar line. Note that although we can effectively reduce most noise using this strategy, some limitations remain: (1) the estimated epipolar lines may include minor errors owing to the noisy corresponding points, and (2) we cannot eliminate the inter-reflected correspondences whose projector and camera pixels are on corresponding epipolar lines. However, the amount of noise caused by these cases is small, so they can be further reduced by the

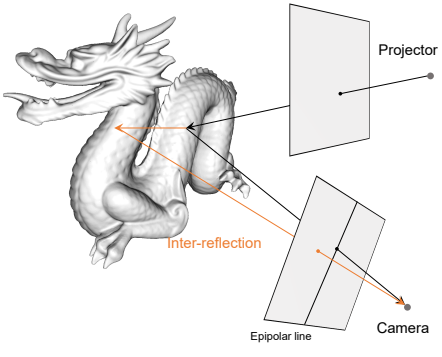


Figure 1: Illustration of pattern misdetection caused by inter-reflection.

photometric supervision introduced in Section 2.4 of the main paper. The effectiveness of this noise-reduction strategy is demonstrated by the ablation study (see Section 4.2 in supplementary material).

2 Details on triangulation

In this section we will explain the details on the calculation of $\boldsymbol{y}_a$ and $\boldsymbol{y}_b$ in Eq. (5) of the main paper. $\boldsymbol{y}_a$ and $\boldsymbol{y}_b$ are the nearest points between the two skew camera rays $R_a(\tau)$ and $R_b(\tau)$ (see the right column of Fig. 4). We denote $R_a(\tau) = \{\boldsymbol{o}_a + t_a\boldsymbol{v}_a \mid t_a \geq 0\}$ and $R_b(\tau) = \{\boldsymbol{o}_b + t_b\boldsymbol{v}_b \mid t_b \geq 0\}$. The cross product of $\boldsymbol{v}_a$ and $\boldsymbol{v}_b$ is perpendicular to the lines:

$$\boldsymbol{n} = \boldsymbol{v}_a \times \boldsymbol{v}_b. \tag{1}$$

The plane formed by the translations of $R_b(\tau)$ along $\boldsymbol{n}$ contains the point $\boldsymbol{o}_b$ and is perpendicular to $\boldsymbol{n}_1 = \boldsymbol{v}_b \times \boldsymbol{n}$. Therefore, the intersecting point of $R_a(\tau)$ with the above-mentioned plane, which is also the point on $R_b(\tau)$ that is nearest to $R_a(\tau)$, is given by

$$\boldsymbol{y}_a = \boldsymbol{o}_a + \frac{(\boldsymbol{o}_b - \boldsymbol{o}_a)\cdot\boldsymbol{n}_1}{\boldsymbol{v}_a\cdot\boldsymbol{n}_1}\boldsymbol{v}_a. \tag{2}$$

Similarly, the point on $R_b(\tau)$ nearest to $R_a(\tau)$ is given by

$$\boldsymbol{y}_b = \boldsymbol{o}_b + \frac{(\boldsymbol{o}_a - \boldsymbol{o}_b)\cdot\boldsymbol{n}_2}{\boldsymbol{v}_b\cdot\boldsymbol{n}_2}\boldsymbol{v}_b, \tag{3}$$

where $\boldsymbol{n}_2 = \boldsymbol{v}_a \times \boldsymbol{n}$.

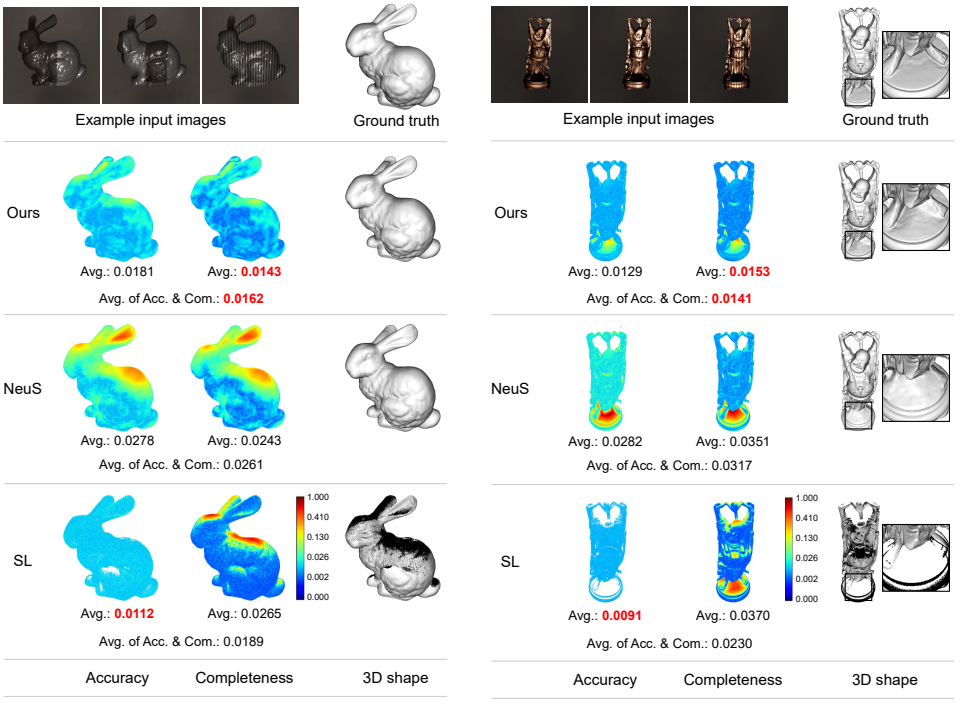
3 Initial camera poses estimation for real-world dataset

In the experiment on real-world scenes, the initial camera poses were measured using 26 AprilTag 16h5 Markers [3] fixed on the turntable. We assume the intrinsic parameters of the cameras are known. After capturing the multi-view input images, the initial camera poses are estimated following four steps.

Step 1. Marker Detection: Given each image containing AprilTag 16h5 Markers, the detection process has to return a list of detected markers. Each detected marker includes the position of its four corners in the image and the id of the marker. This step is implemented using OpenCV ArUco module [1].

Step 2. Camera Pose Initialization: The next thing is to obtain the camera pose from detected markers. First, for each image, the pose of each marker in the camera coordinate system is estimated individually using OpenCV ArUco module [1]. Then using one marker as a reference , all camera poses in one coordinate system can be obtained by calculating the 3D transformation from each camera coordinate systems to the reference marker coordinate system.

Step 3. Camera Pose Optimization: The camera poses obtained by Step 2 usually have large error. Next they are optimized using bundle adjustment while simultaneously updating the marker poses. Specifically, our bundle adjustment jointly refining the camera poses and marker poses by minimizing the reprojection error of four corners of each marker.



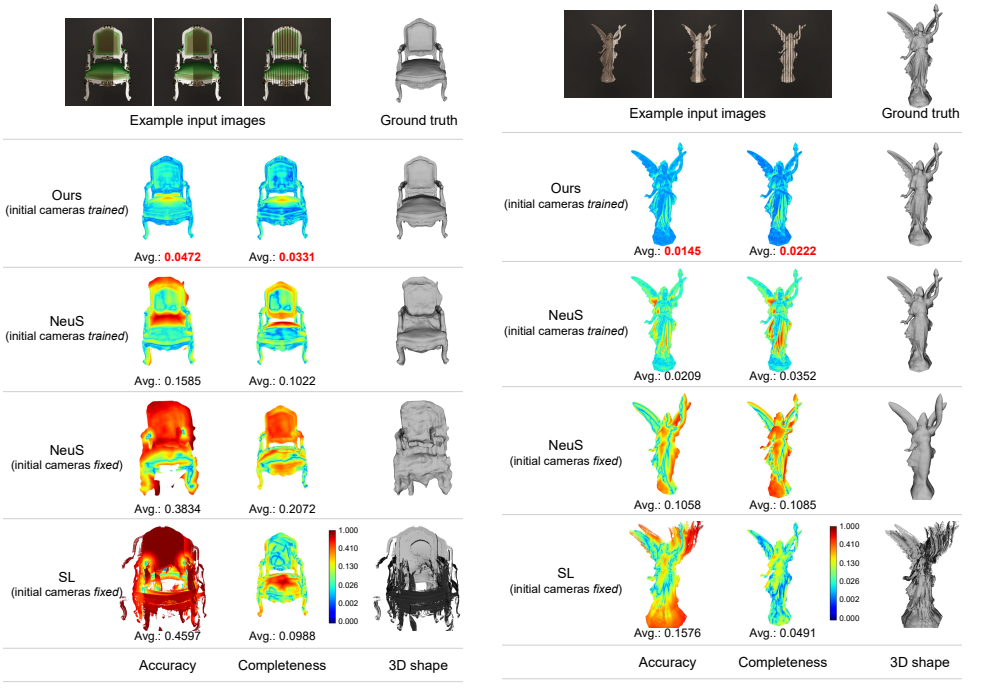
(a) Stanford Bunny (glass) (b) Happy Buddha (metal)

Figure 2: Example input images, 3D reconstruction results, and their completeness and accuracy errors on two additional synthetic scenes with *fixed ground truth* camera poses.

4 Additional experimental results

4.1 Simulation results

In this section, we show additional quantitative simulation results on a Stanford Bunny model (Fig. 2 (a)), Happy Buddha model (Fig. 2 (b)) and a Lucy model (Fig. 3 (b)) obtained from the Stanford 3D Scanning Repository [4] and a Chair model with thin structure downloaded from the Internet [2]. To demonstrate the proposed method on the challenging targets, we rendered the models from the Stanford 3D Scanning Repository with different shiny materials, such as glass (Stanford Bunny), metal (Happy Buddha) and marble (Lucy). For each synthetic scene, the input images are generated using the same setup as described in Section 4.1 of the main paper. We used our method to generate 3D reconstructions in two different setups: (1) *fixed ground-truth* camera poses and (2) trainable camera poses with *noisy* initializations obtained using an SfM approach [5]. Fig. 2 shows the comparisons with baseline methods with *fixed ground truth* camera poses. Fig. 3 shows the comparisons with baseline methods with *noisy* camera poses calculated by Colmap. In Table 1 we show a comparison of camera directions (Dire.) and positions (Posi.) between the noisy initial values and optimized values (Opt.). Note the considerable improvement in optimized camera accuracy over initial values.



(a) Chair (b) Lucy (marble)

Figure 3: Example input images, 3D reconstruction results, and their completeness and accuracy errors on two additional synthetic scenes with *noisy* camera poses.

Table 1: Camera poses accuracy w.r.t the ground truth.

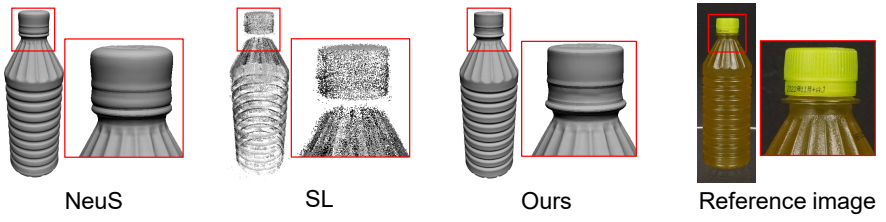
	Chair		Lucy	
	Initial	Opt.	Initial	Opt.
Dire.(deg)	2.832	**0.177**	0.781	**0.106**
Posi.(m)	0.119	**0.037**	0.830	**0.044**

4.2 Ablation studies

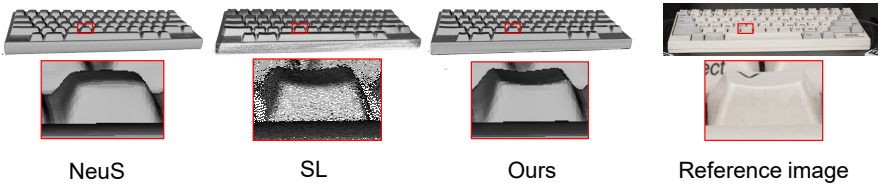
We used the glossy marble Dragon model (the same scene in Fig. 6 of the main paper) to conduct the ablation study. First, to confirm the contribution of the individual loss used for structured-light supervision (reprojection loss $\mathcal{L}_{\text{SR}}$ and triangulation loss $\mathcal{L}_{\text{ST}}$), we test following two cases: (a) w/o $\mathcal{L}_{\text{SR}}$ (by setting $\lambda_{\text{SR}} = 0$), (b) w/o $\mathcal{L}_{\text{ST}}$ (by setting $\lambda_{\text{ST}} = 0$). The quantitative results are shown in Table 2. We can confirm that the (e) full model that uses both of $\mathcal{L}_{\text{SR}}$ and $\mathcal{L}_{\text{ST}}$ achieves the best result. We also studied the effect of the noise reduction of decoding. The noises caused by inter-reflection leads to a deteriorated reconstruction quality as shown in Table 2 (c) when compared with the (e) full model which reduced the noises. In Table 2 (d) we show the result of training with fixed camera poses set to the inaccurate camera initializations obtain with SfM [5]. This indicates that the joint optimization of camera poses and 3D geometry is indeed significant.

Table 2: Quantitative results of ablation studies.

	Avg. of acc.	Avg. of comp.
(a) w/o $\mathcal{L}_{\text{SR}}$	0.0101	0.0157
(b) w/o $\mathcal{L}_{\text{ST}}$	0.0114	0.0160
(c) w/o noise reduction	0.0174	0.0183
(d) initial cameras fixed	0.0191	0.0194
(e) full model	0.0094	0.0155



(a) Plastic bottle



(b) Keyboard

Figure 4: Additional 3D reconstruction results on the real dataset.

4.3 Results for real-world scenes

In Fig. 4 we present additional qualitative results on the real dataset. The data acquisition follows the same setup as described in Section 4.1 of main papaer. We can confirm that proposed method perform better than all baseline methods.

4.4 Limitations

Although our method produces satisfactory results in most cases, it has several limitations. First, the projector pattern will not be captured by the cameras, and no correspondences can be obtained if the material of the object is mirror-like. In this case our method only relies on photometric supervision. In Fig. 5 we show a failure case on a synthetic scene with a textureless and mirror-like reflection. Our method fails to reconstruct an accurate surface owing to the lack of structured-light supervision. It should be noted that this material is also challenging for other state-of-the-art methods. Second, although our method can optimize camera poses, it requires a reasonable camera pose initialization using markers or SfM softwares.

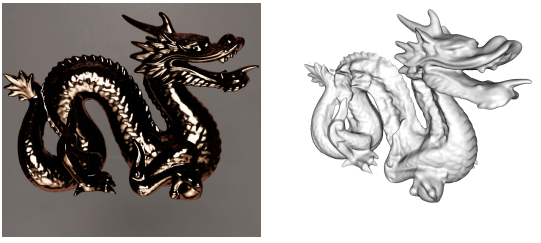


Figure 5: A failure case on a mirror-like object.

References

[1] Detection of ArUco Markers. https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html.

[2] Blend Swap. https://www.blendswap.com/blend/8261.

[3] Richardson Andrew, Strom Johannes, and Olson Edwin. AprilCal: Assisted and repeatable camera calibration. In *IROS*, 2013.

[4] Turk Greg and Levoy Marc. The Stanford 3D Scanning Repository. http://graphics.stanford.edu/data/3Dscanrep/.

[5] Johannes L. Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016.