Multi-View Photometric Stereo Revisited –Supplementary Material

Berk Kaya[1] Suryansh Kumar[1*] Carlos Oliveira[1] Vittorio Ferrari[2] Luc Van Gool[1,3]
ETH Zürich[1], Google Research[2], KU Leuven[3]

Abstract

Our supplementary material extends the experimental analysis presented in the main paper. To this end, we first demonstrated the high-frequency details of our method's 3D reconstructions. Next, an analysis of our method's performance under imaging noise is presented. Additionally, a brief description of the classical image formation PS model is outlined emphasizing the challenges with it to model the object with different material surfaces and the BRDF's role. Finally, we provide a detailed description of our deep-PS and deep-MVS networks for completeness. ***For convincing comparison showing the advantage of our approach, we provide a supplementary video clip. We urge the readers to view our supplementary video.***

1. Additional Results

(a) High-Frequency Details. Here, we extend the qualitative comparison provided in the main paper by demonstrating the high-frequency details of our 3D reconstructions. Fig.1 visually compares the 3D reconstructions obtained by our method and other MVPS methods, focusing on the ear of Buddha object. Clearly, our method recovers fine details that are missing in other methods and provides outstanding 3D reconstructions. For more of such results, refer to our supplementary video.

(b) Effect of Image Noise. The performance of MVPS is indeed affected by the image noise. With this study, we investigate the influence of image noise on our 3D reconstruction quality. To that end, we add zero-mean Gaussian noise to our input images with various standard deviations $\bar{\sigma}$. In Fig.2, we show F-score metric results as a function of its distance error threshold ϵ on Buddha, Pot2 and Reading categories of DiLiGenT-MV [11]. It can be observed that our method performs best under noiseless setting and the reconstruction accuracy slightly drops with increasing noise. From these results, we conclude that our method is robust to noise and provides reliable results under challenging scenarios.

*Corresponding Author (k.sur46@gmail.com)

2. Image Formation and BRDF.

Photometric stereo (PS) methods predict the surface normals of an object from its images captured under different illumination conditions. The classical setup for PS assumes distant j point light sources at direction $\mathbf{l}_j$ and intensity e_j. Under this configuration, the image intensity of surface point $\mathbf{p}_i$ observed at the viewing direction $\mathbf{v}$ is modeled as follows:

$$X_j^v(\mathbf{p}_i) = e_j \cdot \rho\big(\mathbf{n}_i(\mathbf{p}_i), \mathbf{l}_j, \mathbf{v}\big) \cdot \zeta_a\big(\mathbf{n}_i(\mathbf{p}_i), \mathbf{l}_j\big) \cdot \zeta_c(\mathbf{p}_i) \tag{1}$$

Here, $\zeta_a(\mathbf{n}_i(\mathbf{p}_i), \mathbf{l}_j) = \max(\mathbf{n}_i(\mathbf{p}_i)^T \mathbf{l}_j, 0)$ accounts for the attached shadow, and $\zeta_c(\mathbf{p}_i) \in \{0, 1\}$ assigns 0 or 1 value to $\mathbf{p}_i$ depending on whether it lies in the cast shadow region or not. The reflectance of the material is modeled by $\rho()$ which stands for the bidirectional reflectance distribution function (BRDF). Modeling BRDF of a general object surface is a challenging problem while solving PS, and therefore, isotropic material assumption is commonly used.

Isotropy assumes that the reflectance of the material is identical if it is rotated around the surface normal vector (see Fig.3). Although having this assumption helps in the recovery of surface normals of some objects, materials such as wood or brushed metal do not exhibit such symmetric behavior in their BRDF. For these materials, imaging changes substantially when the material is moved or rotated. Therefore, simply applying the provided image formation model for surface normal estimation is not always an option. To overcome this limitation, we resort to volume rendering. Volume rendering approach models the radiance not only at a surface point but along the ray by taking transmittance of the material into account. In contrast to analytical BRDF representations, material properties are implicitly learned by the network. Hence, volume rendering can model complex light phenomena and generalize well to anisotropic and the glossy materials.

3. Network Design

Although we have introduced the deep-PS and deep-MVS networks in the main paper, we provide a detailed description of network designs for completeness.

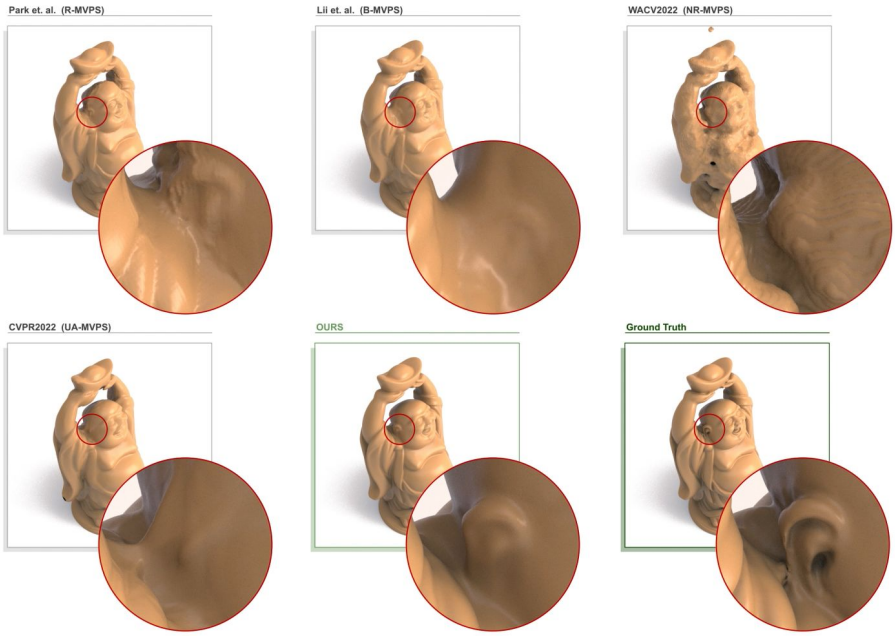

Figure 1. Visual comparison of MVPS reconstructions on Buddha category, demonstrating the benefit of our approach in recovering high-frequency details.

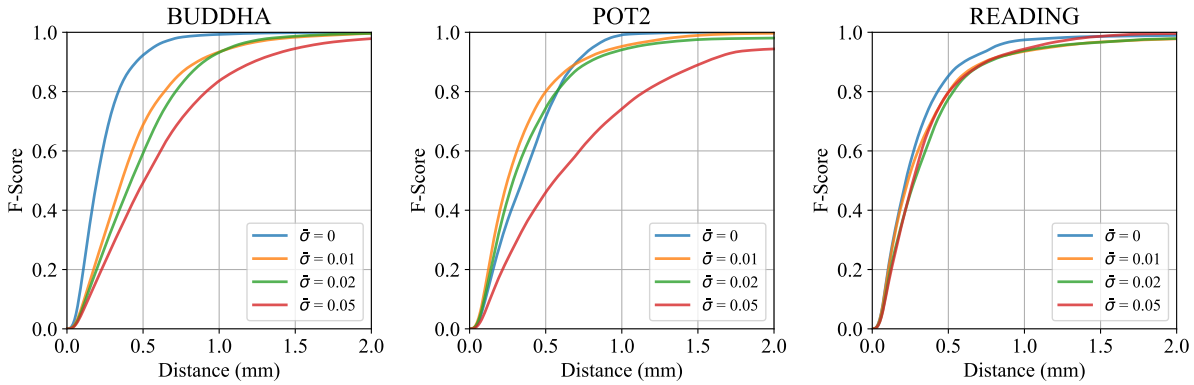

Figure 2. 3D reconstruction quality of Buddha, Pot2 and Reading objects with varying degree of image noise. We use F-score as a function of distance error threshold (ϵ) to report the 3D reconstruction accuracy. It can be observed that as the F-score curves gets substandard with increase in $\bar{\sigma}$ values.

3.1. Deep-PS Network

In this section, we provide details on the observation map strategy, network architecture, and the dataset used to train our Deep-PS network.

Observation Map. As mentioned in the main draft, we use the observation map representation introduced in [8] to predict per-pixel surface normals form PS images $\mathcal{X}_{ps}^{v}$. We define the observation map $\Omega_i^v \in \mathbb{R}^{\omega \times \omega}$ at each pixel i and view v as a 2D matrix containing the intensity values due to different light source illuminations (see Fig.4). Precisely, we use the following formulation to construct an observation map:

$$\Omega_i^v\Big(\xi\big(\omega \cdot \frac{(\mathbf{l}_j(x)+1)}{2}\big), \xi\big(\omega \cdot \frac{(\mathbf{l}_j(y)+1)}{2}\big)\Big) = \eta_i \frac{X_j^v(\mathbf{p}_i)}{e_j} \tag{2}$$

Here, ω stands for the size of the observation map, $\mathbf{l}_j = [\mathbf{l}_j(x), \mathbf{l}_j(y), \mathbf{l}_j(z)]^T \in \mathbb{R}^{3\times 1}$ is the light source direction, $\eta_i = \max(e_1/X_1^v(\mathbf{p}_i), \ldots, e_L/X_L^v(\mathbf{p}_i))$ is a scaling factor for normalization, L is the total number of light sources and $\xi : \mathbb{R} \mapsto \mathbb{Z}_0^+$ is a rounding operation to have integer values. Note that there is a bijective mapping between a light source direction vector and its projection onto $x-y$ coordinate system. This allows us to represent PS image information

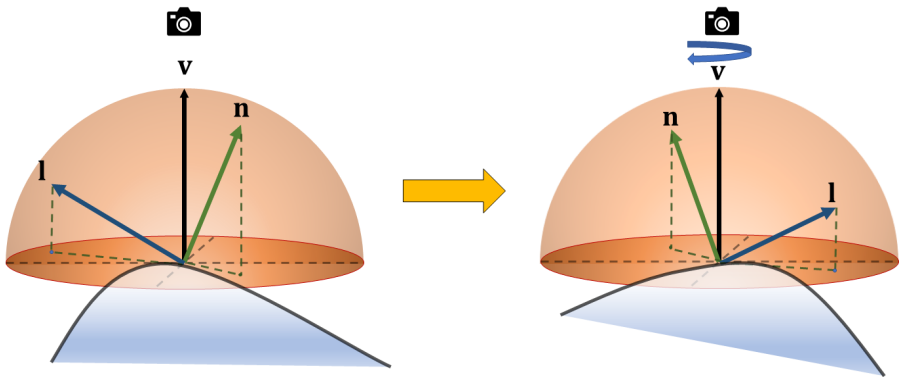

Figure 3. Under the isotropy assumption, the measured pixel intensity is invariant to the joint rotation of surface normal **n** and light source direction **l** around the viewing direction **v**. Such an assumption though suited for a reasonable category of objects but is not applicable for glossy and anisotropic material objects.

in a simple and effective way via an observation map.

Network Architecture. The network first takes a per-pixel observation map Ω_i^v as input and applies a 3×3 convolution layer for feature extraction. Then, a dense block with two convolution layers and ReLU activation is applied. It is later followed by a transition block. The transition block consists of a 1×1 convolution layer with ReLU activation and an average pooling layer. After the transition, a second dense block is used. The features extracted from the second dense block are processed by one convolution and two fully connected layers. The output is normalized to unit length to obtain a surface normal vector. Note that we additionally use dropout with probability $p_{\text{mc}} = 0.2$ after convolution and fully-connected layers to have prediction uncertainty at the test time [5, 6].

Training Dataset. We trained our deep-PS network using the CyclesPS dataset [8]. CyclesPS is a synthetic dataset of 15 object shapes. Each object shape is rendered with specular, diffuse, and metallic material reflectances which exhibit isotropic BRDF properties. The objects are illuminated by 1300 distinct light sources for rendering. To train our network using this dataset, we first constructed observation maps Ω_i^v with size $\omega = 32$ for all the object pixels. Then, we used 90% of the per-pixel observation maps in the training set and used the remaining 10% in the validation set. We trained the network by 10 epochs using Adam optimizer [10] and learning rate of 0.1 to minimize the training loss shown in Eq:(3) of the main draft.

3.2. Deep-MVS Network

In this section, we introduce the PatchMatchNet [13] network design that is used to predict per-pixel depth values from MVS images $\mathcal{Y}_{mv}$. The network mainly consists of multi-scale feature extraction, learning-based Patchmatch, and depth refinement modules. Before describing each module separately, let's denote a reference frame by $Y^r \in \mathcal{Y}_{mv}$, source frames by $Y^s \in \mathcal{Y}_{mv}$ and $\{\mathbf{R}_{r,s}, \mathbf{t}_{r,s}\}$ as the relative rotation and translation between frames r and s. Note that for each reference frame, there exist N_s source frames. Given these inputs, the PatchMatchNet predicts the dense depth map corresponding to the reference view r iteratively using a coarse-to-fine strategy. Precisely, the learning-based Patchmatch is applied on three coarse stages, i.e. $k = 3, 2, 1$ where $k = 3$ is the coarsest stage. At the finest level ($k = 0$) depth refinement module is applied instead of learning-based Patchmatch. For simplicity of our method description, we do not denote the stage number in our notation.

Multi-scale Feature Extraction. Given the reference and source images of spatial resolution $w \times h$, feature maps φ^r and φ^s are extracted at different levels inspired by the Feature Pyramid Network [12]. Precisely, the extracted feature maps have the spatial dimension of $(w/2^k) \times (h/2^k)$ at stage k.

Learning-based Patchmatch. Inspired by the traditional Patchmatch [2], the learning-based Patchmatch module iteratively performs initialization, propagation, and evaluation steps to generate and update depth hypotheses, which are then used for depth regression.

***(i)**Initialization:* At the first iteration of the algorithm, 48 depth hypotheses are randomly generated at each pixel. The hypotheses are sampled such that their distribution is uniform on the inverse depth range. As the initialization is not required for the subsequent iterations, local perturbations are applied instead to expand the existing hypotheses. For that, additional 16 hypotheses are generated at stage $k = 3$ and 8 hypotheses are generated at stages $k = 2$ and $k = 1$.

***(ii)**Propagation:* The generated depth hypotheses are augmented such that the hypotheses belonging to the same physical surface are encouraged to have similar depth values. To that end, a CNN based on Deformable Convolution Network [4] implementation is used to learn a 2D offset for

Observation Map on Glossy Surface Observation Map on Anisotropic Surface

Figure 4. Illustration of observation maps corresponding to the pixel marked with yellow color. Each element of the observation map corresponds to the image intensity measured under different lighting. For a glossy surface, an observation map clearly reveals the specular region, providing strong cues for reliable prediction of surface normals. On the other hand, anisotropic surfaces exhibit more irregular distribution w.r.t. light source directions, which makes it difficult to predict surface normals reliably.

each pixel i using the feature map φ^r. Accordingly, new depth hypotheses $\mathbf{d}_p$ are obtained as follows:

$$\mathbf{d}_p(\mathbf{o}_i) = \{\mathbf{d}(\mathbf{o}_i + \kappa_j + \tilde{\kappa}_j(\mathbf{o}_i))\}_{j=1}^{\mathbf{N}_p} \tag{3}$$

where $\mathbf{o}_i$ is the coordinates of pixel i, $\mathbf{d}$ is depth from previous iteration, κ_j is a fixed offset and $\tilde{\kappa}_j(\mathbf{o}_i)$ is the learned offset for hypothesis j. In this way, $\mathbf{N}_p = 16$ and $\mathbf{N}_p = 8$ hypotheses are generated at stages $k = 3$ and $k = 2$ respectively.

(iii)Evaluation: In this step, the existing depth hypotheses are evaluated to obtain a matching cost which is used for depth regression. To that end, source features $\varphi^s(\mathbf{o}_i)$ are first warped to the reference frame r via differentiable warping. The warped source feature map $\varphi^s(\mathbf{o}_i^{s,j})$ is then used to compute the similarity. For that, the feature channels are divided into $\mathcal{G}$ groups and group-wise similarity is calculated as follows:

$$\Gamma_s^g(\mathbf{o}_i, j) = \frac{\mathcal{G}}{u} < \varphi_g^r(\mathbf{o}_i), \varphi_g^s(\mathbf{o}_i^{s,j}) > \tag{4}$$

Here, u is the number of feature channels and $< \cdot, \cdot >$ is the inner product operation. The initial similarity values are fed to a 3D convolution layer with $1 \times 1 \times 1$ kernel size to have a visibility estimation $W_s \in \mathbb{R}^{w \times h \times \mathcal{H}}$ where $\mathcal{H}$ denotes the number of depth hypotheses. Then, per-pixel visibility weight is computed as $\mathbf{w}_s(\mathbf{o}_i) = \max(\{W_s(\mathbf{o}_i, j)\}_{j=1}^{\mathcal{H}})$. Next, the group similarities $\Gamma_s^g(\mathbf{o}_i, j)$ are weighted over N_s source images using $\mathbf{w}_s(\mathbf{o}_i)$. By applying a 3D convolution with kernel size $1 \times 1 \times 1$, these weighted group similarities are converted into a matching cost per hypothesis $\mathcal{J} \in \mathbb{R}^{w \times h \times \mathcal{H}}$. This cost is aggregated over a spatial window adaptively based on the strategy used in Patchmatch stereo [3] and AANet [14]. The resulting matching cost $\mathbf{J}$ is then used to regress depth as shown in Eq:(2) of the main draft.

Depth Refinement. The finest stage of the learning-based Patchmatch module (k=1) provides depth $\mathbf{D}$ in the spatial resolution of $(w/2) \times (h/2)$. Instead of applying another stage of Patchmatch, the output depth is obtained by a refinement module which uses the RGB image to up-sample the depth to the finest resolution of $w \times h$. For that, a network based on MSG-Net [7] is used to estimate a depth residual. The residual is added to the up-sampled depth $\mathbf{D}$ to have the output refined depth map $\mathbf{D}_{ref}$.

Training. The deep-MVS network is trained on DTU MVS dataset [1] which consists of 80 scenes. For each scene, there exists 49 to 64 images from different viewpoints. The ground-truth depth maps and camera calibrations are provided for each view. The training and test splits are used as introduced in [9]. To train the network, the l_1 loss between the predicted and the ground-truth depth are used at each stage using the following loss function:

$$\mathcal{L}_{pmnet} = \mathcal{L}_{ref} + \sum_{k=1}^{3} \sum_{t=1}^{N_{\text{iter}}^k} \mathcal{L}_t^k \tag{5}$$

Here, $\mathcal{L}_{ref}$ is the refined depth loss and $\mathcal{L}_t^k$ stands for the loss at iteration t of stage k. For training on DTU dataset, the network is trained for 8 epochs using Adam optimizer [10] and a learning rate of 10^{-3}. The number of source images is set to $N_s = 4$ and number of iterations at each stage is set to $N_{\text{iter}}^3 = 2$, $N_{\text{iter}}^2 = 2$, $N_{\text{iter}}^1 = 1$. For more details on the network architecture and training, refer to [13].

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120(2):153–168, 2016. 4

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009. 3

[3] Michael Bleyer, Christoph Rhemann, and Carsten Rother. Patchmatch stereo-stereo matching with slanted support windows. In *Bmvc*, volume 11, pages 1–11, 2011. 4

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 3

[5] Yarin Gal and Zoubin Ghahramani. Bayesian convolutional neural networks with bernoulli approximate variational inference. *arXiv preprint arXiv:1506.02158*, 2015. 3

[6] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR, 2016. 3

[7] Tak-Wai Hui, Chen Change Loy, and Xiaoou Tang. Depth map super-resolution by deep multi-scale guidance. In *European conference on computer vision*, pages 353–369. Springer, 2016. 4

[8] Satoshi Ikehata. Cnn-ps: Cnn-based photometric stereo for general non-convex surfaces. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 3–18, 2018. 2, 3

[9] Mengqi Ji, Juergen Gall, Haitian Zheng, Yebin Liu, and Lu Fang. Surfacenet: An end-to-end 3d neural network for multiview stereopsis. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2307–2315, 2017. 4

[10] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3, 4

[11] Min Li, Zhenglong Zhou, Zhe Wu, Boxin Shi, Changyu Diao, and Ping Tan. Multi-view photometric stereo: A robust solution and benchmark dataset for spatially varying isotropic materials. *IEEE Transactions on Image Processing*, 29:4159–4173, 2020. 1

[12] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 3

[13] Fangjinhua Wang, Silvano Galliani, Christoph Vogel, Pablo Speciale, and Marc Pollefeys. Patchmatchnet: Learned multi-view patchmatch stereo. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14194–14203, 2021. 3, 4

[14] Haofei Xu and Juyong Zhang. Aanet: Adaptive aggregation network for efficient stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1959–1968, 2020. 4