PX-NET: Simple and Efficient Pixel-Wise Training of Photometric Stereo Networks - Supplementary Material

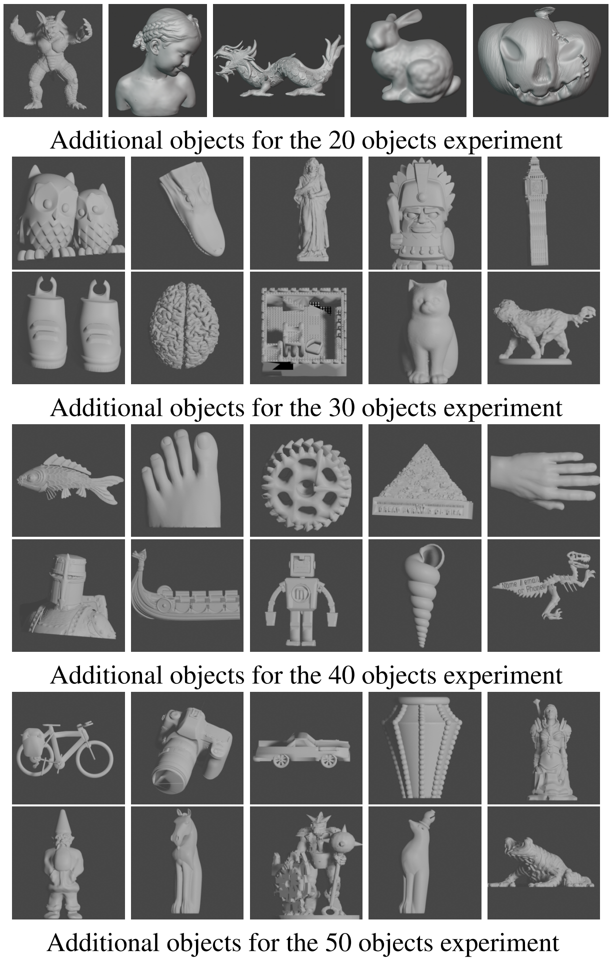
Abstract

This document provides the supplementary material for the main publication. Section 1 provides additional information about the 50 objects dataset used to train CNN-PS [1]. Section 2 provides an in depth explanation of the pixelwise data generation with all the relevant hyperparameters. Section 3 contains additional visualisations of the qualitative results on the DiLiGenT dataset.

1. CNN-PS Training on an Extended Dataset

This section provides additional information about the experiment during which CNN-PS [1] was trained on an extended dataset of globally rendered objects. This experiment is described in Section 5 and Figure 7 of the main publication. The dataset contains 50 objects in total, 15 of which are the original objects of the training portion of the Cycles-PS [1] dataset. These 15 objects are then supplemented with another 35 objects from Thingi10K [3] dataset. CNN-PS is trained under four different setups, using 20, 30, 40 and 50 objects respectively. Figure 1 shows the images of the objects used for corresponding experiments. For each object, original rendering protocol of Cycles-PS [1] is employed to render 3000 images using Blender. 3 material categories (diffuse, specular dieletric and metallic) and 1000 random directional lights (uniform in the upper hemisphere up to 70^o elevation angle) per material category are sampled. The material hyper-parameter ranges are chosen to be slightly more general than the ones used in [1]. They are provided in Figure 2. All images are rendered at 256x256 resolution with each 8x8 pixel patch having a different material (i.e. 32x32 material maps).

Training is performed using the original CNN-PS network architecture and training script of [1] (using the same masking of lights per map and rotation augmentation procedure) with one change - the Euclidean distance loss function is replaced with the angular error loss (see main submission, Section 4) to be fully comparable with our PX-CNN-PS. We also performed an additional set of experiments (with 20, 30, 40, 50 objects), that also included generating additive and multiplicative noise as well as additive ambient light on top of obtained global renderings. These effects were implemented at train time as data augmentation (so as to get a different random sampling at each epoch) with values explained in Table 2.


Additional objects for the 20 objects experiment

Additional objects for the 30 objects experiment

Additional objects for the 40 objects experiment

Additional objects for the 50 objects experiment

Figure 1. Objects used to extend the training portion of the original Cycles-PS [1] dataset.

2. Pixelwise Data Generation

This section provides a more in depth explanation of the pixelwise data generating procedure. The objective is to approximate certain global illumination effects as well as other real world imperfections so as the generated observation maps to be realistic enough to be applicable in real world test data. It is important to note that we do not attempt to exactly replicate the distribution of many real world ef-

Category	Roughness	Specular	Metallic	Subsurface	Other
Diffuse	$0.75-1$	$0-0.25$	0	0	$0-1$
Specular	$0-0.25$	$0.75-1$	$0-1$	0	$0-1$
Metallic	$0-1$	0	$0.75-1$	0	$0-1$

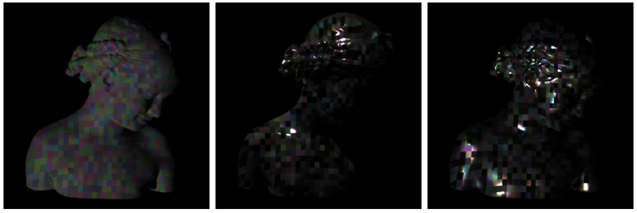


Figure 2. Top: Material categories considered for the extension of the CNN-PS [1] training set. Parameters are uniformly sampled in the respective ranges. *Other* refers to all over parameters of the Disney BRDF namely *BaseColor, SpecularTint, Sheen, SheeTint, Clearcoat, ClearcoatRoughness, IOR*. Bottom: sample images from these categories i.e. Diffuse (*left*), specular dielectric (*middle*) and metallic (*right*).

fects, but instead avoid significantly underestimating them. Thus most of the approximations are aimed to be the reasonable upper bound of an effect and in practice they are likely to be much less (e.g. the sampled ambient effect is up to 0.01, but for most real pixels it is likely to be a lot less). This is a valid procedure as long as the network has enough learning capacity.

As a general principle, we aim to sample most parameters uniformly in the appropriate range, in order to avoid data bias. To simplify the notation, we assume that the pixel value i is a real number $i \in [0, 1]$ with 0 being completely black and 1 being the saturation level (although the division with the light source brightness can lead to observation map values higher than 1). We also denote a uniform real distribution in the interval $[a, b]$ as $\mathcal{U}_{\mathcal{R}}(a, b)$, a uniform integer one in the interval $[k, l]$ as $\mathcal{U}_{\mathcal{I}}(k, l)$ and a normal distribution with mean μ and standard deviation σ as $\mathcal{N}(\mu, \sigma)$.

Normals: We sample normals uniformly in the upper hemisphere so as to maximise the generality of the training data. Note that for some very oblique normals, after all the effects are applied, all of the map pixels may be ending up very small. To avoid numerical instabilities, any map where the maximum RGB pixel value is less than 1e-3 (i.e. 0.1% of the saturation level) is discarded and not included in the training data.

Lights: We trained 2 different networks, aimed to tackle the dense and sparse light settings. For the dense lights settings, we matched the distribution of [1] which was 50-1000 random lights (up to 70^o elevation angle). The sparse light setup was made to match that of [2] namely exactly 10 random lights with elevation angle of up to 45^o. For both dense and sparse light settings, light source brightness ϕ is sampled uniformly and independently (for all lights and channels) in the DiLiGenT range, i.e. $\phi = \mathcal{U}_{\mathcal{R}}(0.28, 3.2)$.

Materials: The material determines the surface albedo (which is essentially the intrinsic color) as well as other BRDF parameters. We sample albedo ρ color components uniformly so $\rho_{red} = \mathcal{U}_{\mathcal{R}}(0, 1)$ and similarly for green and blue channels. In order for our data to be applicable in a range of real world situations, 75% of the data are generated using a random material from the Disney BRDF. All 8 parameters (excluding *subsurface*,*IOR*), namely *metallic, specular, roughness, specularTint, sheen, sheenTint, clearcoat* and *clearcoatRoughness* are sampled uniformly ($\mathcal{U}_{\mathcal{R}}(0, 1)$) and independently. The Dinsey non-linear equation uses both the albedo and BRDF parameters as inputs so the direct reflectance r_d component computation is straightforward. The remaining 25% of the training data are generated using data from the MERL material database. Unfortunately, this database only contains 100 materials with specific albedo (e.g. "blue-acrylic", "green-latex") which is a very limited set for tackling general PS problems. To overcome this limitation, we generate virtual materials superset of the MERL database with the following procedure: firstly a random MERL material $M \sim \mathcal{U}_{\mathcal{I}}(1, 100)$ is selected. Then a random weight $w \sim \mathcal{U}_{\mathcal{R}}(0, 1)$ is sampled. Finally, the material's BRDF B_M is mixed with a Lambertian component $\mathbf{N} \cdot \mathbf{L}$ to get overall reflectance:

$$r_b(\mathbf{N}, \mathbf{L}, \mathbf{V}_0, \rho, M) = \rho\Big(wB_M(\mathbf{N}, \mathbf{L}, \mathbf{V}_0) + (1-w)\mathbf{N}\cdot\mathbf{L}\Big) \tag{1}$$

where in Equation 1, $\mathbf{L}$, $\mathbf{V}_0$ are the light and view direction respectively (as explained in the main text).

We note that the above material sampling procedure is aimed to target general test data. Of course, in the case of a very specific application (e.g industrial inspection), a more constrained set of materials would be more appropriate.

Cast shadows: Cast shadows are observed in real data when a part of the surface is blocking the light, thus turning the direct reflectance to zero. Our aim is to compute a shadow map approximation, (i.e. compute all of the directions were direct refletance is blocked) regardless of which light sources are actually available. We note that that realistic shadows are likely to be piece-wise continuous in scenes with a few discrete objects. In addition, the likelihood of a particular directing to be in shadow increases as its elevation angle increases, and in principle, the central direction $\mathbf{V}_0 = [0, 0, 1]$ is never blocked by shadow. Finally, if a direction is blocked by shadow, it is very likely that all other directions with the same azimuth and more oblique elevation angles to also be blocked by shadow.

Taking all of the above considerations into account, the assumed shadow model consists of a circular "wall" (see Figure3) surrounding the observation map. More specifically, we sample 20 height values (corresponding to az-

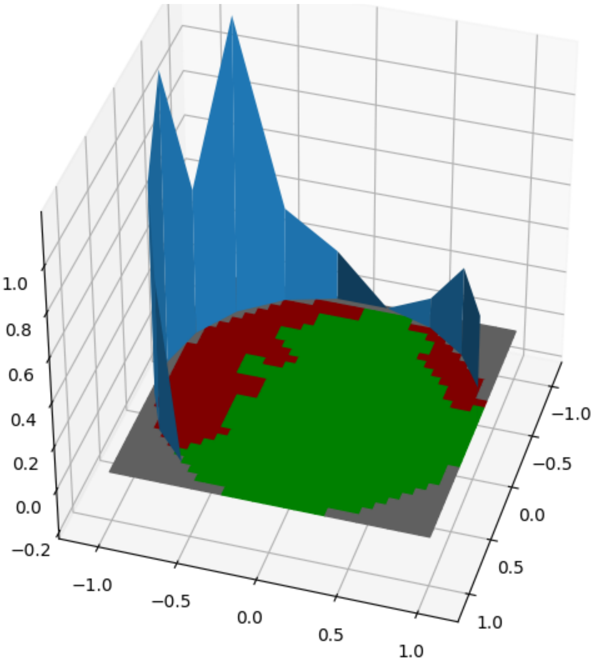


Figure 3. Demonstration of the assumed shadow model consisting of a circular "wall" surrounding the observation map. All shaded direction are marked red whereas non shaded ones are marked green.

Effects	Ball	Bear	Buddha	Cat	Cow	Goblet	Harvest	Pot1	Pot2	Reading	Mean
Up to Ambient	2.08	4.24	8.38	4.59	5.77	7.79	14.88	5.57	6.01	12.46	7.18
+Same mat. reflection	2.40	3.70	7.97	4.46	5.76	7.58	14.51	5.37	5.62	10.91	6.83
+Discontinuity	2.17	3.72	7.78	4.25	5.49	7.56	14.95	5.36	5.69	10.94	6.79
+Diff. mat. reflection	1.99	3.98	8.59	4.65	6.59	8.29	15.01	5.59	6.70	11.24	7.26
+Discontinuity	2.56	3.67	8.13	4.51	6.48	7.69	14.72	5.60	5.95	11.74	7.10

Table 1. Justification for computing self reflection with reflecting points having the same material (but different albedo than the main point). The 3 lines on top are copied from Table 1 of main text whereas the 2 lines on bottom contain the self reflection effect computing with different material for each self reflection point. These different materials were sampled but perturbing the main Diligent material parameters by a random value $\in [-0.1, 0.1]$.

imuth angles which are integer multiples of 36^o) from a Gaussian with mean 0, standard deviation 2 (taking the absolute value). In addition, to allow for the possibility of no shadow in a range of directions, each of these values is set to 0 with a chance of 25%. We also sample 25% of the data with completely empty shadow maps which are aimed to approximate points is convex part of a surface. Then, the height of the "wall" is linearly interpolated in order to get a height value for all azimuth angles. Finally, each pixel of the shadow map is set as 0 if the corresponding direction (extending out of the center) is intersecting the wall.

Self reflections: As explained in the main text, we approximate the self reflection effect by sampling a few directions $\mathbf{L}_R$ (inside the shadow map) and then computing a single light bounce from $\mathbf{L}$ to $\mathbf{L}_R$ to $\mathbf{V}_0$. Table 1 justifies using the same material for these reflecting points (which aims to model the case of an object with piece-wise constant material distribution). We note that in real objects, highly concave regions are most likely to have the maximum amount of self reflection which also corresponds to the maximum amount of shadows. To enforce this positive correlation between the self reflection magnitude and the amount of shadows, we first sample 5 directions $\mathbf{L}_R$ uniformly in the upper hemisphere, and then only keep those that are part of the shadow map, i.e. $S(\mathbf{L}_R) = 0$. Thus, the more shaded pixels the shadow map has, the higher the chance for a higher number of self reflection points. Note that this sampling procedure only makes a subset of data points that have a non empty shadow map to have any self reflection directions; this is consistent with real data as significant self reflection is only present in a small portion of the data (corresponding to highly concave regions of surfaces or nearby very reflective objects).

Surface discontinuity: For this step, we allow 15% of the training data to be a combination of 2 or 3 'subpixels', each having a different normal $\mathbf{N}_k$ and albedo ρ_k. As explained in the main text, all direct reflection r_d and self reflection r_r components computed and averaged. We note that the overall normal (used for training the network) is simply the average of the subpixel's normals.

Ambient light: Ambient light aims to address any additional reflection, such as from objects in the background or the even the atmosphere. Literature usually assumes a constant reflection for all light sources as the multiple bounces tend to average out the effect. None the less, as this reflectance component is caused by the light source, the ambient effect has to be proportional to its brightness ϕ . In addition, it is reasonable to assume high correlation with the surface albedo (ambient reflection at a dark point should be dark, ambient reflection at a red point should be red etc.) as well as diminished reflection at oblique angles so we assume correlation with $\mathbf{N} \cdot \mathbf{V}_0$). Therefore, we set $a = \rho \mathbf{N} \cdot \mathbf{V}_0 \mathcal{U}_{\mathcal{R}}(0, 0.01)$ and apply this effect to 75 % of the generated data points (i.e. 25% of out training data are made ambient free). We note that a is the same for all light sources and its contribution to the total reflectance is multiplied by the light brightness ϕ (see Equation 5 in main text).

Noises: As explained in the main text, we apply four different types of noise namely additive and multiplicative, uniform and Gaussian. The most important component is the uniform multiplicative as it is aimed to address several unmodeled effects (i.e. near light attenuation) which effect pixel intensities multiplicatively, and thus was set to 5%, i.e. $n_{MU} = \mathcal{U}_{\mathcal{R}}(0.95, 1.05)$. The rest of the hyper-parameters for Equation 5 were: multiplicative Gaussian noise $n_{MG} =$

Effect	Probability	Magnitude
Number of Lights	N/A	$\mathcal{U}_{\mathcal{I}}(50, 1000)$ dense /10 sparse
Light Brightness	N/A	$\mathcal{U}_{\mathcal{R}}(0.28, 3.2)$
MERL Materials	0.25	N/A
Shadow	0.75	$\lvert\mathcal{N}(0, 2)\rvert$
Self reflections	*	≤ 5
Surface Discontinuity	0.15	$\mathcal{U}_{\mathcal{I}}(2, 3)$
Ambient a	0.75	$\rho\mathbf{N} \cdot \mathbf{V}_0\mathcal{U}_{\mathcal{R}}(0, 0.01)$
Noise Multiplicative	1	$\mathcal{U}_{\mathcal{R}}(0.95, 1.05)\mathcal{N}(1, 10^{-4})$
Noise Additive	1	$\mathcal{U}_{\mathcal{R}}(-10^{-4}, 10^{-4}) + \mathcal{N}(0, 10^{-4})$
Quantisation	1	16 bits

Table 2. Summary of all the data generation hyperparameters. The probability column determines the proportion of the data with that particular effect doing applied (some effects like discretisation and noise are always used). the Note that self reflections are only sampled when the shadow map is non empty, and the corresponding magnitude is a positively correlated with the amount of shaded pixels.

$\mathcal{N}(1, 10^{-3})$, Gaussian additive noise $n_{AG} = \mathcal{N}(0, 10^{-4})$ and uniform additive noise $n_{AU} = \mathcal{U}_{\mathcal{R}}(-10^{-4}, 10^{-4})$.

All of the relevant hyperparameters are summarised in Table 2 and their application order is shown in Equation 5 in the main document .

3. Diligent Results

This section contains full visual comparison with CNN-PS [1] for all Diligent objects at Figures 4 and 5.

References

[1] Satoshi Ikehata. Cnn-ps: Cnn-based photometric stereo for general non-convex surfaces. In *ECCV*, 2018. 1, 2, 4, 5, 6

[2] Junxuan Li, Antonio Robles-Kelly, Shaodi You, and Yasuyuki Matsushita. Learning to minify photometric stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7568–7576, 2019. 2

[3] Q. Zhou and A. Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv*, 1605.04797, 2016. 1

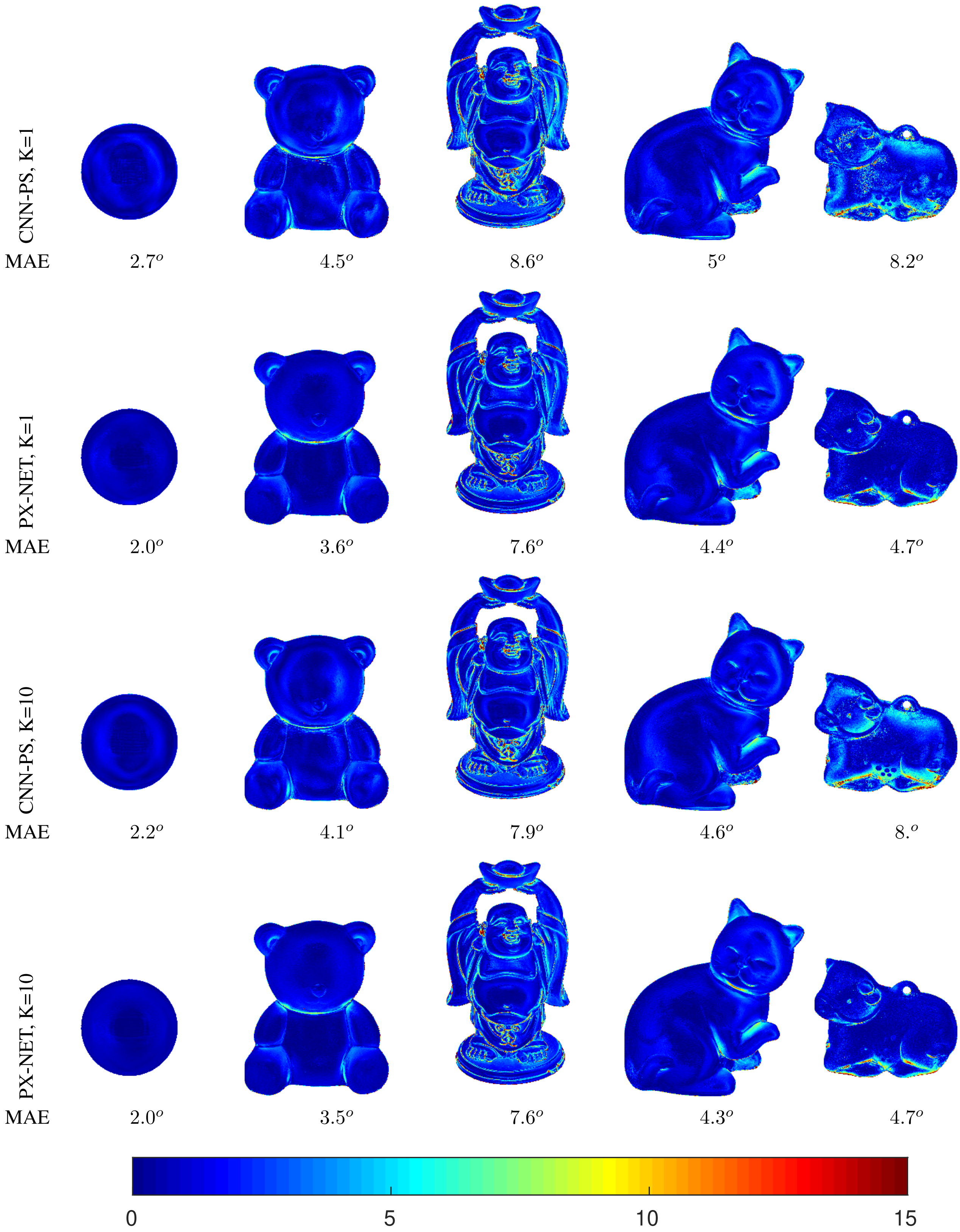


Figure 4. Visual comparison [1/2] of CNN-PS [1] with the proposed PX-NET (Table 2 of the main paper) for both K=1 and K=10 on the Diligent dataset.

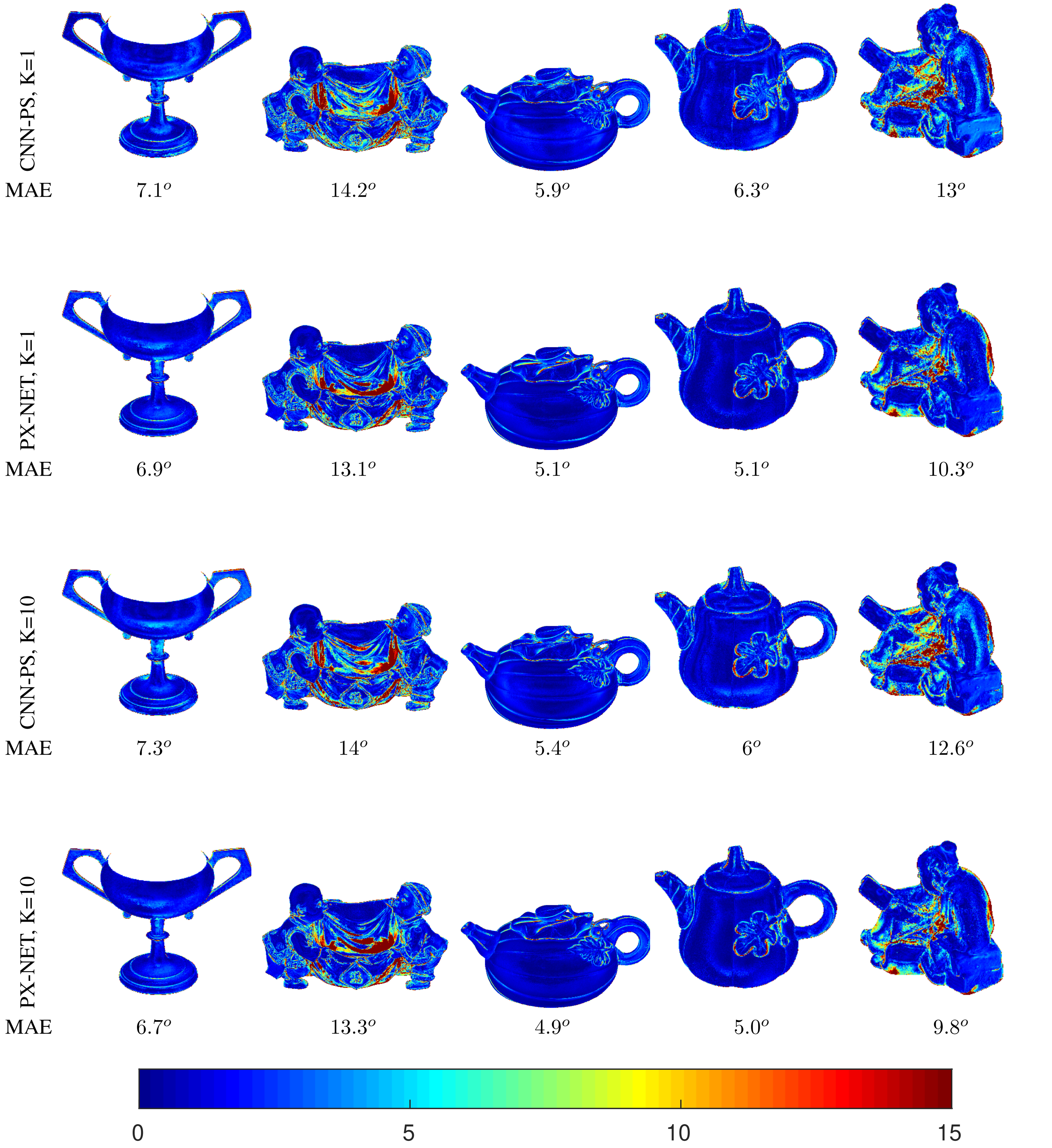


Figure 5. Visual comparison [2/2] of CNN-PS [1] with the proposed PX-NET (Table 2 of the main paper) for both K=1 and K=10 on the Diligent dataset.