ISSN 2464-4617 (print)
ISSN 2464-4625 (online)
Computer Science Research Notes - CSRN 3301
http://www.wscg.eu
WSCG 2023 Proceedings

MS-PS: A Multi-Scale Network for Photometric Stereo With a New Comprehensive Training Dataset

Clément Hardy
Normandie Univ,
UNICAEN, GREYC
Caen, France
clement.hardy@unicaen.fr

Yvain Quéau
Normandie Univ, CNRS,
GREYC
Caen, France
yvain.queau@ensicaen.fr

David Tschumperlé
Normandie Univ, CNRS,
GREYC
Caen, France
david.tschumperle@unicaen.fr

ABSTRACT

The photometric stereo (PS) problem consists in reconstructing the 3D-surface of an object, thanks to a set of photographs taken under different lighting directions. In this paper, we propose a multi-scale architecture for PS which, combined with a new dataset, yields state-of-the-art results. Our proposed architecture is flexible: it permits to consider a variable number of images as well as variable image size without loss of performance. In addition, we define a set of constraints to allow the generation of a relevant synthetic dataset to train convolutional neural networks for the PS problem. Our proposed dataset is much larger than pre-existing ones, and contains many objects with challenging materials having anisotropic reflectance (e.g. metals, glass). We show on publicly available benchmarks that the combination of both these contributions drastically improves the accuracy of the estimated normal field, in comparison with previous state-of-the-art methods.

Keywords
Photometric stereo, 3D-recontruction, normal map estimation, multi-scale achitecture, new dataset

1 INTRODUCTION

Photometric stereo (PS) is a 3D-reconstruction technique that estimates the 3D normal at each point of the surface of an object, using three or more photographs taken from the same viewpoint but with different lighting directions. Early works in this field (e.g. [38]) considered the ideal case of a perfect Lambertian surface. However, most images of real world objects exhibit a wide variety of complex lighting effects, which are not well predicted by Lambert's law. Especially, objects' reflectance often includes a specular component, giving a more or less *shiny* appearance to the image surface. Translucent surfaces, such as glass and acrylic, do not respect Lambert's law either. These kind of materials remain in most cases, poorly managed by traditional photometric stereo solutions [31]. In order to manage non-Lambertian surfaces, deep learning methods based on convolutional neural networks have recently emerged as the most efficient ones [31, 34]. The quality of results obtained by such approaches relies on two main factors:

1. The architecture of the network, which must ensure a good capacity for generalization on new data, including data with a different size from the training set.
2. The quality of the learning dataset, which must be as representative as possible of the diversity of observable light phenomena, for the network to be able to differentiate materials from each other.

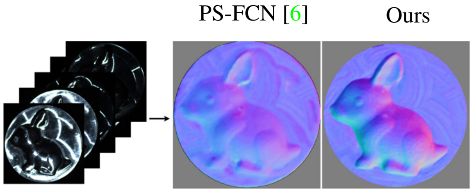


Figure 1: From a set of images taken under different illumination directions (left), photometric stereo estimates a normal map (right). Our proposed method is particularly efficient when used on challenging anisotropic materials, e.g. metal and glass as with this aluminium bunny from [31].

Contributions

Here, we propose a deep learning-based method for the problem of calibrated PS (known lighting direction and intensities), with the following features:

- A multi-scale network architecture for PS, which analyzes the input images simultaneously at different scales;
- A new synthetic training set featuring a wide variety of geometry and non-Lambertian reflectance.

https://www.doi.org/10.24132/CSRN.3301.23

Using these two contributions together, we show that challenging materials with anisotropic reflectance (e.g. metal, glass) can be handled appropriately in the PS problem (Fig. 1). The underlying core idea is that information over the *whole* image is indeed necessary to infer the 3D normal. Otherwise, complex lighting effects like inter-reflections in metallic objects or sub-surface scattering inside glass cannot be analyzed. On the contrary, our proposed multi-scale architecture takes advantage of all available complex geometric/lighting information and long-distance pixel correlations when inferring the 3D normal map.

2 RELATED WORK

Deep learning techniques for photometric stereo are all based on the use of Convolutional Neural Networks (CNN). Typically, a fully CNN architecture requires a fixed number of input images. However, in photometric stereo, the number of images depends on the acquisition procedure. To avoid having to train a different network model for each possible number of input images, two alternatives have been considered in the literature.

Observation map VS pooling

The first alternative consists in using an observation map [11, 13, 22, 25, 41], which projects all observations of each pixel under different illuminations into a fixed-size space - typically a sampled hemisphere. Therefore, an observation map makes a fixed-size summary of the information contained in a variable-size set of images. However, the spatial information (intra image) is lost, and the performance drops when the number of input images is small (typically, <10) [12].

The second alternative rather resorts to specific pooling modules [5, 6, 16, 18, 37], which aggregate the different features of each image extracted by previous convolution layers. This allows to obtain fixed-size image features from a variable number of input images. Different pooling methods can be considered. It is shown in [6] that max pooling performs better than average pooling as soon as the number of images exceeds 16. The latter tends to over-smooth the salient features and to be too sensitive to the regions of images with little interest, although a max pooling can also sometimes ignore a large proportion of the features extracted [17]. Still, in contrast to the observation map approach, pooling methods pay attention to intra image information, despite using less the variations of pixel values across the images.

Architectural variants

To overcome the drawbacks of both these approaches, Yao et al. [40] introduced a graph method called GPS-NET. It first aggregates the inter-image information by using a graph structure, and then uses a CNN to predict a 3D normal map. This graph structure therefore allows to preserve the spatial information. More recently, Ikehata [12] proposed a dual-branch transformer (PS-transformer). One branch takes as input the pixels under different illuminations to get the inter-information, the other branch processes the images to get the spatial one. The features extracted are then aggregated, and a CNN finally gives the 3D normal map. However, as mentioned in [12] transformers are not particularly suitable for dense problems (in our case, a large number of input images).

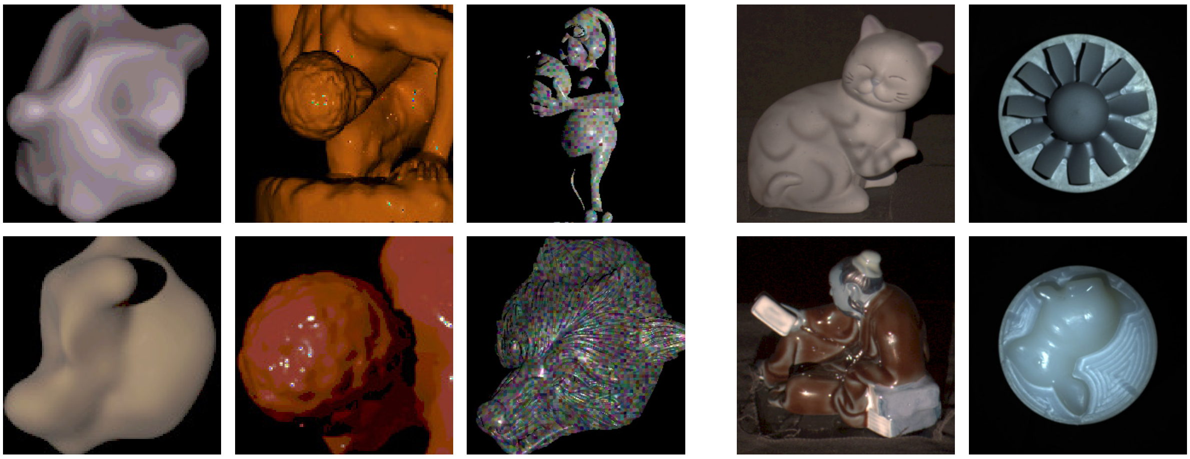
In this paper, we rather consider the pooling-based scheme from [6] as a baseline model, and broaden it to a multi-scale architecture. Multi-scale architecture for photometric stereo has already been used, e.g., by Lichy et al. in the context of directional lighting with few images (no more than 6 images in inputs) [24], or for near (non-parallel) lighting [23]. On the contrary, we design our method to handle the directional lighting case with a large number of input images (e.g. 96 images).

Existing training datasets

Regardless of its architecture, a neural network needs to be trained on a proper dataset to perform well. In practice though, it is very difficult to acquire a large dataset of real images with 3D ground truths of photographed objects. For this reason, deep photometric stereo networks proposed in the literature often rely on training datasets of synthetic 3D objects, notably the *Blobby* and *Structure* datasets introduced in [6], and *CyclePS* in [11].

The *Blobby* dataset is composed of 10 geometric shapes, each one observed from 1296 distinct viewpoints. As the name suggests, the shapes in *Blobby* are rather smooth and regular (Fig. 2a). The *Structure* dataset consists in objects with complex geometry containing fine details (Fig. 2b). It is composed of 8 objects, rendered in 3D from 1387 to 6874 viewpoints. To simulate surfaces with non-Lambertian light reflectance, a material from the *MERL* [28] dataset is randomly drawn and applied in each rendering, providing a total of 25920 samples for *Blobby* and 59292 for *Structure*. In both cases, each sample is rendered under 64 different light directions, randomly selected on the hemisphere (Fig. 3c).

Finally, the *CyclePS* [11] dataset is also composed of complex objects, but contains only 18 objects rendered from 10 views (Fig. 2c). However, the number of materials available is substantial because *Disney's principled BSDF* [3] parametric reflectance model is used. It allows the variation of the base colour, roughness, proportion of specular reflectance, etc., thus the objects can be rendered using a near infinite number of materials. The training dataset presented in the present paper will also feature the possibility to generate as many materials as needed, while also considering much more geometric shapes than in existing sets.



(a) Blobby [15] (b) Structure [15] (c) CyclePS [11] (d) DiLiGenT [34] (e) DiLiGenT10^2 [31]

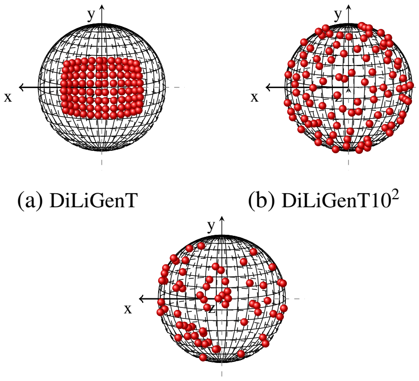
Figure 2: Samples from existing datasets. The first three [15, 11] are synthetic ones, used for training the neural networks. Both the last ones [34, 31] are real-world datasets used for benchmarking. Our proposed multi-scale architecture is evaluated on both benchmarking datasets, and trained on a new synthetic training set, which contains much more objects with *non-Lambertian* reflectance.

Existing benchmarking datasets

To validate the relevance of the training datasets, as well as to verify that the models trained on these synthetic data are able to generalize to real images, two real-world datasets exist: *DiLiGenT* [34] and *DiLiGenT10*^{2} [31].

The *DiLiGenT* dataset comprises 10 different objects, taken from the same viewpoint under 96 different illuminations (Fig. 3a). The reflectance of the objects in this dataset goes from quasi-Lambertian to moderately specular. For each photographed object, the ground truth normal map is provided, as well as the calibrated lighting directions and intensities. Therein, the ground truth geometry was acquired by manually registering laser scans with the images.

The *DiLiGenT10*^{2} dataset contains 10 different objects. Each object was explicitly fabricated with 10 different materials and photographed under 100 calibrated illuminations (Fig. 3b). The ground truth was not obtained by scanning the objects, but from the 3D digital models used to machine the objects. This real dataset is particularly interesting for evaluating performances on highly specular materials and translucent ones. Indeed, it contains metallic materials, such as aluminium or steel, and a translucent one (acrylic). This dataset also contains diffuse and slightly specular materials, hence most of real-world material characteristics are present. The diversity of object shapes is also high as it contains objects with simple geometry like balls but also complex ones like turbines. It offers the opportunity to test the impact of diverse inter-reflection, shadow and shading effects. Today, it is the most complete dataset composed of *real* images available in PS.



(a) DiLiGenT (b) DiLiGenT10^2

(c) Random

Figure 3: Distribution of illumination directions in the real *DiLiGenT* and *DiLiGenT10*^{2} datasets, and an example of a random distribution. The z-axis corresponds to the optical axis of the camera, with the imaged object at coordinates $(0,0,0)$.

Uncalibrated PS

In all the methods discussed above, the light directions and intensities are assumed to be known, i.e. we consider the *calibrated* PS problem. When these acquisition parameters are unknown, the problem is called *uncalibrated*. Uncalibrated PS has been studied e.g. in [5, 14, 20], and partially solved by defining a first neural network that predicts the lighting parameters associated with each acquired image. This estimated data is then fed into a second network that solves the problem of calibrated PS. Managing non-directional lighting, e.g. near point-light sources [26, 32] or natural illumination [9, 14, 29], is another ongoing research problem. In this paper we focus on the case of *calibrated* PS with known *directional* light sources.

3 A NEW MULTI-SCALE ARCHITECTURE FOR PS

The multi-scale architecture we propose builds upon the normal estimation network introduced in [6]. Therein, each image is first normalized by the calibrated lighting intensity, and then concatenated with the calibrated direction. The resulting "image" forms the input to the feature extractor which processes each (image, direction) pair independently. Then, all the independent features are aggregated through a feature aggregation module, and lastly a regression module predicts the normal map.

In order for the normal estimation to perform equivalently well on low-frequency geometry and high-frequency details, we propose to embed this network in a *multi-scale* approach which progressively refines the result as the spatial scale increases. Thus, our model first focuses on the *global* aspect of the object, then progressively insert *details* such as cracks, slight bumps, or holes as illustrated in Fig. 4.

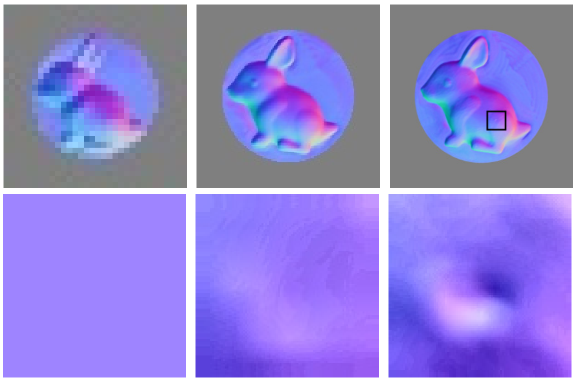


Figure 4: Multi-scale normal estimation at three different scales (bottom row is a contrast-enhanced zoom on the rectangle area). Low-detail geometry is reconstructed from the first levels. High-frequency details get refined as the scale increases.

The proposed multi-scale network combines two independent architectures (Fig. 5). The first stage takes as inputs the calibrated lighting directions and the images (downsampled from the original images to some initial resolution r_0), and outputs a low-resolution normal map with the same resolution r_0. This first stage is essentially similar to the normal estimation network proposed in [6]. In the second stage, the low-resolution 3D normal map is up-sampled to a resolution $r_1 = 2r_0$ (using bilinear interpolation followed by normalization to enforce the unit-length constraint on normal vectors), and concatenated with the images (down-sampled from the original input images to resolution r_1) and lighting directions. The process is then repeated until the resolution of the original images is reached. In these sequential stages, the inputs differ from the first stage, thus a new, independent architecture is obviously necessary. Yet, let us emphasize that since this new architecture is completely convolutional (except the pooling layer) and as only the spatial resolution changes from stage to stage, we can share the weights between each processed scale. Therefore, only two networks actually need being trained, independently from the number of scales. The network formed by these two sub-networks is trained by minimizing the cosine similarity, which measures the angular difference between the estimated 3D normals and the ground truth ones. It is defined as follows:

$$l_{normal} = 1 - \sum_{ij} N_{ij}^{\top} \hat{N}_{ij}, \tag{1}$$

where $\hat{N}_{ij}$ is the estimated normal at pixel (i, j), and N_{ij} is the ground truth one. In terms of computational cost, our multi-scale CNN has 4.4 millions parameters. In comparison with the mono-scale approach, it uses only 5% more memory and takes 14% more time for inference.

As remarked in [24], one of the most interesting features of a multi-scale architecture is its ability to process images with arbitrary size (small or large) without loss of performance. Indeed, even if a single-scale model is fully convolutional and so can process high-resolution images, such a model with a fixed number of convolution layers may not have enough convolutions to synthesize the information over a whole, potentially large image. And, a network trained to handle a specific resolution may not behave well for much larger images. For example, information from the bottom left of the image may not be used to infer the normal at the top right. Yet, such an ability would be particularly useful for handling non-local reflectance effects such as translucency. See for instance the acrylic ball shown in the experiments section in Fig. 10, where light passes through the object. By propagating global information at different scales, such a limitation of local methods is overcome.

More importantly, the proposed multi-scale architecture with shared weights allows one to process images with higher resolution than the ones used during training. For example, in our implementation the first processing resolution is 8×8 pixels. By taking a resolution multiplier of two between two scales, four scales are necessary to reach a resolution of 128×128 pixels (which is the training resolution in our tests), and seven scales for the *DiLiGenT10*^{2} images which have a resolution of 1001×1001 pixels. Yet, the same weights are used in both cases, hence a resolution-specific training is not necessary. In practice, this removes the need for either rescaling the input images to the resolution of the training images, or resorting to a (too local) patch-based approach.

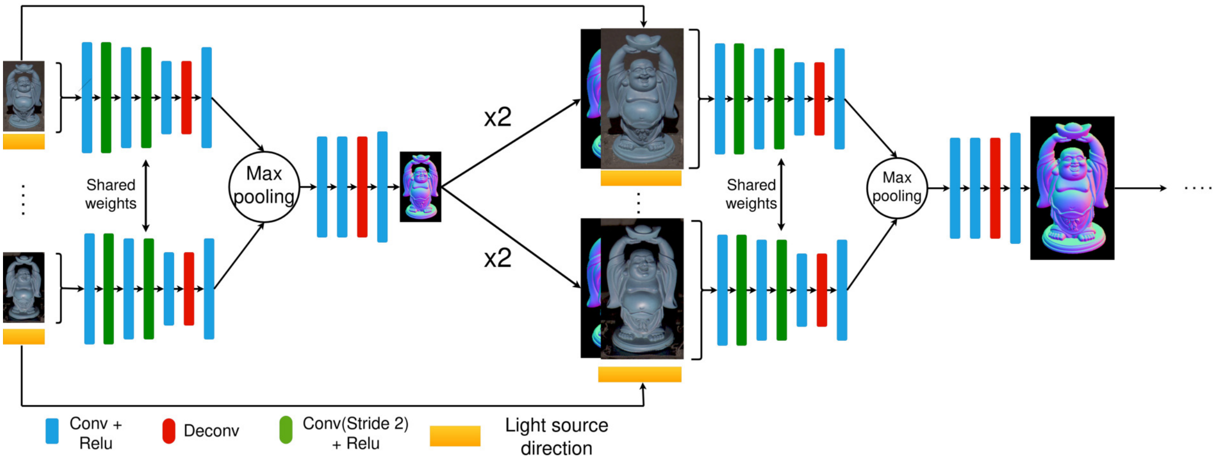


Figure 5: First two stages of the proposed multi-scale architecture. A first architecture, inspired by the PS-FCN method [6], takes as inputs the calibrated lighting directions and downsampled images, and outputs a low-resolution 3D normal map. The latter is then up-sampled and concatenated with lighting directions and higher-resolution images. A second architecture then infers higher-resolution normals, and this part of the process is repeated until the resolution of the original images is reached (network weights being shared by all scales).

4 PROPOSED LEARNING DATASET

As discussed in Section 2, the existing *Blobby* and *Structure* synthetic datasets lack of diversity in terms of geometry and textures. For example, although the *Structure* dataset is composed of complex objects, all these objects are statues. Similarly, the number of different materials in the MERL material base is only 100. This is clearly not enough to model the huge diversity of materials present in the nature. The *CyclePS* dataset partially solves this issue, by allowing to generate infinitely many materials by randomly selecting parameters from a parametric BSDF model. Still, it remains limited in terms of geometry. Overall, a greater diversity of shapes and materials in the images of the training dataset would be beneficial for training networks for photometric stereo. For these reasons, we propose here a new dataset, which includes a large variety of shapes and materials.

In order to create this dataset, we implemented our own image data generation pipeline. We used the *Blender* [8] software with the Cycles rendering engine. As a result, our new dataset is composed of two parts:

- *Our Blobby* contains objects with smooth surfaces;
- *Our Object* contains objects with complex geometry: strong discontinuities, edges, corners, textures details, etc.

Samples from our training dataset are shown in Fig. 6.

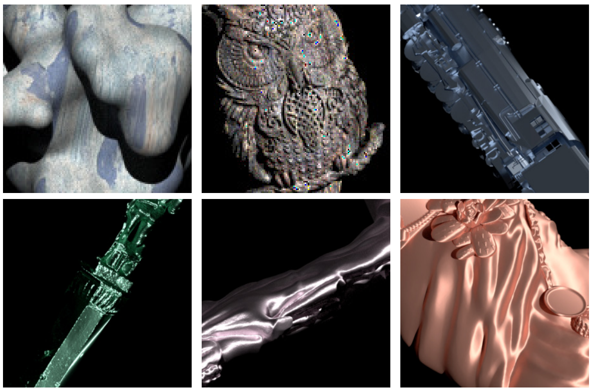


Figure 6: Examples of images from the proposed dataset.

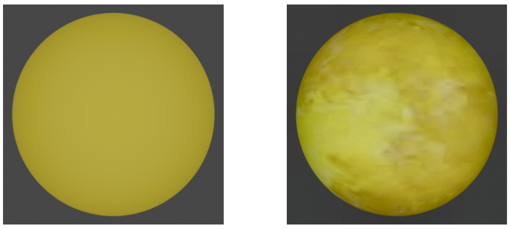
Our Blobby has 3000 distinct objects, generated by the sum of random Gaussian potentials, followed by iso-surface extraction using the *Marching Cubes* algorithm [27]. *Our Object* contains 76 detailed objects which are 3D meshes from the *Sketchfab* [2] website. Moreover, to allow the learning of non-Lambertian surfaces, more than 1100 different real materials, extracted from the *ambientCG* [1] website, are randomly applied to the objects, much more than the 100 materials of *Structure* and *Blobby*. To complete a lack of diversity of the most complicated materials (metals, glasses, etc.) that could persist, we generated additional materials by randomly setting the values of somes parameters (metallic, specular, roughness, anisotropic, etc.) of Disney's principled BSDF [3]. To ensure that all possible materials are represented, during the rendering we choose to apply to the object with a probabilty of 50% a real material (from ambientCG), with 17% a glass material and with 17% a metal one. The remaining 16% materials are constructed by randomly selecting all possible parameters in the principled BSDF (which may result in non-realistic materials).

	# objects	# views	# total number of samples	# lighting	# materials
Blobby	10	1 296	25 920	64	100
Structure	8	1387-6874	59 292	64	100
CyclePS	18	10	180	1 300	90 000
DiLiGenT	10	1	10	96	10
DiLiGenT10^2	10	10	100	100	10
Our Blobby	3000	5	15 000	100	1 100 + infinity
Our Object	76	267	45 000	100	1 100 + infinity

Table 1: Summary of the characteristics of the different learning datasets used in photometric stereo.

If we set a single value for each parameter of the principled BSDF, we would obtain a material which is spatially uniform in terms of reflectance, as in the example of Fig. 7a. Yet, many real-world objects exhibit a spatially-varying reflectance, which is a known limitation of existing PS techniques [6]. To solve this problem in our generation pipeline, we rather incorporated a few spatially-varying material maps, as in the example of Fig. 7b. This technique was used for 50% of the renderings. It allowed us to create both objects with uniform reflectance, and others with spatially-varying one, as illustrated in Fig. 6.

Finally, to generate data having realistic lighting conditions, we rendered all the images with both random illumination direction (Fig.3c) and random intensity. In total, 15 000 *blobby* samples and 45 000 *object* samples were generated this way. Table 1 summarizes the characteristics of the existing datasets, versus the ones we propose. In order to ensure the reproducibility of our results, the code and these learning datasets will be made publicly available online.


Uniform reflectance

Spatially-varying reflectance

Figure 7: Rendering of the same ball with a uniform base color, or with a spatially-varying one.

5 EXPERIMENTS

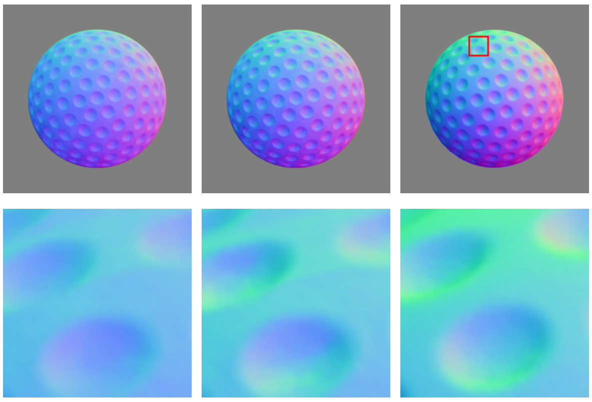
In this section, we demonstrate the effectiveness of our proposed multi-scale architecture on publicly available benchmarks, namely DiLiGenT [34] and DiLiGenT10^2 [31]. To evaluate the impact of our new training dataset, we trained our network both on the pre-existing training datasets *Blobby* and *Structure* (this training is referred to as "DS1" in the following) and on our new training dataset ("DS2" in the following). In the rest of this section, "*Mono* (DS1)" will thus refer to the mono-scale architecture trained on the pre-existing dataset, "*Multi* (DS1 + DS2)" to the multi-scale architecture trained on both the pre-existing and the new datasets, etc. We will first provide a few qualitative results to illustrate the importance of the two building blocks of our contribution, and then provide a thorough quantitative evaluation on the two benchmarks.

5.1 Implementation details

Both the "*Mono*" and the "*Multi* architectures were implemented in Pytorch. The Adam optimizer [21] was used with a learning rate of 10^{-4}. We trained both the multi-scale and the mono scale architecture by taking 32 patches of size 128 by 128 as inputs. The training took a few days on a single Nvidia GeForce GTX 1080 Ti with a batch size of 3 (the maximun we can fit in our GPU). The inference time depends on the number of input images and their resolution. For example, by taking 100 images of 256 by 256 pixels, it takes approximately 1.6 seconds for our multi-scale methods on our GPU. The inference time scales linearly with the number of images, while it seems to be roughly multiplied by a factor of 4 when the resolution is multiplied by 2.

5.2 Qualitative evaluation

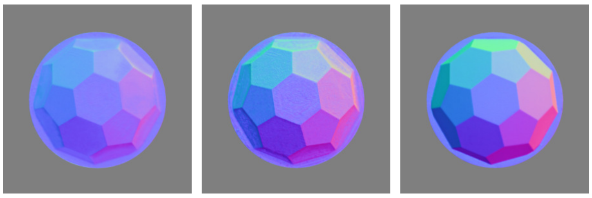
Let us start by showing two illustrative results on the DiLiGenT10^2 [31] benchmark, on challenging metallic objects (the copper golf ball and the copper hexagon). As we shall see, both the new training dataset and the new multi-scale architecture contribute to improving the estimation performances on such objects exhibiting an anisotropic reflectance. Since we do not have access to the ground truth normals, for visual purpose we show as "ground truth" the result we obtained with our *Multi* (DS1+DS2) approach, applied to the same object but fabricated in PVC (a matte material). The example of Fig. 8 shows that, independently from the training set, the multi-scale architecture largely contributes to improving the results on metals. In this example, the same dataset is used for training both the mono-scale and multi-scale architectures, and the latter offers visually more accuate results. Likely, the ability of the multi-scale architecture to propagate information in a global manner helps interpreting the anisotropic behavior.



Mono (DS1) *Multi* (DS1) "Ground truth"

Figure 8: Results of our mono- and multi-scale architectures (both trained on the pre-existing dataset DS1) on the copper golf ball from [31]. The multi-scale architecture yields much sharper results, especially around the holes.

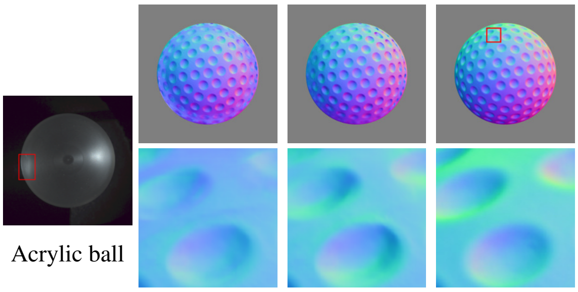
The example of Fig. 9, on the contrary, shows the importance of the presence of metallic objects in the training dataset, independently from the network architecture. It can be observed that the network performs much better when it is trained on our new training dataset, even without considering the multi-scale architecture.



Mono (DS1) *Mono* (DS1+DS2) "Ground truth"

Figure 9: Results of our mono-scale architecture on the copper hexagon from [31]. Since the new dataset (DS2) contains much more metallic objects than the existing one (DS1), training on our new dataset yields largely improved results.

Fig. 10 illustrates a particularly visible improvement brought by the multi-scale architecture, which is the correct handling of translucent materials. In this example, we consider again the gulf ball from [31], but this time coated with an acrylic material. Acrylic is a glass-like material, with some of the light passing through the object. As can be seen in the top of Fig. 10, even when light comes from the right side of the ball, part of its left side appears illuminated. Without seeing the whole object the model could not imagine that there exists a path underneath the surface that lets the light go through. On the contrary, the multi-scale approach being global by construction, such non-local phenomena are better managed by the network and the overall reconstruction is clearly more accurate.

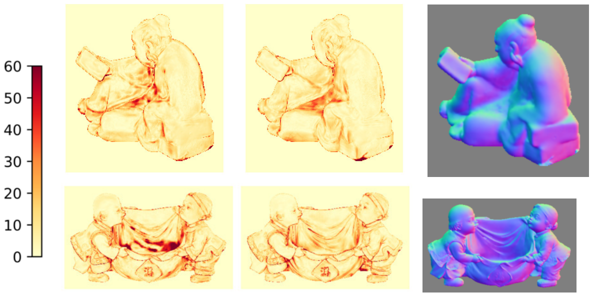


Acrylic ball

Mono (DS1+DS2) *Multi* (DS1+DS2) "Ground truth"

Figure 10: An image of an acrylic ball from [31], illuminated from the right, and results of our mono- and multi-scale architectures (both trained on the new dataset DS2) on the acrylic golf ball from [31]. The reconstruction of translucent objects is improved a lot by using the multi-scale approach.

Others common phenomenas which are cast-shadows and inter-reflections are also better handled by our multi-scale architecture, as Fig. 11 shows.



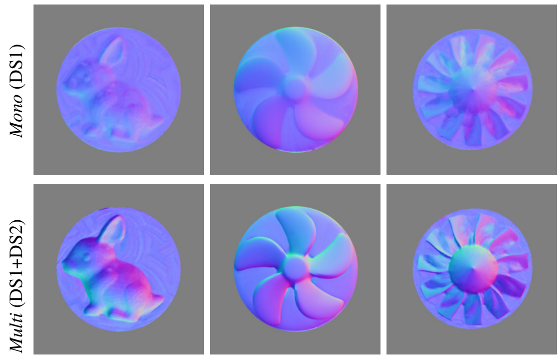
Mono (DS1) *Multi (DS1+DS2)* *Multi (DS1+DS2)*

Figure 11: Angular error map and predicted normal map for the "reading" and "harvest" objects from [34]. The concave parts, where cast shadows and inter-reflections occur, are better handled by our approach.

Fig. 12 shows several additional qualitative comparisons of the result obtained with our baseline (mono-scale architecture trained on the existing dataset) and with both our building blocks included (multi-scale architecture trained on the new dataset). The convex objects (*Bunny* and *Propeller*) are very well reconstructed, despite being fabricated with anisotropic (Aluminium) or moderately specular (ABS, a type of plastic) materials. The steel turbine reconstruction is also improved, although on this object our approach shows its limitations. Indeed, this object exhibits concavities, which create many inter-reflections which are not very well handled by the network.

	ball	bear	buddha	cat	cow	goblet	harvest	pot1	pot2	reading	average
L2 (Baseline)[38]	4.10	8.39	14.92	8.41	25.60	18.5	30.62	8.89	14.65	19.80	15.39
GPS-NET [40]	2.92	5.07	7.77	5.42	6.14	9.00	15.14	6.04	7.01	13.58	7.81
CHR-PSN [19]	2.26	6.35	7.15	5.97	6.05	8.32	15.32	7.04	6.76	12.52	7.77
PS-transformer (10 images) [12]	3.27	4.88	8.65	5.34	6.54	9.28	14.41	6.06	6.97	11.24	7.66
MT-PS-CNN [4]	2.29	5.87	**6.92**	5.79	6.89	**6.85**	**7.88**	11.94	7.48	13.71	7.56
PS-FCN [7]	2.67	7.72	7.52	4.75	6.72	7.84	12.39	6.17	7.15	10.92	7.39
CNN-PS [11]	2.2	4.6	7.9	4.1	8.0	7.3	14.0	5.4	6.0	12.6	7.2
Mono (DS1)	2.63	6.66	8.27	4.47	4.77	8.24	12.78	6.00	5.38	9.68	6.88
Multi (DS1)	**1.60**	7.82	7.55	4.33	4.18	7.85	12.36	5.22	5.36	9.04	6.54
OB-Cnn [10]	2.49	3.59	7.23	4.69	4.89	6.89	12.79	5.10	**4.98**	11.08	6.37
PX-NET [25]	2.03	**3.58**	7.61	4.39	4.69	6.90	13.10	5.08	5.10	10.26	6.28
Multi (DS1+DS2)	2.05	4.24	7.03	**3.9**	**4.00**	7.57	11.01	**4.94**	5.22	**8.47**	**5.84**

Table 2: Mean angular error (in degrees) on the DiLiGenT [34] benchmark.The best result for each object is indicated in bold, and the second best one is underlined. The lines in blue indicate our results. Combining the proposed multi-scale architecture "*Multi*" and proposed training dataset "*DS2*" yields state-of-the-art results, by a large margin.



(a) Bunny - Aluminium (b) Propeller - ABS (c) Turbine - Steel

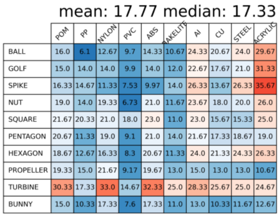
Figure 12: Visual comparison of the improvements brought by the combination of the new architecture and our new training set, on three objects from [31]. All three objects are much better reconstructed, although the steel turbine remains challenging.

5.3 Quantitative evaluation on DiLiGenT [34]

Next, we compare in Table 2 our results against the most recent state-of-the-art methods, on the DiLiGenT benchmark [34]. Let us however remark that PS-transformer [12] takes as inputs no more than 10 images, hence the comparison is biased. Besides, we emphasize that our mono-scale architecture is largely inspired from PS-FCN [6, 7], hence *Mono* (DS1) can be considered as an optimized version of [6, 7], where we let the training phase run for much longer. This table shows that the proposed multi-scale architecture provides a significant gain of 4.6%, in comparison with the mono-scale approach – compare *Mono* (DS1) and *Multi* (DS1). And, as soon as our new training dataset is considered, the state-of-the-art is outperformed and we reach an average angular error below 6°, with a particularly visible improvement on the most difficult "reading" object (Fig. 11).

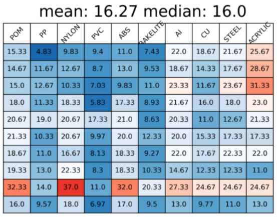
5.4 Quantitative evaluation on DiLiGenT 10^2 [31]

We now quantatively evaluate the impact of the multi-scale architecture on the DiLiGenT 10^2 benchmark [31]. Note that we process images at their full resolution (1024 pixels by 1024), requiring 7 scales in the multi-scale architecture. To this end, we show in Table 3 the difference between the mono- and the multi-scale approaches, when they are both trained on the pre-existing dataset. As can be observed, a significant gain of 9.3% is observed with the multi-scale architecture. The gain is most visible on objects which have a spherical shape and anisotropic material (top right of Tab 3c, see also Fig. 8 for a qualitative result on the Golf - CU object), as well as for the most challenging "acrylic" material, which is translucent.


mean: 17.77 median: 17.33

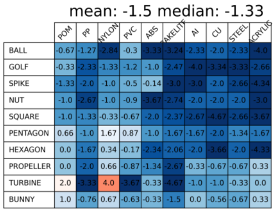
	POM	PP	NYLON	PVC	ABS	BAKELITE	Al	CU	STEEL	ACRYLIC
BALL	16.0	6.1	12.67	9.7	14.33	10.67	24.33	20.67	24.0	29.67
GOLF	15.0	14.0	14.0	9.9	14.0	12.0	22.67	17.67	21.0	31.33
SPIKE	16.33	14.67	11.33	7.53	9.97	14.0	26.33	13.67	26.33	35.67
NUT	19.0	14.0	19.33	6.73	21.0	11.67	23.67	18.0	20.0	26.0
SQUARE	21.67	20.33	21.0	18.0	23.0	11.0	23.0	15.67	15.33	25.0
PENTAGON	20.67	11.33	19.0	9.1	21.0	14.0	21.67	17.33	18.67	19.0
HEXAGON	18.67	12.67	16.33	8.3	20.67	11.33	24.0	21.33	24.33	26.33
PROPELLER	19.33	15.0	21.67	9.17	19.67	13.0	15.0	13.0	13.0	10.67
TURBINE	30.33	17.33	33.0	14.67	32.33	25.0	28.33	25.67	25.0	24.67
BUNNY	15.0	10.33	17.33	7.6	17.33	11.0	13.0	10.33	11.67	12.67

(a) *Mono* (DS1)


mean: 16.27 median: 16.0

	POM	PP	NYLON	PVC	ABS	BAKELITE	Al	CU	STEEL	ACRYLIC
BALL	15.33	4.83	9.83	9.4	11.0	7.43	22.0	18.67	21.67	25.67
GOLF	14.67	11.67	12.67	8.7	13.0	9.53	18.67	14.33	17.67	28.67
SPIKE	15.0	12.67	10.33	7.03	9.83	11.0	23.33	11.67	23.67	31.33
NUT	18.0	11.33	18.33	5.83	17.33	8.93	21.67	16.0	18.0	23.0
SQUARE	20.67	19.0	20.67	17.33	21.0	8.63	20.33	11.0	12.67	21.33
PENTAGON	21.33	10.33	20.67	9.97	20.0	12.33	20.0	15.33	17.33	17.33
HEXAGON	18.67	11.0	16.67	8.13	18.33	9.27	22.0	17.67	22.33	22.0
PROPELLER	19.33	13.0	22.33	8.3	18.33	10.33	14.67	12.33	12.33	11.0
TURBINE	32.33	14.0	37.0	11.0	32.0	20.33	27.33	24.67	24.67	24.67
BUNNY	16.0	9.57	18.0	6.97	17.0	9.5	13.0	9.77	11.0	13.0

(b) *Multi* (DS1)

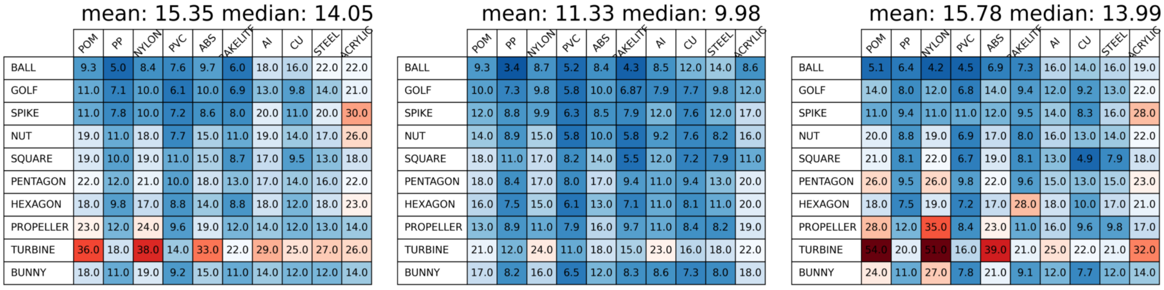

mean: -1.5 median: -1.33

	POM	PP	NYLON	PVC	ABS	BAKELITE	Al	CU	STEEL	ACRYLIC
BALL	-0.67	-1.27	-2.84	-0.3	-3.33	-3.24	-2.33	-2.0	-2.33	-4.0
GOLF	-0.33	-2.33	-1.33	-1.2	-1.0	-2.47	-4.0	-3.34	-3.33	-2.66
SPIKE	-1.33	-2.0	-1.0	-0.5	-0.14	-3.0	-3.0	-2.0	-2.66	-4.34
NUT	-1.0	-2.67	-1.0	-0.9	-3.67	-2.74	-2.0	-2.0	-2.0	-3.0
SQUARE	-1.0	-1.33	-0.33	-0.67	-2.0	-2.37	-2.67	-4.67	-2.66	-3.67
PENTAGON	0.66	-1.0	1.67	0.87	-1.0	-1.67	-1.67	-2.0	-1.34	-1.67
HEXAGON	0.0	-1.67	0.34	-0.17	-2.34	-2.06	-2.0	-3.66	-2.0	-4.33
PROPELLER	0.0	-2.0	0.66	-0.87	-1.34	-2.67	-0.33	-0.67	-0.67	0.33
TURBINE	2.0	-3.33	4.0	-3.67	-0.33	-4.67	-1.0	-1.0	-0.33	0.0
BUNNY	1.0	-0.76	0.67	-0.63	-0.33	-1.5	0.0	-0.56	-0.67	0.33

(c) *Multi* (DS1) - *Mono* (DS1)

Table 3: Mean angular on the *DiLiGenT10*^{2} benchmark, considering either the mono-scale architecture or the multi-scale one, both trained on the pre-existing dataset DS1. The multi-scale approach yields a significant gain, most visible on the top-right part of the table (spherical shapes with anisotropic reflectance).

https://www.doi.org/10.24132/CSRN.3301.23



mean: 15.35 median: 14.05

	POM	PP	NYLON	PVC	ABS	BAKELITE	AI	CU	STEEL	ACRYLIC
BALL	9.3	5.0	8.4	7.6	9.7	6.0	18.0	16.0	22.0	22.0
GOLF	11.0	7.1	10.0	6.1	10.0	6.9	13.0	9.8	14.0	21.0
SPIKE	11.0	7.8	10.0	7.2	8.6	8.0	20.0	11.0	20.0	30.0
NUT	19.0	11.0	18.0	7.7	15.0	11.0	19.0	14.0	17.0	26.0
SQUARE	19.0	10.0	19.0	11.0	15.0	8.7	17.0	9.5	13.0	18.0
PENTAGON	22.0	12.0	21.0	10.0	18.0	13.0	17.0	14.0	16.0	22.0
HEXAGON	18.0	9.8	17.0	8.8	14.0	8.8	18.0	12.0	18.0	23.0
PROPELLER	23.0	12.0	24.0	9.6	19.0	12.0	14.0	12.0	13.0	14.0
TURBINE	36.0	18.0	38.0	14.0	33.0	22.0	29.0	25.0	27.0	26.0
BUNNY	18.0	11.0	19.0	9.2	15.0	11.0	14.0	12.0	12.0	14.0

(a) *Mono* (DS1+DS2)

mean: 11.33 median: 9.98

	POM	PP	NYLON	PVC	ABS	BAKELITE	AI	CU	STEEL	ACRYLIC
BALL	9.3	3.4	8.7	5.2	8.4	4.3	8.5	12.0	14.0	8.6
GOLF	10.0	7.3	9.8	5.8	10.0	6.87	7.9	7.7	9.8	12.0
SPIKE	12.0	8.8	9.9	6.3	8.5	7.9	12.0	7.6	12.0	17.0
NUT	14.0	8.9	15.0	5.8	10.0	5.8	9.2	7.6	8.2	16.0
SQUARE	18.0	11.0	17.0	8.2	14.0	5.5	12.0	7.2	7.9	11.0
PENTAGON	18.0	8.4	17.0	8.0	17.0	9.4	11.0	9.4	13.0	20.0
HEXAGON	16.0	7.5	15.0	6.1	13.0	7.1	11.0	8.1	11.0	20.0
PROPELLER	13.0	8.9	11.0	7.9	16.0	9.7	11.0	8.4	8.2	19.0
TURBINE	21.0	12.0	24.0	11.0	18.0	15.0	23.0	16.0	18.0	22.0
BUNNY	17.0	8.2	16.0	6.5	12.0	8.3	8.6	7.3	8.0	18.0

(b) *Multi* (DS1+DS2)

mean: 15.78 median: 13.99

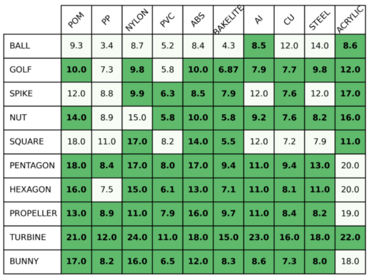
	POM	PP	NYLON	PVC	ABS	BAKELITE	AI	CU	STEEL	ACRYLIC
BALL	5.1	6.4	4.2	4.5	6.9	7.3	16.0	14.0	16.0	19.0
GOLF	14.0	8.0	12.0	6.8	14.0	9.4	12.0	9.2	13.0	22.0
SPIKE	11.0	9.4	11.0	11.0	12.0	9.5	14.0	8.3	16.0	28.0
NUT	20.0	8.8	19.0	6.9	17.0	8.0	16.0	13.0	14.0	22.0
SQUARE	21.0	8.1	22.0	6.7	19.0	8.1	13.0	4.9	7.9	18.0
PENTAGON	26.0	9.5	26.0	9.8	22.0	9.6	15.0	13.0	15.0	23.0
HEXAGON	18.0	7.5	19.0	7.2	17.0	28.0	18.0	10.0	17.0	21.0
PROPELLER	28.0	12.0	35.0	8.4	23.0	11.0	16.0	9.6	9.8	17.0
TURBINE	54.0	20.0	51.0	16.0	39.0	21.0	25.0	22.0	21.0	32.0
BUNNY	24.0	11.0	27.0	7.8	21.0	9.1	12.0	7.7	12.0	14.0

(c) CNN-PS [11] (DS1)

Table 4: Mean angular error on the *DiLiGenT10*^{2} benchmark, with the results of CNN-PS [11] indicated for comparison. When incorporating both the new dataset and the multi-scale architecture, the state-of-the-art is largely outperformed.

We repeat this experiment in Table 4, but this time with our networks trained on the new dataset. Comparing Tables 3 and 4 allows one to quantify the benefits of using our new training dataset: the mono-scale architecture gets improved by 14%, and the multi-scale one by 30%. Comparing Tables 4a and 4b also allows one to quantify the impact of switching to the multi-scale architecture: the results improve by 26%. Particularly large improvements can be observed on the *Turbine* and *Acrylic Gulf* objects (see also Figs. 12c and 10). For such objects with non-local light transport (due to interreflections or anisotropic reflectance), the ability of tne multi-scale approach to get access to a global information is indeed of primary importance.

Overall, the combination of the new architecture and dataset allows one to reach an average error of 11.33° on this benchmark. This is to be compared with the 15.78° achieved by CNN-PS [11] (Table 4c), which was the best performing method so far [31]. By comparing our results with all available state-of-the-art methods [5, 6, 11, 30, 33, 35, 36, 39, 40, 41], we found out that the proposed method is the best performer on 73% of the objects of this benchmark, as indicated in Table 5.

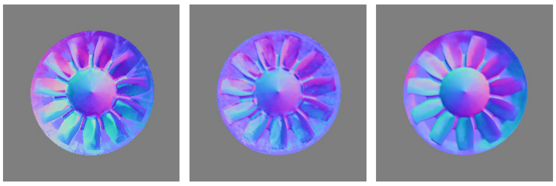


	POM	PP	NYLON	PVC	ABS	BAKELITE	AI	CU	STEEL	ACRYLIC
BALL	9.3	3.4	8.7	5.2	8.4	4.3	**8.5**	12.0	14.0	**8.6**
GOLF	**10.0**	7.3	**9.8**	5.8	**10.0**	**6.87**	**7.9**	**7.7**	**9.8**	**12.0**
SPIKE	12.0	8.8	**9.9**	**6.3**	**8.5**	**7.9**	12.0	**7.6**	12.0	**17.0**
NUT	**14.0**	8.9	15.0	**5.8**	**10.0**	**5.8**	**9.2**	**7.6**	**8.2**	**16.0**
SQUARE	18.0	11.0	**17.0**	8.2	**14.0**	**5.5**	12.0	7.2	7.9	**11.0**
PENTAGON	**18.0**	**8.4**	**17.0**	**8.0**	**17.0**	**9.4**	**11.0**	**9.4**	**13.0**	20.0
HEXAGON	**16.0**	7.5	**15.0**	**6.1**	**13.0**	**7.1**	**11.0**	**8.1**	**11.0**	20.0
PROPELLER	**13.0**	**8.9**	**11.0**	**7.9**	**16.0**	**9.7**	**11.0**	**8.4**	**8.2**	19.0
TURBINE	**21.0**	**12.0**	**24.0**	**11.0**	**18.0**	**15.0**	**23.0**	**16.0**	**18.0**	**22.0**
BUNNY	**17.0**	**8.2**	**16.0**	**6.5**	**12.0**	**8.3**	**8.6**	**7.3**	**8.0**	18.0

Table 5: Mean angular error achieved by the best performer among [5, 6, 11, 30, 33, 35, 36, 39, 40, 41] and us, on the 100 objects of [31]. Green cases indicate when the proposed architecture, combined with the new dataset, gives the best results.

5.5 Limitations

Even if the combination of our multi-scale and our new training dataset improves the results on non-Lambertian materials, some shortcomings remain. For example, we notice that the normals at the border of some translucent objects are incorrectly predicted (Fig. 13). As shown in Fig. 14, in this example the the opposite side of the incoming light is the most shiny part of the image. Although our multi-scale approach better handles such anisotropic than the mono-scale one or existing methods such as CNN-PS, it shows its limitations when the anisotropy is this much important.

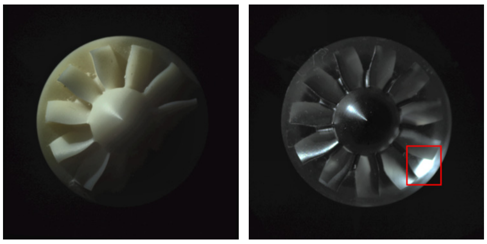


CNN-PS (DS1)
Error: 32°

Mono (DS1)
Error: 26°

Multi (DS1+DS2)
Error: 22°

Figure 13: Results of CNN-PS, our mono-scale and our multi-scale architecture on the acrylic turbine.



ABS

Acrylic

Figure 14: Same turbine, fabricated either with a diffuse (ABS) or an anisotropic (acrylic) material, and illuminated from the same direction (coming from "top left"). The bottom-right area, which is shadowed in the diffuse case, appears much shinier on the anisotropic object.

6 CONCLUSION

In this paper, we have proposed a novel deep normal estimation framework for the calibrated photometric stereo problem. It builds upon a multi-scale architecture which is independent from the resolution of the images, as well as a new comprehensive learning dataset. We have shown on publicly available benchmarks that the combination of these two features yields state-of-the-art results, with performances particularly improved on challenging anisotropic materials. In the future, we plan to extend our approach to handle observation maps [11] as well, which have recently been shown to benefit from physical interpretability [13].

7 REFERENCES

[1] AmbientCG. https://ambientcg.com/.

[2] Sketchfab. https://sketchfab.com.

[3] B. Burley and W. D. Studios. Physically-based shading at Disney. *ACM SIGGRAPH Courses*, 2012.

[4] Y. Cao, B. Ding, Z. He, J. Yang, J. Chen, Y. Cao, and X. Li. Learning inter-and intraframe representations for non-Lambertian photometric stereo. *OLEN*, 150:106838, 2022.

[5] G. Chen, K. Han, B. Shi, Y. Matsushita, and Kwan-Yee K. Wong. Self-Calibrating Deep Photometric Stereo Networks. In *CVPR*, 2019.

[6] G. Chen, K. Han, and K. Wong. PS-FCN: A Flexible Learning Framework for Photometric Stereo. In *ECCV*, 2018.

[7] G. Chen, Kai Han, Boxin S., Yasuyuki M., and K. W. Deep Photometric Stereo for Non-Lambertian Surfaces. *PAMI*, 44(1), 2022.

[8] Blender Online Community. *Blender - a 3D modelling and rendering package*, 2018.

[9] B. Haefner, Z. Ye, M. Gao, T. Wu, Y. Quéau, and D. Cremers. Variational uncalibrated photometric stereo under general lighting. In *ICCV*, 2019.

[10] D. Honzátko, E. Türetken, P. Fua, and L. Dunbar. Leveraging Spatial and Photometric Context for Calibrated Non-Lambertian Photometric Stereo. In *3DV*, 2021.

[11] S. Ikehata. CNN-PS: CNN-based Photometric Stereo for General Non-Convex Surfaces. In *ECCV*, 2018.

[12] S. Ikehata. PS-transformer: Learning sparse photometric stereo network using self-attention mechanism. In *BMVC*, 2021.

[13] S. Ikehata. Does Physical Interpretability of Observation Map Improve Photometric Stereo Networks? In *ICIP*, 2022.

[14] S. Ikehata. Universal photometric stereo network using global lighting contexts. *CVPR*, 2022.

[15] M. Johnson and E. Adelson. Shape Estimation in Natural Illumination. In *CVPR*, 2011.

[16] Y. Ju, J. Dong, and S. Chen. Recovering Surface Normal and Arbitrary Images: A Dual Regression Network for Photometric Stereo. *TIP*, 30:3676–3690, 2021.

[17] Y. Ju, M. Jian, J. Dong, and K. Lam. Learning Photometric Stereo via Manifold-based Mapping. In *VCIP*, 2020.

[18] Y. Ju, K. Lam, Y. Chen, L. Qi, and J. Dong. Pay Attention to Devils: A Photometric Stereo Network for Better Details. In *IJCAI*, 2020.

[19] Y. Ju, Y. Peng, M. Jian, F. Gao, and J. Dong. Learning conditional photometric stereo with high-resolution features. *CVM*, 8(1):105–118, 2022.

[20] B. Kaya, S. Kumar, C. Oliveira, V. Ferrari, and Van G. Uncalibrated Neural Inverse Rendering for Photometric Stereo of General Surfaces. In *CVPR*, 2021.

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

[22] J. Li, A. Robles-Kelly, S. You, and Y. Matsushita. Learning to Minify Photometric Stereo. In *CVPR*, 2019.

[23] D. Lichy, S. Sengupta, and D. Jacobs. Fast light-weight near-field photometric stereo. In *CVPR*, 2022.

[24] D. Lichy, J. Wu, S. Sengupta, and D. Jacobs. Shape and Material Capture at Home. In *CVPR*, 2021.

[25] F. Logothetis, I. Budvytis, R. Mecca, and R. Cipolla. PX-net: Simple and efficient pixel-wise training of photometric stereo networks. In *ICCV*, 2021.

[26] F. Logothetis, R. Mecca, I. Budvytis, and R. Cipolla. A CNN based approach for the point-light photometric stereo problem. *IJCV*, 131(1):101–120, 2023.

[27] W. Lorensen and H. Cline. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH*, 1987.

[28] W. Matusik. *A data-driven reflectance model*. PhD thesis, Massachusetts Institute of Technology, 2003.

[29] Z. Mo, B. Shi, F. Lu, S.-K. Yeung, and Y. Matsushita. Uncalibrated photometric stereo under natural illumination. In *CVPR*, 2018.

[30] T. Papadhimitri and P. Favaro. A Closed-Form, Consistent and Robust Solution to Uncalibrated Photometric Stereo Via Local Diffuse Reflectance Maxima. *IJCV*, 2014.

[31] J. Ren, F. Wang, J. Zhang, Q. Zheng, M. Ren, and B. Shi. DiLiGenT10^2: A Photometric Stereo Benchmark Dataset with Controlled Shape and Material Variation. In *CVPR*, 2022.

[32] H. Santo, M. Waechter, and Y. Matsushita. Deep near-light photometric stereo for spatially varying reflectances. In *ECCV*, 2020.

[33] B. Shi, P. Tan, Y. Matsushita, and K. Ikeuchi. Bi-polynomial modeling of low-frequency reflectances. *PAMI*, 36(6):1078–1091, 2013.

[34] B. Shi, Z. Wu, Z. Mo, D. Duan, S. Yeung, and P. Tan. A Benchmark Dataset and Evaluation for Non-Lambertian and Uncalibrated Photometric Stereo. In *CVPR*, 2016.

[35] B. Shi, Z. Wu, Z. Mo, D. Duan, S. Yeung, and P. Tan. A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. In *CVPR*, 2016.

[36] T. Taniai and T. Maehara. Neural Inverse Rendering for General Reflectance Photometric Stereo. In *ICML*, 2018.

[37] X. Wang, Z. Jian, and M. Ren. Non-Lambertian Photometric Stereo Network Based on Inverse Reflectance Model With Collocated Light. *TIP*, 29, 2020.

[38] R. J. Woodham. Photometric Method For Determining Surface Orientation From Multiple Images. *Opt. Eng.*, 19, 1980.

[39] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma. Robust photometric stereo via low-rank matrix completion and recovery. In *ACCV*, 2011.

[40] Z. Yao, K. Li, Y. Fu, H. Hu, and B. Shi. GPS-Net: Graph-based Photometric Stereo Network. In *NIPS*, 2020.

[41] Q. Zheng, Y. Jia, B. Shi, X. Jiang, L. Duan, and A. Kot. SPLINE-Net: Sparse Photometric Stereo Through Lighting Interpolation and Normal Estimation Networks. In *ICCV*, 2019.

https://www.doi.org/10.24132/CSRN.3301.23