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

Clément Hardy, Yvain Quéau, David Tschumperle
Normandie Univ, UNICAEN, GREYC
Caen, France
{clement.hardy, yvain.queau, 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.

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. [36]) 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 [29]. In order to manage non-Lambertian surfaces, deep learning methods based on convolutional neural networks have recently emerged as the most efficient ones [29, 32]. The quality of results obtained by such approaches relies on two main factors:

• 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.
• 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.

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 an unprecedented variety of geometry and non-Lambertian reflectance.

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, 21, 23, 39], which projects all observations corresponding to each same 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, 7, 16, 18, 35], 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 [7] 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. [38] 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 the present paper, we will rather consider the pooling-based scheme from [7] as a baseline model, and broaden it to a multi-scale architecture.

**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 [7], 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

![Figure 2. Samples from existing datasets. The first three [11, 15] are synthetic ones, used for training the neural networks. Both the last ones [29, 32] 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.](image-url)
DiLiGenT 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.

Existing benchmarking datasets To validate the relevance of the training datasets, and to verify that the models trained on these synthetic data are able to generalize to real images, two real-world datasets exist: DiLiGenT [32] and DiLiGenT10° [29]. The DiLiGenT dataset contains 10 different objects, taken from the same viewpoint under 96 different illuminations (Fig. 3a). 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° dataset contains 10 different objects. Each object was explicitly machined 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.

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 [24, 30] or natural illumination [9, 27], is another ongoing research problem. In this paper we focus on the case of calibrated PS with known directional light sources.

3. Proposed Multi-Scale Architecture

The multi-scale architecture we propose builds upon the normal estimation network introduced in [7]. 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 3. Distribution of illumination directions in the real DiLiGenT and DiLiGenT10° 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).

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.
Figure 5. First two stages of the proposed multi-scale architecture. A first architecture, inspired by the PS-FCN method [7], 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).

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 [7]. 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:

\[
l_{\text{normal}} = 1 - \sum_{ij} N_{ij} \hat{N}_{ij}, \tag{1}\]

where \( \hat{N}_{ij} \) is the estimated normal at pixel \((i, j)\), and \( N_{ij} \) is the ground truth one.

As remarked in [22], 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, 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 \( 8 \times 8 \) pixels. By taking a resolution multiplier of two between two scales, four scales are necessary to reach a resolution of \( 128 \times 128 \) (training resolution in our test), and seven scales for the DiLiGenT10 images which have a resolution of \( 1001 \times 1001 \). 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.
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. **Our Blobby** has 3000 distinct objects, generated by the sum of random Gaussian potentials, followed by iso-surface extraction using the Marching Cubes algorithm [25]. **Our Object** contains 76 detailed objects which are 3D meshes from the Sketchfab [2] website. Moreover, to allow the rendering 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 some 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 probability 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).

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 [7]. 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 (not done yet, to preserve author’s anonymity).

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

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

In this section, we demonstrate the effectiveness of our proposed multi-scale architecture on publicly available benchmarks, namely DiLiGenT [32] and DiLiGenT10² [29]. 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). Similarly, to quantify the gain due to the multi-scale architecture we trained both a single-scale version of our network on $128 \times 128$ patches (“Mono” in the following) and the multi-scale one (“Multi” 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. Qualitative evaluation

Let us start by showing two illustrative results on the DiLiGenT10² [29] 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 machined 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 accurate results. Likely, the ability of the multi-scale architecture to propagate information in a global manner helps interpreting the anisotropic behavior.

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.

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 golf ball from [29], 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.
Table 2. Mean angular error (in degrees) on the DiLiGenT [32] 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.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Acrylic ball</td>
<td>4.10</td>
<td>2.92</td>
<td>3.27</td>
<td>2.26</td>
<td>3.72</td>
<td>2.29</td>
<td>2.67</td>
<td>2.63</td>
<td>1.60</td>
<td>2.03</td>
</tr>
<tr>
<td>“Ground truth”</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
<td>4.00</td>
</tr>
<tr>
<td>Bunny</td>
<td>6.39</td>
<td>5.57</td>
<td>6.92</td>
<td>5.79</td>
<td>6.89</td>
<td>6.85</td>
<td>7.88</td>
<td>6.90</td>
<td>4.69</td>
<td>2.03</td>
</tr>
<tr>
<td>Aluminium</td>
<td>14.92</td>
<td>7.77</td>
<td>8.65</td>
<td>7.15</td>
<td>5.34</td>
<td>6.66</td>
<td>7.52</td>
<td>8.27</td>
<td>4.73</td>
<td>7.03</td>
</tr>
<tr>
<td>Reading</td>
<td>9.41</td>
<td>5.42</td>
<td>4.33</td>
<td>4.72</td>
<td>4.73</td>
<td>6.89</td>
<td>13.00</td>
<td>6.90</td>
<td>4.69</td>
<td>4.00</td>
</tr>
<tr>
<td>Harvest</td>
<td>8.41</td>
<td>5.42</td>
<td>4.33</td>
<td>4.72</td>
<td>4.73</td>
<td>6.89</td>
<td>13.00</td>
<td>6.90</td>
<td>4.69</td>
<td>4.00</td>
</tr>
<tr>
<td>Cow</td>
<td>25.60</td>
<td>9.00</td>
<td>9.00</td>
<td>6.14</td>
<td>6.14</td>
<td>8.32</td>
<td>14.14</td>
<td>9.28</td>
<td>7.82</td>
<td>7.82</td>
</tr>
<tr>
<td>Cat</td>
<td>18.5</td>
<td>15.14</td>
<td>15.32</td>
<td>8.32</td>
<td>8.32</td>
<td>15.32</td>
<td>18.44</td>
<td>14.41</td>
<td>11.94</td>
<td>11.94</td>
</tr>
<tr>
<td>Pot1</td>
<td>30.02</td>
<td>6.04</td>
<td>7.04</td>
<td>7.04</td>
<td>7.04</td>
<td>12.67</td>
<td>13.71</td>
<td>6.06</td>
<td>11.74</td>
<td>11.74</td>
</tr>
<tr>
<td>Pot2</td>
<td>8.89</td>
<td>7.01</td>
<td>7.04</td>
<td>7.04</td>
<td>7.04</td>
<td>12.67</td>
<td>13.71</td>
<td>6.06</td>
<td>11.74</td>
<td>11.74</td>
</tr>
<tr>
<td>Average</td>
<td>15.39</td>
<td>7.81</td>
<td>7.77</td>
<td>7.77</td>
<td>7.77</td>
<td>7.77</td>
<td>7.77</td>
<td>7.77</td>
<td>7.77</td>
<td>7.77</td>
</tr>
</tbody>
</table>

5.2. Quantitative evaluation on DiLiGenT [32]

Next, we compare in Table 2 our results against the most recent state-of-the-art methods, on the DiLiGenT benchmark [32]. 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.

5.3. Quantitative evaluation on DiLiGenT 10^2 [29]

We now quantitatively evaluate the impact of the multi-scale architecture on the DiLiGenT 10^2 benchmark [29]. To this end, we show in Table 4 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 4c, 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.
Table 3. Mean angular error on the DiLiGenT102 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.

Table 4. Mean angular on the DiLiGenT102 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).

We repeat this experiment in Table 3, but this time with our networks trained on the new dataset. Comparing Tables 4 and 3 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 3a and 3b also allows one to quantify the impact of switching to the multi-scale architecture: the results improve by 26%. A particularly large improvement can be observed on the Turbine object, which is partly concave with many inter-reflections between the blades. For such an object, a model needs to get access to the whole geometry to be able to analyze the light path. This is probably why the multi-scale approach yields such an improvement, although the results remain perfectible, as illustrated in Fig. 11c. Another impressive example of improvement is the Acrylic Gulf object: by using a global approach, the results improve on such translucent materials with non-local light transport, as also illustrated in Fig. 10.

Overall, the combination of the new architecture and dataset allows one to reach an unprecedented average error of 11.33° on this benchmark. This is to be compared with the 15.78° achieved by CNN-PS [11] (Table 3c), which was the best performing method so far [29]. By comparing our results with all available state-of-the-art methods [5, 7, 11, 28, 31, 33, 34, 37–39], we found out that the proposed method is the best performer on 73% of the objects of this benchmark, as indicated in Table 5.

6. Conclusion and Future Work

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].
References


