Learning to Minify Photometric Stereo

Junxuan Li[1,2] Antonio Robles-Kelly[3] Shaodi You[1,2] Yasuyuki Matsushita[4]

[1]Australian National University, College of Eng. and Comp. Sci., Acton, ACT 2601, Australia
[2]Data61-CSIRO, Black Mountain Laboratories, Acton, ACT 2601, Australia
[3]Deakin University, Faculty of Sci., Eng. and Built Env., Waurn Ponds, VIC 3216, Australia
[4]Osaka University, Graduate School of Information Science and Technology, Osaka 565-0871, Japan

Abstract

Photometric stereo estimates the surface normal given a set of images acquired under different illumination conditions. To deal with diverse factors involved in the image formation process, recent photometric stereo methods demand a large number of images as input. We propose a method that can dramatically decrease the demands on the number of images by learning the most informative ones under different illumination conditions. To this end, we use a deep learning framework to automatically learn the critical illumination conditions required at input. Furthermore, we present an occlusion layer that can synthesize cast shadows, which effectively improves the estimation accuracy. We assess our method on challenging real-world conditions, where we outperform techniques elsewhere in the literature with a significantly reduced number of light conditions.

1. Introduction

Photometric stereo is a technique that estimates the surface normal of an object from numbers of photographs taken under different illumination conditions from a fixed viewpoint. Due to the diverse appearances of real-world materials, a method for dealing with general reflectances is desired. Recent methods toward this direction improve the estimation accuracy by *robust* estimation at the cost of increasing the number of images, such as [21]; however, it complicates the data acquisition setup and calibration procedure.

Recently, deep learning-based methods appeared in the context of photometric stereo [16, 8, 3]. These methods are shown effective for surfaces with diverse reflectances, indicating that the mapping from input images to surface normal can be well established even with the diversity. Unlike these methods that primarily focus on the estimation accuracy, we study the problem of *reducing* the number of required images so as to minify the photometric stereo input. We approach this problem by learning the relative importance of different illuminant directions, which opens up a possibility of optimally selecting input illuminant directions for lightweight data acquisition.

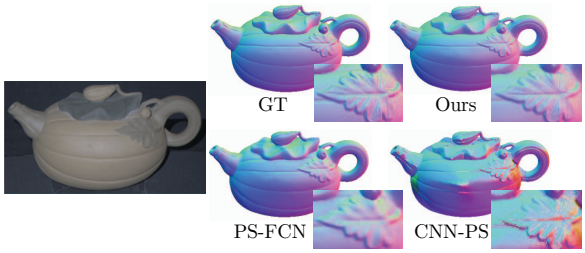

Figure 1. Performance with only 8 inputs for our method, PS-FCN [3] and CNN-PS [8] on the "pot1" from DiLiGenT [17]. Note we outperform the alternatives.

However, reducing the number of input illuminant directions without loss in performance is a non-trivial task because the surface radiance is determined by the illumination intensity, illuminant direction, surface normal and BRDF function. For instance, Argyriou *et al*. [2] use a sparse representation of illuminations for reducing the required number of lights to 5 under the assumption of Lambertian surfaces. The task is further complicated by the presence of cast shadows, specularities, and inter-reflections.

In this paper, we propose a deep learning approach, which applies a connection table that can select those illuminant directions that are most relevant to the surface normal prediction process. To this end, we employ a connection table so as to introduce a trainable input mapping. This mapping is trained by making use of the ℓ_1-norm and a sparsity-loss. Once the model is trained, it can efficiently estimate the surface normal only given a much-reduced number of input images without loss in accuracy. For the sake of the scalability of our method, here we select the illuminant directions making use of an observation map [8] and estimate the surface normal in a pixel-wise fashion.

978-1-7281-3293-8/19/$31.00 ©2019 IEEE
DOI 10.1109/CVPR.2019.00775

Furthermore, to account for the effect of global illumination, we tackle the problem of cast shadows explicitly. Specifically, by regarding cast shadows as a localized zero pattern on the observation map, we introduce an occlusion layer into the network to deal with cast shadows. In summary, our contributions are:

1. We propose a connection table in the network input together with a suitable loss function and a rank selection process so as to select those illuminant directions that are the most relevant to the surface normal prediction process.
2. We propose an occlusion layer that can simulate object cast shadows. This occlusion layer can be applied on both augmented and real-world datasets, and enhance performance on shadowed areas.
3. Our end-to-end deep neural network for photometric stereo can predict the surface normal with a much-reduced number of light source directions. This is consistent with our results on the DiLiGenT benchmark [17], where we can predict the surface normal with as few as 8 input images.

2. Related Work

The literature of photometric stereo is vast, it can be divided, however, into the following groups:

Least-squares methods: Introduced by Woodham [20], least-squares approaches aim at solving the problem under the Lambertian assumption, *i.e.*, the pixel intensity is proportional to the cosine of the angle between the illuminant direction and the surface normal. Moreover, they often assume the surface is convex and devoid of cast shadows with a uniform, Lambertian reflectance. The Lambertian assumption is important since it allows for the image formation model to be cast into a linear system of equations that can be solved in a closed-form manner.

Robust approaches: These approaches can cope with specularities and cast shadows on the object under study by viewing non-Lambertian regions as outliers. Wu *et al*. [21] add an additional term to the image formation model so as to naturally represent the error, which accounts for those pixels that deviate from the Lambertian assumption and solve the photometric stereo problem using rank minimization. Many other techniques are also used for solving the problem under the assumptions above, such as RANSAC [14], expectation maximization [22], sparse regression [9] and variational optimization approaches [15].

Example based methods: These approaches take exemplars of materials and surface geometry as the reference to estimate the surface normal of an unknown object with the same or similar BRDF function which are in the example set. Hertzmann and Seitz [5] propose a method that first clusters the materials on the object. More recently, Hui and Sankaranarayanan [7] use a virtually rendered sphere instead of a physical one for normal estimation.

Deep learning: These methods are the ones that, recently, have achieved the best performance in photometric stereo. Santo *et al*. [16] propose the first deep network-based method for pixel-wise estimation. In their model, they assume that the light directions are all known and consistent between the training and prediction phases. Nonetheless effective, this assumption on the consistency of the training and testing set greatly limits the generalization capabilities of the method and makes it overly sensitive to the training set. Ikehata [8] has proposed a deep net which employs an observation map for unstructured photometric stereo inputs. This map is intended to account for every illuminant direction at each surface pixel. Despite effective, the author assumes a dense illumination map, making the method prone to corruption with a lesser amount of illuminant directions at input. In a different approach, Taniai and Maehara [18] propose an unsupervised network that can take the whole images at input and predict the dense normal in an end-to-end fashion. This method has the drawback, however, of being computationally very costly. More recently, Xu *et al*. [23] propose a deep net to learn the optimal samples for image-based relighting.

3. Preliminaries

In this section, we commence by providing the basic underpinnings and notations regarding photometric stereo.

3.1. Reflected Irradiance

Assume that a light source illuminates a surface point from the direction $\mathbf{l} \in \mathcal{S}^2$ (the space of 3-dimensional unit vectors) and its irradiance is e. And, respectively, the reflected radiance of this surface is denoted as r. The surface normal is denoted as $\mathbf{n} \in \mathcal{S}^2$. The BRDF function is then denoted as $\rho(\mathbf{l}, \mathbf{v}, \mathbf{n})$, *i.e.*, a function of the incident and reflected radiance vectors $\mathbf{l}$, $\mathbf{v}$ and the surface normal $\mathbf{n}$.

Due to the variation of the normals and light source directions, shadows may appear at the reflecting surface. If the normal is opposite to the light direction, *i.e.*, $\mathbf{l} \cdot \mathbf{n} \leq 0$, the attached shadow occurs. If the surface is occluded by the object itself, the cast shadow also occurs, as shown in Fig. 2. As a result, the observed reflected radiance $r(\mathbf{v})$ can be written as

$$r(\mathbf{v}) = \mathbb{Q}\left(\mathrm{e}\rho(\mathbf{l}, \mathbf{v}, \mathbf{n}) \max(\mathbf{l} \cdot \mathbf{n}, 0)\right), \tag{1}$$

where $\mathbb{Q} \in \{0, 1\}$ is a binary variable with a value of 0 at cast shadows, assuming that there are no inter-reflections in shadowed areas. Hence, as long as the illumination is occluded by objects, the indicator variable $\mathbb{Q}$ will be set to zero regardless.

Following the conventional calibrated photometric stereo problem, we assume that the light direction $\mathbf{l}$ and

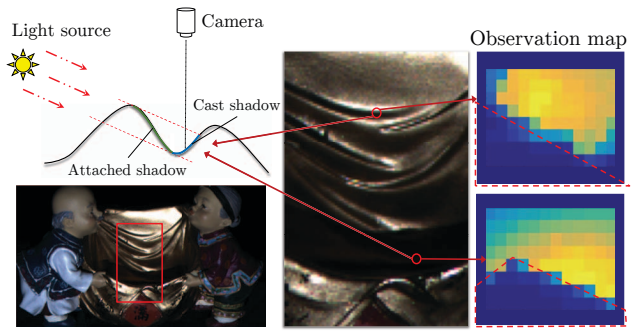

Figure 2. Shadowed area on "harvest" from the DiLiGenT [17] dataset. Note cast shadows are caused by occlusions, while attached shadowed areas are facing back with respect to the light source. On the right-hand side of the figure, we show two observation maps obtained from cast shadows. The red dash line on the observation map indicates the shape that occludes the lights.

intensity e are given at each illumination condition. Also, we assume that the light source is in distance over the images. In order to simplify the ensuing equations, we consider the camera to be in orthographic position and, hence, $\mathbf{v}$ is $[0, 0, 1]$. Finally, here we use, without any loss of generality, the relative intensity $m = \frac{r}{e}$ to rewrite the Eq. (1) as follows

$$m = \mathbb{Q}\big(\rho(\mathbf{l}, \mathbf{v}, \mathbf{n}) \max(\mathbf{l} \cdot \mathbf{n}, 0)\big). \tag{2}$$

3.2. Observation Map Computation

We follow Ikehata [8] and adopt the use of an observation map. Recall that an observation map is a 2-D projection of all the lighting directions in a 3-D hemisphere. More formally, assume that a light direction is given in the x, y, z -coordinate system by the vector $\mathbf{l} = [l_x, l_y, l_z]^\top$. In contrast with Ikehata [8], we do not use the Cartesian coordinates directly, but rather compute the observation map by first projecting the Cartesian vector $\mathbf{l}$ into a polar coordinate system whose components are given by $\theta = \arcsin(l_x / \sqrt{l_x^2 + l_z^2}), \varphi = \arcsin(l_y / \sqrt{l_y^2 + l_z^2})$. Here, we have used a different size of observation maps as compared to Ikehata [8], and have noted that the performance of the polar coordinates outperforms Cartesian ones for this size of observation maps.

The entry indexed $\mathrm{int}\left(\frac{\theta+b}{\Delta\theta}\right), \mathrm{int}\left(\frac{\varphi+b}{\Delta\varphi}\right)$ of the observation map $\boldsymbol{O} \in \mathbb{R}^{w \times w}$ with a window size w can be then set to m. In the indexes above, b is the upper and lower bound of the range of θ and ϕ and $\mathrm{int}(\cdot)$ is an operator the rounding operator that delivers the closest integer to its argument, and $\Delta\theta$ and $\Delta\varphi$ are the gap between each of the map indexes given by $\Delta\theta = \Delta\varphi = \frac{2b}{w-1}$.

In this manner, the observation map converts un-ordered inputs into a meaningful feature map which captures the relationship between the reflectance properties of the object under different illuminant directions. As shown in Fig. 2, each value on the map naturally encodes the information of the surface reflectance r, illuminant direction $\mathbf{l}$ and intensity e. Moreover, the map can also be viewed as a group of functions of the normal $\mathbf{n}$, the BRDF ρ and the cast shadow indicator variable $\mathbb{Q}$.

3.3. illuminant direction Selection

We now turn our attention to the learning of those illuminant directions that are the most relevant to the photometric stereo process so as to greatly minimize the number of views required to compute the surface normal. Mathematically, if we have k different illuminations with directions $\boldsymbol{L} = [\mathbf{l}^{(1)}, \mathbf{l}^{(2)}, \cdots, \mathbf{l}^{(k)}]^\top \in \mathbb{R}^{k \times 3}$, and for each direction of the lights the relative intensities on a surface point are given by $\boldsymbol{m} = [m^{(1)}, m^{(2)}, \cdots, m^{(k)}]^\top \in \mathbb{R}^k$. Then for each surface point we have a system of equations given by

$$\boldsymbol{m} = \mathbb{Q} \circ \rho(\mathbf{l}, \mathbf{v}, \mathbf{n}) \circ \mathrm{vmax}(\boldsymbol{L}\mathbf{n}, \mathbf{0}), \tag{3}$$

where $\mathbb{Q} \in \{0, 1\}^k$ is a vector encoding the cast shadow information under different illuminations, $\circ$ denotes the Hadamard product, *i.e.*, an element-wise multiplication operator, and similarly, vmax is an element-wise $\max$ operator. Our goal is hence to solve this system of equation with k being as small as possible.

4. Proposed Method

Here, we propose a neural network that can take sparse illuminations as inputs and predict the surface normal with a marginal loss in accuracy. As shown in Fig. 3, the observation map can encode different lighting directions into a feature map, making the selection over the input space feasible. Then an occlusion layer renders the system robust to cast shadows. For the network itself, we have used a variation of the DenseNet [6] architecture.

To the effect of selecting the relevant illuminant directions, we employ a learnable connection table at the input of our deep network. As the network undergoes training, our method can predict the surface normal accurately by taking at input a significantly less number of illuminant directions than other methods elsewhere in the literature.

4.1. Connection Table

To the best of our knowledge, there are no learning based approaches aiming to find the relevant input feature maps for photometric stereo. However, as related to deep networks, Alvarez *et al.* [1] use sparse constraints for regularizing the training process of a deep neural network. Koray *et al.* [11] reduce the number of parameters and, hence, the complexity of deep networks using a binary connection table to "disconnect" a random set of feature maps.

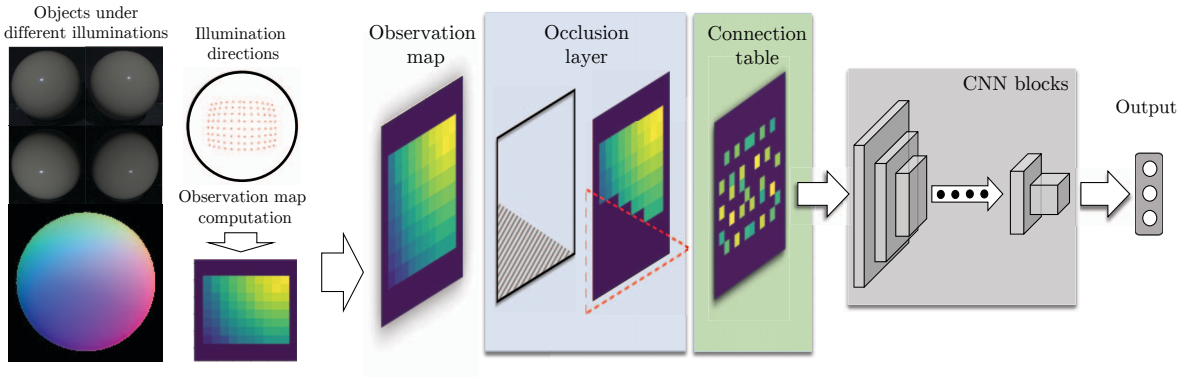

Figure 3. Overview of our network. Given the images of objects under different illuminations, we can compute its corresponding observation map at each pixel. Then we apply an occlusion layer which operates as a binary filter mask so as to zero out an area of the observation map. We then employ a learned connection table to weigh the feature map at input. After training, only a few illuminations corresponding to the non-zero weights of the connection table are necessary for our network to predict the surface normal.

Wei *et al*. [19] employ a one-dimensional connection table to perform band reduction in imaging spectroscopy. These contrast with our method, which aims at recovering a two-dimensional sparse input connection table with non-binary weights over the observation map.

4.1.1 Sparse Table Computation

Let the input connection table be denoted by $\boldsymbol{C}$. This table is applied to the observation map after an occlusion layer. For now, we concentrate on the connection table and will elaborate further on the occlusion layer later in the paper. This table effectively acts as an input selector whereby the connection weights are learned at training. Note that the connection table has the same size as the observation map, $\boldsymbol{C} \in \mathbb{R}^{w \times w}$, with all its parameters larger or equal to zero, $\boldsymbol{C}_{i,j} \geqslant 0$. The sparse observation map is then yielded by an element-wise product with the observation map after it has been "filtered" by the occlusion layer. This generates a sparse observation map as the input for the network.

The loss function of the network is then given by two terms. The first term accounts for the surface normal prediction, where we use the mean square error. The second term is a regularization one over the connection table. These yield the loss given by

$$\mathcal{L} = \|\mathbf{n} - \mathbf{n}'\|_2^2 + \lambda g(\boldsymbol{C}), \tag{4}$$

where $\mathbf{n}$ and $\mathbf{n}'$ are the predicted and ground truth surface normals, respectively, $g(\cdot)$ denotes the regularization function and λ is a hyper-parameter that controls the contribution of the regularize to the overall loss. Thus, a large λ will enforce a more sparse connection table and, conversely, a smaller λ will yield a denser one.

Note that, in this manner, the connection table can be used to select the most relevant illuminant directions at input. These directions in observation map are effectively defined by the angular intervals *i.e.*, $\Delta\theta = \Delta\varphi = \frac{2b}{w-1}$ with respect to the illuminant direction. This is an important observation since the connection table can also be interpreted as relevance for the corresponding range of the illuminant direction, which renders the method more robust to small variations in the illuminant direction.

4.1.2 Connection Table Training

Here, to make the connection table sparse, we employ the regularization function given by

$$g(\boldsymbol{C}) = \sum_{i,j} \left(2\boldsymbol{C}_{i,j} - \frac{\boldsymbol{C}_{i,j}^2}{2\alpha} \right), \tag{5}$$

where α is the maximum value of the map, *i.e.*, $\alpha = \max(\boldsymbol{C}_{i,j}) \; \forall \, i, j$.

Note that the function above is minimum as the connection table weights tend to zeros. Moreover, at each step of the back-propagation, the connection table can be updated in a straightforward manner using the derivative given by

$$\frac{\partial g(\boldsymbol{C})}{\partial \boldsymbol{C}_{i,j}} = 2 - \frac{\boldsymbol{C}_{i,j}}{\alpha}. \tag{6}$$

Since the connection table by definition is non-negative, the derivative $\frac{\partial g(\boldsymbol{C})}{\partial \boldsymbol{C}_{i,j}}$ takes values in the range of $[1, 2]$.

With the connection table in hand, we can apply a selection strategy to select the k-most important inputs to the network. To this end, we use a recurrent training scheme and, at each training operation, we apply a rank selection with exponential decay, *i.e.*, $k = \text{int}(c + \tau e^{-\beta t})$, where $t = \{0, 1, 2, \ldots\}$ is the training operation index and c, β and τ are scalar parameters. It is worth noting in passing that our choice of exponential decay in the number of table entries that are not null is reminiscent of the exponential cooling strategy in annealing methods and rank selection approaches elsewhere in the literature.

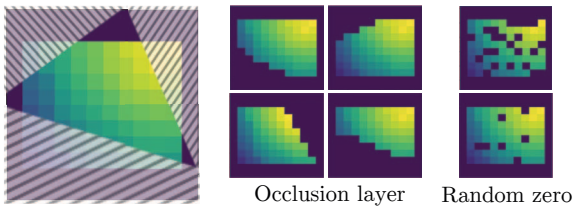

Figure 4. Observation map processed by an occlusion layer as compared to being randomly zeroed [16]. Note that the occlusion layer effectively simulates the pattern in the observation map induced by cast shadows.

4.2. Occlusion Layer

Recall that cast shadows are one of the most challenging problems in photometric stereo. It is particularly true when the number of input illuminant directions is small. Santo *et al*. [16], tackle this problem by making use of a shadow layer in their network which randomly sets some of the inputs to zero.

Despite effective, their approach does not take into account that cast shadows are often consistent with respect to the illuminant direction rather than occurring randomly. To illustrate this, we show, in Fig. 2 the observation map obtained from a shadowed area in the "harvest" object. The cast shadow is caused by the wrinkles on the cloth and shows a consistent pattern with a relatively sharp boundary. Locally, a smooth or flat occlusion results in a straight boundary while sudden changes in the object geometry often yield more abrupt boundaries.

Thus, here we employ an occlusion layer that can effectively simulate the cast shadows by randomly drawing a line across the observation map to obtain a mask that can then be used to set the corresponding entries of the map to zero. Specifically, the layer randomly selects two sides of the map, and randomly picks a point on each side. Then the line connecting these two points divides the map into two regions. The smaller of these two regions is then set to zero. This is illustrated in Fig. 4. By training the network with this occlusion layer, our proposed method can effectively learn the pattern of cast shadows and predict the surface normal more robustly.

5. Implementation

Observation Map Parameters For the observation map, we have set the window size w to 14. And the upper and lower bounds of θ and φ to $\frac{\pi}{4}$. With these parameters in hand, we then randomly select 5% of the observation maps to be put through our occlusion layer, which simulates the effect of cast shadows. Besides, we also randomly set about 5% points of observation map to be zero. We discuss further the choice of this fraction of the observation maps in Sec. 6.2.

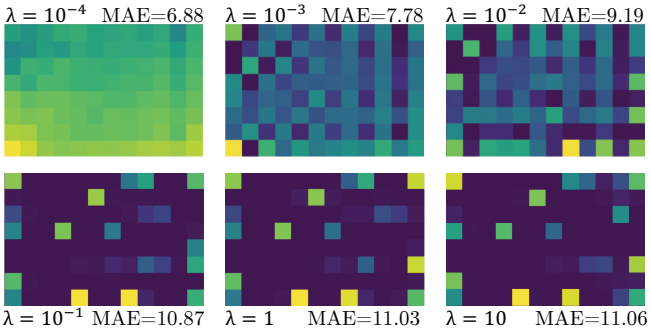

Figure 5. Visualization of the connection tables (cropped) and the mean angular errors on the validation set as a function of λ.

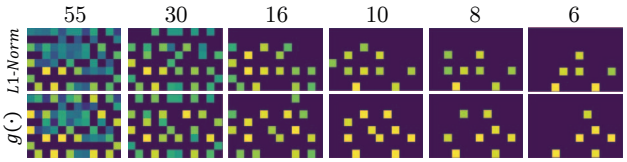

Figure 6. Connection tables (cropped) after rank selection. From left-to-right, each panel shows the table as the number of non-zero values decreases down to 6. The top row shows the maps yielded using an ℓ_1-norm regularizer. The second row shows the maps obtained using the regularization function in Eq. (5).

Pre-training For the connection table, as mentioned earlier, we perform rank selection as we recursively train the network. To this end, we initialize the table to be all ones and proceed to pre-train it. For the pre-training, we use an alternative loss function given by

$$\mathcal{L} = \|\mathbf{n} - \mathbf{n}'\|_2^2 + \lambda f(\boldsymbol{C}), \tag{7}$$

where $f(\cdot)$ is given by the ℓ_1-norm regularizer.

The reason for using the ℓ_1-norm at pre-training and the regularizer $g(\cdot)$ later is due to the fact that, in contrast with the behavior of $g(\cdot)$, the ℓ_1-norm decreases as the magnitude of the connection table entries tends to zero. Thus, we have employed the ℓ_1-norm to provide a good initialization for the training and employed the regularizer in Eq. (5) to favor values of the table that are either close to unity or to zero. Moreover, nonetheless λ can control the sparsity of the connection table when the ℓ_1-norm is used, it quickly becomes ineffective as it increases. To illustrate this behavior, we show, in Fig. 5, the connection table obtained with different values of λ at pre-training. We can see from the figure that, as λ increases beyond 10^{-1}, the connection table shows virtually no change.

Connection Table Rank Selection As mentioned above, we commence by pre-training the network. Here, we have set $\lambda = 10^{-3}$ for the pre-training operation and then applied, rank selection to the connection table over a set of subsequent, recurrent training steps. As described earlier,

Methods \ Inputs	96	16	10	8	6
Ours-$g(\cdot)$	8.43	9.66	**10.02**	**10.39**	**12.16**
Ours-L1		10.29	11.06	11.33	12.57
Ours-Random		10.50	11.39	11.85	12.58
PS-FCN [3]	8.39	**9.37**	10.51	11.42	12.54
CNN-PS [8]	**7.20**	10.49	14.34	19.50	30.28
L1-RES [9]	14.08	15.47	16.37	16.84	18.45
Baseline [20]	15.39	16.65	17.31	17.47	18.60

Table 1. Performance on the DiLiGenT dataset with different number of input light directions (see Sec. 6.1 for more details).

in this manner we select the top k largest weights, with k steadily decreasing over the training process from all illuminations in the dataset (96) down to 6 following the sequence 55, 30, 16, 10, 8, 6 in k.

In Fig. 6 we present the connection tables at each training step after the rank selection has been affected. In the left-hand panel, we show the connection table obtained at the first training step after the 55 most significant elements of the learned table are selected and the rest are set to zero. The second panel, from left-to-right, shows the table where the 30 most significant entries have been selected after the second training operation and so on. For all training operations, we have set $\lambda = 10^{-3}$. Note this is consistent with our pre-training operation.

5.1. Dataset

To train our network, we generate a synthetic dataset using the 10 shapes from the Blobby Shape Dataset [10] and 100 reflectances from MERL BRDF dataset [13]. To do this, we randomly select 8 BRDFs to repeatedly render each object using 144 randomly distributed light directions. For the sake of consistency, we have used the same range of light directions as that used in the DiLiGenT [17] dataset.

We choose 9 shapes for training and the remaining one for validation. Thus, our synthetic dataset comprises approximately $9 \times 10^4 \times 8 = 720,000$ observation maps for training and about $80,000$ maps for validation. For the testing, we use the DiLiGenT [17] dataset, which is a real-world public benchmark containing the images of 10 distinct objects. Each object is observed under 96 different known light directions, of which the horizontal direction ranges from $-45°$ to $45°$ and vertically from $-30°$ to $30°$ over a hemisphere centered at the viewing direction. The ground truth surface normals have been acquired using a laser scanner and are available for quantitative evaluation.

6. Experiments

For all our experiments we have trained our network using Adam [12] optimizer with the default setting. The network is implemented by Keras with Tensorflow as the backend on a workstation using Ubuntu-14 with an NVIDIA GTX 1080Ti 11G.

Training	Methods	0%	5%	10%	15%
Synthesis	Occlusion layer	11.42	**10.42**	10.68	10.57
	Random zeroing		10.75	10.87	10.94
Real-world	Occlusion layer	8.28	7.99	8.04	**7.85**
	Random zeroing		8.33	8.13	8.37

Table 2. Comparison of our occlusion layer and random zeroing [16] on the DiLiGenT dataset for 10 illumination directions.

For the quantification of the error, in all our experiments we have used the mean angular error (MAE) for surface normal evaluation. Here, we compare the results by our method to those obtained using a number of alternatives [16, 3, 8, 9] that can be used with sparse inputs. We also illustrate the effectiveness of our loss function and our occlusion layer and show results on pixel-wise normal estimation.

6.1. Effectiveness of the Regularization Function

To illustrate the effectiveness of our regularizer $g(\cdot)$ in the loss function, we now compare the performance of our network when other regularization functions are used. In Table 1, "Ours-$g(\cdot)$" denotes the network trained using the function $g(\cdot)$ described in Eq. (5). "Ours-L1" corresponds to the results yielded by our network when the ℓ_1-norm regularizer is used as an alternative to $g(\cdot)$. Finally, the "Ours-Random" corresponds to the case where no regularizer is included and the loss function and the connection table is randomly initialized rather than pre-trained. All the other methods are tested with randomly selected inputs with all the results shown are averaged over 10 trails. As we can see from the table, our model with the sparsity regularization function $g(\cdot)$ performs the best and clearly outperforms the ℓ_1-norm and random initialization.

6.2. Evaluation of the Occlusion Layer

Now we turn our attention to the effects of the occlusion layer. To this end, we have trained the network with both, our occlusion layer and that proposed by Santo *et al*. [16] and set $k = 10$, *i.e*., used 10 light directions at input. Here, we have trained the network on both, our synthetic dataset and DiLiGenT.

In Table 2, we show the effects of our occlusion layer and that proposed by Santo *et al*. [16]. Note that in our synthetic dataset there are no cast shadows. This contrasts with the real-world objects in the DiLiGenT dataset, which exhibit a diversity of shadows and inter-reflections. The purpose of the comparison shown here is not only showing the benefits of our occlusion layer as compared to that in [16] but also showing that it can benefit the training of real-world and synthetic datasets alike.

Also, note that the percentages in the table have a different meaning for both approaches. For our occlusion layer, the percentages indicate how many of the observation maps are "masked" with our occlusion layer during training. While the percentages of randomly zeroing follow the

Figure 7. Performance comparison of our occlusion layer with respect to random zeroing [16]. The blue color in the panels corresponds to the area where the occlusion layer outperforms the random zeroing. The converse applies to the red regions. We can see that, in the "Cat" image, the occlusion layer is better in most of the shoulder region. In the "Buhhda" image, which is quite textured, our method outperforms the random zeroing on all the cloth wrinkles. All of these areas correspond to cast shadows.

	methods	ball	cat	pot1	bear	pot2	buddha	goblet	reading	cow	harvest	Avg.
10 Inputs	Ours	**3.97**	**6.69**	**7.30**	8.73	**9.74**	11.36	**10.46**	**14.37**	**10.19**	**17.33**	**10.02**
	PS-FCN[3]	4.02	8.30	10.14	**7.18**	9.85	**9.79**	11.58	15.03	10.51	18.70	10.51
	CNN-PS[8]	9.11	11.71	13.23	14.08	14.65	14.58	15.48	16.99	14.04	19.56	14.34
	L1-RES[9]	4.92	9.68	10.69	9.58	17.64	14.53	18.39	18.23	27.01	33.05	16.37
	Baseline [20]	5.09	9.66	11.32	11.59	18.03	16.25	19.97	19.86	27.90	33.41	17.31
8 Inputs	Ours	**3.65**	**7.07**	**8.11**	**8.91**	**9.92**	12.03	**10.67**	**15.27**	**10.14**	**18.15**	**10.39**
	PS-FCN[3]	5.14	9.47	10.65	9.16	10.05	**9.66**	12.39	16.47	11.14	20.05	11.42
	CNN-PS[8]	22.74	21.06	18.87	16.89	19.18	19.95	18.62	17.77	16.04	23.90	19.50
	L1-RES[9]	4.70	9.57	11.96	10.10	17.33	15.62	19.07	18.96	28.23	32.82	16.84
	Baseline [20]	5.50	9.89	12.01	11.57	17.49	16.53	19.73	20.41	28.64	32.96	17.47

Table 3. Results on the DiLiGenT dataset with 10 and 8 input illumination directions. All the alternatives are tested with randomly selected inputs. All the mean angular errors shown correspond to the average performance over 10 trails.

setting in [16], indicating how many pixels at each input map are randomly set to zero.

As we can see from the table, our occlusion layer performs better when 5% of the input maps are occluded. For the random zeroing strategy, the best performance appears to be also about 5%. Nonetheless, regardless of the percentage used, the lowest error is always achieved by our occlusion layer. Finally, we show the error map for a number of sample objects in the DiLiGenT dataset in Fig. 7. The error maps shown correspond to the difference between the MAE yielded by our occlusion layer and random zeroing. Hence the negative areas, *i.e.*, blue regions in Fig. 7 are those where our occlusion layer outperforms the alternative. From the figure, it is clear that the occlusion layer performs much better at most of the rugged and wrinkled areas, where cast shadows are stronger. This is consistent with the notion that our occlusion layer can help the network cope with cast shadows.

6.3. Benchmark Comparison Under Sparse Inputs

We commence by comparing our method with other state-of-art approaches using the DiLiGenT dataset. For the results yielded by the alternatives in sparse illuminations, and in contrast with our method, we have randomly selected input illuminant directions up to the k under consideration. Note that this is in accordance with the sparse input setting proposed by the authors.

The MAE for the dataset over 10 trails is shown in Table 1. Note that in Table 1, all the methods except PS-FCN [3] estimate surface normal in a pixel-wise fashion. Since pixel-wise methods naturally discard the connection between pixels and shape, a sparse input setting like the one used here is expected to decrease the performance dramatically. As we can see, the performance of CNN-PS [8] decreases from 7.20 to 30.28 at 6 inputs. In comparison, our method do not exhibit such loss in performance, outperforming the other pixel-wise methods. We can see that from 16 inputs down, our method starts to show its ability to select the illuminant directions to avoid an overly loss in performance. At 10 and 8 input directions, our method shows a large improvement in performance with respect to the other methods. Moreover, our method shows a decrease in performance as low as 1.25 degrees on MAE from 96 to 16 input directions and only 0.3 degrees from 16 to 10 inputs.

Also, note that PS-FCN [3] is a dense normal estimation method which takes the whole image into account. Hence, PS-FCN [3] naturally has advantages on sparse inputs as it can utilize the structural information across pixels. Nonetheless, our method can perform better than PS-FCN [3] in highly sparse illumination settings with 10 or less light source directions. In Table 3, we also show the performance of our method and the alternatives on each object in DiLiGenT dataset for 10 and 8 inputs. In Fig. 8, we show the surface normal and error maps of the "Goblet" object from DiLiGenT.

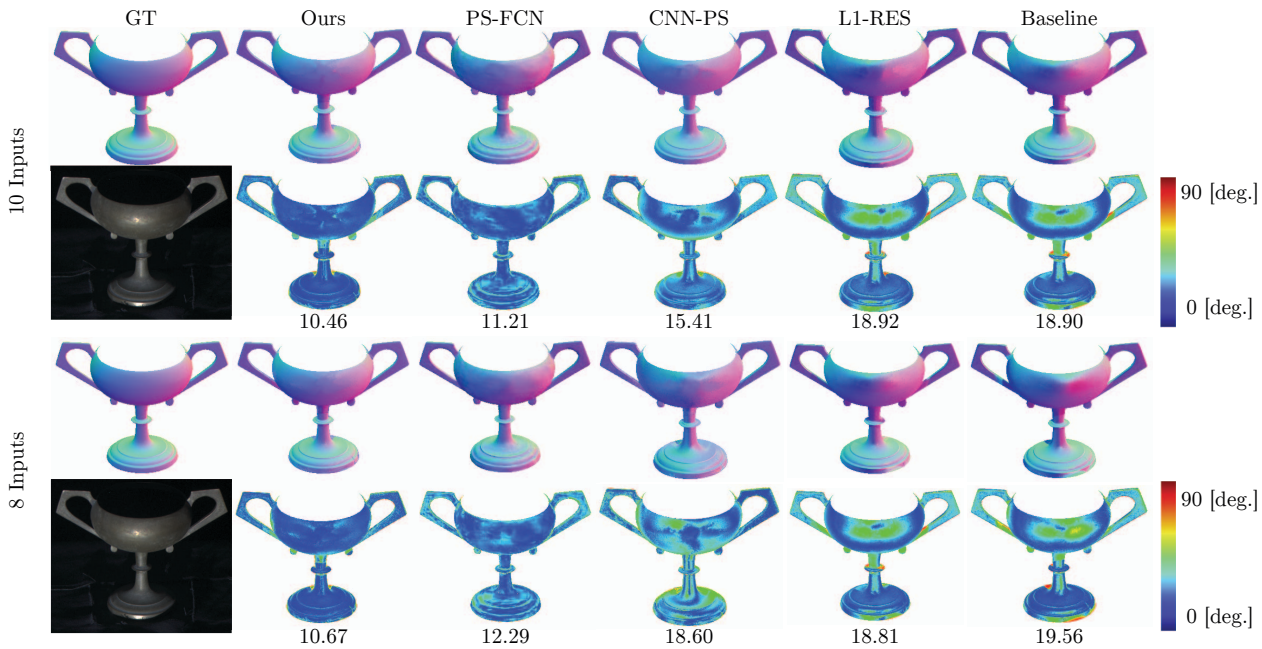

Figure 8. Performance of "Goblet" in DiLiGenT dataset in only 10 and 8 inputs.

Light-Config	Ours	PS-FCN[3]	CNN-PS[8]	L1-RES[9]	Baseline[20]
Random	**10.02**	10.51	14.34	16.37	17.31
Ours		11.35	13.02	15.83	17.12
Optimal [4]		**8.73**	13.35	15.50	16.57

Table 4. Mean angular errors on the DiLiGenT benchmark using the same 10 lights directions obtained using random sampling (Random), our approach (Ours) and the method in [4].

Table 4 shows the mean angular errors for our and other methods under similar light configurations. The first row (Random) shows the result when 10 lights are randomly sampled from a uniform distribution over the hemisphere. The second row (Ours) is the result under the light directions that correspond to the non-zero entries of the connection map learned by our method. The third row (Optimal) correponds to the optimal light configuration proposed by Drbohlav and Chantler [4]. Note that, these light directions do not precisely match those provided in DiLiGenT, we use the closest ones in the DiLiGenT dataset for testing. Our method outperforms others except for the PS-FCN [3] under the optimal light configuration.

Pixel-wise normal estimation. Finally, we present a comparison between pixel-wise and dense normal estimation. In Fig. 9, we show the results yielded by our method and PS-FCN [3] on a challenging synthetic object. In Fig. 9, we use one object from Blobby [10] which has been excluded from our training dataset and render it by randomly selecting a BRDF function from MERL [13] for each pixel. We then test both methods using only 8 inputs. Note that PS-FCN [3] suffers from the relationship between the BRDF and the geometry of the object whereas our method, despite some noise, still recovers a surface normal in good accordance with the object itself, outperforming the PS-FCN [3].

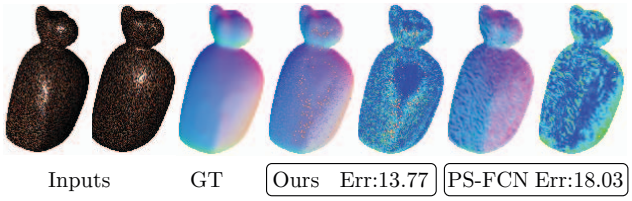

Figure 9. Performance of our method and PS-FCN for a spatially varying BRDF when 8 input light source directions are shown.

7. Conclusion

In this paper, we have proposed a deep learning approach which applies a connection table that can select those illuminant directions that are most relevant to the surface normal prediction process. To this end, we have employed a connection table and showed how, once the model is trained, it can efficiently estimate the surface normal only given a much-reduced number of input images without loss in accuracy. Furthermore, we have tackled the problem of cast shadows explicitly by introducing an occlusion layer into the network. We have illustrated the utility of our method for photometric stereo by comparing our results with those yielded by a number of alternatives.

Acknowledgments: Part of this work was supported by JST CREST Grant Number JP17942373, Japan.

References

[1] Jose M Alvarez and Mathieu Salzmann. Learning the number of neurons in deep networks. In *Advances in Neural Information Processing Systems*, pages 2270–2278, 2016. 3

[2] Vasileios Argyriou, Stefanos Zafeiriou, Barbara Villarini, and Maria Petrou. A sparse representation method for determining the optimal illumination directions in photometric stereo. *Signal Processing*, 93(11):3027–3038, 2013. 1

[3] Guanying Chen, Kai Han, and Kwan-Yee K Wong. Ps-fcn: A flexible learning framework for photometric stereo. In *European Conference on Computer Vision*, pages 3–19. Springer, 2018. 1, 6, 7, 8

[4] Ondrej Drbohlav and Mike Chantler. On optimal light configurations in photometric stereo. In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1707–1712. IEEE, 2005. 8

[5] Aaron Hertzmann and Steven M Seitz. Example-based photometric stereo: Shape reconstruction with general, varying brdfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1254–1264, 2005. 2

[6] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 2261–2269. IEEE, 2017. 3

[7] Zhuo Hui and Aswin C Sankaranarayanan. Shape and spatially-varying reflectance estimation from virtual exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(10):2060–2073, 2017. 2

[8] Satoshi Ikehata. Cnn-ps: Cnn-based photometric stereo for general non-convex surfaces. In *European Conference on Computer Vision*, pages 3–19. Springer, 2018. 1, 2, 3, 6, 7, 8

[9] Satoshi Ikehata, David Wipf, Yasuyuki Matsushita, and Kiyoharu Aizawa. Robust photometric stereo using sparse regression. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 318–325. IEEE, 2012. 2, 6, 7, 8

[10] Micah K Johnson and Edward H Adelson. Shape estimation in natural illumination. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2553–2560. IEEE, 2011. 6, 8

[11] Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann L Cun. Learning convolutional feature hierarchies for visual recognition. In *Advances in neural information processing systems*, pages 1090–1098, 2010. 3

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[13] Wojciech Matusik, Hanspeter Pfister, Matt Brand, and Leonard McMillan. A data-driven reflectance model. *ACM Transactions on Graphics*, 22(3):759–769, July 2003. 6, 8

[14] Yasuhiro Mukaigawa, Yasunori Ishii, and Takeshi Shakunaga. Analysis of photometric factors based on photometric linearization. *JOSA A*, 24(10):3326–3334, 2007. 2

[15] Yvain Quéau, Tao Wu, François Lauze, Jean-Denis Durou, and Daniel Cremers. A non-convex variational approach to photometric stereo under inaccurate lighting. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 350–359. IEEE, 2017. 2

[16] Hiroaki Santo, Masaki Samejima, Yusuke Sugano, Boxin Shi, and Yasuyuki Matsushita. Deep photometric stereo network. In *Computer Vision Workshop (ICCVW), 2017 IEEE International Conference on*, pages 501–509. IEEE, 2017. 1, 2, 5, 6, 7

[17] Boxin Shi, Zhipeng Mo, Zhe Wu, Dinglong Duan, Sai Kit Yeung, and Ping Tan. A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018. 1, 2, 3, 6

[18] Tatsunori Taniai and Takanori Maehara. Neural inverse rendering for general reflectance photometric stereo. In *International Conference on Machine Learning*, pages 4864–4873, 2018. 2

[19] Ran Wei, Antonio Robles-Kelly, and José Álvarez. Context free band reduction using a convolutional neural network. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 86–96. Springer, 2018. 4

[20] Robert J Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):191139, 1980. 2, 6, 7, 8

[21] Lun Wu, Arvind Ganesh, Boxin Shi, Yasuyuki Matsushita, Yongtian Wang, and Yi Ma. Robust photometric stereo via low-rank matrix completion and recovery. In *Asian Conference on Computer Vision*, pages 703–717. Springer, 2010. 1, 2

[22] Tai-Pang Wu and Chi-Keung Tang. Photometric stereo via expectation maximization. *IEEE transactions on pattern analysis and machine intelligence*, 32(3):546–560, 2010. 2

[23] Zexiang Xu, Kalyan Sunkavalli, Sunil Hadap, and Ravi Ramamoorthi. Deep image-based relighting from optimal sparse samples. *ACM Transactions on Graphics (TOG)*, 37(4):126, 2018. 2