CNN-PS: CNN-based Photometric Stereo for General Non-Convex Surfaces

Satoshi Ikehata

National Institute of Informatics, Tokyo, Japan
sikehata@nii.ac.jp

Abstract. Most conventional photometric stereo algorithms inversely solve a BRDF-based image formation model. However, the actual imaging process is often far more complex due to the global light transport on the non-convex surfaces. This paper presents a photometric stereo network that directly learns relationships between the photometric stereo input and surface normals of a scene. For handling unordered, arbitrary number of input images, we merge all the input data to the intermediate representation called *observation map* that has a fixed shape, is able to be fed into a CNN. To improve both training and prediction, we take into account the rotational pseudo-invariance of the observation map that is derived from the isotropic constraint. For training the network, we create a synthetic photometric stereo dataset that is generated by a physics-based renderer, therefore the global light transport is considered. Our experimental results on both synthetic and real datasets show that our method outperforms conventional BRDF-based photometric stereo algorithms especially when scenes are highly non-convex.

Keywords: photometric stereo, convolutional neural networks

1 Introduction

In 3-D computer vision problems, the input data is often *unstructured* (*i.e.*, the number of input images is varying and the images are unordered). A good example is the multi-view stereo problem where the scene geometry is recovered from unstructured multi-view images. Due to this unstructuredness, 3-D reconstruction from multiple images less relied on the supervised learning-based algorithms except for some structured problems such as binocular stereopsis [1] and two-view SfM [2] whose number of input images is always fixed. However, recent advances in deep convolutional neural network (CNN) have motivated researchers to address unstructured 3-D computer vision problems with deep neural networks. For instance, a recent work from Kar *et al.* [3] presented an end-to-end learned system for the multi-view stereopsis while Kim *et al.* [4] presented a learning-based surface reflectance estimation from multiple RGB-D images. Either work intelligently merged all the unstructured input to a structured, intermediate representation (*i.e.*, 3-D feature grid [3] and 2-D hemispherical image [4]).

Photometric stereo is another 3-D computer vision problem whose input is unstructured, where surface normals of a scene are recovered from appearance variations under different illuminations. Photometric stereo algorithms typically solved an inverse problem of the pointwise image formation model which was based on the Bidirectional Reflectance Distribution Function (BRDF). While effective, a BRDF-based image formation model generally cannot account the global illumination effects such as shadows and inter-reflections, which are often problematic to recover non-convex surfaces. Some algorithms attempted the robust outlier rejection to suppress the non-Lambertian effects [5–8], however the estimation failed when the non-Lambertian observation was dominant. This limitation inevitably occurs due to the fact that multiple interactions of light and a surface are difficult to be modeled in a mathematically tractable form.

To tackle this issue, this paper presents an end-to-end CNN-based photometric stereo algorithm that learns the relationships between surface normals and their appearances without physically modeling the image formation process. For better scalability, our approach is still pixelwise and rather inherit from conventional robust approaches [5–8], which means that we learn the network that automatically "neglects" the global illumination effects and estimate the surface normal from "inliers" in the observation. To achieve this goal, we will train our network on as much as possible synthetic patterns of the input that is "corrupted" by global effects. Images are rendered with different complex objects under the diverse material and illumination condition.

Our challenge is to apply the deep neural network to the photometric stereo problem whose input is unstructured. In similar with recent works [3, 4], we merge all the photometric stereo data to an intermediate representation called *observation map* that has a fixed shape, therefore is naturally fed to a standard CNN. As many photometric stereo algorithms were, our work is also primarily concerned with isotropic materials, whose reflections are invariant under rotation about the surface normal. We will show that this isotropy can be taken advantages of in a form of the *rotational pseudo-invariance* of the observation map for both augmenting the input data and reducing the prediction errors. To train the network, we create a synthetic photometric stereo dataset (*CyclesPS*) by leveraging the physics-based Cycles renderer [9] to simulate the complex global light transport. For covering diverse real-world materials, we adopt the Disney's principled BSDF [10] that was proposed for artists to render various scenes by controlling small number of parameters.

We evaluate our algorithm on the DiLiGenT Photometric Stereo Dataset [11] which is a real benchmark dataset containing images and calibrated lightings. We compare our method against conventional photometric stereo algorithms [12–15, 5, 6, 16, 7, 17, 18, 8, 19–21] and show that our end-to-end learning-based algorithm most successfully recovers the non-convex, non-Lambertian surfaces among all the algorithms concerned.

The summary of contributions is following:
(1) We firstly propose a supervised CNN-based calibrated photometric stereo algorithm that takes unstructured images and lighting information as input.

(2) We present a synthetic photometric stereo dataset (*CyclesPS*) with a careful injection of the global illumination effects such as cast shadows, inter-reflections. (3) Our extensive evaluation shows that our method performs best on the DiLi-GenT benchmark dataset [11] among various conventional algorithms especially when the surfaces are highly non-convex and non-Lambertian.

Henceforth we rely on the classical assumptions on the photometric stereo problem (*i.e.*, fixed, linear orthographic camera and known directional lighting).

2   Related Work

Diverse appearances of real world objects can be encoded by a BRDF ρ, which relates the observed intensity I_j to the associated surface normal $\boldsymbol{n} \in \mathbb{R}^3$, the j-th incoming lighting direction $\boldsymbol{l}_j \in \mathbb{R}^3$, its intensity $L_j \in \mathbb{R}$, and the outgoing viewing direction $\boldsymbol{v} \in \mathbb{R}^3$ via

$$I_j = L_j \rho(\boldsymbol{n}, \boldsymbol{l}_j, \boldsymbol{v}) \max\left(\boldsymbol{n}^\top \boldsymbol{l}_j, 0\right) + \epsilon_j, \tag{1}$$

where $\max\left(\boldsymbol{n}^\top \boldsymbol{l}_j, 0\right)$ accounts for attached shadows and ϵ_j is an additive error to the model. Eq. (1) is generally called *image formation model*. Most photometric stereo algorithms assumed the specific shape of ρ and recovered the surface normals of a scene by inversely solving Eq. (1) from a collection of observations under m different lighting conditions ($j \in 1, \cdots, m$). All the effects that are not represented by a BRDF (image noises, cast shadows, inter-reflections and so on) are typically put together in ϵ_j. Note that when the BRDF is Lambertian and the additive error is removed, it is simplified to the traditional Lambertian image formation model [12].

Since Woodham firstly introduced the Lambertian photometric stereo algorithm, the extension of its work to non-Lambertian scenes has been a problems of significant interest. Photometric stereo approaches to dealing with non-Lambertian effects are mainly categorized into four classes: (a) robust approach, (b) reflectance modeling with non-Lambertian BRDF, (c) example-based reflectance modeling and (d) learning-based approach.

Many photometric stereo algorithms recover surface normals of a scene via a simple diffuse reflectance modeling (*e.g.*, Lambertian) while treating other effects as outliers. For instance, Wu *et al.* [5] have proposed a rank-minimization based approach to decompose images into the low-rank Lambertian image and non-Lambertian sparse corruptions. Ikehata *et al.* extended their method by constraining the rank-3 Lambertian structure [6] (or the general diffuse structure [7]) for better computational stability. Recently, Queau *et al.* [8] have presented a robust variational approach for inaccurate lighting as well as various non-Lambertian corruptions. While effective, a drawback of this approach is that if it were not for dense diffuse inliers, the estimation fails.

Despite their computational complexity, various algorithms arrange the parametric or non-parametric models of non-Lambertian BRDF. In recent years, there has been an emphasis on representing a material with a small number

of fundamental BRDF. Goldman *et al.* [22] have approximated each fundamental BRDF by the Ward model [23] and Alldrin *et al.* [13] later extended it to non-parametric representation. Since the high-dimensional ill-posed problem may cause the instability of the estimation, Shi *et al.* [18] presented a compact biquadratic representation of isotropic BRDF. On the other hand, Ikehata *et al.* [17] introduced the sum-of-lobes isotropic reflectance model [24] to account all frequencies in isotropic observations. For improving the efficiency of the optimization, Shen *et al.* [25] presented a kernel regression approach, which can be transformed to an eigen decomposition problem. This approach works well as far as a resultant image formation model is correct without model outliers.

A few amount of photometric stereo algorithms are grouped into the *example-based* approach, which takes advantages of the surface reflectance of objects with known shape, captured under the same illumination environment with the target scene. The earliest example-based approach [26] requires a reference object whose material is exactly same with that of target object. Hertzmann *et al.* [27] have eased this restriction to handle uncalibrated scenes and spatially varying materials by assuming that materials can be expressed as a small number of basis materials. Recently, Hui *et al.* [20] presented an example-based method without a physical reference object by taking advantages of virtual spheres rendered with various materials. While effective, this approach also suffers from model outliers and has a drawback that the lighting configuration of the reference scene must be taken over at the target scene.

Machine learning techniques have been applied in a few very recent photometric stereo works [21, 19]. Santo *et al.* [19] presented a supervised learning-based photometric stereo method using a neural network that takes as input a normalized vector where each element corresponds to an observation under specific illumination. A surface normal is predicted by feeding the vector to one dropout layer and adjacent six dense layers. While effective, this method has limitation that lightings remain the same between training and test phases, making it inapplicable to the unstructured input. One another work by Taniai and Maehara [21] presented an unsupervised learning framework where surface normals and BRDFs are predicted by the network trained by minimizing reconstruction loss between observed and synthesized images with a rendering equation. While their network is invariant to the number and permutation of the images, the rendering equation is still based on a point-wise BRDF and intolerant to the model outliers. Furthermore, they reported slow running time (*i.e.*, 1 hour to do 1000 SGD iterations for each scene) due to its self-supervision manner.

In summary, there is still a constant struggle in the design of the photometric stereo algorithm among its complexity, efficiency, stability and robustness. Our goal is to solve this dilemma. Our end-to-end learning-based algorithm builds upon the deep CNN trained on synthetic datasets, abandoning the modeling of complicated image formation process. Our network accepts the unstructured input (*i.e.*, our network is invariant to both number and order of input images) and works for various real-world scenes where non-Lambertian reflections are intermingled with global illumination effects.
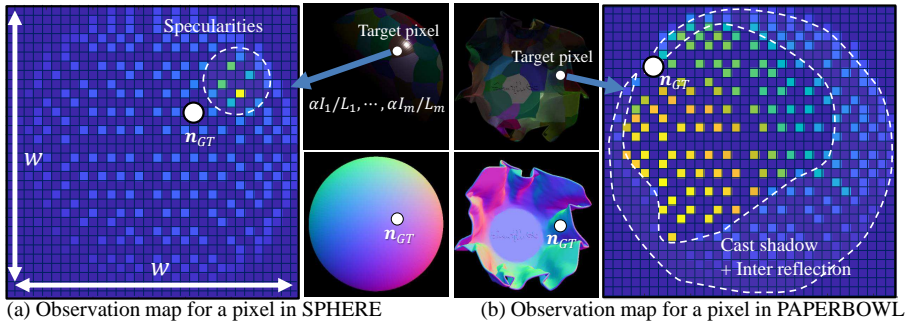
Fig. 1. We project pairs of images and lightings to a fixed-size observation map based on the bijective mapping of a light direction from a hemisphere to the 2-D coordinate system perpendicular to the viewing axis. This figure shows observation maps for (a) a point on a smooth convex surface and (b) a point on a rough non-convex surface. We also projected the true surface normal at the point onto the same coordinate system of the observation map for reference.

3   Proposed Method

Our goal is to recover surface normals of a scene of (a) spatially-varying isotropic materials and with (b) global illumination effects (*e.g.*, shadows and interreflections) (c) where the scene is illuminated by unknown number of lights. To achieve this goal, we propose a CNN architecture for the calibrated photometric stereo problem which is invariant to both the number and order of input images. The tolerance to global illumination effects is learned from the synthetic images of non-convex scenes rendered with the physics-based renderer.

3.1   2-D observation map for unstructured photometric stereo input

We firstly present the *observation map* which is generated by a pixelwise hemispherical projection of observations based on known lighting directions. Since a lighting direction is a vector spanned on a unit hemisphere, there is a bijective mapping from $\boldsymbol{l}_j \triangleq [l_x^j \ l_y^j \ l_z^j]^\top \in \mathbb{R}^3$ to $[l_x^j \ l_y^j]^\top \in \mathbb{R}^2$ (s.t., $l_x^2 + l_y^2 + l_z^2 = 1$) by projecting a vector onto the x-y coordinate system which is perpendicular to a viewing direction (*i.e.*, $\boldsymbol{v} = [0\ 0\ 1]$).[2] Then we define an observation map $O \in \mathbb{R}^{w \times w}$ as

$$O_{\text{int}(w(l_x+1)/2),\text{int}(w(l_y+1)/2)} = \alpha I_j / L_j \ \ \forall j \in 1, \cdots, m, \tag{2}$$

where "int" is an operator to round a floating value to an integer and α is a scaling factor to normalize data (*i.e.*, we simply use $\alpha = \max L_j / I_j$). Once all the observations and lightings are stored in the observation map, we take it as an input of the CNN. Despite its simplicity, this representation has three

---

[2] We preliminarily tried the projection on the spherical coordinate system (θ, ϕ), but the performance was worse than one on the standard x-y coordinate system.
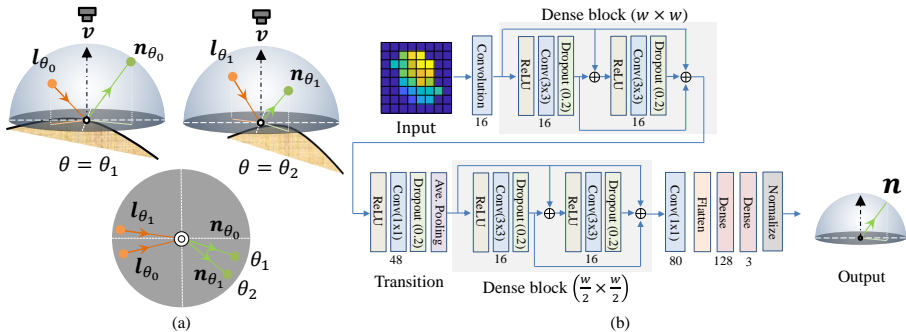
Fig. 2. (a) Isotropy guarantees that the appearance of a surface from v is invariant of the rotation of l and n around the view axis. (b) Our network architecture is a variation of DenseNet [28] that outputs a normalized surface normal from a 32×32 observation map. Numbers of the filter are presented below each layer.

major benefits. First, its shape is independent of the number and size of input images. Second, the projection of observations is order-independent (*i.e.*, the observation map does not change when swapping i-th and j-th images). Third, it is unnecessary to explicitly feed the lighting information into the network.

Fig. 1 illustrates examples of the observation map of two objects namely SPHERE and PAPERBOWL, one is purely convex and the other is highly non-convex. Fig. 1-(a) indicates that the target point could be on the convex surface since the values of the observation map gradually decrease to zero as the light direction is going apart from the true surface normal (n_{GT}). The local concentration of large intensity values also indicates the narrow specularity on the smooth surface. On the other hand, the abrupt change of values in Fig. 1-(b) evidences the presence of cast shadows or inter-reflections on the non-convex surface. Since there is no local concentration of intensity values, the surface is likely to be rough. In this way, an observation map reasonably encodes the geometry, material and behavior of the light at around a surface point.

3.2 Rotation pseudo-invariance for the isotropy constraint

An observation map O is sparse in a general photometric stereo setup (*e.g.*, assuming that $w = 32$ and we have 100 images as input, the ratio of non-zero entries in O is about 10%). The missing data is generally considered problematic as CNN input and often interpolated [4]. However, we empirically found that smoothly interpolating missing entries degrades the performance since an observation map is often non-smooth and zero values have an important meaning (*i.e.*, shadows). Therefore we alternatively try to improve the performance by taking into account the isotropy of the material.

Many real-world materials exhibit identically same appearance when the surface is rotated along a surface normal. The presence of this behavior is referred to as isotropy [29, 30]. Isotropic BRDFs are parameterized in terms of three values

instead of four [31] as

$$\rho = f(\boldsymbol{n}^\top \boldsymbol{l}, \boldsymbol{n}^\top \boldsymbol{v}, \boldsymbol{l}^\top \boldsymbol{v}), \tag{3}$$

where f is an arbitrary reflectance function.[3] Combining Eq. (3) with Eq. (1), we get following image formation model.

$$I = Lf(\boldsymbol{n}^\top \boldsymbol{l}, \boldsymbol{n}^\top \boldsymbol{v}, \boldsymbol{l}^\top \boldsymbol{v}) \max(\boldsymbol{n}^\top \boldsymbol{l}, 0). \tag{4}$$

Note that lighting index and model error are omitted for brevity. Let's consider the rotation of surface normal $\boldsymbol{n}$ and lighting direction $\boldsymbol{l}$ around the z-axis (i.e., viewing axis) as $\boldsymbol{n}' = [(R[n_x \ n_y]^\top)^\top \ n_z]^\top, \boldsymbol{l}' = [(R[l_x \ l_y]^\top)^\top \ l_z]^\top$ where $\boldsymbol{n} \triangleq [n_x \ n_y \ n_z]^\top$ and $R \in SO(2)$ is an arbitrary rotation matrix. Then,

$$\boldsymbol{n'}^\top \boldsymbol{l'} = [(R[n_x \ n_y]^\top)^\top \ n_z][(R[l_x \ l_y]^\top)^\top \ l_z]^\top \tag{5}$$
$$= [n_x \ n_y]R^\top R[l_x \ l_y]^\top + n_z l_z = \boldsymbol{n}^\top \boldsymbol{l},$$

$$\boldsymbol{n'}^\top \boldsymbol{v'} = [(R[n_x \ n_y]^\top)^\top \ n_z][0 \ 0 \ 1]^\top = n_z = \boldsymbol{n}^\top \boldsymbol{v}, \tag{6}$$

$$\boldsymbol{l'}^\top \boldsymbol{v'} = [(R[l_x \ l_y]^\top)^\top \ l_z][0 \ 0 \ 1]^\top = l_z = \boldsymbol{l}^\top \boldsymbol{v}. \tag{7}$$

Feeding them into Eq. (4) gives following equation,

$$I = Lf(\boldsymbol{n'}^\top \boldsymbol{l'}, \boldsymbol{n'}^\top \boldsymbol{v}, \boldsymbol{l'}^\top \boldsymbol{v}) \max(\boldsymbol{n'}^\top \boldsymbol{l'}, \boldsymbol{0}) \tag{8}$$
$$= Lf(\boldsymbol{n}^\top \boldsymbol{l}, \boldsymbol{n}^\top \boldsymbol{v}, \boldsymbol{l}^\top \boldsymbol{v}) \max(\boldsymbol{n}^\top \boldsymbol{l}, 0).$$

Therefore, the rotation of lighting and surface normal around z-axis does not change the appearance as illustrated in Fig. 2-(a). Note that this theorem holds even for the indirect illumination in non-convex scenes by rotating all the geometry and environment illumination around the viewing axis. This result is important for our CNN-based algorithm. We suppose that a neural network is a mapping function $g : x \mapsto g(x)$ that maps x (i.e., a set of images and lightings) to $g(x)$ (i.e., a surface normal) and r is a rotation operator of lighting/normal at the same angle around z-axis. From Eq. (8), we get $r(g(x)) = g(r(x))$. We call this relationship as *rotational pseudo-invariance* (the standard rotation invariance is $g(x) = g(r(x))$). Note that this rotational pseudo-invariance is also applied on the observation map since the rotation of lightings around the viewing axis results in the rotation of the observation map around the z-axis[4].

We constrain the network with the rotational pseudo-invariance in the similar manner that the rotation invariance is achieved. Within the CNN framework, two approaches are generally adopted to encode the rotation invariance. One is applying rotations to the input image [33] and the other is applying rotations to the convolution kernels [34]. We adopt the first strategy due to its simplicity. Concretely, we augment the training set with many rotated versions of lightings and surface normal, which allows the network to learn the invariance without explicitly enforcing it. In our implementation, we rotate the vectors at 10 regular intervals from 0 to 360.

---

[3] Note that there are other parameterizations of an isotropic BRDF [32].

[4] Strictly speaking, we rotate the lighting directions instead of the observation map itself. Therefore, we do not need to suffer from the boundary issue unlike the standard rotational data augmentation.

Input images   2-D Observation maps   Surface normal prediction via CNN   Recovered surface normal map
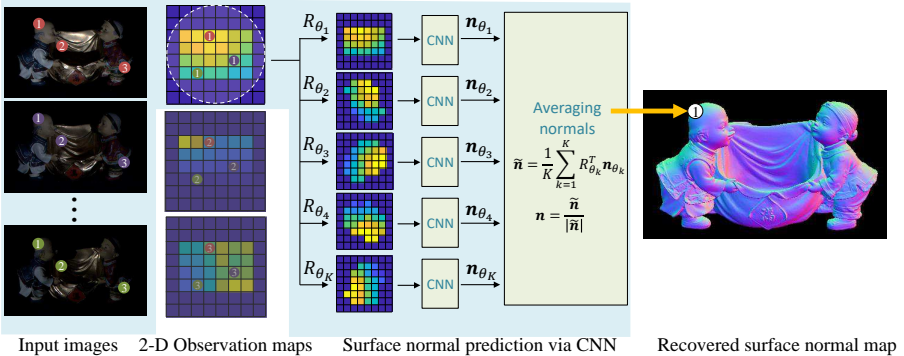
Fig. 3. The illustration of the prediction module. For each surface point, we generate K observation maps taking into account the rotational pseudo-invariance. Each observation map is fed into the network and all the output normals are averaged.
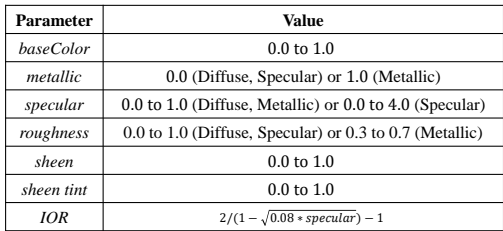
3.3   Architecture details

In this section, we describe the framework of training and prediction. Given images and lightings, we produce observation maps followed by Eq. (2). Data is augmented to achieve the rotational pseudo-invariance by rotating both lighting and surface normal vectors around the viewing axis. Note that a color image is converted to a gray-scale image. The size of the observation map (w) should be chosen carefully. As w increases, the observation map becomes sparser. On the other hand, the smaller observation map has less resprensentability. Considering this trade-off, we empirically found that $w = 32$ is a reasonable choice (we tried $w = 8, 16, 32, 64$ and $w = 32$ showed the best performance when the number of images is less than one thousand).

A variation of densely connected convolutional neural network (DenseNet [28]) architecture is used to estimate a surface normal from an observation map. The network architecture is shown in Fig. 2-(b). The network includes two 2-layer dense blocks, each consists of one activation layer (relu), one convolution layer (3×3) and a dropout layer (20% drop) with a concatenation from the previous layers. Between two dense blocks, there is a transition layer to change feature-map sizes via convolution and pooling. We do not insert a batch normalization layer that was found to degrade the performance in our experiments. After the dense blocks, the network has two dense layers followed by one normalization layer which convert a feature to an unit vector. The network is trained with a simple mean squared loss between predicted and ground truth surface normals. The loss function is minimized using Adam solver [35]. We should note that since our input data size is relatively small (*i.e.*, $32 \times 32 \times 1$), the choice of the network architecture is not a critical component in our framework.[5]
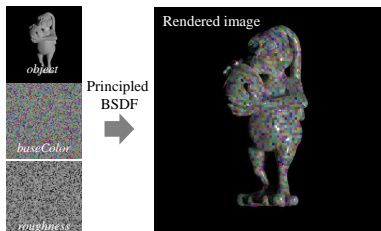
---

[5] We compared architectures of AlexNet, VGG-NET and densenet as well as much simpler architectures with only two or three convolutoinal layers and the dense layer(s). Among the architectures we tested, the current architecture was slightly better.

Parameter	Value
baseColor	0.0 to 1.0
metallic	0.0 (Diffuse, Specular) or 1.0 (Metallic)
specular	0.0 to 1.0 (Diffuse, Metallic) or 0.0 to 4.0 (Specular)
roughness	0.0 to 1.0 (Diffuse, Specular) or 0.3 to 0.7 (Metallic)
sheen	0.0 to 1.0
sheen tint	0.0 to 1.0
IOR	$2/(1 - \sqrt{0.08 * specular}) - 1$

(a)

(b)

Fig. 4. (a) The range of each parameter in the principled BSDF [10] is restricted by three different material configurations (Diffuse, Specular, Metallic). (b) The material parameters are passed to the renderer in the form of a 2-D texture map.

The prediction module is illustrated in Fig. 3. Given observation maps, we predict surface normals based on the trained network. Since it is practically impossible to train the perfect rotational pseudo-invariant network, estimated surface normals for differently rotated observation maps were not identical (typically the difference of angular errors between every two different rotations was less than 10%-20% of their average). For further emphasizing the rotational pseudo-invariance, we again augment the input data by rotating lighting vectors at a certain angle $\theta \in \theta_1, \cdots \theta_K$ and then merge the outputs into one. Suppose the surface normal $(\boldsymbol{n}_\theta)$ is a prediction from the input data rotated by R_θ, then we simply average the inversely rotated surface normals as follows,

$$\bar{\boldsymbol{n}} = \frac{1}{K} \sum_{k=1}^{K} R_{\theta_k}^\top \boldsymbol{n}_{\theta_k},$$

$$\boldsymbol{n} = \bar{\boldsymbol{n}} / \|\bar{\boldsymbol{n}}\|. \tag{9}$$

3.4   Training dataset (*CyclesPS* dataset)

In this section, we present our *CyclesPS* training dataset. *DiLiGenT* [11], the largest real photometric stereo dataset contains only ten scenes with fixed lighting configuration. Some works [18, 17, 19] attempted to synthesize images with MERL BRDF database [29], however only one hundred measured BRDFs cannot cover the tremendous real-world materials. Therefore, we decided to create our own training dataset that has diverse materials, geometries and illumination.

For rendering scenes, we collected high quality 3-D models under royalty free license from the internet.[6] We carefully chose fifteen models for training and three models for test whose surface geometry is sufficiently complex to cover the diverse surface normal distribution. Note that we empirically found 3-D models in ShapeNet [36] which was used in a previous work [4] are generally too simple (*e.g.*, models are often low-polygonal, mostly planar) to train the network.

The representation of the reflectance is also important to make the network robust to wide varieties of real-world materials. Due to its representability,

---

[6] References to each 3-D model are included in supplementary.

we choose Disney's principled BSDF [10] which integrates five different BRDFs controlled by eleven parameters (*baseColor, subsurface, metallic, specular, specularTint, roughness, anisotropic, sheen, sheenTint, clearcoat, clearcoatGloss*). Since our target is isotropic materials without subsurface scattering, we neglect parameters such as *subsurface* and *anisotropic*. We also neglect *specularTint* that artistically colorizes the specularity and *clearcort* and *clearcoatGloss* that does not strongly affect the rendering results. While principled BSDF is effective, we found that there are some unrealistic combinations of parameters that we want to skip (*e.g.*, metallic = 1 and roughness = 0, or metallic = 0.5). For avoiding those unrealistic parameters, we divide the entire parameter sets into three categories, (a) Diffuse, (b) Specular and (c) Metallic. We generate three datasets individually and evenly merge them when training the network. The value of each parameter is randomly selected within specific ranges for each parameter (see Fig. 4-(a)). To realize spatially varying materials, we divide the object region in the rendered image into P (*i.e.*, 5000 for the training data) superpixels and use the same set of parameters at pixels within a superpixel (See Fig. 4-(b)).

For simulating complex light transport, we use *Cycles* [9] renderer bundled in Blender [37]. The orthographic camera and the directional light are specified. For each rendering, we choose a set of an object, BSDF parameter maps (one for each parameter), and lighting configuration (*i.e.*, Once roughly 1300 lights are uniformly distributed on the hemisphere, small random noises are added to each light). Once images were rendered, we create *CyclesPS* dataset by generating observation maps pixelwisely. For making the network robust to the test data of any number of images, observation maps are generated from a pixelwisely different number of images. Concretely, when generating an observation map, we pick a random subset of images whose number is whithin 50 to 1300 and whose corresponding elevation angle of the light direction is more than a random threshold value within 20 to 90 degrees.[7] The training process takes 10 epochs for 150 image sets (*i.e.*, 15 objects × 10 rotations for the rotational pseudo-invariance). Each image set contains around 50000 samples (*i.e.*, number of pixels in the object mask).

4   Experimental Results

We evaluate our method on synthetic and real datasets. All experiments were performed on a machine with 3×GeForce GTX 1080 Ti and 64GB RAM. For training and prediction, we use Keras library [38] with Tensorflow background and use default learning parameters. The training process took around 3 hours.

4.1   Datasets

We evaluated our method on three datasets, two are synthetic and one is real.

---

[7] The minimum number of images is 50 for avoiding too sparse observation map and we only picked the lights whose elevation angles were more than 20 degrees since it is practically less possible that the scene is illuminated from the side.
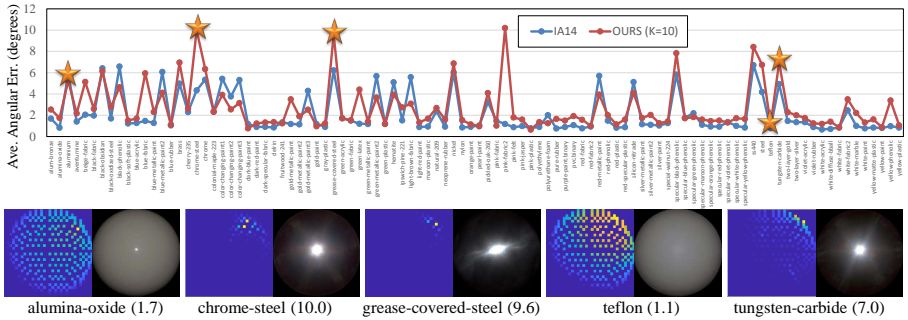
Fig. 5. Evaluation on the *MERLSphere* dataset. A sphere is rendered with 100 measured BRDF in MERL BRDF database [29]. Our CNN-based method was compared against a model-based algorithm (IA14 [7]) based on the mean angular errors of predicted surface normals in degree. We also showed some examples of rendered images and observation maps for further analysis (See Section 4.2).

MERLSphere is a synthetic dataset where images are rendered with one hundred isotropic BRDFs in MERL database [29] from diffuse to metallic. We generated 32-bit HDR images of a sphere (256×256) with a ground truth surface normal map and a foreground mask. There is no cast shadow and inter-reflection.

CyclesPSTest is a synthetic dataset of three objects, SPHERE, TURTLE and PAPERBOWL. TURTLE and PAPERBOWL are non-convex objects where the inter-reflection and cast shadow appear on rendered images. This dataset was generated in the same manner with the *CyclesPS* training dataset except that the number of superpixels in the parameter map was 100 and the material condition was either Specular or Metallic (Note that objects and parameter maps in *CyclesPSTest* are NOT in *CyclesPS*). Each data contains 16-bit integer images with a resolution of 512×512 under 17 or 305 known uniform lightings.

DiLiGenT [11] is a public benchmark dataset of 10 real objects of general reflectance. Each data provides 16-bit integer images with a resolution of 612×512 from 96 different known lighting directions. The ground truth surface normals for the orthographic projection and the single-view setup are also provided.

4.2   Evaluation on *MERLSphere* dataset

We compared our method (with $K = 10$ in Eq. (9)) against one of the state-of-the-art isotropic photometric stereo algorithms (IA14 [17][8]) on the *MERLSphere* dataset. Without global illumination effects, we simply evaluate the ability of our network in representing wide varieties of materials compared to the sum-of-lobes BRDF [24] introduced in IA14. The results are illustrated in Fig. 5. We observed that our CNN-based algorithm performs comparably well, though not

---

[8] We used the authors' implementation of [17] with $N_1 = 2$, $N_2 = 4$ and turning on the retro-reflection handling. Attached shadows were removed by a simple thresholding. Note that our method takes into account all the input information unlike [17].

Table 1. Evaluation on the *CyclesPSTest* dataset. Here m is the number of input images in each dataset and $\{S, M\}$ are types of material *i.e.*, Specular (S) or Metallic (M) (See Fig. 4 for details). For each cell, we show the average angular errors in degrees.

OBJECT	m=17						m=305					
	SPHERE (S)	SPHERE (M)	TURTLE (S)	TURTLE (M)	PAPERBOWL (S)	PAPERBOWL (M)	SPHERE (S)	SPHERE (M)	TURTLE (S)	TURTLE (M)	PAPERBOWL (S)	PAPERBOWL (M)
OURS (K=1)	4.9	11.9	12.9	20.2	22.2	36.8	1.6	2.0	4.2	6.5	6.8	11.1
OURS (K=10)	3.3	9.0	9.9	17.8	20.1	34.2	0.9	1.4	3.3	5.7	6.0	9.5
IA14 [17]	2.7	41.0	13.4	39.1	37.2	43.1	2.8	50.7	16.5	39.0	41.7	42.0
IW12 [6]	1.6	31.0	9.0	35.1	27.7	42.0	1.6	41.7	9.3	37.8	27.6	39.6
IW14 [7]	1.4	23.1	10.5	28.4	36.9	44.2	1.2	20.5	10.2	25.9	37.4	39.5
ST14 [18]	4.4	36.3	12.0	40.6	31.4	41.1	1.0	19.0	19.9	25.5	51.0	50.3
BASELINE[12]	4.6	35.4	12.3	39.7	29.4	39.3	5.0	44.5	12.6	40.2	28.8	37.0

Table 2. Evaluation on the *DiLiGenT* dataset. We show the angular errors averaged within each object and over all the objects. (*) Our method discarded first 20 images in BEAR since they are corrupted (We explain about this issue in the supplementary).

	BALL	BEAR	BUDDHA	CAT	COW	GOBLET	HARVEST	POT1	POT2	READING	AVE. ERR	RANK
OURS (K=10)	2.2	4.1 *	7.9	4.6	8.0	7.3	14.0	5.4	6.0	12.6	7.2	1
OURS (K=1)	2.7	4.5 *	8.6	5.0	8.2	7.1	14.2	5.9	6.3	13.0	7.6	2
HS17 [20]	1.3	5.6	8.5	4.9	8.2	7.6	15.8	5.2	6.4	12.1	7.6	2
TM18 [21]	1.5	5.8	10.4	5.4	6.3	11.5	22.6	6.1	7.8	11.0	8.8	4
IW14 [7]	2.0	4.8	8.4	5.4	13.3	8.7	18.9	6.9	10.2	12.0	9.0	5
SS17 [20]	2.0	6.3	12.7	6.5	8.0	11.3	16.9	7.1	7.9	15.5	9.4	6
ST14 [18]	1.7	6.1	10.6	6.1	13.9	10.1	25.4	6.5	8.8	13.6	10.3	7
SH17 [25]	2.2	5.3	9.3	5.6	16.8	10.5	24.6	7.3	8.4	13.0	10.3	7
IA14 [17]	3.3	7.1	10.5	6.7	13.1	9.7	26.0	6.6	8.8	14.2	10.6	9
GC10 [14]	3.2	6.6	14.9	8.2	9.6	14.2	27.8	8.5	7.9	19.1	12.0	10
BASELINE [12]	4.1	8.4	14.9	8.4	25.6	18.5	30.6	8.9	14.7	19.8	15.4	-

better than IA14, for most of materials, which indicates that Disney's principled BSDF [10] covers various real-world materials. We should note that as was commented in [10], some of very shiny materials, particularly the metals (*e.g.*, chrome-steel and tungsten-carbide), exhibited asymmetric highlights suggestive of lens flare or perhaps anisotropic surface scratches. Since our network was trained on purely isotropic materials, they inevitably degrade the performance.

4.3   Evaluation on *CyclesPSTest* dataset

To evaluate the ability of our method in recovering non-convex surfaces, we tested our method on *CyclesPSTest*. Our method was compared against two robust algorithms IW12 [6] and IW14 [7][9], two model-based algorithms ST14 [18][10] and IA14 [17] and BASELINE [12]. When running algorithms except for ours, we discarded samples whose intensity values were less than 655 in a 16-bit integer image for the shadow removal. In this experiment, we also studied the effects of number of images and rotational merging in the prediction.[11] Concretely, we tested our method on 17 or 305 images with $K = 1$ and $K = 10$ in Eq. (9).

---

[9] We used authors' implementation and set parameters of [6] as $\lambda = 0, \sigma = 1.0^{-6}$ and parameters of [7] as $\lambda = 0, p = 3, \sigma_a = 1.0$.

[10] We used our implementation of [18] and set $T_{low} = 0.25$.

[11] We still augment data by rotations in the training step.
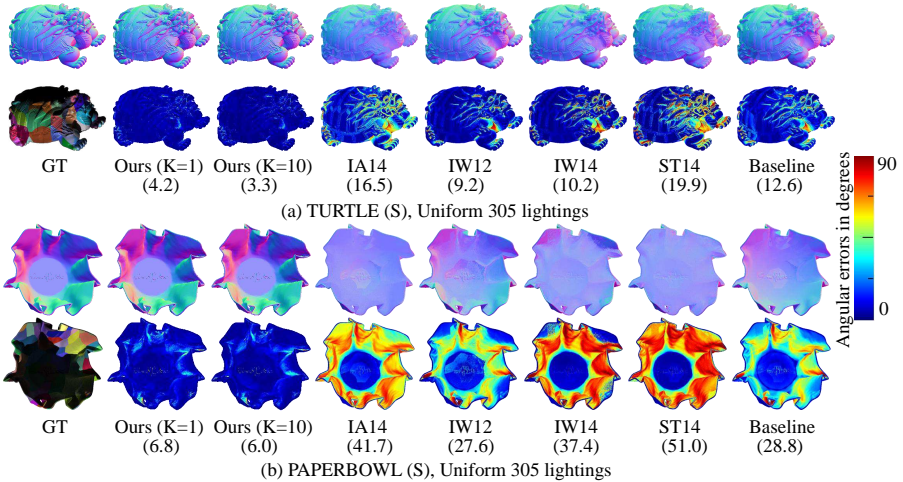
Fig. 6. Recovered surface normals and error maps for (a) TURTLE and (b) PAPER-BOWL of Specular material. Images were rendered under uniform 305 lightings.
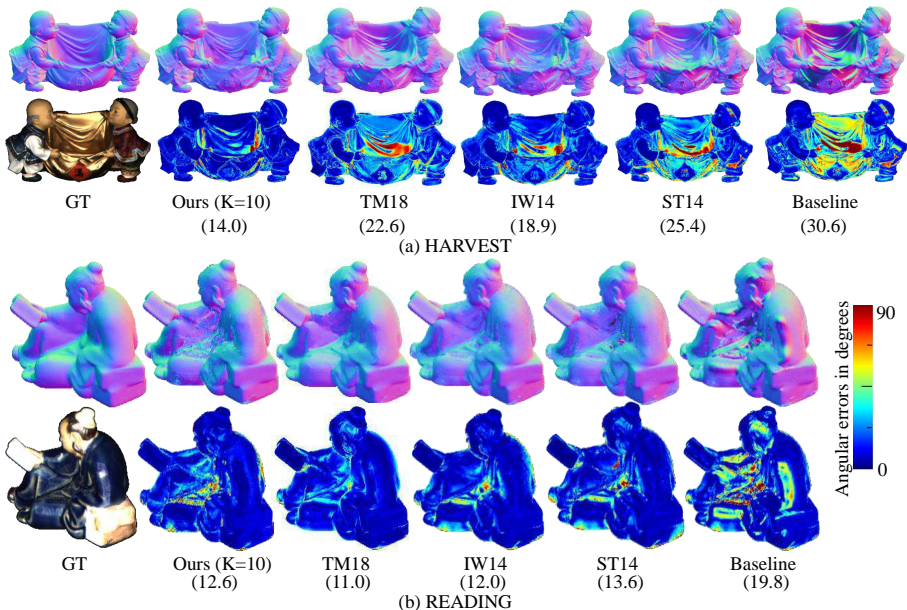
We show the results in Table 1 and Fig. 6. We observed that all the algorithms worked well on the convex specular SPHERE dataset. However, when surfaces were non-convex, all the algorithms except ours failed in the estimation due to strong cast shadow and inter-reflections. It is interesting to see that even the robust algorithms (IA12 [6] and IA14 [7]) could not deal with the global effects as outliers. We also observed that the rotational averaging based on the rotational pseudo-invariance definitely improved the accuracy, though not very much.

4.4 Evaluation on *DiLiGenT* dataset

Finally, we present a side-by-side comparison on the *DiLiGenT* dataset [11]. We collected existing benchmark results for the calibrated photometric stereo algorithms [12–15, 5, 6, 16, 7, 17, 18, 8, 19–21]. Note that we compared the mean angular errors of [12–15, 5, 16–18] reported in [11], ones reported in their own works [19–21] and ones from our experiment using authors' implementation [6–8].[12] The results are illustrated in Table 2. Due to the space limit, we only show the top-10 algorithms[13] w.r.t the overall mean angular, and BASELINE [12]. We observed that our method achieved the smallest errors averaged over 10 objects, best scores for 6 of 10 objects. It is valuable to note that other top-ranked algorithms [20, 21] are time-consuming since HS17 [20] requires the dictionary learning for every different light configuration and TM18 [21] needs the unsupervised training for every estimation while our inference time is less than five seconds (when $K = 1$) for each dataset on CPU. Taking a close look at each

---

[12] As for [8], we used the default setting of their package except that we gave the camera intrinsics provided by [11] and changed the noise variance to zero.

[13] Please find the full comparison in our supplementary.

GT	Ours (K=10)	TM18	IW14	ST14	Baseline
	(14.0)	(22.6)	(18.9)	(25.4)	(30.6)

(a) HARVEST

GT	Ours (K=10)	TM18	IW14	ST14	Baseline
	(12.6)	(11.0)	(12.0)	(13.6)	(19.8)

(b) READING

Fig. 7. Recovered surface normals and error maps for (a) HARVEST and (b) READING in the *DiLiGenT* dataset.

object, Fig. 7 provides some important insights. HARVEST is the most non-convex scene in *DiLiGenT* and other state-of-the art algorithms (TM18 [21], IW14[7], ST14 [18]) failed in the estimation of normals inside the "bag" due to strong shadow and inter-reflections. Our CNN-based method estimated much more reasonable surface normals there thanks to the network trained based on the carefully created *CyclesPS* dataset. On the other hand, our method did not work best (though not bad) for READING which is another non-convex scene. Our analysis indicated that this is because of the inter-reflection of *high-intensity narrow specularities* that were rarely observed in our training dataset (Narrow specularities appear only when *roughness* in the principled BSDF is near zero).

5   Conclusion

In this paper, we have presented a CNN-based photometric stereo method which works for various kind of isotropic scenes with global illumination effects. By projecting photometric images and lighting information onto the observation map, unstructured information is naturally fed into the CNN. Our detailed experimental results have shown the state-of-the-art performance of our method for both synthetic and real data especially when the surface is non-convex. To make better training set for handling narrow inter-reflections is our future direction.

References

1. Kendall, A., Martirosyan, H., Dasgupta, S., Henry, P., Kennedy, R., Bachrach, A., Bry, A.: End-to-end learning of geometry and context for deep stereo regression. Proc. ICCV (2017)
2. Vijayanarasimhan, S., Ricco, S., Schmid, C., Sukthankar, R., Fragkiadaki, K.: Sfm-net: Learning of structure and motion from video. arXiv preprint arXiv:1704.07804 (2017)
3. Kar, A., Häne, C., Malik, J.: Learning multi-view stereo machine. Proc. NIPS (2017)
4. Kim, K., Gu, J., Tyree, S., Molchanov, P., Niessner, M., Kautz, J.: A lightweight approach for on-the-fly reflectance estimation. Proc. ICCV (2017)
5. Wu, L., Ganesh, A., Shi, B., Matsushita, Y., Wang, Y., Ma, Y.: Robust photometric stereo via low-rank matrix completion and recovery. In: Proc. ACCV. (2010)
6. Ikehata, S., Wipf, D., Matsushita, Y., Aizawa, K.: Robust photometric stereo using sparse regression. In: Proc. CVPR. (2012)
7. Ikehata, S., Wipf, D., Matsushita, Y., Aizawa, K.: Photometric stereo using sparse bayesian regression for general diffuse surfaces. IEEE Trans. Pattern Anal. Mach. Intell. **36**(9) (2014) 1816–1831
8. Quau, Y., Wu, T., Lauze, F., Durou, J.D., Cremers, D.: A non-convex variational approach to photometric stereo under inaccurate lighting. In: Proc. CVPR. (2017)
9. Cycles. https://www.cycles-renderer.org/
10. Burley, B.: Physically-based shading at disney, part of practical physically based shading in film and game production. SIGGRAPH 2012 Course Notes (2012)
11. Shi, B., Mo, Z., Wu, Z., D.Duan, Yeung, S.K., Tan, P.: A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. IEEE Trans. Pattern Anal. Mach. Intell. (2018) (to appear)
12. Woodham, P.: Photometric method for determining surface orientation from multiple images. Opt. Engg **19**(1) (1980) 139–144
13. Alldrin, N., Zickler, T., Kriegman, D.: Photometric stereo with non-parametric and spatially-varying reflectance. In: Proc. CVPR. (2008)
14. Goldman, D.B., Curless, B., Hertzmann, A., Seitz, S.M.: Shape and spatially-varying brdfs from photometric stereo. IEEE Trans. Pattern Anal. Mach. Intell. **32**(6) (2010) 1060–1071
15. Higo, T., Matsushita, Y., Ikeuchi, K.: Consensus photometric stereo. In: Proc. CVPR. (2010)
16. Shi, B., Tan, P., Matsushita, Y., Ikeuchi, K.: Elevation angle from reflectance monotonicity. In: Proc. ECCV. (2012)
17. Ikehata, S., Aizawa, K.: Photometric stereo using constrained bivariate regression for general isotropic surfaces. In: Proc. CVPR. (2014)
18. Shi, B., Tan, P., Matsushita, Y., Ikeuchi, K.: Bi-polynomial modeling of low-frequency reflectances. IEEE Trans. Pattern Anal. Mach. Intell. **36**(6) (2014) 1078–1091
19. Santo, H., Samejima, M., Sugano, Y., Shi, B., Matsushita, Y.: Deep photometric stereo network. In: International Workshop on Physics Based Vision meets Deep Learning (PBDL) in Conjunction with IEEE International Conference on Computer Vision (ICCV). (2017)
20. Hui, Z., Sankaranarayanan, A.C.: Shape and spatially-varying reflectance estimation from virtual exemplars. IEEE Trans. Pattern Anal. Mach. Intell. **39**(10) (2017) 2060–2073

21. Taniai, T., Maehara, T.: Neural Inverse Rendering for General Reflectance Photometric Stereo. In: Proc. ICML. (2018)
22. Goldman, D., Curless, B., Hertzmann, A., Seitz, S.: Shape and spatially-varying brdfs from photometric stereo. In: Proc. ICCV. (October 2005)
23. Ward, G.: Measuring and modeling anisotropic reflection. Computer Graphics **26**(2) (1992) 265–272
24. Chandraker, M., Ramamoorthi, R.: What an image reveals about material reflectance. In: Proc. ICCV. (2011)
25. Shen, H.L., Han, T.Q., Li, C.: Efficient photometric stereo using kernel regression. IEEE Transactions on Image Processing **26**(1) (2017) 439–451
26. Silver, W.M.: Determining shape and reflectance using multiple images. Master's thesis, MIT (1980)
27. Hertzmann, A., Seitz, S.: Example-based photometric stereo: shape reconstruction with general, varying brdfs. IEEE Trans. Pattern Anal. Mach. Intell. **27**(8) (2005) 1254–1264
28. G. Huang, Z. Liu, L.M.K.W.: Densely connected convolutional networks. In: Proc. CVPR. (2017)
29. Matusik, W., Pfister, H., Brand, M., McMillan, L.: A data-driven reflectance model. ACM Trans. on Graph. **22**(3) (2003) 759–769
30. Alldrin, N., Kriegman, D.: Toward reconstructing surfaces with arbitrary isotropic reflectance: A stratified photometric stereo approach. In: Proc. ICCV. (2007)
31. Stark, M., Arvo, J., Smits, B.: Barycentric parameterizations for isotropic brdfs. IEEE Trans. on Visualization and Computer Graphics **11**(2) (2011) 126–138
32. Montes, R., Urena, C.: An overview of brdf models. Technical report, LSI-2012-001 en Digibug Coleccion: TIC167 - Articulos (2012)
33. Simard, P.Y., Steinkraus, D., Platt, J.C.: Best practices for convolutional neural networks applied to visual document analysis. In Proc. ICDAR (2003)
34. Schmidt, U., Roth, S.: Learning rotation-aware features: From invariant priors to equivariant descriptors. Proc. CVPR (2012)
35. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In proc. ICLR (2014)
36. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015)
37. Blender. https://www.cycles-renderer.org/
38. Chollet, F., et al.: Keras. https://github.com/keras-team/keras (2015)