# Deep Photometric Stereo Network

Hiroaki Santo*[1], Masaki Samejima†[1], Yusuke Sugano‡[1], Boxin Shi§[2], and Yasuyuki Matsushita¶[1]

[1]Graduate School of Information Science and Technology, Osaka University
[2]Artificial Intelligence Research Center, National Institute of AIST

## Abstract

*This paper presents a photometric stereo method based on deep learning. One of the major difficulties in photometric stereo is designing can appropriate reflectance model that is both capable of representing real-world reflectances and computationally tractable in terms of deriving surface normal. Unlike previous photometric stereo methods that rely on a simplified parametric image formation model, such as the Lambert's model, the proposed method aims at establishing a flexible mapping between complex reflectance observations and surface normal by the use of a deep neural network. As a result we propose a deep photometric stereo network (DPSN) that takes reflectance observations under varying light directions and infers the corresponding surface normal per pixel. To make the DPSN applicable to real-world objects, a database of measured bidirectional reflectance distribution functions (MERL BRDF database) has been used for training the network. Evaluation using simulation and real-world scenes shows effectiveness of the proposed approach over previous techniques.*

## 1. Introduction

Photometric stereo estimates surface normal of an object from a set of measurements that are observed under different light conditions. The basic idea of photometric stereo was introduced in 1980s by Woodham [25] and Silver [22] based on the Lambertian reflectance assumption [13]. To make photometric stereo applicable to real-world objects, it is of interest to use a more flexible reflectance function, for which in a general form it is represented by bidirectional reflectance distribution functions (BRDFs).

While the image formation model with a BRDF representation has greater flexibility and representation power, it is known difficult to directly work with general non-parametric BRDFs in the context of photometric stereo. To ease the problem, there have been studies to use parametric representations to approximate BRDFs. However, so far, known parametric models have been only accurate for a limited class of materials, and the solution methods suffer from unstable optimization, which prohibits obtaining accurate estimates. Thus, it is needed to develop a photometric stereo method that is both computationally tractable and capable of handling diverse BRDFs.

To achieve this goal, we propose an end-to-end learning approach to photometric stereo using a deep neural network (DNN). The proposed method, which we call a deep photometric stereo network (DPSN), uses a DNN for establishing a flexible mapping from reflectance observations to surface normal. To make the DPSN applicable to diverse real-world materials, a database of measured BRDFs (MERL BRDF database [14]) has been used for training the network. In addition, we propose a *shadow layer* that accounts for the non-local shadowing effect using a dropout strategy.

In our method, we assume that the light directions are pre-defined and remain the same between training and prediction phases, which is the case in many photometric stereo apparatuses. The DPSN operates in a per-pixel manner, by taking reflectance observations of a surface point (corresponding to a certain pixel) under varying light conditions, and infers the surface normal of the point. The result shows the effectiveness of our method validated using both simulation and real-world images.

## 2. Related work

Conventional photometric stereo [25, 22] is based on the Lambert's reflectance model. Because the Lambert's model is an ideal reflectance model that may not represent well real-world reflectances, extending photometric stereo to work with non-Lambertian surfaces has been of interest for its practical use. Existing studies on non-Lambertian photometric stereo can be classified into three categories.

The first category is methods based on robust estima-

*santo.hiroaki@ist.osaka-u.ac.jp
†samejima@ist.osaka-u.ac.jp
‡sugano@ist.osaka-u.ac.jp
§boxin.shi@aist.go.jp
¶yasumat@ist.osaka-u.ac.jp

tion, where the non-Lambertian reflectances are treated as outliers. They assume that the majority of reflectance observations obeys, or is close to, the Lambert's model so that the non-Lambertian reflectances, such as specular reflections, can be regarded as anomalies. Wu *et al*. [26] formulate the robust estimation problem as rank minimization. They exploit the fact that the Lambertian observations form a low-rank subspace [2] and treat the non-Lambertian reflectances as sparse outliers. Mukaigawa *et al*. [16] use the random sample consensus (RANSAC) scheme for discarding outliers, which essentially approximates the $\ell_0$ residual minimization. Other robust estimation methods, such as expectation maximization [27], taking the median values [15], $\ell_1$ residual minimization and sparse Bayesian learning [10], are also shown effective for dealing with sparse outliers. Since the robust estimation methods are built upon statistical outlier rejection, they generally require a lot of input images, *e.g*., 40 images in [26], recorded under distinct light directions.

The second category is methods based on more sophisticated reflectance models than the Lambertian model to better approximate non-Lambertian reflectance observations. Georghiades [4] uses the Torrance-Sparrow model [24], and Ruiters *et al*. [18] use Cook-Torrance model [3] in photometric stereo. More recently, Shi *et al*. [20] propose a bi-polynomial BRDF model, which is capable of representing low-frequency non-Lambertian reflectances, and it shows greater accuracy in surface normal estimation. Holroyd *et al*. [8] propose another approach for generalizing reflectance properties based on the reflective symmetry of the halfway vector across the normal-tangent and normal-binormal planes, which does not require estimating a surface reflectance model, and performs well on anisotropic reflectance surfaces.

The third category is example-based methods, which determines surface normal with reference objects. Hertzmann and Seitz [6] propose an example-based method using a reference sphere that has the same reflectance as the target object. From the observations that are consistent between the target and reference objects, their method determines the surface normal of the target object by simply mapping the corresponding one from the reference object. The example-based method naturally avoids solving a complex optimization problem, but it requires a reference object, of which the shape is known and reflectance is the same as the target object.

Our method is somewhere between the second and third categories. As with the methods in the second category, our method is able to deal with diverse BRDFs. Instead of estimating both the BRDF parameters and surface normal in the previous approaches, our method directly establishes mappings from reflectance observations to surface normal using a deep learning framework. Our DPSN is trained using a database of measured BRDFs of various materials (MERL BRDF database [14]); therefore it shares the spirit of the example-based methods in the third category, while DPSN does not require a reference object to be placed together with the target object.

## 3. Preliminaries

When a Lambertian surface with albedo-scaled surface normal $\mathbf{n} \in \mathbb{R}^3$ is illuminated by a directional light $\mathbf{l} \in \mathbb{R}^3$[1] the measurement $m \in \mathbb{R}_+$ can be described as

$$m = \mathbf{l}^\top \mathbf{n}.$$

For a vector of measurements $\mathbf{m} \in \mathbb{R}_+^f$ observed under $f$ distinct light directions, the above equation can be written with a light matrix $\mathbf{L} = [\mathbf{l}_1, \ldots, \mathbf{l}_f] \in \mathbb{R}^{3\times f}$ as

$$\mathbf{m} = \mathbf{L}^\top \mathbf{n}.$$

The conventional photometric stereo method [25, 22] determines the surface normal $\mathbf{n}$ using the above image formation model by

$$\mathbf{n}^* = \mathbf{L}^{-1}\mathbf{m},$$

when $f = 3$ and $\mathrm{rank}(L) = 3$, or, with more than three distinct observations, a least-squares approximate solution $\mathbf{n}^*$ can be obtained by a pseudo-inverse of $\mathbf{L}$ as

$$\mathbf{n}^* = \left(\mathbf{L}\mathbf{L}^\top\right)^{-1}\mathbf{L}\mathbf{m}.$$

Unfortunately, a pure Lambertian surface rarely exists in the real world; therefore, making photometric stereo work with non-Lambertian surfaces is one of the major interests for its practical use.

With a BRDF function $\rho$, the appearance of a surface under a local illumination model can be described more flexibly. The appearances of a surface observed from a fixed viewing direction $\mathbf{v}$ under varying distant light directions $\mathbf{L}$ can be written as

$$\mathbf{m} = \mathbf{b} \circ (\mathbf{L}^\top \mathbf{n}),$$

where $\mathbf{b} \in \mathbb{R}_+^f$ is a vector of reflectances sampled from the BRDF function $\rho$ as $\mathbf{b} = \rho(\mathbf{L}, \mathbf{n}, \mathbf{v})$, and the operator $\circ$ represents element-wise multiplication, and $\left(\mathbf{L}^\top \mathbf{n}\right)$ represents the irradiances at the surface point under the corresponding light directions.

The above equation assumes a shadow-free world, while in the real-world the surface patches facing away from the lighting direction are in attached shadow and light path being occluded causes cast shadow. Such shadowing processes can be written as

$$\mathbf{m} = \mathbf{s} \circ \left[\mathbf{b} \circ \max(\mathbf{L}^\top \mathbf{n}, \mathbf{0})\right], \tag{1}$$

[1] Throughout this paper, we assume $\|\mathbf{l}\|_2 = 1$ and the input image has been normalized by its corresponding light intensity.

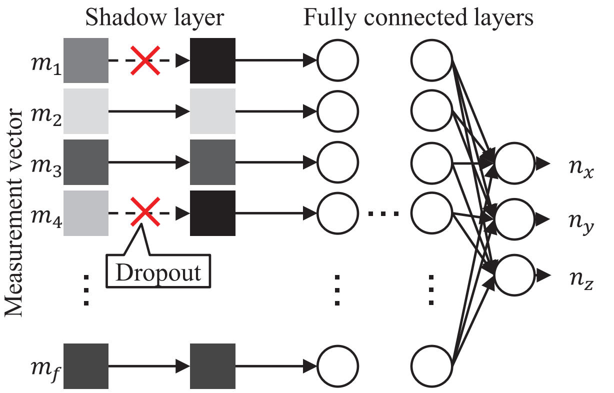

Figure 1. Overview of the proposed network. It consists of two components, the shadow layer and fully connected layers. In the shadow layer, some of the measurement vector elements are randomly dropped to simulate the cast shadow effect. In this figure, $m_1$ and $m_4$ are dropped and corresponding values of the input vector are set to 0.

where $\mathbf{s} \in \{0,1\}^f$ is a boolean vector with 0 indicating observations in cast shadows and 1 otherwise. The effect of attached shadow is accounted by the element-wise $\max$ operator.

## 4. Proposed method

The proposed DPSN is a differentiable multi-layer neural network, which learns a mapping from a measurement vector $\mathbf{m} \in \mathbb{R}^f$ obtained under $f$ different light directions to the surface normal $\mathbf{n} \in \mathbb{R}^3$. It operates in a per-pixel manner for both training and prediction. As stated in the introduction, we assume that the light directions $\mathbf{L} = [\mathbf{l}_1, \ldots, \mathbf{l}_f] \in \mathbb{R}^{3\times f}$ are known and consistent between the training and prediction phases. Our method uses simulated observations that are generated using diverse surface normals rendered with the MERL BRDF database [14], which stores BRDFs of 100 different real-world materials. In what follows, we explain the structure of the proposed network, and training and prediction procedures.

### 4.1. Network architecture

The proposed DPSN learns the mapping from a measurement vector $\mathbf{m}$ of a pixel to the corresponding surface normal $\mathbf{n}$ at that pixel using a fully connected deep neural network. The DPSN takes as input a measurement vector $\mathbf{m} \in \mathbb{R}_+^f$, in which each element corresponds to an observation under a certain light direction, and outputs the prediction of the surface normal $\mathbf{n} \in \mathbb{R}^3$. The measurement vector $\mathbf{m}$ is linearly normalized so that $||\mathbf{m}||_2 = 1.0$ before being fed to the network. While the MERL BRDF dataset contains three color channels, we treat each color channel independently.

One of the major challenges in photometric stereo is cast shadow. Different from attached shadow, cast shadow is due to a global illumination effect, which cannot be modeled by a local illumination model regardless of the representation ability of a BRDF model. To simulate the cast shadow effect in the training phase, we introduce a *shadow layer* that is based on a variant of the dropout scheme [23]. Dropout is a technique, which randomly drops units from the network during training (or could be used for testing as well) to prevent learned weights from excessive adaptation. Our shadow layer applies dropout to input nodes and randomly drops a part of the input measurement vector, namely setting them to 0, so that the dropped nodes can be regarded as shadowed observations. By training the network with the shadow layer, the proposed DPSN effectively learns mapping from observations to surface normal with accounting for diverse BRDFs and cast shadow.

While in conventional dropout, output from the dropout layer is scaled by $1/(1-r)$ with a dropout rate $r \in [0.0, 1.0)$ to avoid shrinkage of the output magnitude, our shadow layer does not apply the scaling but simply sets the selected elements of the measurement vector to 0 to mimic the shadowing effect. The dropout parameter $r$ corresponds to the ratio of shadowed observations in our context. Obviously, the parameter $r$ depends on the object shape and the light distribution, which is inaccessible in general; therefore, we use varying values of $r$ for training. Specifically, we fluctuate the dropout rate by sampling from a binomial distribution $r \sim B(f, p)$, where the probability of each observation being shadowed $p$ is set to $p = 0.05$.

The DPSN structure is summarized in Table 1. The DPSN consists of 7 layers, shadow layer and 6 dense layers. The each dense layers include ReLU and Dropout during training. The DPSN is trained with the following loss function

$$\mathcal{L} = ||\mathbf{n} - \hat{\mathbf{n}}||_2^2, \tag{2}$$

where $\mathbf{n}$ is the ground truth of normal vector and $\hat{\mathbf{n}}$ is the predicted normal vector by the network. $\mathcal{L}$ is minimized using Adam [12] with the suggested default settings.

### 4.2. Training

The training set for DPSN consists of pairs of an observation vector and the corresponding surface normal, *i.e.*, $\{(\mathbf{m}, \mathbf{n})\}$. Instead of collecting real-world observations, we generate the observation vectors $\{\mathbf{m}\}$ using the MERL BRDF dataset [14] for a diverse set of surface normal $\{\mathbf{n}\}$ viewed under pre-defined light directions $\mathbf{L}$. The MERL BRDF dataset consists of measured BRDFs of 100 different materials, and we form reflectance vectors $\{\mathbf{b}\}$ in Eq. (1) from the set of surface normal $\{\mathbf{n}\}$ and light directions $\mathbf{L}$. We ignore the cast shadows when generating a training set, which means the shadow mask $\mathbf{s}$ in Eq. (1) is set to $\mathbf{1}$.

Table 1. The DPSN structure. The number in the parentheses represents the number of nodes in each layer. The shadow layer is used in training, but not for prediction. Dropout rates for training the Dense layers are set to all $0.5$. The input and output dimensions of the shadow layer is $96 (= f)$ in our setting.

| Layer | |
|---|---|
| 1 | Shadow Layer |
| 2 | Dense-(4096), ReLU, Dropout |
| 3 | Dense-(4096), ReLU, Dropout |
| 4 | Dense-(2048), ReLU, Dropout |
| 5 | Dense-(2048), ReLU, Dropout |
| 6 | Dense-(2048), ReLU, Dropout |
| 7 | Dense-(3) |

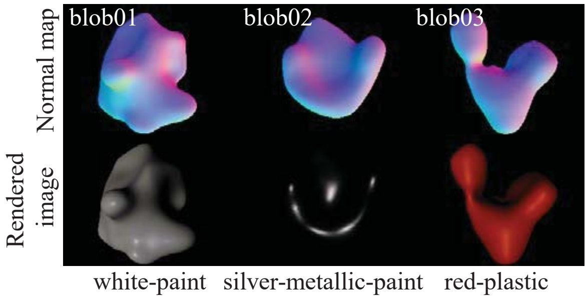

Figure 2. Examples of rendered images that are used for training. `blob01` to `blob03` are a part of shapes from Blobby Shape Dataset [11], "white-paint", "silver-metallic-paint" and "red-plastic" are the material names in the MERL BRDF database [14]. Here, three are shown out of 100 different materials. As seen in the figures, the rendered images contain specularity and attached shadows.

While any distributions of surface normal can be used for generating the surface normal set $\{\mathbf{n}\}$, for this work, we used the Blobby Shape Dataset [11], which consists of various shapes. The shapes are rendered with the MERL BRDFs under light directions $\mathbf{L}$, and observations at each pixel location of the rendered images for each color channel form a measurement vector $\mathbf{m}$. Figure 2 shows the sample images of training data generated under the certain light directions. As shown in the figure, the rendered images contain complex reflectance components that do not obey a simple parametric model.

## 4.3. Prediction

In the prediction phase, given a set of observations under the light directions $\mathbf{L}$, DPSN estimates surface normal in a per-pixel fashion. In a similar manner to the training phase, color channels are treated independently. For an RGB image, the DPSN estimates three surface normals per-pixel and merges them to obtain the final estimate. Namely, say $\mathbf{n}_r$, $\mathbf{n}_g$, and $\mathbf{n}_b$ are the surface normal estimates from the RGB color channels that are independently estimated, we merge them by the following procedure. We put them together by taking the mean vector of the normalized surface normal estimates as

$$\bar{\mathbf{n}} = \frac{1}{3}\left(\frac{\mathbf{n}_r}{||\mathbf{n}_r||_2} + \frac{\mathbf{n}_g}{||\mathbf{n}_g||_2} + \frac{\mathbf{n}_b}{||\mathbf{n}_b||_2}\right).$$

Finally, the merged surface normal $\bar{\mathbf{n}}$ is further normalized to obtain the final surface normal estimate $\hat{\mathbf{n}}$ as

$$\hat{\mathbf{n}} = \frac{\bar{\mathbf{n}}}{||\bar{\mathbf{n}}||_2}.$$

# 5. Experiments

We evaluate the experimental result of the proposed method using both simulation and real-world datasets. We will explain the training data and implementation details for both experiments before showing the results.

**Training data and implementation details** For experiments, eight shapes out of ten in the Blobby Shape Dataset are used for generating the training data, and we kept the remaining two shapes for testing. The shapes are rendered under pre-defined light directions for each of $100$ BRDFs from MERL dataset. The pre-defined light directions are the same as the $96$ light directions defined in the DiLiGenT dataset [21]. As a result, we generated $96$ images for each object and material; therefore, and in total we rendered $8 \times 100 \times 96 = 76800$ images. The resolution of each image is $256 \times 192$, and the total number of the training set $\{(\mathbf{m}, \mathbf{n})\}$ becomes about $3.9 \times 10^7$.

DPSN was implemented using TensorFlow[2], and trained for $5000$ steps with the batch size $1000$. The model which achieves the highest accuracy for test data is used for evaluation.

**Evaluation procedure** We compare the proposed DPSN with Lambertian photometric stereo based on conventional $\ell_2$ residual minimization (L2) [25] and that with $\ell_1$ residual minimization (L1) [10]. For these methods, surface normal $\mathbf{n}$ for each pixel is computed by

$$\text{minimize}_{\mathbf{n}} ||\mathbf{m} - \mathbf{L}^\top \mathbf{n}||_F^2,$$

and

$$\text{minimize}_{\mathbf{n}} ||\mathbf{m} - \mathbf{L}^\top \mathbf{n}||_1$$

from observations $\mathbf{m}$ and known lighting directions $\mathbf{L}$, respectively. For real-world scenes, we also assess the performance of our method using the DiLiGenT benchmark that covers the state-of-the-art methods of non-Lambertian photometric stereo. To see the effect of shadow layer, we compare DPSN without a shadow layer (denoted as "Proposed") and with a shadow layer ("Proposed W/ SL").

[2] TensorFlow: https://www.tensorflow.org

**Experimental result for synthetic dataset** To generate scenes with unseen BRDFs that are not used for training, we synthesize new BRDFs by applying nonlinear transformation over a linearly combined BRDFs that are sampled from the MERL BRDF database. Specifically, a new BRDF $\tilde{\rho}(\mathbf{L}, \mathbf{n}, \mathbf{v})$ is generated as the following:

$$\tilde{\rho}(\mathbf{L}, \mathbf{n}, \mathbf{v}) = (\alpha\rho_1(\mathbf{L}, \mathbf{n}, \mathbf{v}) + (1-\alpha)\rho_2(\mathbf{L}, \mathbf{n}, \mathbf{v}))^{\gamma}, \quad (3)$$

where $\rho_1$ and $\rho_2$ are BRDFs sampled from the MERL BRDF database, and $\alpha$ and $\gamma$ are constant parameters. In this experiment, $\rho_1$ and $\rho_2$ are selected randomly from the database, and the parameters are set to $\alpha = 0.5$ and $\gamma = 0.8$. For target shapes, we used three objects, `sphere`, `blob02`, and `blob08`, which have not been used for training. We used the same light directions as the training for generating the input data.

The evaluation results are summarized in Fig. 3. The two material names used for $\rho_1$ and $\rho_2$ are shown in the top of the figure. “Lambert” means the Lambertian reflectance, for which the reflectance function $\rho$ is a constant. Since the generation of synthetic scenes neglects rendering of cast shadows, our method without a shadow layer (“Proposed”) shows superior performance to the one with the shadow layer (“Proposed W/ SL”) in most scenes.

**Prediction using real-world dataset** Then we evaluate the performance of our method using the DiLiGenT dataset. The DiLiGenT benchmark includes the evaluation of the following eight methods: WG10[17], IW12 [10], GC10 [5], AZ08 [1], ST12 [19], HM10 [7], ST14 [20], and IA14 [9]. We add our results in addition to these eight methods, as shown in Fig. 4 and Table 2.

Figure 4 shows the estimated normal maps and corresponding error maps for “Proposed”, “Proposed W/ SL”, “L2”, and “L1”. Here, we show 6 objects out of 10 in the DiLiGenT dataset (`buddha`, `cow`, `goblet`, `harvest`, `pot2` and `reading`). `buddha`, `pot2` and `reading` are objects made of pottery material, whose reflections are mostly Lambertian except for some specular outliers; therefore, “L2” and especially “L1” based on methods work well. For objects like `cow` and `goblet`, which are made of metallic materials and exhibit strong specular reflection in a wide area, the estimation error becomes rather large for Lambertian based methods such as “L2” and “L1”, while the proposed method can yield highly accurate estimates. For `harvest`, the estimation accuracy is poor in all methods due to the complexity of the shape of the object. For objects with complicated shapes, various reflection phenomenon besides cast shadow such as interreflections occur. We discuss more about this issue in following section.

**The effect of the shadow layer** Figure 5 shows the difference map of error map between “Proposed” and “Proposed W/ SL”. Here, we pick up 4 objects (`ball`, `pot2`, `goblet` and `harvest`) as examples. For all these objects, the accuracy is generally improved in boundary areas where shadows often occur. The accuracy improves in larger areas for `ball` and `pot2` than for `goblet` and `harvest`. For metallic objects with strong interreflections like `goblet` and `harvest`, the measurement value is larger than $0$ for shadows, as the example shown in Fig. 6. Shadow layer assumes that measurement values in the shadowing parts become $0$, so it shows degraded performance when strong interreflections exist.

**Discussion on shadowing probability $p$** In the above section, we use the result of shadowing probability $p = 0.05$, however, the optimal $p$ depends on the shape of object. In this section, we discuss the effect of shadowing probability $p$. We show the results of $p = 0.05$ and $p = 0.9$ on `ball` in Fig. 7. Comparing (a) and (b), we can see that although (b) improves more in the peripheral parts of sphere (shadowing parts) than (b), it deteriorates in the central part.

With shadow layer, the model is optimized for the shadow, and considers that the accuracy deteriorates in the area where the shadow does not exist. In the case of $p = 0.05$, since the dropout rate $r$ sampled in the manner of Sec. 4.1 can be $0$, the estimation accuracy does not deteriorate in areas without shadow. On the other hand, in the case of $p = 0.9$, dropout would be applied to all inputs, so the accuracy deteriorates in the area where the shadow does not exist. Based on such observation, we use $p = 0.05$ in the evaluation.

**Benchmark comparison** In this section, we compare proposed methods (“Proposed” and “Proposed W/ SL”) with benchmark results shown in [21]. Table 5 shows the Mean Angular Error (MAE) in degree of each method for each objects and the average of them. Green color represents good result and Red color represents bad result. For each objects, the best result is highlighted using bold font. For most objects, the best result is obtained by existing methods, but “Proposed W/ SL” significantly improves the accuracy of `harvest` and obtains the best result of the average over all objects. This results show that DPSN generally achieves high accuracy for objects consisting of various materials (real BRDF), which are not seen during training (synthetic BRDF). The high accuracy is partially due to that DPSN an handle the cast shadow effectively, which is not addressed by most existing methods.

## 6. Discussion

This paper proposed a photometric stereo method based on deep learning. The proposed method uses deep neural network for establishing a flexible mapping from shading observations to surface normal. Inspired by the suc-

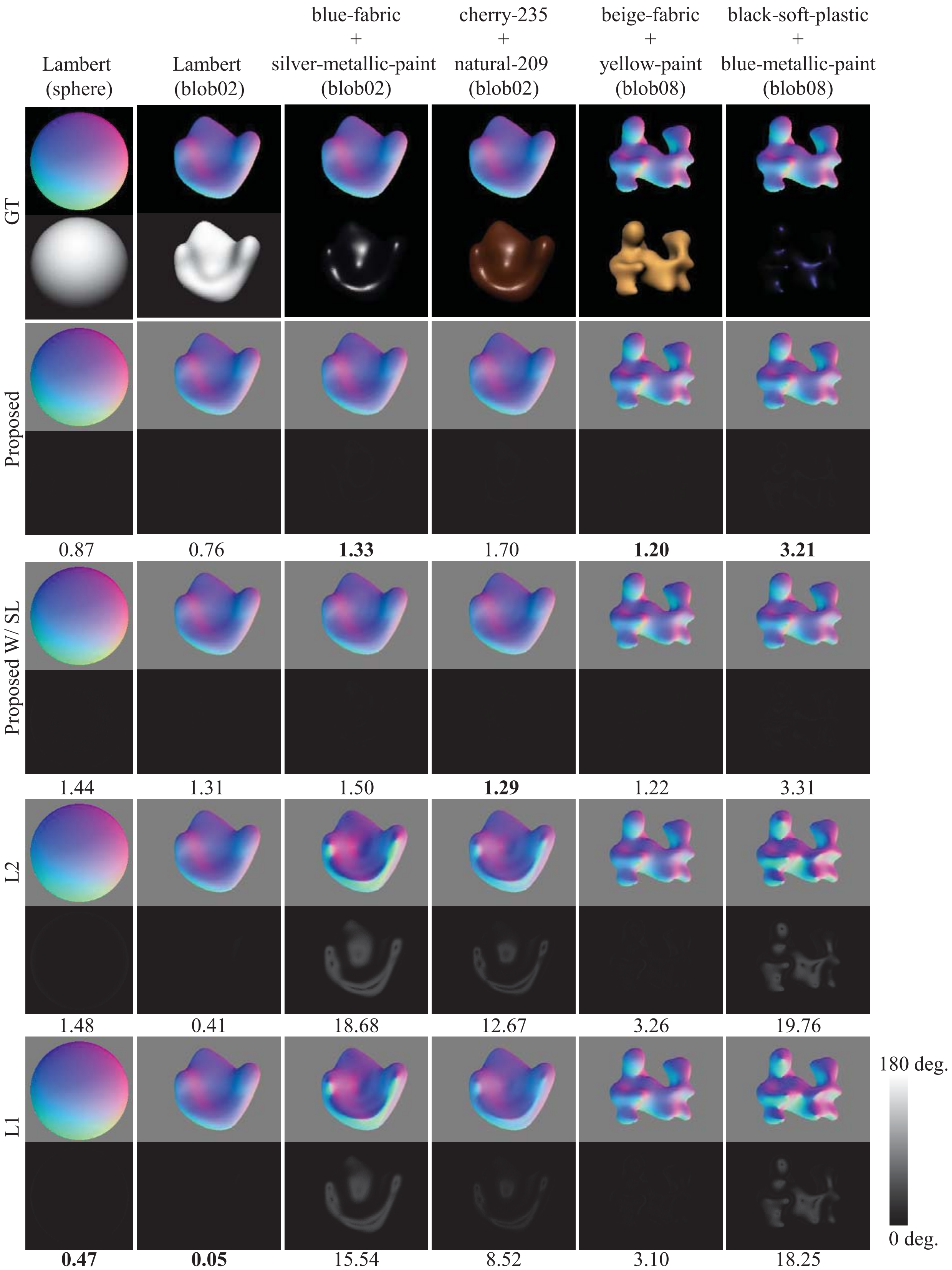

Figure 3. Experimental result for synthetic scenes. In each row, a normal map is shown on top of an error map. The numbers represent Mean Angular Error (MAE) in degree. In the top row, GT means the ground truth, the images below the normal maps are examples of observation images. On the top, material and shape names that are used for synthesizing the data are displayed.

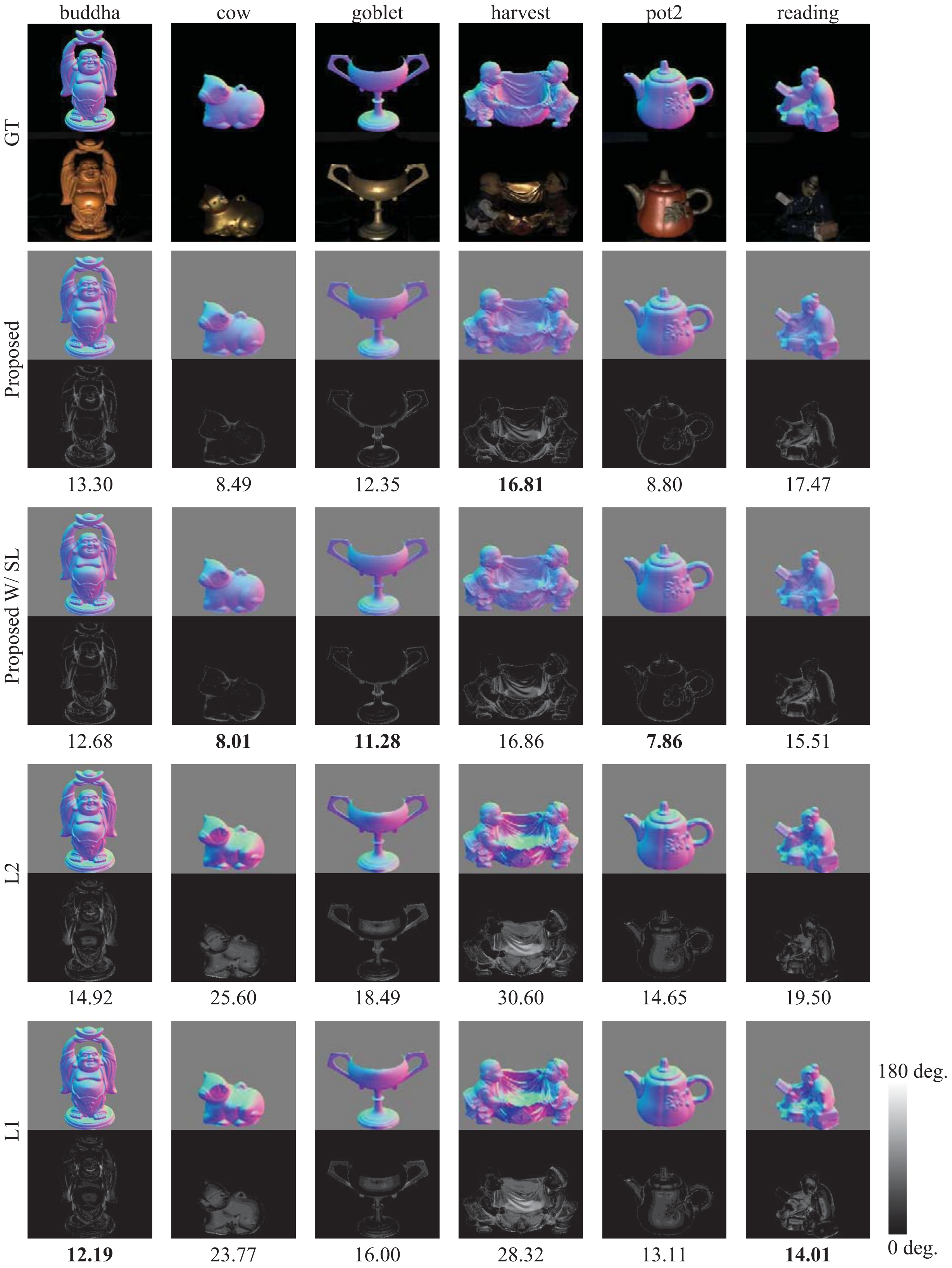

Figure 4. Estimation result for real-world scenes from DiLiGenT [21]. In each row, a normal map is shown on top of an error map. The numbers represent Mean Angular Error (MAE) in degree. GT means "ground truth" and below figures are one of observation images. The contrast of observation images is adjusted for easy viewing.

Table 2. Comparison with benchmark [21].

| | ball | cat | pot1 | bear | buddha | cow | goblet | harvest | pot2 | reading | AVG. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Proposed | 3.44 | 7.21 | 7.90 | 7.20 | 13.30 | 8.49 | 12.35 | **16.81** | 8.80 | 17.47 | 10.30 |
| Proposed W/ SL | 2.02 | 6.54 | 7.05 | 6.31 | 12.68 | 8.01 | 11.28 | 16.86 | **7.86** | 15.51 | **9.41** |
| ST14 | **1.74** | **6.12** | **6.51** | 6.12 | 10.60 | 13.93 | 10.09 | 25.44 | 8.78 | **13.63** | 10.30 |
| IA14 | 3.34 | 6.74 | 6.64 | 7.11 | **10.47** | 13.05 | **9.71** | 25.95 | 8.77 | 14.19 | 10.60 |
| WG10 | 2.06 | 6.73 | 7.18 | 6.50 | 10.91 | 25.89 | 15.70 | 30.01 | 13.12 | 15.39 | 13.35 |
| AZ08 | 2.71 | 6.53 | 7.23 | **5.96** | 12.54 | 21.48 | 13.93 | 30.50 | 11.03 | 14.17 | 12.61 |
| HM10 | 3.55 | 8.40 | 10.85 | 11.48 | 13.05 | 14.95 | 14.89 | 21.79 | 16.37 | 16.82 | 13.22 |
| IW12 | 2.54 | 7.21 | 7.74 | 7.32 | 11.11 | 25.70 | 16.25 | 29.26 | 14.09 | 16.17 | 13.74 |
| ST12 | 13.58 | 12.34 | 10.37 | 19.44 | 18.37 | **7.62** | 17.80 | 19.30 | 9.84 | 17.17 | 14.58 |
| GC10 | 3.21 | 8.22 | 8.53 | 6.62 | 14.85 | 9.55 | 14.22 | 27.84 | 7.90 | 19.07 | 12.00 |
| BASELINE | 4.10 | 8.41 | 8.89 | 8.39 | 14.92 | 25.60 | 18.50 | 30.62 | 14.65 | 19.80 | 15.39 |

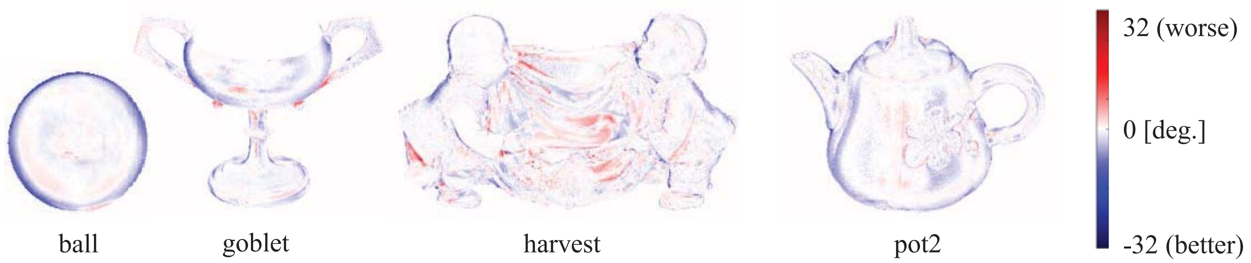

Figure 5. Improvement by Shadow Layer. We show difference map of error map between "Proposed" and "Proposed W/ SL". Pixels whose normal estimation accuracy is improved by shadow layer are shown in blue, otherwise in red.

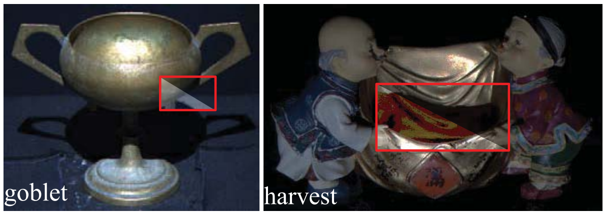

Figure 6. We adjust the left half of red box area with the gamma correction. The shadowing parts in the box shows larger value than 0 due to strong interreflections.

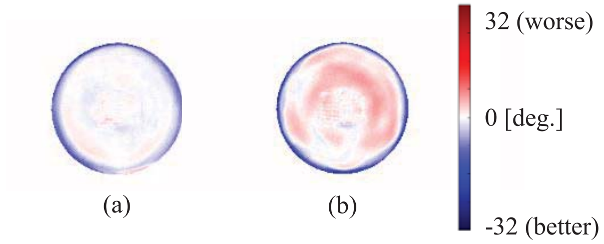

Figure 7. Comparison shadowing probability $p$ on "Ball". (a) is $p = 0.05$ ("Proposed W/ SL") and (b) is $p = 0.9$. Both are difference maps of error map with "Proposed". Pixels whose normal estimation accuracy is improved by shadow layer are shown in blue, otherwise in red.

cess of example-based methods to deal with challenging reflectance, we use modern neural network structures and train it end-to-end with simulated observations rendered by the MERL BRDF database [14] to replace the functionality of reference objects. In addition, we proposed a shadow layer that simulates the non-local shadowing effect using the dropout strategy. Evaluation using real-world scenes shows more accurate estimation of the surface normal in comparison to previous techniques.

**Limitations** One of limitations in our method is the assumption that light directions are pre-defined and remain the same between training and test phases. Our method needs to train models for each lighting condition, because our training requires simulated observations rendered under a specific lighting condition. Our method is particularly useful if the photometric stereo data capture uses a device with fixed light sources and camera, so that we only need to perform the training for that device. It is our future work to deal with lighting condition which is not pre-defined, such as the data capture using a randomly waving light source.

## Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP16H01732.

## References

[1] N. Alldrin, T. Zickler, and D. Kriegman. Photometric stereo with non-parametric and spatially-varying reflectance. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2008. 5

[2] R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2), 2003. 2

[3] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24, 1982. 2

[4] A. Georghiades. Incorporating the torrance and sparrow model of reflectance in uncalibrated photometric stereo. pages 816–823, 2003. 2

[5] D. B. Goldman, B. Curless, A. Hertzmann, and S. M. Seitz. Shape and spatially-varying brdfs from photometric stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1060–1071, 2010. 5

[6] A. Hertzmann and S. 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

[7] T. Higo, Y. Matsushita, and K. Ikeuchi. Consensus photometric stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1157–1164. IEEE, 2010. 5

[8] M. Holroyd, J. Lawrence, G. Humphreys, and T. Zickler. A photometric approach for estimating normals and tangents. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2008)*, 27(5):133, 2008. 2

[9] S. Ikehata and K. Aizawa. Photometric stereo using constrained bivariate regression for general isotropic surfaces. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2179–2186, 2014. 5

[10] S. Ikehata, D. Wipf, Y. Matsushita, and K. Aizawa. Robust photometric stereo using sparse regression. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 318–325. IEEE, 2012. 2, 4, 5

[11] M. K. Johnson and E. H. Adelson. Shape estimation in natural illumination. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2553–2560, 2011. 4

[12] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *Proceedings of International Conference on Learning Representations (ICLR)*, 2014. 3

[13] J. Lambert. Photometria. *Augustae Vindelicorum*, 1760. 1

[14] W. Matusik, H. Pfister, M. Brand, and L. McMillan. A data-driven reflectance model. *ACM Transactions on Graphics*, 22(3):759–769, July 2003. 1, 2, 3, 4, 8

[15] D. Miyazaki, K. Hara, and K. Ikeuchi. Median photometric stereo as applied to the segonko tumulus and museum objects. *International Journal of Computer Vision*, 86(2–3):229–242, 2010. 2

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

[17] T. Papadhimitri and P. Favaro. A closed-form, consistent and robust solution to uncalibrated photometric stereo via local diffuse reflectance maxima. *International journal of computer vision*, 107(2):139–154, 2014. 5

[18] R. Ruiters and R. Klein. Heightfield and spatially varying brdf reconstruction for materials with interreflections. *Computer Graphics Forum*, 28(2):513–522, Apr. 2009. 2

[19] B. Shi, P. Tan, Y. Matsushita, and K. Ikeuchi. Elevation angle from reflectance monotonicity: Photometric stereo for general isotropic reflectances. *Proceedings of European Conference on Computer Vision (ECCV)*, pages 455–468, 2012. 5

[20] B. Shi, P. Tan, Y. Matsushita, and K. Ikeuchi. Bi-polynomial modeling of low-frequency reflectances. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1078–1091, 2014. 2, 5

[21] B. Shi, Z. Wu, Z. Mo, D. Duan, S.-K. Yeung, and P. Tan. A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4, 5, 7, 8

[22] W. M. Silver. Determining shape and reflectance using multiple images. Master's thesis, Massachusetts Institute of Technology, Cambridge, MA, U.S., 1980. 1, 2

[23] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014. 3

[24] K. E. Torrance and E. M. Sparrow. Theory for off-specular reflection from roughened surfaces. *JOSA*, 57(9):1105–1114, 1967. 2

[25] R. J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical engineering*, 19(1):139–144, 1980. 1, 2, 4

[26] L. Wu, A. Ganesh, B. Shi, Y. Matsushita, Y. Wang, and Y. Ma. Robust photometric stereo via low-rank matrix completion and recovery. *Proceedings of Asian Conference on Computer Vision (ACCV)*, pages 703–717, 2011. 2

[27] T.-P. Wu and C.-K. Tang. Photometric stereo via expectation maximization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(3):546–560, 2010. 2