 DOI: 10.3217/978-3-85125-524-9-32

Photometric Stereo in Multi-Line Scan Framework under Complex Illumination via Simulation and Learning

Dominik Hirner[1,2], Svorad Štolc[1], Thomas Pock[2]

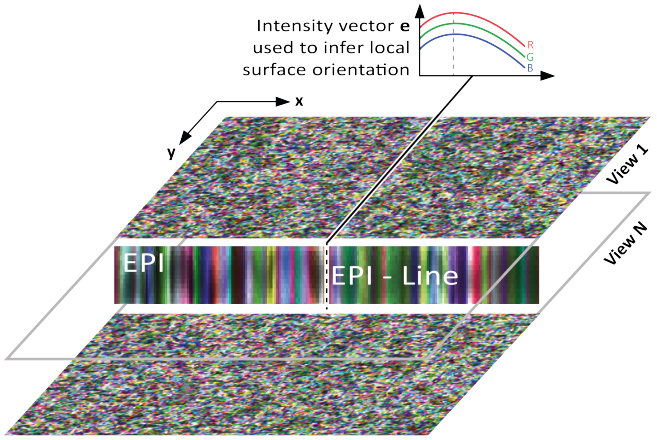


Fig. 1: Visualization of the image stack created by the multi-line scan acquisition. The middle part shows the EPI-lines (here a slice through the image stack). The dashed line represents the read out of one such EPI-line with the respective RGB intensity vector **e**, which is used in order to infer (by training the network) the surface gradient in transport direction ∇x.

***Abstract*— This paper presents a neural network implementation of photometric stereo formulated as a regression task. Photometric stereo estimates the surface normals by measuring the irradiance of any visible given point under different lighting angles. Instead of the traditional setup, where the object has a fixed position and the illumination angles changes around the object, we use two constant light sources. In order to produce different illumination geometries, the object is moved under a multi-line scan camera. In this paper we show an approach where we present a multi-layer perceptron with a number of intensity vectors (i.e. points with constant albedo under different illumination angles) from randomly chosen pixels of six materials with different reflectance properties. We train it to estimate the gradient of the surface normal along the transport direction of the given point. This completely eliminates the need of knowing the light source configuration while still remaining a competitive accuracy even when presented with materials which have non-Lambertian surface properties. Due to the random pooling of the pixels our implementation is also independent from spatial information.**

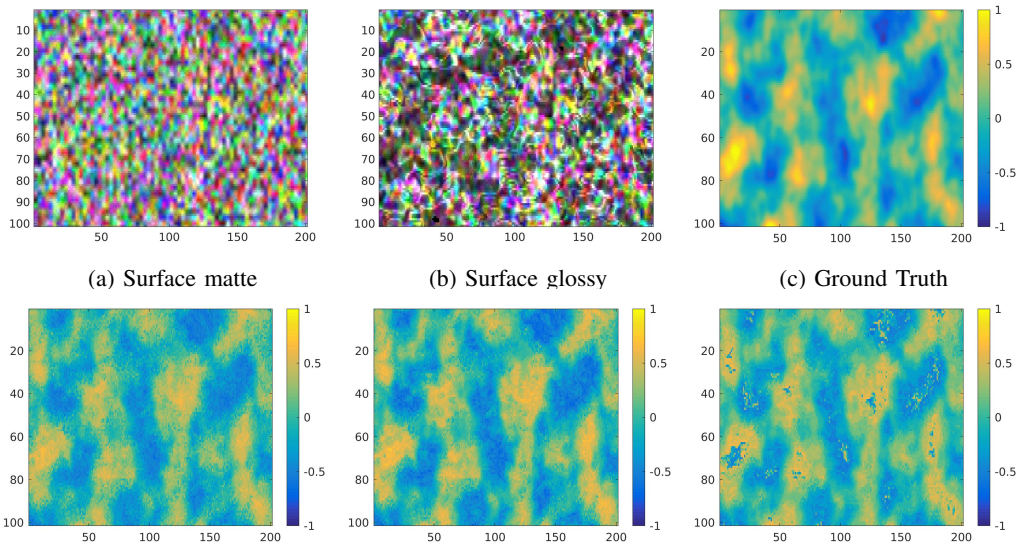
I. INTRODUCTION

The goal of photometric stereo is to estimate the surface normals (and therefore 3D information) of an object using 2D images. This is done by exploiting Lambert's cosine law [1], which states that the intensity of the light at a point is directly proportional to the cosine of its surface normal and the angle of the incident light (see Eq. (1)). By measuring the light intensity of each point under different known and fixed illumination angles the surface normal of each point can be calculated. This approach was first introduced by Woodham in 1980 [2]. However, this equation only holds with the assumption of a Lambertian surface, i.e. a surface that scatters the light in all directions equally. In case of specular reflections the observed intensity of a point also dependents on the position of the observer and therefore the basic approach of photometric stereo does not hold. In the standard photometric stereo approach the orientation and position of the observer (i.e. camera) is known and fixed. Light-field processing via light-field cameras can be seen as an add-on to the general photometric stereo idea. A light-field is a 4-D radiance function written as $L(u,v,s,t)$, where (u,v) denotes the angle, and (s,t) denotes the position of each light ray respectively. To capture a light-field with a camera, a number of different approaches exist, for instance commercially avaliable plenoptic cameras such as the Lytro [3] or by using an array of cameras (multi-camera array) [4]. Using multi-line scan acquisition with a light-field in order to create 2.5/3D surface structure was first introduced in [5]. The same multi-line scan light field camera was used in this approach, which acquires multiple single lines (in our implementation 13) with different viewing angles at one time. Between the active lines on the sensor there are a number of predefined inactive lines (in our implementation 40), so that different viewing angles are produced within one acquisition step without the need of placing several cameras (as e.g. in a multi-camera array).

In our setup an object is placed underneath the camera and is transported in a defined direction over time with two constant light strips placed orthogonal to the transport direction. Between two acquisition steps s_i and s_{i+1} the object has to move the distance equivalent by exactly one pixel. After the acquisition process, the single lines acquired by one such step of each active line on the sensor are concatenated and thus all possible lighting angles and a number of different views are created. This produces a 3D light field structure (two spatial and one directional dimension), instead of the usual 4D structure. This 3D light field can be represented as an image stack that can be seen in Fig. 1. This allows for a fast in-line acquisition suitable for industrial inspection. However, since different lighting responses are dependent on the movement of the object, only inference in the transport

[1]AIT Austrian Institute of Technology GmbH, Vision, Automation & Control, Vienna, Austria {dominik.hirner, svorad.stolc}@ait.ac.at

[2]Graz University of Technology, Institute for Computer Graphics and Vision, Graz, Austria pock@icg.tugraz.at



(a) Surface matte (b) Surface glossy (c) Ground Truth

(d) Prediction Lambertian (e) Prediction Semi-glossy (f) Prediction glossy

Fig. 2: Yellow pixels show positive and blue pixels show negative gradients. Predictions of the surface gradient in transport direction are shown as follows: (a) Surface of the Lambertian material (first view of the 3D light field data), (b) Surface of the glossy material (first view of the 3D light field data), (c) ground truth surface normal gradient in transport direction ∇x used as labels for the regression network, (d) surface normal gradient in transport direction of a Lambertian material learned by the network, (e) surface normal gradient in transport direction of a semi-glossy (gloss = 0.25, roughness = 0.75, see Fig. 4) material learned by the network, (f) surface normal gradient in transport direction of a very glossy material learned by the network. The properties of the different materials can be seen in Fig. 4. One can see that the peaks of the specular lobes (i.e. areas of the biggest negative or positive gradients in (c)) can produce wrong gradient signs.

direction is possible.

The basic method of photometric stereo uses the fact that the observed intensity (or light response) of a given point is dependent on the surface normal orientation as well as the direction of the light, under the assumption of viewing a Lambertian material and a constant albedo. This can be formulated as follows:

$$\mathbf{e} = \mathbf{L} \cdot \mathbf{n} \cdot a \tag{1}$$

where $\mathbf{e} = [e_1 \dots e_n]^T$ is a vector of observed intensities, $\mathbf{L}$ is a matrix describing the light directions and n denotes the surface normal $\mathbf{n} = [n_x, n_y, n_z]^T$. a denotes the albedo which is a scalar value in range $a \in [0,1]$. Inverting this linear equation system yields:

$$\mathbf{n} \cdot a = \mathbf{L}^+ \cdot \mathbf{e} \tag{2}$$

Solving this over-determined least squares problem produces an estimation of the surface normal (Note: L^+ is the Pseudo-Inverse of the light direction matrix using, e.g. the Moore-Penrose method [6]). Instead of solving Eq. (1) directly we use a multi-layer perceptron in order to learn a mapping between the intensity responses ($e_R = [e_{R1} \dots e_{R13}]^T, e_G = [e_{G1} \dots e_{G13}]^T, e_B = [e_{B1} \dots e_{B13}]^T$) in each pixel to the gradient of the surface normal in transport direction $\nabla x = {}^{an_x}/_{an_y}$.

Some results of the learned mapping are visualized in Fig. 2. The figure depicts the same small area in all six images. The first two images are examples of how the surfaces that where used for inference from two different material types (matte and glossy) looks like. For both images the first view (i.e. the first illumination angle) of the 3D light field structure was taken. The remaining images show a color-coded visualization of the surface normal gradient ∇x.

The intensity vectors e_R, e_G and e_B correspond to the observed intensity values of the different illumination angles (here referred to as views) for each channel of the RGB pixel value respectively. These three vectors (e_R, e_G and e_B) are then stacked vertically for each pixel in order to create the data samples for the network, which then has the form $E = [e_{R1} \dots e_{R13}, e_{G1} \dots e_{G13}, e_{B1} \dots e_{B13}]^T$, where $E \in \mathbb{R}^{39}$ (three color-channels a 13 illumination angles). These data-points of all six datasets are then randomly shuffled in order to avoid a spatial bias due to, e.g. non-constant lighting before presenting it to the network. Since the cumulative number of all points from all datasets is very large (around 3 million samples), a batch based training approach with a batch size of 1000 was used rather then an online learning approach. We used the TensorFlow library [7] for the implementation of the network as well as for the cost function and optimizer.

II. Related Work

3D reconstruction using 2D images has been a well studied problem in the field of computer vision. Over the years many different methods to solve this problem arose. In [8] range scanning with stripe patterns were combined with

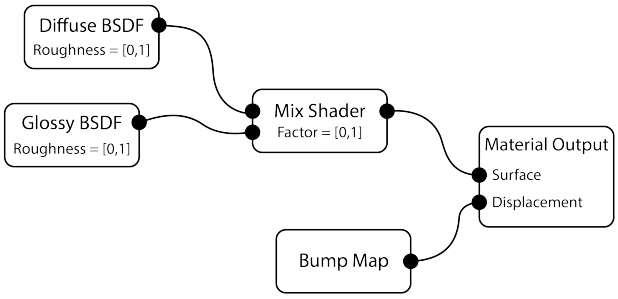


Fig. 3: Schematic illustration of the Blender Node Setup for creating different materials.

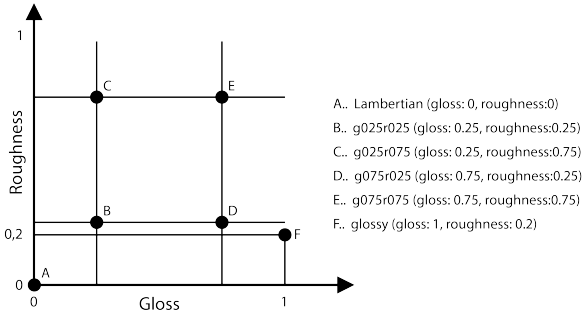


Fig. 4: Visualization of the six different datasets created by changing the roughness and the percentage of which the glossy or diffuse node is taken.

photometric stereo with five light sources in order to recover the 3D surface of an object. Using epipolar plane image (EPI) structures from motion analysis for depth reconstruction was introduced in [9]. The paper by Tao deals with incorporating a shading term to depth from defocus with correspondence cues in order to refine the shape estimation [10]. In [11] Hayakawa used a singular-value decomposition (SVD) of a formulated matrix in order to get a surface normal estimation without the need of a-priori knowledge of the light source direction under the Lambertian assumption. Some machine learning approaches have been explored, such as [12] where a multi-layered neural network was used in order to learn the mapping between image intensities and the surface normal orientation, using a Gaussian sphere with average reflectance as the training data. In [13] Cheng used a symmetrical 6-layer neural network to train a mapping between the vectorized image and a reflectance value for each pixel. Another machine learning approach has been investigated in [14], where a neural network was used in order to solve the shape from shading problem, previously introduced by [15].

III. Experiments

A. Generating Ground Truth Data

Blender 2.78 [16] Cycles Renderer was used to generate the ground truth data. This artificial ground truth data has some advantages over real-world acquisition, such as the ease of modification of the setup, feasibility of generating many images quickly as well as being less prone to errors. However, in order to make the resulting images more realistic, some artefacts, such as jitter or salt&pepper noise, can be taken into consideration. The goal while creating the ground truth data was to cover as much ground as possible with the synthetic data regarding the task. The network should learn a mapping between the RGB intensity vectors of the different views and surface properties, to the surface normal gradient. As it is infeasible to cover all possible mappings between color, light reflectance and surface normals, a random approach was chosen. A uniformly distributed, 8-bit random color pattern was created (each RGB color channel uniformly distributed between 0-255) and used as a texture. The blender-intern noise texture and displacement map node was used in order to create a random surface normal structure on a flat surface. With this approach the possible mapping space is sparsely covered. Furthermore we created six different material datasets with different gloss values using a mixture of the Diffuse BSDF and Glossy BSDF node shaders. In this model two parameters can be changed, namely the gloss factor (controlled by the mix node) and the roughness of the two BSDF nodes. For the sake of simplicity, the roughness is the same for both, the Diffuse and the Glossy BSDF node. A schematic illustration of this setup can be seen in Fig. 4. This model is based on a presentation from Gastaldo [17], where he states:

$$\mathscr{R} + \mathscr{T} + \mathscr{A} = 1 \tag{3}$$

where $\mathscr{R}$ denotes reflectivity, $\mathscr{T}$ denotes transparency and $\mathscr{A}$ denotes absorption. Furthermore he states that reflectivity can be divided into diffuse reflectivity ($\mathscr{R}_d$) and specular reflectivity ($\mathscr{R}_s$). With this he derives:

$$\mathscr{R}_d + \mathscr{R}_s + \mathscr{T} \leq 1 \tag{4}$$

In our setup $\mathscr{R}_d$ correlates to the Diffuse BSDF node and $\mathscr{R}_s$ to the Glossy BSDF node. Transparency was not taken into consideration (i.e. is always 0) as we exclude glass like materials from our data. The roughness parameter of the Diffuse BSDF node corresponds with the roughness of the Oren-Nayar reflectance model [18]. The model used for the glossy factor of the material was GGX [19]. The roughness parameter of the GGX model simulates microscopic bumps in the surface, so that the reflections of the material look blurrier the higher the roughness parameter is.

We excluded a glossy dataset with a roughness value of 0, which would imitate a mirror like behavior. However,

a material with a roughness of 0.2 already shows highly specular behavior.

The multi-line scan camera setup as described in Sec. I was recreated in Blender, where a plane with a random color texture and a bump map (see Fig. 3) was moved underneath the camera. During each animation step the plane was moved by exactly one pixel. The resulting images were concatenated and reshaped in order to create a 3D image stack representation of the light field. Each image plane is then shifted to the left in the following manner:

$$\forall x,y,i : I'_i(x,y) = I_i(x-40i,y) \tag{5}$$

where $i \in [0\dots12]$ denotes the index of the image in the 3D light field structure, $I_i \in \{width \times height \times 3\}$ is the spatial image domain of the i'th view and I'_i denotes the new translated image. Since the disparity (i.e. the gap between active lines on the camera sensor) is 40 pixels it was used as the shifting constant. The resulting overlap (at most 12×40 in the last view) is then cropped. This is done so that the EPI-lines are vertical with no slope, as they would be with an object with true 3D geometries.

B. Network Parameters Evaluation

For the optimal performance of a neural network some parameter evaluation and tuning, such as changing the number of hidden neurons, using different activation functions or cost functions, is needed. In our evaluation, we looked at 3 different activation functions, namely linear, Sigmoid and rectified linear unit (RELU). The input layer, which consists of 39 neurons is fully connected with the hidden layer. We tried different numbers of neurons in the hidden layer for each evaluated activation function respectively. The results of these experiments can be seen in Table I. For the read-out of the output layer, which consists of one neuron since we only regress the gradient in the transport direction, a linear activation function was used.

Given the problem we want to solve and our material properties, one can expect that a low number of hidden neurons will suffice and already give a good performance, as the Lambertian reflectance function is low dimensional. The low dimensionality of a Lambertian reflectance has been proven and explored in [20]. Having a less complex network architecture can be beneficial for both the runtime as well as the generalization of the network. Here 1, 3, 10 and 20 neurons of one hidden layer were used.

The cost function used to measure the quality of the regressed prediction (therefore also the value used for the optimization of the network) was the mean square error (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2 \tag{6}$$

For optimization the batch based gradient descent algorithm with a learning rate of $\eta = 0.001$ was used. The dataset was split into 80% training set and 20% testing set, as proposed by the Pareto Principle by J.M. Juran [21]. The network was trained for 100 epochs.

Table I shows that the Sigmoid has both the best overall, as well as the best performance in a single run with 20 neurons in the hidden layer. As the aforementioned experiments were performed only to show the overall tendency and convergence of the network structure, a small learning rate η was used for all the experiments. However, [22] shows that exploring this parameter further is important for the overall network accuracy. For this task we found that a learning rate of $\eta = 0.2$ works best which improved the overall accuracy of the network to $MSE_{train} = 0.020464$ and $MSE_{test} = 0.02052$ when trained for 100 epochs.

TABLE I: Training and testing MSE with different numbers of neurons and activation functions.

Training set MSE					
# hidden neurons	1	3	10	20	avg
act. fct.					
linear	0.05903	0.05988	0.05760	0.05857	0.05877
Sigmoid	0.05429	0.05285	0.05263	**0.04792**	**0.05192**
RELU	0.05605	0.05543	0.05283	0.05150	0.05395
Testing set MSE					
linear	0.05902	0.05972	0.05777	0.05855	0.05877
Sigmoid	0.05402	0.05276	0.05266	**0.04768**	**0.05178**
RELU	0.05608	0.05571	0.05312	0.05147	0.054095

C. Network Performance

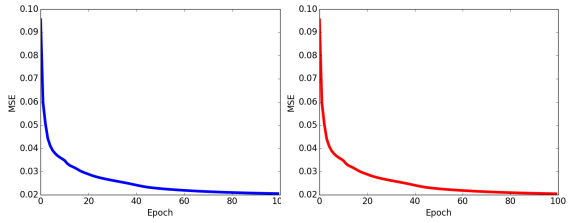


Fig. 5: Evolution of the mean square error over epochs with a learning rate of $\eta = 0.2$. Left: Training set (80% randomly chosen from all sets), right: Testing set (20% randomly chosen from all sets).

Fig. 5 shows the convergence of the overall accuracy on the training and test set, combining and shuffling all six created datasets. This was done in order to generalize the network as much as possible regarding the material type (matte, semi-glossy or glossy). Once the network was learned it was applied to each material type individually and the accuracy of the prediction on the whole set was reported. For simplicity we use acronyms for each created dataset, as shown in Fig. 3. For the sake of simplicity we took the liberty of reporting the error on the whole dataset (data points used for training and testing combined). As the errors on the training and on the testing set are very close together and there is no sign of overfitting the network, this liberty can be taken without distorting the results. The best performance was achieved on the semi-glossy datasets. The larger error on the glossy dataset is due to the fact that the sign of the surface normal is sometimes predicted wrong if the specular lobe is narrow and outside of the observed range. This can

TABLE II: MSE of each individual whole dataset applied to the network. On the left we report the accuracy of our neural network, then the accuracy of the Lambertian model when 80% of the Lambertian dataset was used to estimate the illumination matrix **L** from Eq. (1) as an analogy of learning (L.m.L stands for Lambertian model Lambertian datasets). In the last column 80% of all datasets were used (Lambertian model all dataset).

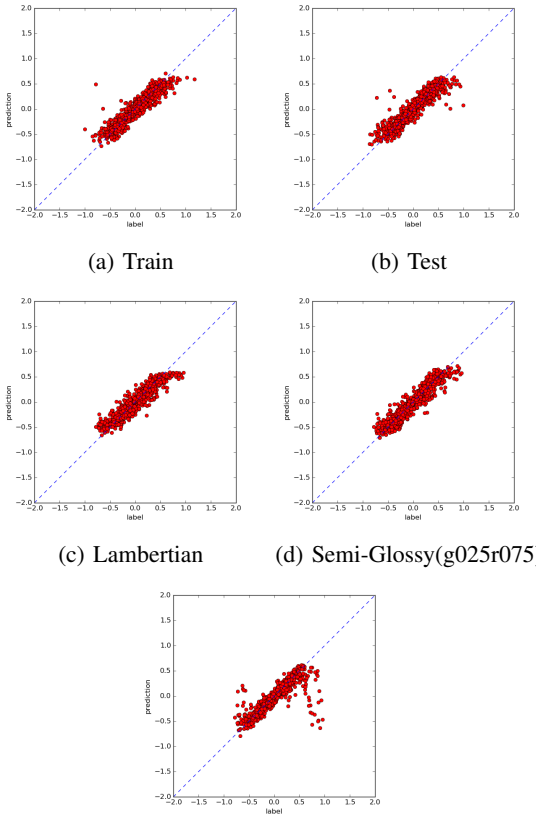
Dataset	$MSE_{network}$	$MSE_{L.m.L}$	$MSE_{L.m.a}$
Lambertian	0.01637	0.02435	0.02804
g025r025	0.01537	0.05550	0.03268
g025r075	0.01835	0.02063	0.02741
g075r025	0.01760	0.24619	0.10237
g075r075	0.01795	0.03233	0.02930
glossy	0.03722	0.89302	0.37912
avg.	0.02047	0.21200	0.09982

also be seen in the correlation plots in Fig. (6) where some of the outliers from the glossy dataset also show up in the correlation plot for the whole train and test dataset.

We compare our results with the model-based Lambertian approach by solving Eq. (1) for **L** as an analogy of learning with the same dataset training/testing split as for our machine learning approach. For this the assumption of an constant albedo with a value of 1 was taken. Despite it can be argued that the Lambertian model only works for Lambertian materials. The quantitative results are reported in Table II. It can be seen that the L.m.L. approach completely failed for the glossier material datasets. On the other hand the L.m.a. approach proved to perform in average about twice as good improving significantly especially on the glossy cases. Last but not least, we show that our neural network approach outperforms the traditional photometric stereo by far for the given task, especially for glossier material.

IV. CONCLUSIONS AND FUTURE WORK

In this paper we showed a neural network based machine learning approach in order to learn a mapping between intensity vectors (i.e different illumination angles) of points with different reflectance properties to a surface normal gradient. We showed that in our approach we do not need to know the position and direction of the light source as well as no spatial information and were still able to produce competitive accuracy. The proposed machine learning approach outperformed the standard photometric stereo based on the Lambertian model by 5-10 times. We tested the network on synthetically generated data and showed that our implementation works well even for very glossy surface properties. In our simulations the train error converges very fast which suggests that we did not yet reach the absolute best accuracy possible and increasing the number of features as well as training the network for longer may still increase the overall prediction of the multilayer perceptron. The mean absolute error (MAE) can be advantageous as it is more robust against outliers [23], however since we excluded strong outliers manually in our datasets beforehand we did not need to use MAE. Nevertheless, exploring this cost function in the future should be done.



(a) Train (b) Test

(c) Lambertian (d) Semi-Glossy(g025r075)

(e) Glossy

Fig. 6: (a-e) Show the correlation plot between label and prediction of ∇x for the respective datasets of 100 samples uniformly taken from the set. (a) combines 80% of all datasets (which were randomly chosen). (b) combines 20% of all datasets (which were randomly chosen). (e) shows some outliers where the sign of the gradient was wrongly predicted due to the high specular response. The stronger outliers on (a) and (b) also come from this set.

For future work we intend to extend this approach to perform material classification (e.g. classify matte, glossy, semi-glossy material etc.) as well as learning the albedo of the created datasets. In this paper we only used synthetic data in order to prove the correctness of the method, however an evaluation on real-world data for the trained networks would be the next step. Additionally, we want to investigate the possibilities of inference on the surface normal gradient orthogonal to the transport direction.

REFERENCES

[1] J.H. Lambert. *Photometria sive De mensura et gradibus luminis, colorum et umbrae*. Sumptibus viduae Eberhardi Klett, typis Christophori Petri Detleffsen, 1760.

[2] Robert J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):191139–191139–, 1980.

[3] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light Field Photography with a Hand-Held Plenoptic Camera. Technical report, April 2005.

[4] Bennett Wilburn, Neel Joshi, Vaibhav Vaish, Eino-Ville Talvala, Emilio Antunez, Adam Barth, Andrew Adams, Mark Horowitz, and Marc Levoy. High performance imaging using large camera arrays. *ACM Trans. Graph.*, 24(3):765–776, July 2005.

[5] Svorad Štolc, Reinhold Huber-Mörk, Branislav Holländer, and Daniel Soukup. Depth and all-in-focus images obtained by multi-line-scan light-field approach. In *IS&T/SPIE Electronic Imaging*, pages 902407–902407. International Society for Optics and Photonics, 2014.

[6] Jonathan S Golan. Moore–penrose pseudoinverses. In *The Linear Algebra a Beginning Graduate Student Ought to Know*, pages 441–452. Springer, 2012.

[7] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[8] Diego Nehab, Szymon Rusinkiewicz, James Davis, and Ravi Ramamoorthi. Efficiently combining positions and normals for precise 3d geometry. In *ACM transactions on graphics (TOG)*, volume 24, pages 536–543. ACM, 2005.

[9] Robert C Bolles, H Harlyn Baker, and David H Marimont. Epipolar-plane image analysis: An approach to determining structure from motion. *International Journal of Computer Vision*, 1(1):7–55, 1987.

[10] Michael W Tao, Pratul P Srinivasan, Jitendra Malik, Szymon Rusinkiewicz, and Ravi Ramamoorthi. Depth from shading, defocus, and correspondence using light-field angular coherence. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1940–1948, 2015.

[11] Hideki Hayakawa. Photometric stereo under a light source with arbitrary motion. *JOSA A*, 11(11):3079–3089, 1994.

[12] KV Rajaram, Guturu Parthasarathy, and MA Faruqi. A neural network approach to photometric stereo inversion of real-world reflectance maps for extracting 3-d shapes of objects. *IEEE transactions on systems, man, and cybernetics*, 25(9):1289–1300, 1995.

[13] Wen-Chang Cheng. Neural-network-based photometric stereo for 3d surface reconstruction. In *Neural Networks, 2006. IJCNN'06. International Joint Conference on*, pages 404–410. IEEE, 2006.

[14] Siu-Yeung Cho and Tommy WS Chow. Shape recovery from shading by a new neural-based reflectance model. *IEEE Transactions on Neural Networks*, 10(6):1536–1541, 1999.

[15] Berthold KP Horn and Michael J Brooks. *Shape from shading*. MIT press, 1989.

[16] Blender Online Community. *Blender - a 3D modelling and rendering package*. Blender Foundation, Blender Institute, Amsterdam,

[17] Franois Gastaldo. How to make your own physically correct shading. Blender Converence, 2012.

[18] Michael Oren and Shree K Nayar. Generalization of lambert's reflectance model. In *Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 239–246. ACM, 1994.

[19] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, pages 195–206, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[20] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, Feb 2003.

[21] F John Reh. Pareto's principle-the 80-20 rule. *BUSINESS CREDIT-NEW YORK THEN COLUMBIA MD-*, 107(7):76, 2005.

[22] Igiri Chinwe Peace, Anyama Oscar Uzoma, and Silas Abasiama Ita. Effect of learning rate on artificial neural network in machine learning. In *International Journal of Engineering Research and Technology*, volume 4. IJERT, 2015.

[23] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.