A Design of Contractive Appearance Flow for Photometric Stereo: Supplemental Materials

BMVC 2020 Submission # 1305

1 Sampling an Incident Light Field

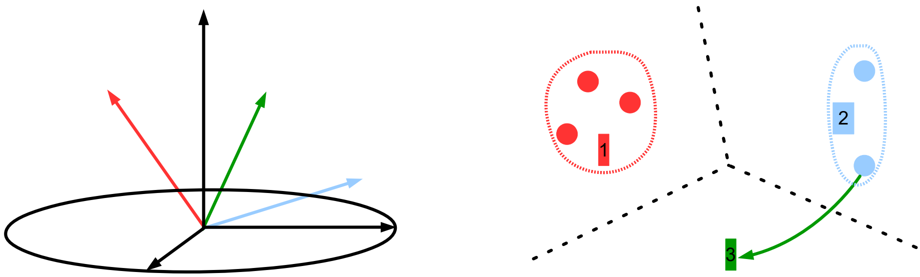
Due to page limit, here we elaborate on how to present a consistent Representation for an incident light field that is briefly discussed in Section 4.1 in the manuscript.

An good representation for an incident light field in a data-driven scenario is always a trade-off between expressivity and stability: on the one hand, it has to be expressive enough so that reflectance signals sampled from different surface normals $\vec{n}$ can be differentiated; on the other hand, it has to be stable enough with respect to different sampling schemes to avoid overfitting. In other words, samples of a well-generalized representation in training data accounts for what is likely to occur during testing, for the input signal can effectively reduce the size of training data. This work qualitatively addresses the former aspect as it is related to the concept of appearance flow, and a rigourous analysis on both aspects will be the focus of our future work.

Similar to "observation map" [1], we partition the hemisphere into multiple disjoint regions. Specifically, we randomly select a set of K unit vectors, $\bar{L}$, and each sample generated falls in a neighborhood of one of the elements of $\bar{L}$. Consequently, the samplers generated a vector $X \in R^K$ that serves as a standard representation for the appearance flow and can be fed into a neural network directly. In contrast to the design of "observation map" sets a grid on the fronto-parallel plane (XY-plane) to describe the lighting distribution, our design defines the neighborhood using the angular distance (*i.e.* inner product between two unit vectors) directly.

During this partitioning process, two scenarios may arise need extra processing: (1) there is no sampler in the neighborhood, (2) there are multiple samplers fall into the same neighborhood. As illustrated in Figure 1, in the first case we simply find the nearest neighbor to interpolate, whereas in the second case we average all samples in the neighborhood. It is worth noting that this design allows for multiple independent partition schemes, which result in multiple representations to describe the same incident light field. In reference to Figure 3 in the manuscript, they are processed by different **channels** in the latent space Z_1, where $Z_1(\theta_r)$ is the cross-channel average. Formally speaking, we can set up $\bar{L}_1, \bar{L}_2 \ldots, \bar{L}_N$ over N channels to produce $X_1(\theta_r), X_2(\theta_r) \ldots, X_N(\theta_r)$ and

$$Z_1(\theta_r) = \frac{1}{N}\sum_{i=1}^{N} \Phi^i_{tr} \cdot \Phi^i_{en}(X_i(\theta_r)). \tag{1}$$



(1) set up $\bar{L}$ (2) sampling scheme

Figure 1: (1) If $\bar{L}$ contains three elements, the flow is represented by vectors in R^3. (2) To obtain a standard representation out of a set of randomly distributed samples, the spherical surface is partitioned into three regions measured by angular distance. Circles indicate actual samples and square indicates the center of each region. Each region takes the average value of the samples in its neighborhood. When no sampler is falling into a region (*e.g.* region 3), then we fill this region with the value taken from its nearest neighbor as a replacement.

and in our experiment we use 4 channels to illustrate this idea.

2 Generating Training Data

The training set includes as many combinations of $\vec{n}$,θ_r and L as possible. L denoting the distribution of the samples is independent of $\bar{L}$ and is characterized by two measures: **sparsity** and **coverage**, where the former has been investigated in details [4], and we do not have much insight into the latter. To prepare the training set, we make the same assumption as the "observation map" does, which assumes that an incident light field can be sufficiently sampled by the samplers over the positive hemisphere for shape analysis. As a rule of thumb,we limit the coverage of both L and $\bar{L}$ to the positive hemisphere. Specifically, during training, for each sample, for each sample we randomly select a value $Z_{min} \in [0.2, 0.7]$ to determine a range $[Z_{min}, 1]$, from which L is sampled and a vector is then generated over the unit-sphere [3]. In this case, the distribution of the azimuth angles is approximately uniform.

The surface normal, $\vec{n}$, is generated by from a normal map of a sphere, which is dense and uniformly distributed and represented by 30000 R^3 unit vectors. The source of our input is simply the tabulated MERL reflectance data [2] consisting of 100 different materials.

3 Network Architecture

The network architecture is illustrated in Figure 2. It consists of all MLP implementation, where the input signal consists of 128 pixel values under 128 lightings specified by $\bar{L}$. The output signal is another 128-tuple representing its corresponding Lambertian reflectance.

The first encode-transfer procedure can be carried out over multiple channels. Given N channels in design (Equation 1), we deploy N distinct instances to produce N latent signals in R^{64}, and all these signals are averaged to encode one single latent signal in R^{48}. Correspondingly, there are also N instances of Φ_{de} that generates $128 \times N$ pixel values.

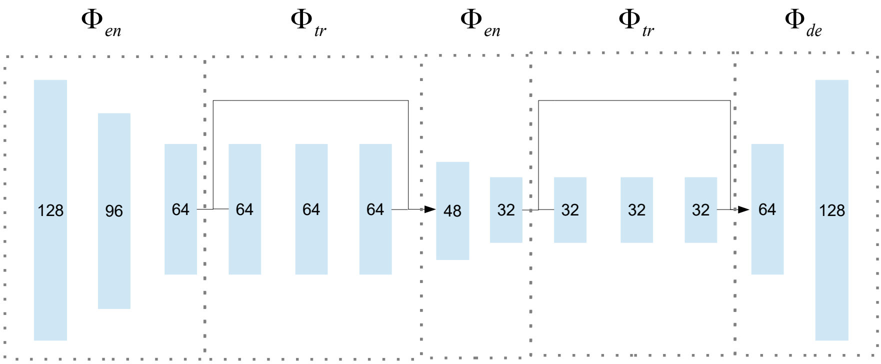


Figure 2: Network architecture.The input space is in R^{128}, and the two latent spaces are spanned by vectors in R^{64} and R^{32}, respectively. Two transfer operator, Φ_{tr}, are implemented by residual blocks as they are space invariant.The first encode-transfer process, namely, from R^{128} to R^{64}, can be carried out in multiple channels for one sample, and a cross-channel average is performed to encode a single latent signal in R^{48}.

4 Experiment Settings

The experiment is conducted on a standard PC with a single RTX-2070 GPU of 8G RAM. Batch size is set to 30000, epoch number is 12, learning rate is 0.001. Each sample vector is normalized by its maximum value, so all of all of its entries are in $[0,1]$. We select 128 directions over the hemisphere to generate $\bar{L}$. As there are 4 independent channels, it takes 512 variables to represent an input incident light field. The entire pipeline is implemented using Matlab Deep Learning Toolbox.

5 Benchmark Performance

With an element in $\bar{L}$ we can synthesize an image of the scene with Lambertian reflectance. In addition to comparison of normal map, for illustration purpose, we pick 8 of the total 512 images produced for each scene. L has 96 nights.

The sequence for the **normal map** comparison is: (1) **ground truth**, (2) **estimation**, (3) **error map**. The color saturates at error of 40 **degrees**.

The sequence for the **Lambertian reflectance** comparison is: (1) **ground truth**, (2) **estimation by the network**, (3) **error map** of the network estimation. The color saturates at error of **1**. (4) **reconstruction** from the estimation and $\bar{L}$.

References

[1] Satoshi Ikehata. Cnn-ps: Cnn-based photometric stereo for general non-convex surfaces. In *Proc. of European Conference on Computer Vision*, 2018.

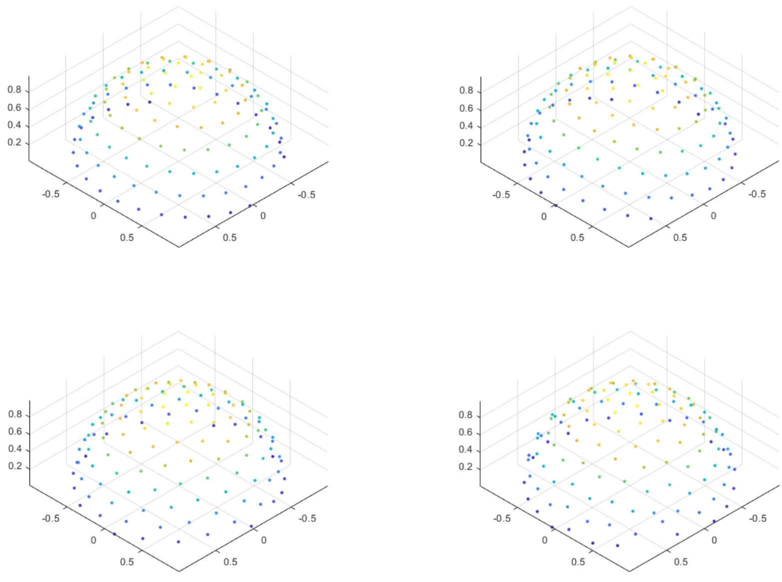


Figure 3: Four different light distributions generated for $\bar{L}$, one for a channel. Each distribution takes 128 samples, so the network decodes one latent signals into four separate vectors in R^{128}.

[2] Wojciech Matusik and Matt Brand. A data-driven reflectance model. *ACM Transactions on Graphics (Proc. of ACM SIGGRAPH)*, 22(3), 2003.

[3] Evguenii A Rakhmanov, EB Saff, and YM Zhou. Minimal discrete energy on the sphere. *Mathematical Research Letters*, 1(6), 1994.

[4] Qian Zheng, Yiming Jia, Boxin Shi, Xudong Jiang, Ling-Yu Duan, and Alex C Kot. Spline-net: Sparse photometric stereo through lighting interpolation and normal estimation networks. In *Proc. of International Conference on Computer Vision*, 2019.

Figure 4: ball

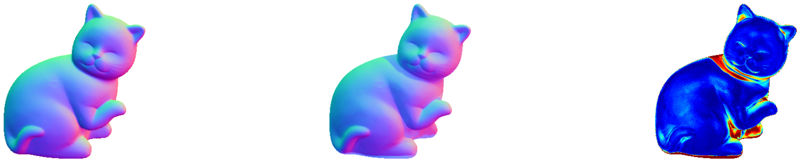


Figure 5: cat

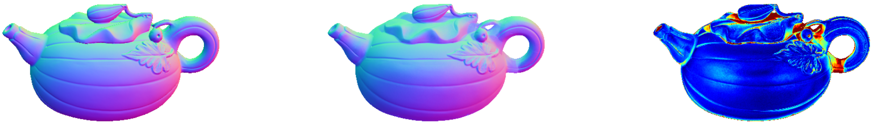


Figure 6: pot1

Figure 7: bear

Figure 8: pot2

Figure 9: buddha

Figure 10: goblet

Figure 11: reading

Figure 12: cow

Figure 13: harvest

Figure 14: ball

Figure 15: ball

Figure 16: ball

Figure 17: ball

Figure 18: ball

Figure 19: ball

Figure 20: ball

Figure 21: ball

Figure 22: cat

Figure 23: cat

Figure 24: cat

Figure 25: cat

Figure 26: cat

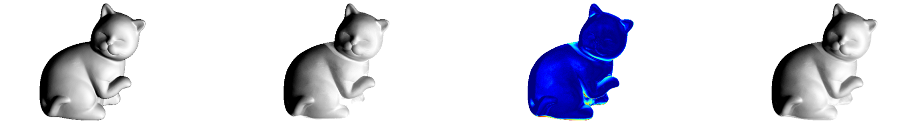


Figure 27: cat

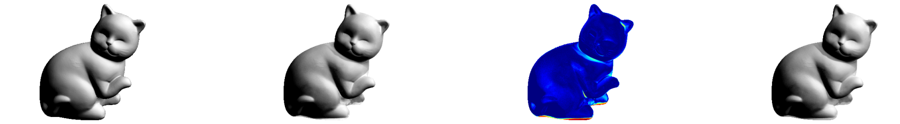


Figure 28: cat

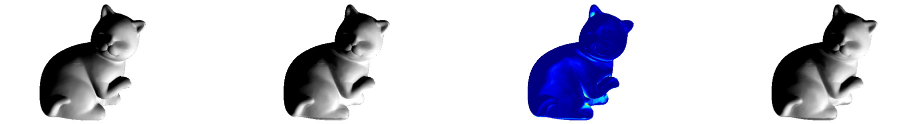


Figure 29: cat

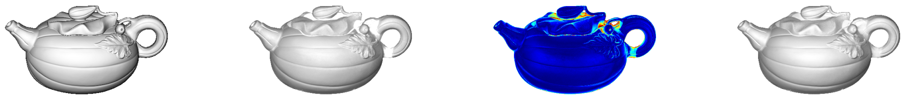


Figure 30: pot1

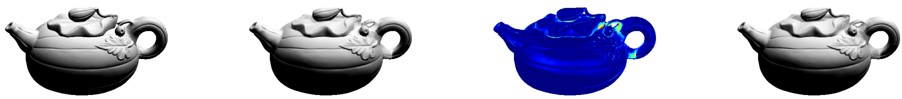


Figure 31: pot1

Figure 32: pot1

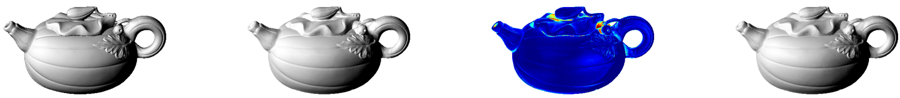


Figure 33: pot1

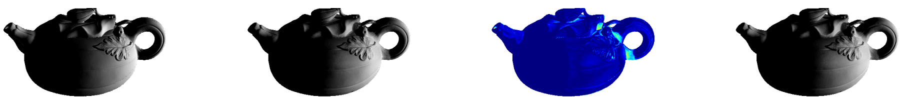


Figure 34: pot1

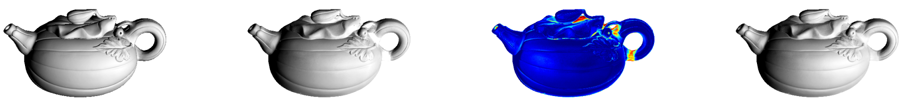


Figure 35: pot1

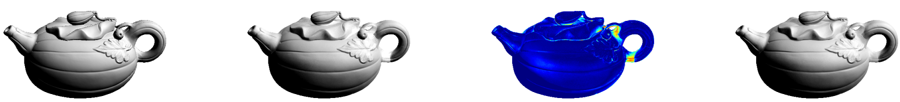


Figure 36: pot1

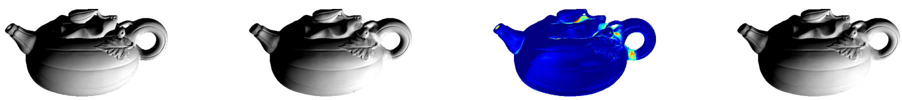


Figure 37: pot1

Figure 38: bear

Figure 39: bear

Figure 40: bear

Figure 41: bear

Figure 42: bear

Figure 43: bear

Figure 44: bear

Figure 45: bear

Figure 46: pot2

Figure 47: pot2

Figure 48: pot2

Figure 49: pot2

Figure 50: pot2

Figure 51: pot2

Figure 52: pot2

Figure 53: pot2

Figure 54: buddha

Figure 55: buddha

Figure 56: buddha

Figure 57: buddha

Figure 58: buddha

Figure 59: buddha

Figure 60: buddha

Figure 61: buddha

Figure 62: goblet

Figure 63: goblet

Figure 64: goblet

Figure 65: goblet

Figure 66: goblet

Figure 67: goblet

Figure 68: goblet

Figure 69: goblet

Figure 70: reading

Figure 71: reading

Figure 72: reading

Figure 73: reading

Figure 74: reading

Figure 75: reading

Figure 76: reading

Figure 77: reading

Figure 78: cow

Figure 79: cow

Figure 80: cow

Figure 81: cow

Figure 82: cow

Figure 83: cow

Figure 84: cow

Figure 85: cow

Figure 86: harvest

Figure 87: harvest

Figure 88: harvest

Figure 89: harvest

Figure 90: harvest

Figure 91: harvest

Figure 92: harvest

Figure 93: harvest