Image Based Relighting Using Neural Networks

Peiran REN* Yue DONG Stephen LIN Xin TONG Baining GUO
Microsoft Research Asia

Figure 1: *Relighting of various scenes using light transport captured by our method from a small number of images.*

Abstract

We present a neural network regression method for relighting real-world scenes from a small number of images. The relighting in this work is formulated as the product of the scene's light transport matrix and new lighting vectors, with the light transport matrix reconstructed from the input images. Based on the observation that there should exist non-linear local coherence in the light transport matrix, our method approximates matrix segments using neural networks that model light transport as a non-linear function of light source position and pixel coordinates. Central to this approach is a proposed neural network design which incorporates various elements that facilitate modeling of light transport from a small image set. In contrast to most image based relighting techniques, this regression-based approach allows input images to be captured under arbitrary illumination conditions, including light sources moved freely by hand. We validate our method with light transport data of real scenes containing complex lighting effects, and demonstrate that fewer input images are required in comparison to related techniques.

CR Categories: I.2.10 [Artificial Intelligence]: Vision and Scene Understanding—intensity, color, photometry and thresholding; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Color, shading, shadowing, and texture; I.4.1 [Image Processing and Computer Vision]: Digitization and Image Capture—radiometry, reflectance, scanning

Keywords: image based relighting, light transport, neural network, clustering

*e-mail:renpeiran@gmail.com

1 Introduction

The appearance of a scene arises from the transport of light within it. In realistic rendering algorithms, this light transport is computed from complete scene information including geometry, reflectance properties, and lighting environment. With this information, the new appearance of the scene under different illumination can be readily determined. For a real-world scene where such data is usually unavailable, the effects of light transport can instead be inferred from images that exhibit scene appearance under different lighting conditions. Represented in a *light transport matrix* that relates image radiance to lighting condition, this light transport information can be used to relight real-world scenes through computation of a matrix-vector product [Ng et al. 2003]:

$$\mathbf{I} = \mathbf{M}\,\mathbf{L}, \tag{1}$$

where the outgoing radiance $\mathbf{I}$ is expressed as a vector over N_p image pixels, the lighting condition $\mathbf{L}$ is modeled by a vector of incident radiance from N_s light sources, and the light transport matrix $\mathbf{M}$ is of dimension $N_p \times N_s$. Image-based relighting in this manner produces high realism without the need for scene modeling. The key challenge of this approach is in reconstructing the light transport matrix from images acquired of the scene.

Various techniques have been presented in the literature for reconstruction of the light transport matrix. A brute-force solution is to directly measure the entries of the light transport matrix [Debevec et al. 2000; Wenger et al. 2005] by capturing an image of the scene under each of the canonical light sources (corresponding to the columns of the light transport matrix). This requires acquisition of a considerable number of images, and specialized devices are needed to accurately control the lighting. Another approach is to exploit sparseness [Zongker et al. 1999; Peers et al. 2009; Sen and Darabi 2009] or coherence [Fuchs et al. 2007; Wang et al. 2009; O'Toole and Kutulakos 2010] in the light transport matrix to reduce the number of images needed for light transport reconstruction. However, these methods are either designed for specific lighting effects [Zongker et al. 1999] or rely on special hardware to capture images under particular lighting and/or viewing conditions [Fuchs et al. 2007; Wang et al. 2009; O'Toole and Kutulakos 2010]. For scenes with occlusions and high-frequency lighting effects (e.g., hard shadows, sharp specular reflections, and caustics), these techniques require numerous images to reconstruct a high-resolution light transport matrix.

In this paper, we present a method for relighting a real-world scene from a small number of easily acquired images. The key observation in this work is that the entries of the light transport matrix should generally exhibit a local, non-linear coherence, since scene points or light sources in proximity to each other often share significant commonalities in their light transport, yet non-linear variations may exist due to high-frequency lighting effects or surface features. Based on this observation, we model the light transport as a function of pixel coordinates and light source position, and approximate this function with a set of neural networks. The neural networks are trained on images of the scene captured under different known lighting conditions, and then discrete samples of the regressed function are taken to reconstruct the light transport matrix.

Neural networks have been used to compress the low-frequency shadowing [Nowrouzezahrai and Snyder 2009] and indirect light transport [Ren et al. 2013] of synthetic scenes. Both methods assume that the light transport data is completely known, and they utilize neural networks together with physical properties of the scene to compress the data into a compact form. Different from these light transport compression techniques, our method aims to reconstruct the light transport matrix of a real-world scene from just a small subset of the light transport data. This task presents significant challenges. On one hand, light transport in scenes with complex high-frequency lighting effects are difficult to approximate with a neural network. On the other hand, the reconstruction should utilize only a small number of image samples in order to simplify data acquisition. In addition, the physical properties of the imaged scene are unknown.

To tackle these challenges, we develop a neural network model that provides high-quality approximations of light transport with a minimal number of captured images. This design includes four main elements. The first is the use of neural network ensembles [Hansen and Salamon 1990], which are a set of neural networks independently trained on different subsets of the training data. In contrast to a single neural network trained on all the data, a neural network ensemble yields a collective prediction of light transport that is less sensitive to local optima in neural network training. Secondly, we augment the input of each neural network with the average color of image values at each pixel. The average color of pixels provides an indication of their similarity in material and geometric properties. Accounting for this similarity can facilitate neural network modeling of scenes with rich material and geometric variations [Ren et al. 2013]. The third element is the modeling of different parts of the light transport function with different neural network ensembles. We present an adaptive fuzzy clustering scheme for partitioning the light transport space into more coherent segments which can be more effectively approximated by neural networks.

The fourth element of our model, which represents a key step in our solution, is the design of a neural network structure with a suitable ensemble size and an optimized number of nodes. For a larger number of nodes, a greater number of images need to be captured to train the neural network. At the same time, a neural network with more nodes can model a larger segment of the light transport space. Through an empirical analysis, we analyze the quantitative relationship between these factors on a set of representative scenes and derive a neural network configuration for effective reconstruction of light transport in similar scenes with a small number of captured images.

With the proposed design and usage of neural networks, our method effectively exploits the non-linear local coherence in light transport to reconstruct the light transport matrix from a small number of images. We validate this relighting method on existing light transport data, and verify the need for fewer images compared to other light transport reconstruction methods [Wang et al. 2009; O'Toole and Kutulakos 2010] on scenes with complex lighting effects. Moreover, our regression-based reconstruction method allows images captured under any known illumination to be used for training, which obviates the need for special lighting devices. For the lighting in our image acquisition, we freely move a point or linear light source by hand. Results demonstrate this simple manner of capturing images to be practical for our reconstruction method.

2 Related Work

The light transport of a scene can be modeled by an 8D reflectance field [Debevec et al. 2000], which describes the mapping of radiance from an incident light field to an outgoing light field. For ease of capture and processing, most image-based relighting methods consider only a simplified 4D reflectance field with a fixed viewpoint and 2D incident illumination. The light transport matrix provides a discrete representation of this reflectance field.

2.1 Light Transport Reconstruction

Existing methods for light transport reconstruction can be classified into three categories [Wang et al. 2009]: brute force, sparsity based, and coherence based.

Brute force methods directly sample all the entries of the light transport matrix from the scene. Debevec et al. [2000] designed a light stage for capturing the reflectance field of the human face from a fixed viewpoint and with directional lighting. Each matrix column is measured in an image of the subject's face illuminated by a directional light, and the matrix is filled by uniformly sampling the lighting directions over a hemisphere. Wenger et al. [2005] later developed a light stage with high-speed multiplexed illumination to enable reflectance field capture of dynamic characters. The light stage concept was also extended by Hawkins et al. [2005] into a dual light stage where Helmholtz reciprocity is exploited to reverse the roles of the camera and light source. Dense sampling of the lighting domain is obtained with this reversal, which facilitates reflectance field capture of highly reflective objects. O'Toole et al. [2012] presented a primal-dual coding technique that provides control over which light paths contribute to an image. This allows for individual elements of the light transport matrix to be measured. Brute force sampling is limited in practice though by the need for specialized acquisition hardware and a large number of images to be captured.

Sparsity based methods model light transport using a sparse representation that is recovered from images of the scene lit with designed illumination patterns. Several methods assume the light transport matrix itself to be sparse [Masselus et al. 2003; Sen et al. 2005] or data-sparse [Garg et al. 2006]. To accelerate acquisition, multiplexed illumination patterns are devised for capturing multiple columns of the light transport matrix from each image. In environment matting, the light transport of transparent, translucent and glossy objects is modeled as a sparse sum of basis functions, such as rectangular kernels recovered using light stripe hierarchies [Zongker et al. 1999], Gaussian kernels estimated with light stripe sweeps [Chuang et al. 2000], and a wavelet basis computed using wavelet illumination patterns [Peers and Dutré 2003]. Light transport has also been reconstructed in general scenes using sparse basis representations, including hierarchical box functions estimated from various natural illumination conditions [Matusik et al. 2004] and wavelets from scenes lit with wavelet noise patterns [Peers and Dutré 2005]. Sparse basis representations of light transport have also been recovered using compressive sensing techniques, employed in a hierarchical manner [Peers et al. 2009]

and in a dual photography setting [Sen and Darabi 2009]. Reddy et al. [2012] decomposed the light transport of a diffuse scene into direct, near-range, and far-range components that have a sparse representation in either the spatial or frequency domain. Each component is then captured with corresponding optimal illumination patterns.

For scenes with more complicated lighting effects, sparse representations become less adequate for modeling the corresponding light transport matrices of greater complexity. Moreover, these methods require specific illumination patterns defined on the 2D light domain for image acquisition. Our method instead exploits non-linear local coherence to reduce image capture, and sets no particular requirements on lighting. It thus does not rely on special lighting hardware and can handle illumination with higher degrees of freedom.

Coherence based methods exploit the data coherence in light transport to reconstruct the light transport matrix from a subset of rows/columns sampled from the scene. Malzbender et al. [2001] proposed polynomial texture maps for modeling fixed-view images of object surfaces captured under different lighting directions. Vasilescu and Terzopoulos [2004] formulated light transport as a high-dimensional tensor and approximated it with a low-rank multilinear model. Masselus et al. [2004] fit a multilevel B-spline to the measured samples to obtain a smooth approximation of reflectance functions. Fuchs et al. [2005] relit an object under novel lighting using a linear combination of object images captured under different illuminations. The linear combination coefficients are computed from images of a light probe placed near to the object. Later, they presented a reconstruction scheme in which a subset of matrix columns is adaptively captured and then images for nearby light samples are interpolated to recover the unknown columns [Fuchs et al. 2007]. Wang et al. [2009] used a non-linear kernel to map the light transport matrix into a matrix of low rank and then reconstructed it from sparsely sampled rows and columns via a generalized Nystrom scheme. They used two coaxial camera/projector pairs to capture rows and columns of the light transport matrix. O'Toole and Kutulakos [2010] directly measured the low-rank approximation of the light transport matrix with two coaxial camera/projector pairs. A sequence of illumination patterns is used to recover the eigenvectors and eigenvalues of the light transport matrix. These methods can greatly reduce the number of images required for reconstructing the light transport matrix, but they all require special devices for accurate control of lighting. For the high-rank light transport matrix of a scene with complex high-frequency lighting effects, hundreds or thousands of images are required for accurate reconstruction.

Our work also takes advantage of data coherence in light transport, but considers the coherence to be both non-linear and local. The greater flexibility of non-linear modeling permits fewer images to be used in reconstructing high-frequency variations among matrix elements. In addition, by modeling locally in segments of the light transport space that have greater coherence, we obtain more efficient representations that can be reconstructed with fewer images. With the proposed design and usage of neural networks, our method capitalizes on the non-linear and local coherence of light transport for reconstruction from a small set of images. Moreover, it can use images of the scene lit with a freely moving light source as input, thus simplifying image acquisition.

For synthetic scenes, many-lights methods [Walter et al. 2005; Hašan et al. 2007; Ou and Pellacini 2011] approximate one-bounce reflections of light from scene surfaces with millions of virtual lights and then compute their contributions to the image by a multiplication of the light transport matrix and a virtual lights vector, where each matrix element records the direct light transport from a virtual light source to the surface point of an image pixel. Lightcuts [Walter et al. 2005] hierarchically clusters the virtual lights into a light tree and then selects a set of representative lights for each pixel to render the final image. Matrix row-column sampling [Hašan et al. 2007] assumes that the light transport matrix is low rank and samples a sparse set of representative rows and columns for approximating the sum of all matrix columns. LightSlice [Ou and Pellacini 2011] groups the matrix rows into several low-rank submatrices and then applies a row-column sampling scheme to approximate the sum of each submatrix. Different from these methods that exploit local or global linear coherence for reconstructing the light transport of one-bounce indirect lighting, our method models the full lighting effects of light transport in a scene by exploiting the nonlinear coherence of local light transport segments.

2.2 Light Transport Compression

Several methods have been presented for compressing full light transport matrices captured from real-world scenes or precomputed from synthetic scenes. Precomputed radiance transfer methods [Sloan et al. 2002; Ramamoorthi 2009] project the light transport data onto a set of basis functions (e.g., spherical harmonics [Sloan et al. 2002], wavelets [Ng et al. 2003], or spherical radial basis functions [Tsai and Shih 2006]) and then compress the coefficients with clustered PCA [Sloan et al. 2003] or other data compression schemes [Tsai and Shih 2006]. Nowrouzezahrai et al. [2009] use neural networks for approximating low-frequency precomputed visibility data of dynamic objects and rendering low-frequency self-shadowing in a dynamic scene. All of these methods assume the entire light transport matrix to be known, and exploit coherence for compression. On the contrary, our method takes a small subset of the light transport data and infers the rest of the matrix entries based on the non-linear coherence revealed in the images.

Recently, Ren et al. [2013] proposed radiance regression functions (RRF) based on neural networks for rendering the indirect light transport of a synthetic scene. To model the indirect transport, they augmented the RRF input with surface normal and material properties, partitioned the input space, and constructed an RRF for each subdivision. In contrast to the RRF method which employs neural networks as a compact representation of densely-sampled light transport data, our method uses neural networks in an inverse manner to predict the light transport of unobserved lighting conditions from a small number of image samples. The demonstrated ability of neural networks to model light transport as shown by Ren et al. [2013] suggests potential in using neural networks as an instrument for light transport inference.

Mahajan et al. [2007] theoretically analyzed the relationship between the dimensionality of local light transport and the underlying image region size, and derived optimal parameters for clustered PCA. Since our method uses non-linear neural network ensembles to model light transport, their analysis cannot directly be applied in our partitioning of the light transport space. We instead empirically analyze the relationship between the node count of neural networks and the region size. Based on this analysis, we infer an optimal neural network structure for recovering the light transport matrix from a minimal number of images.

3 Neural Networks for Light Transport

In this section, we describe how to formulate the light transport matrix as discrete samples of a continuous light transport function, and then show how we approximate the transport function using neural networks. Without loss of generality, we consider light transport for a fixed viewpoint and with point light sources that lie on a 2D

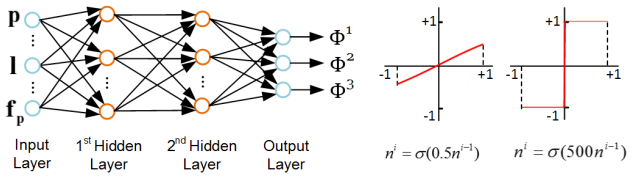

Figure 2: *Left: Structure of the acyclic feed-forward neural network used in modeling a light transport function. An input vector* $(\mathbf{p},\mathbf{l})$ *is mapped to an output of RGB values in the light transport matrix* $\mathbf{M}$*. Right: Examples of activation functions with different weight values.*

plane. Our solution can be directly extended to handle other light transport configurations.

Light transport function The light transport matrix represents the proportion of radiance from each light source that reaches each image pixel. Given a real-world scene, we model the light transport matrix as discrete samples of a continuous light transport function $\Psi(\mathbf{p},\mathbf{l})$:

$$\mathbf{M}(i,j) = \Psi(\mathbf{p}(i),\mathbf{l}(j)), \tag{2}$$

where $\mathbf{M}(i,j)$ is an element in the light transport matrix that corresponds to pixel i and light source j, $\mathbf{p}(i)$ denotes the image coordinates of pixel i, and $\mathbf{l}(j)$ is the position of light source j in the 2D light domain. By expressing the 2D light transport matrix as a continuous 4D light transport function, the coherence of light transport in both the image domain and light domain can be more readily exploited.

Neural network approximation We approximate the light transport function with multilayer acyclic feed-forward neural networks. As a universal function approximator, a neural network can fit any function with arbitrary accuracy given adequate network size and training data. As shown in Figure 2, a multilayer acyclic feed-forward neural network can be illustrated as a weighted and directed graph with layers of nodes. The first layer is the input layer, which consists of nodes representing each element of the input vector $(\mathbf{p},\mathbf{l})$ of the light transport function. The final layer, called the output layer, consists of three nodes whose outputs are taken as the RGB components of light transport matrix element $\mathbf{M}(i,j)$. The layers in-between are referred to as two hidden layers, which transform the input into values useful to the output layer. At the j-th node in the i-th layer, the node output n_j^i is computed from a weighed sum z_j^i of the outputs from the preceding $(i-1)$-th layer:

$$n_j^i = \sigma(z_j^i), \quad z_j^i = w_{j0}^i + \sum_{k>0} w_{jk}^i n_k^{i-1}, \tag{3}$$

where n_k^{i-1} is the output of the k-th node in the $(i-1)$-th layer, w_{j_k} is the weight of the edge from node k to node j, and w_{j0} is a bias value. The weighted sum is transformed by an activation function σ to obtain the node output. The activation function we use here is a hyperbolic tangent function $\sigma(z) = 2/(1+e^{-2z}) - 1$, which can model both smooth functions and step functions with sharp changes as shown in Figure 2. With this model, the output of a neural network can be determined by the input vector $(\mathbf{p},\mathbf{l})$ and the weights $\mathbf{w}$ of all the nodes. Thus we can approximate the light transport function $\Psi(\mathbf{p},\mathbf{l})$ by a neural network function $\Phi(\mathbf{p},\mathbf{l},\mathbf{w})$. Please refer to Section 6 for neural network design details.

Augmentation of neural network input Figure 3 illustrates 2D slices of 4D light transport functions approximated by neural networks with two hidden layers. Each slice image corresponds to

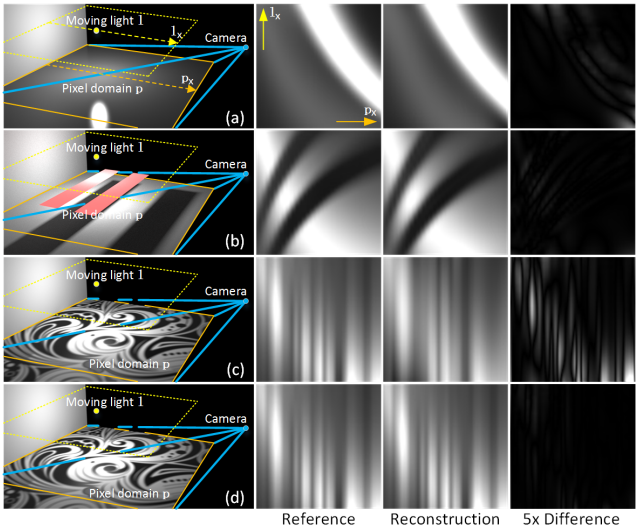

Figure 3: *2D slices of 4D light transport functions approximated by neural networks. The first column shows the scene and capture configuration. A neural network can accurately represent high-frequency variations in light transport functions with glossy reflections as in (a) and hard shadows as in (b). However, it poorly models the rich material variations in (c). With the help of average color, light transport with rich material variations can be well reconstructed as in (d).*

light transport between a 1D light sample and a 1D image scanline. It is shown that a neural network can accurately represent high-frequency variations such as those caused by hard shadows in a light transport function. However, this neural network cannot effectively model the more complex light transport of a scene with rich material variations. On synthetic data, Ren et al. [2013] demonstrated that by augmenting the neural network inputs with the geometry and material properties of surface points, the neural network can much more accurately model the complex light transport of such scenes.

In our case, since the geometry and material information of the scene are unknown, we augment the neural network input with the average color value $\mathbf{f}$ of pixel $\mathbf{p}$ in the captured images, which results in the neural network function $\Phi(\mathbf{p},\mathbf{l},\mathbf{f}_\mathbf{p},\mathbf{w})$. The average color of a pixel over all the captured images provides an indication of its similarity to other pixels in material and geometric properties, as discussed in Appendix A. This information can aid a neural network in modeling the light transport of scenes with rich material variations, as shown in Figure 3(d).

4 Light Transport Reconstruction

To reconstruct the light transport matrix, we recover the light transport function through neural network regression on captured images. In this regression, we solve for the weight vector $\mathbf{w}$ of the neural network such that it best approximates the images. This is non-trivial task because the optimization is highly non-linear with many local minima. Although increasing the number of acquired images can help to avoid a suboptimal result or overfitting, this would run counter to the goal of simplifying data capture.

To deal with this issue, we model light transport functions with neural network ensembles [Hansen and Salamon 1990]. A neural network ensemble Φ_E is composed of several *base* neural networks, each of which is independently trained from a different subset of the captured images. A light transport matrix element is then ap-

proximated by averaging the outputs of all the base neural networks Φ_n:

$$\Phi_E(\mathbf{p},\mathbf{l},\mathbf{f_p}) = \frac{1}{N_e}\sum_{n=1}^{N_e}\Phi_n(\mathbf{p},\mathbf{l},\mathbf{f_p},\mathbf{w}_n), \quad (4)$$

where N_e is the number of base neural networks in the ensemble, and $\mathbf{w}_n$ is the weight vector of base neural network Φ_n. By determining the output collectively from multiple neural networks that have been separately trained, the effect of local optima is reduced in a statistical manner. Note that when considering the fuzzy adaptive clustering in Section 5, the set of ensembles can be different for pixels in different clusters. For simplicity, we will assume in this section that the neural networks are trained with all the pixels of the scene.

Given a set of images $\mathbf{I}_1 \ldots \mathbf{I}_{N_m}$ captured from the scene with N_m different lighting conditions $\mathbf{L}_1 \ldots \mathbf{L}_{N_m}$, we generate each base neural network Φ_n in the ensemble by computing weight vectors $\mathbf{w}_n$ that minimize the following error function:

$$E(\mathbf{w}_n) = \sum_{m=1}^{N_m} ||\mathbf{I}_m - \mathbf{M}_n\mathbf{L}_m||^2 \quad (5)$$

where m is an index of the N_m measurements, and $\mathbf{M}_n$ is the light transport matrix modeled by base neural network Φ_n:

$$\mathbf{M}_n(i,j) = \Phi_n(\mathbf{p}(i),\mathbf{l}(j),\mathbf{f_p}(i),\mathbf{w}_n). \quad (6)$$

The distance between a measured image and a reconstructed image is computed as a sum of L2 distances between all pairs of pixel colors.

The weight vector $\mathbf{w}_n$ is initialized using the Nguyen and Widrow [1990] method, which normalizes a randomly initialized weight vector to guarantee that each node will be active for at least part of the training data. After initialization, we solve for the weight vector using the Levenberg-Marquardt (LM) algorithm [Hagan and Menhaj 1994], which performs well on the small-scale neural networks used in our network ensembles. In each step of the optimization, we determine the Hessian and gradient by computing the Jacobian matrix of $E(\mathbf{w}_n)$ with respect to $\mathbf{w}_n$ using the standard back-propagation scheme [Hinton 1989]. The training is performed in batch mode, where all the given training data is used at each iteration. To avoid overfitting, we randomly select 70% of the captured images as training images and leave the remaining 30% for cross-validation [Beale et al. 2012]. This random selection of training data can also be used to generate different base neural networks in the network ensemble.

After regression, we can reconstruct the light transport matrix from the resulting base neural networks in the ensemble:

$$\mathbf{M}(i,j) = \frac{1}{N_e}\sum_{n=1}^{N_e}\Phi_n(\mathbf{p}(i),\mathbf{l}(j),\mathbf{f_p}(i),\mathbf{w}_n). \quad (7)$$

5 Adaptive Fuzzy Clustering

A neural network ensemble can accurately represent the coherent light transport in a local region of the light transport space. But as light transport decreases in coherence more globally, it becomes more difficult to model effectively with a single neural network ensemble. We therefore partition the light transport space into coherent segments and regress a neural network ensemble for each of them. Since light transport generally exhibits less coherence in image space than in lighting space, we partition the light transport in only the image domain.

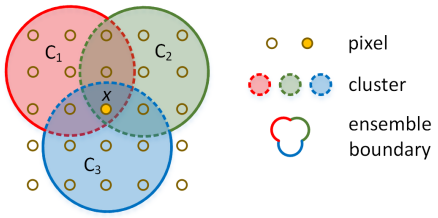

Figure 4: *Fuzzy clustering, illustrated for three base neural networks ($N_e = 3$). For a pixel x, its input image values are used in training the neural networks for the nearest three clusters, C_1, C_2 and C_3. The resulting three neural networks are used as a neural network ensemble for determining the value of pixel x.*

A straightforward partitioning method is to divide the image space evenly into uniform regions. However, this solution ignores the coherence between pixels across region boundaries, which may lead to artifacts in the reconstructed light transport. Also, it would be inefficient to divide a large region with coherent light transport variations, as this would lead to a redundant representation.

To address these issues, we develop a fuzzy clustering scheme for image space partitioning. As shown in Figure 4, our method assigns each pixel to several clusters whose centers are nearest to the current pixel, with the distance measured in terms of pixel position[1]. Since neighboring clusters will thus have overlapping image regions, they share training data from within the overlaps, leading to smooth transitions in light transport between clusters. Moreover, since the light transport of each pixel is to be modeled by the neural network ensembles of several neighboring clusters, we can instead train a single neural network for each cluster and use the set of neural networks from all the clusters that the pixel belongs to to form its neural network ensemble. The fuzzy clustering is performed in an adaptive, hierarchical manner that allows for differently-sized regions to emerge. Starting with clusters at a coarse level, they are refined into smaller clusters only if their light transport cannot be accurately reconstructed with the existing neural network ensembles.

In the remainder of this section, we first present the basics of our fuzzy clustering method and then describe the details of our adaptive fuzzy clustering scheme.

Fuzzy clustering We cluster all pixels defined in the image plane according to their 2D image-space distances. Given the number of clusters N_c, we uniformly sample N_c pixels in the image plane as initial cluster centers and then group all the pixels via standard k-means clustering. The number of clusters is determined adaptively as described later. After the centers of the final clusters are determined, we assign each pixel to the N_e nearest clusters according to its distance to the cluster centers and train a single neural network for each cluster.

Adaptive fuzzy clustering To partition the image space adaptively according to its light transport variations, we first cluster all the pixels at N_h different levels ($N_h = 4$ in our implementation). The finest level H_0 contains a maximal number of clusters ${N_c}^0$, while each higher level h contains a quarter as many clusters as in level $h-1$, i.e., ${N_c}^h = {N_c}^{h-1}/4$.

The maximal number of clusters ${N_c}^0$ is determined such that each cluster would contain enough pixels for neural network training.

[1] We found that the pixel's appearance has little effect on our clustering results, so it is not included in the distance computation.

input : N_h levels of clusters from finest level 0 to coarsest level $N_h - 1$;
error threshold $\varepsilon = 0.03$.
output: Clustering level $h(\mathbf{p})$ and cluster IDs $\mathbf{c}(\mathbf{p})$ for each pixel $\mathbf{p}$;
Trained neural networks $\Phi_{(i,h)}$ for cluster i at level h.

initialization: $h(\mathbf{p}) := N_h - 1$, $\mathbf{c}(\mathbf{p}) :=$ nearest N_e clusters in level $h(\mathbf{p})$;
for *all pixels* **p do**
  // train one neural network with all the data in each cluster
  Train $\Phi_{(\mathbf{c}(\mathbf{p}),h(\mathbf{p}))}$;
  Measure training error $E(\mathbf{p})$;
  while $\frac{E(\mathbf{p})}{||\mathbf{I}(\mathbf{p})||^2} > \varepsilon$ ***and*** $h(\mathbf{p}) > 0$ **do**
    $h(\mathbf{p}) := h(\mathbf{p}) - 1$;
    $\mathbf{c}(\mathbf{p}) :=$ nearest N_e clusters in level $h(\mathbf{p})$;
    // train a neural network with all the data in each cluster, if the cluster contains one or more flagged pixels.
    Train $\Phi_{(\mathbf{c}(\mathbf{p}),h(\mathbf{p}))}$;
    Update training error $E_{\mathbf{p}}$;

Algorithm 1: Adaptive clustering scheme

As later explained in Section 6, the minimal number of pixels N_t required for training is calculated as $N_t = \frac{25N_w}{N_m}$, where N_w is the number of neural network weights and N_m is the number of captured images (i.e., the number of samples in the light domain). The maximal number of clusters ${N_c}^0$ is then computed from N_t and the total number of image pixels N_p as ${N_c}^0 = \lfloor \frac{N_p}{N_t} \rfloor$.

After computing the fuzzy clusters at each level, we determine the level at which each pixel should be modeled. We start by training neural networks for all the clusters at the coarsest level. Then we compute the training error $E(\mathbf{p}_i)$ of each pixel $\mathbf{p}_i$ as the squared difference between the values predicted by the light transport of its neural network ensemble and the true values $\mathbf{I}(\mathbf{p}_i)$ in the captured images:

$$E(\mathbf{p}_i) = \sum_{m=1}^{N_m} ||\mathbf{I}_m(\mathbf{p}_i) - \mathbf{M}(i,\cdot)\mathbf{L}_m||^2, \tag{8}$$

where $\mathbf{M}(i,\cdot)$ is the row vector of the light transport matrix that corresponds to pixel $\mathbf{p}_i$. Pixels whose relative training error $\frac{E(\mathbf{p}_i)}{||\mathbf{I}(\mathbf{p}_i)||^2}$ is smaller than a threshold (0.03 in our implementation) are assigned to this level, while the others are flagged as having a poorly approximated light transport. These flagged pixels are then evaluated again at the next finer level of the hierarchy. At the next level, neural networks are trained for the clusters that contain flagged pixels. We repeat this process until no flagged pixels remain or the algorithm reaches the finest level. Algorithm 1 summarizes this adaptive fuzzy clustering scheme. We note that for each cluster, we train its neural network with the captured image values of all of its pixels regardless of whether a pixel is flagged, in order to ensure a sufficient number of samples for neural network training.

After adaptive clustering, we find for each pixel the N_e clusters nearest to it at its assigned level h. The neural networks of these N_e clusters are taken as the base neural networks for light transport reconstruction at this pixel.

Through this hierarchical processing, our adaptive fuzzy clustering method identifies for each pixel a clustering level with sufficient local coherence to accurately model its light transport. By sharing neural network ensembles with neighboring pixels, the coherence within the light transport is well preserved. In addition, this method does not require training a neural network ensemble for each cluster. The total number of neural networks used for light transport reconstruction increases with the number of clusters but is independent of the number of base neural networks in an ensemble. Figure 5(c) illustrates relighting results generated by our adaptive fuzzy clustering scheme.

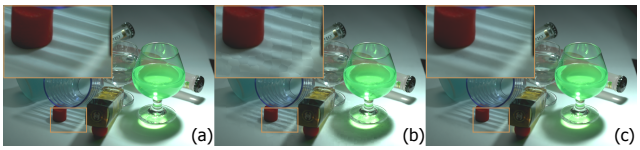

Figure 5: *Fuzzy vs. non-fuzzy clustering. (a) Ground truth. (b) Result from our adaptive method but with hard clustering (i.e., $N_e = 1$). (c) Result from our adaptive fuzzy clustering scheme, with $N_e = 5$.*

6 Design of Neural Network Ensembles

In the previous two sections, we described the adaptive fuzzy clustering scheme and the neural network regression method for light transport reconstruction. What remains to be presented is the structure of the base neural networks and the number of base neural networks in an ensemble. Since the aim of this work is to simplify image acquisition, we design them so that light transport can be reconstructed from a minimal number of images.

Design of Base Neural Networks For the acyclic feed-forward neural network described in Section 2, we utilize two hidden layers that have an equal number of nodes, as this configuration has been shown to be effective for light transport modeling [Ren et al. 2013]. What we seek to determine here is the number of nodes in each hidden layer. Since a neural network with more nodes provides greater representational power but requires more training data, a proper balance between these two considerations needs to be found.

To determine the number of nodes N_n that we will use in each hidden layer, we solve for the value of N_n that minimizes the number of required training images $\frac{T(N_n)}{R(N_n)}$, where $T(N_n)$ is the number of samples needed to train a base neural network with N_n nodes in each hidden layer, and $R(N_n)$ is the cluster size (i.e., number of pixels per cluster) that can be modeled by a base neural network with N_n nodes in each hidden layer.

Previous analysis [Turmon and Fine 1995] indicates that the number of samples T required to train a neural network is proportional to the number of neural network weights N_w. For our base neural networks with seven nodes in the input layer and three nodes in the output layer, the number of weights can be calculated as $N_w = 7N_n + N_n + {N_n}^2 + N_n + 3N_n + 3$, so the number of samples needed in our case is

$$T(N_n) = \rho N_w = \rho(12N_n + {N_n}^2 + 3), \tag{9}$$

where ρ is a constant scale factor. In Figure 6(d) we show a plot of $T(N_n)$, which increases quadratically with the number of nodes N_n.

The function $R(N_n)$ is more challenging to analyze since it depends on scene properties and our clustering scheme in addition to neural network structure. We determine it empirically by examining how cluster sizes change when using neural networks with different numbers of nodes. This analysis is conducted on three representative light transport datasets with different kinds of lighting effects: the Plant dataset from a synthetic scene with high frequency shadows (Figure 6(a)), the Glass dataset captured from a real scene with

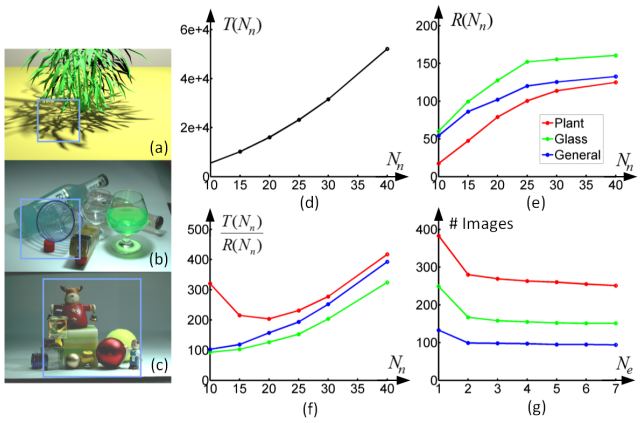

Figure 6: *Effects of different node numbers N_n. (a-c) Scenes with various lighting effects that are used for analysis. (d) Number of needed training samples $T(N_n)$. (e) Cluster size $R(N_n)$. (f) Number of required training images $\frac{T(N_n)}{R(N_n)}$. (g) Required training images with respect to different ensemble sizes.*

complex caustics (Figure 6(b)), and the General dataset sampled from another real scene with a mixture of different lighting effects (Figure 6(c)). For efficient experimentation, we select light transport data defined in an image region (enclosed by blue boxes) that covers the major lighting effects of the scene, and resample the data to a 256×256 image resolution. After that, we execute our clustering and regression scheme on each dataset with different N_n settings for base neural networks. For each N_n sampled from 10 to 40, we set $N_e = 1$ and use the full light transport matrix as training data for regression. The total number of clusters N_c used for light transport reconstruction is recorded, and we compute the average number of pixels in the resulting clusters by

$$R(N_n) = \frac{256^2}{N_c}, \tag{10}$$

where 256^2 is the resolution of the light transport datasets. In Figure 6(e), $R(N_n)$ is plotted for each of the three datasets and is shown to exhibit consistent behavior, with $R(N_n)$ increasing sub-linearly with the number of nodes N_n.

With this analysis of $R(N_n)$ and $T(N_n)$, we can compute the number of required training images for different N_n settings as $\frac{T(N_n)}{R(N_n)}$. We plot this function in Figure 6(f) for the three light transport datasets. Though the datasets contain different kinds of lighting effects, the number of required training images is approximately minimized at $N_n = 15$ and gradually increases for larger N_n. Here, we generated the plots with $\rho = 25$, but the location of the minima is independent of this scale factor. Based on this analysis, we use 15 nodes per hidden layer in our base neural networks. We also considered using different numbers of nodes for the two hidden layers, but we empirically found that this does not reduce the number of required images.

Ensemble Size Optimization After designing the base neural network structure, we optimize the ensemble size to further reduce the number of images required for training. To this end, we again use the three light transport datasets of Figure 6(a-c) to empirically determine the minimal number of images needed for training neural network ensembles with different numbers of base neural networks. For each ensemble size N_e, we compute clusters and train neural network ensembles starting with the full light transport data. Then we iteratively reduce the number of input images in the training data and find the minimal number of images at which the relative training error of each pixel remains smaller than a pre-defined threshold (0.03 in our implementation). Figure 6(g) displays how the number of images required for training decreases as the ensemble size increases for the three light transport datasets. Note that for all three datasets, the number of images required for training becomes stable after the ensemble size becomes larger than 3. However, we found that the resulting light transport exhibits some visual artifacts when the ensemble size is smaller than 5. Therefore, we set the number of base neural networks to $N_e = 5$ in our implementation.

Scene	Image Resolution	Light Resolution	Data from
Glass	1752x1168	34x34	[Wang et al. 2009]
General	1752x1168	34x34	[Wang et al. 2009]
Waldorf	696x464	32x32	[O'Toole and Kutulakos 2010]
Bull	696x464	32x32	[O'Toole and Kutulakos 2010]

Table 1: *Properties of four light transport datasets used for validation.*

Once the values of N_n and N_e are determined, we use this neural network configuration for reconstructing the light transport of all scenes described in the paper.

7 Validation

We validate our method using four light transport matrices acquired in previous works [Wang et al. 2009; O'Toole and Kutulakos 2010; Ren et al. 2013]. This data consists of various lighting effects, including caustics, specular inter-reflections, hard shadows, and low-frequency diffuse inter-reflections. The datasets are all captured with a fixed viewpoint and with lighting densely sampled over a uniform 2D grid. Table 1 lists the properties of the four data sets.

For each dataset, we randomly select a sparse set of columns as input images and then reconstruct the full light transport matrix with our clustering and training scheme. For comparison, we also reconstruct the light transport matrix with the same number of input images using the kernel Nystrom [Wang et al. 2009], optical Arnoldi [O'Toole and Kutulakos 2010], and radiance regression function (RRF) [Ren et al. 2013] methods. Since the kernel Nystrom method requires sampling both rows and columns of the light transport matrix, we randomly select $N/2$ rows and $N/2$ columns as the N input images. For the optical Arnoldi method, we use $N/4$ optical Arnoldi iterations for reconstruction since each iteration requires four images. The relative reconstruction error is computed as

$$\varepsilon = \sqrt{\frac{\sum_j ||\mathbf{I}_j - \tilde{\mathbf{I}}_j||^2}{\sum_j ||\mathbf{I}_j||^2}}, \tag{11}$$

where $\mathbf{I}_j$ is the image of the ground truth light transport matrix column $\mathbf{M}(\cdot, j)$, and $\tilde{\mathbf{I}}_j$ is the reconstructed one.

In Figure 7, the reconstruction error curves of the three methods are plotted with respect to the number of captured images. The results are shown for each dataset. As the number of input images increases, the reconstruction errors of all the methods are quickly reduced. With the same number of input images, our method provides reconstructions with less reconstruction error. Figure 7 shows a comparison of the ground truth image and images reconstructed by the four methods for a matrix column that is not among the input images. The number of input images used for each dataset is indicated by the vertical yellow line in the error plots of Figure 7. While the result generated by our method looks almost indistinguishable from the ground truth, the results generated by the other methods

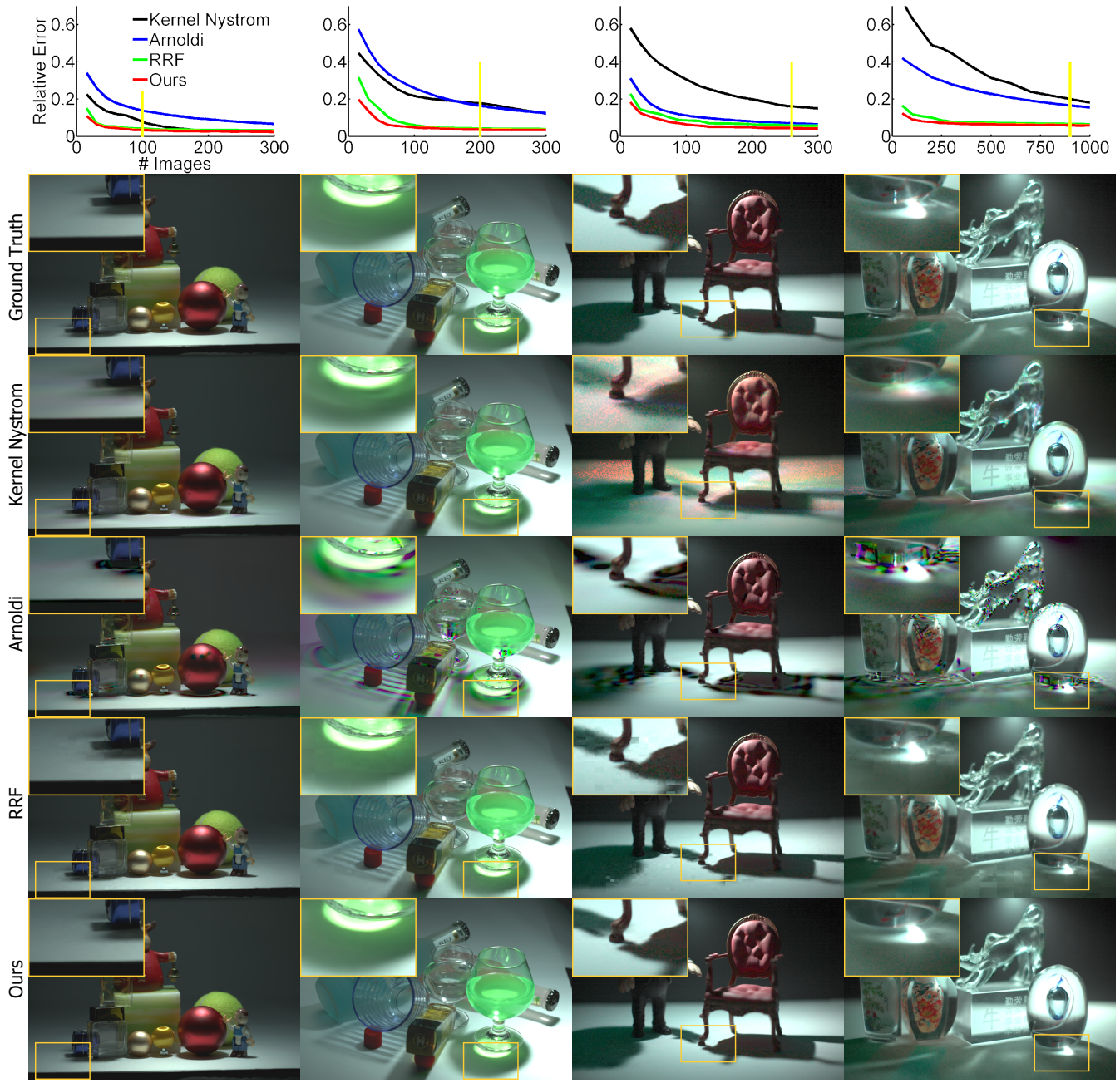

Figure 7: *Comparisons of light transport matrix reconstruction. Left to right columns: General, Glass, Waldorf, and Bull scenes. The first row shows the relative reconstruction errors for different numbers of input images. Red: our method; Green: Radiance Regression Functions (RRF) method; blue: kernel Nystrom method; and black: optical Arnoldi method. The second row shows ground truth images of a light transport matrix column that is not in the input image set. The last four rows are images of the same matrix column reconstructed using the kernel Nystrom method, the optical Arnoldi method, the RRF method, and our method, respectively, where the number of input images used for reconstruction is 100 for the General scene, 200 for the Glass scene, 260 for the Waldorf scene, and 900 for the Bull scene.*

exhibit visual artifacts. To achieve relative reconstruction errors similar to ours, the kernel Nystrom, optical Arnoldi, and radiance regression function methods need considerably more images than our method does, as shown in the supplemental material.

To examine the sensitivity of our method to different input images, neural network initializations, and clustering initializations, we repeated our reconstruction five times, each time with different input images randomly selected from the light transport data and with randomly initialized neural networks and clusters. The relative reconstruction errors and their variances are plotted in Figure 8 for different numbers of input images. As the number of input images increases, both the reconstruction error and its variance decreases quickly.

We also note that the Waldorf and Bull scenes contain significant noise due to low-dynamic range image acquisition. Although the reconstruction error of our method may increase with higher levels

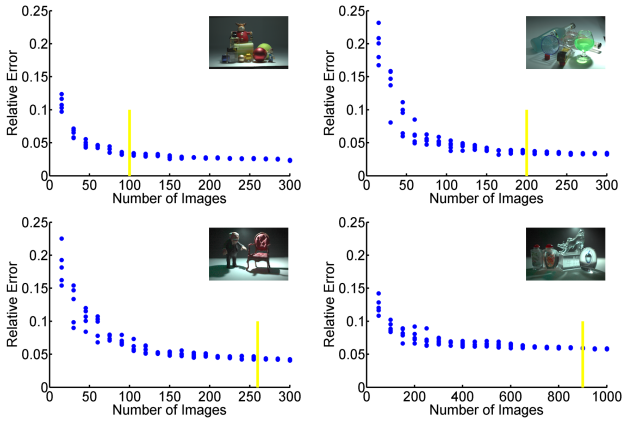

Figure 8: *Relative reconstruction errors and their variance for our method with different training samples and initializations. The yellow line indicates the number of images needed to reconstruct the light transport matrix with a predefined reconstruction error threshold of 0.03.*

of noise, it is seen for these two examples in Figure 7 that the error levels are still relatively low for a small number of input images.

8 Experimental Results

In this section, we demonstrate the ability of our method to model high-frequency light transport and use input images with easily produced lighting conditions. The three scenes shown in Figure 10 are used for this experiment. The Toolset scene includes sharp specular reflections and glossy interreflections. The Horse scene consists of sharp specular reflections and caustics, and the Indoor scene exhibits hard shadows and sharp caustics.

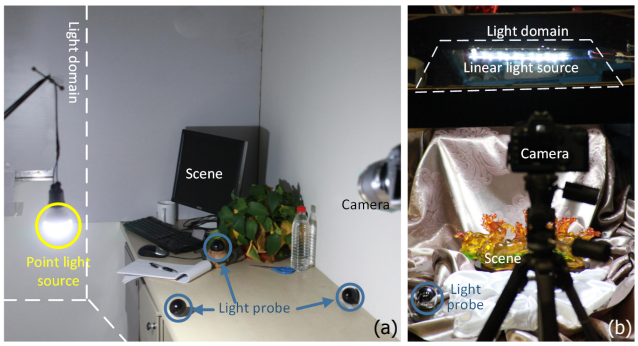

Figure 9: *Imaging setup. (a) For a three-dimensional light domain. (b) For a two-dimensional light domain.*

Image Acquisition The three datasets are captured from a fixed viewpoint with a Canon 5D Mark II camera. The light sources are moved freely by hand. Figure 9 displays the imaging setup. To recover the light source position, we placed three light probes near the target scene and compute the light source position using the method of Nayar [1989]. To constrain light movement to a 2D plane, we placed a glass panel above the scene and kept the light source in contact with the glass. At each light source position, we captured a set of RAW images for construction of a high dynamic range image. For the Toolset scene, we captured 200 images with a point light source moved freely by hand on the 2D glass panel. For the Horse scene, we illuminated the scene with a linear light source composed of 12 LEDs that was also manually moved on the glass panel. A total of 300 images were taken for reconstructing light transport between image pixels and point light sources that lie on the same 2D plane. For the Indoor scene, we captured 400 images with a point light source moved by hand inside a 3D volume. The acquisition process for each scene took about ten minutes.

Performance We reconstructed the full light transport matrix from the captured images on a PC cluster with 100 nodes, each of which is configured with two Quadcore Intel Xeon L5420 2.50G CPUs. Table 2 lists the performance of our reconstruction algorithm for the three datasets as well as statistics on the resulting neural network ensembles used for light transport reconstruction. For all three datasets, the clustering and neural network regression takes about one to two hours on the cluster. Though somewhat long, this training time comes with significant savings in acquisition effort. Reconstructing the light transport matrix with 1024^2 pixels and 32^2 lightings from the neural network ensembles takes about 30 minutes on a single PC. By adapting the GPU rendering code in Ren et al. [2013], the speed can likely be increased by at least one order of magnitude.

To evaluate the accuracy of a reconstructed light transport matrix, we captured a set of test images from each scene, each of which (including the Horse scene) is lit with a point light source randomly sampled in the light domain. The number of test images is set to be equal to the number of training images. We computed reconstruction errors for the three datasets according to Equation 11, and list them in Table 2. Figure 10 compares images captured from the real scenes with images rendered from the reconstructed light transport matrix with the same light source position. It can be seen that our method effectively reconstructs the light transport matrix of the scenes and accurately reproduces all of the high-frequency lighting effects. Moreover, this experiment demonstrates the high-quality reconstructions that are obtained even with simple handheld lighting.

Figure 11 compares our method with simple linear interpolation method. Note that the light transport generated by our method faithfully reproduces the image of the scene under new lighting, while the result generated by linear interpolation illustrates clear ghosting artifacts.

Scene	Number of Images	Image Resolution	Cluster Number	Model Storage	Training Time	Error
Toolset	200	1047×776	4617	32MB	1.6h	3.8%
Horse	300	972×709	3896	29MB	1.0h	3.9%
Indoor	400	874×636	2927	23MB	0.8h	2.1%

Table 2: *Properties of the three light transport datasets, and statistics of our reconstruction algorithm on the three datasets. Number of images represents how many images are used in light transport reconstruction. Cluster number shows the total number of clusters in the trained model. Model storage represents the storage size for neural network weights, pixel IDs for each cluster, and average color values used as neural network input. Training time is for adaptive fuzzy clustering and neural network regression on a PC cluster. Error is the reconstruction error of the light transport matrix computed from a set of test images.*

Relighting Results After the light transport matrix of a scene is reconstructed, we can relight the scene with new illumination conditions. Figure 12 (first row) illustrates image-based relighting results of the Toolset scene rendered with three rotating point light sources. The sharp anisotropic highlights, hard shadows and glossy inter-reflections in the scene are faithfully reproduced by our reconstructed light transport matrix.

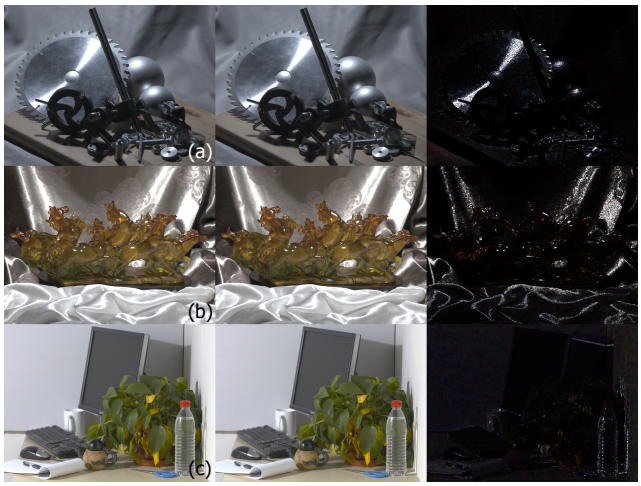

Figure 10: *Reconstruction results for the three light transport datasets. The first column shows ground truth images captured of real scenes. The second column shows images rendered from the reconstructed light transport matrix with the same point light source position as in the ground truth. The third column shows error maps amplified by 5×. (a) Toolset scene. (b) Horse scene. (c) Indoor scene.*

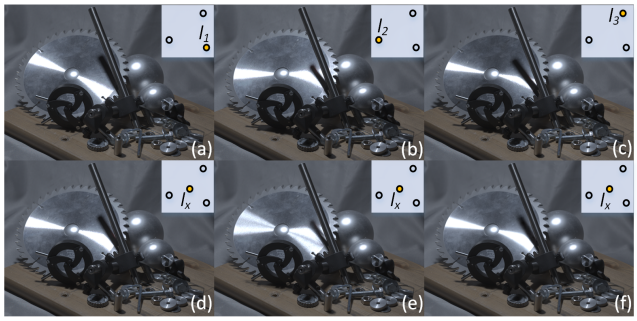

Figure 11: *Comparison between our method and simple interpolation. (a-c) Three input images lit with light sources at l_1, l_2, and l_3, respectively. (d) Ground truth image for light at a new position l_x. (e) Result rendered by linearly interpolating the three nearby inputs shown in (a-c). (f) Result generated by our method. The training set contains no image illuminated by a light inside the triangle formed by l_1, l_2, and l_3.*

In Figure 12 (second row), we relight the Horse scene with three rotating point light sources. The fine-scale changes in the cloth highlights and the volumetric scattering of the horse model are well reproduced.

Relighting in a 3D light domain is especially challenging because of the more dramatic changes in lighting effects. In Figure 12 (third row), we show relighting results of the Indoor scene illuminated with a point light source that is moved in the 3D light domain. The positions of the light source used for rendering are different from those of the input images used for reconstruction. Computed from 400 input images, the reconstructed light transport matrix captures both sharp shadows and caustics, as well as the low-frequency interreflections of the scene. Please see the accompanying video for relighting results of the three scenes under dynamic illumination.

Limitations Though our method exploits local coherence to recover light transport variations, this may not be enough in some cases to recover subtle lighting effects that are not present in the training images. Figure 13 illustrates one example, where the caustic details are lost in the relighting result rendered using our reconstructed light transport matrix. To avoid losing such details, our method may need hundreds of images for training, as shown for the three scenes with high-frequency lighting effects in Figure 10. To faithfully reconstruct the light transport matrix of a scene, our technique also requires the light samples of the training images to be well-distributed over the light domain, since it may not adequately extrapolate beyond the convex hull of the sampled light positions. Although the neural network configuration derived from the three datasets works well for all the scenes described in the paper, it may not be as effective in modeling the light transport of a scene with different lighting effects. Fortunately, we can derive a suitable neural network configuration by simply performing the analysis of Section 6 with the new light transport data included.

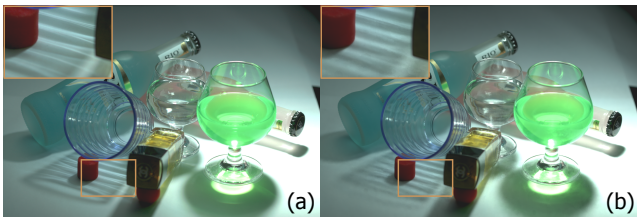

Figure 13: *Failure case. (a) Real photo. (b) Reconstructed image with missing caustic details that were not present in the training images and not recovered from local coherence.*

9 Conclusion

We presented a neural network based regression method for image based relighting. Our method exploits the non-linear coherence of light transport in local image regions and models the light transport with neural network ensembles built within an adaptive fuzzy clustering framework. We studied the relationship between the number of neural network nodes and the size of image region that it can model, and use this analysis in designing a neural network structure that can can be used to capture the light transport matrix from a small number of input images. Compared to other light transport acquisition methods, ours requires fewer input images for the same level of reconstruction quality, and does not need special lighting devices.

We believe that our method makes relighting much more accessible to practitioners, as it can utilize simple lighting conditions (such as manually moved point or linear light sources) and requires relatively few images. Moreover, it can easily support 3D light domains (as shown with the Indoor scene) without the specialized light control devices or substantial image data needed in previous methods. An important yet challenging future direction is to avoid training for each new scene by using a learned dictionary of NNs, where each NN models the reflectance field of a small patch as in Marwah et al. [2013]. We also hope that our analysis of model parameters and light transport complexity will encourage further study of other regression models to better understand what kind of representations are most suitable for light transport.

Acknowledgements

The authors thank Zheng ZHANG, Jiaxing ZHANG, Dong YU, Zhiheng HUANG for insightful discussions on deep neural networks, Yi MA, Gong CHENG, Jinyu LI on robust PCA, and Yang LIU on non-linear optimization. The authors also thank the anonymous reviewers for their helpful suggestions and comments. The Waldorf and the Bull scenes are from the public data shared by

Figure 12: *Relighting results. First row: Toolset scene. Second row: Horse scene. Third row: Indoor scene. The three results of each scene are rendered with a point light source at different positions.*

Matthew O'TOOLE and Kiriakos N. KUTULAKOS.

References

BEALE, M. H., HAGAN, M. T., AND DEMUTH, H. B. 2012. Neural network toolbox users guide.

CHUANG, Y.-Y., ZONGKER, D. E., HINDORFF, J., CURLESS, B., SALESIN, D. H., AND SZELISKI, R. 2000. Environment matting extensions: Towards higher accuracy and real-time capture. In *Proceedings of SIGGRAPH 2000*, 121–130.

DEBEVEC, P., HAWKINS, T., TCHOU, C., DUIKER, H.-P., SAROKIN, W., AND SAGAR, M. 2000. Acquiring the reflectance field of a human face. In *Proceedings of SIGGRAPH 2000*, 145–156.

FUCHS, M., BLANZ, V., AND SEIDEL, H.-P. 2005. Bayesian Relighting. In *Eurographics Symposium on Rendering (2005)*, The Eurographics Association, K. Bala and P. Dutre, Eds.

FUCHS, M., BLANZ, V., LENSCH, H. P., AND SEIDEL, H.-P. 2007. Adaptive sampling of reflectance fields. *ACM Trans. Graph. 26*, 2 (June).

GARG, G., TALVALA, E.-V., LEVOY, M., AND LENSCH, H. P. 2006. Symmetric photography: Exploiting data-sparseness in reflectance fields. In *Proceedings of the EGSR 2006*, 251–262.

HAGAN, M., AND MENHAJ, M.-B. 1994. Training feedforward networks with the marquardt algorithm. *Neural Networks, IEEE Transactions on 5*, 6 (Nov), 989–993.

HANSEN, L. K., AND SALAMON, P. 1990. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell. 12*, 10 (Oct.), 993–1001.

HAŠAN, M., PELLACINI, F., AND BALA, K. 2007. Matrix row-column sampling for the many-light problem. *ACM Trans. Graph. 26*, 3 (July).

HAWKINS, T., EINARSSON, P., AND DEBEVEC, P. 2005. A dual light stage. In *Proceedings of EGSR 2005*, 91–98.

HINTON, G. E. 1989. Connectionist learning procedures. *Artif. Intell. 40*, 1-3 (Sept.), 185–234.

MAHAJAN, D., SHLIZERMAN, I. K., RAMAMOORTHI, R., AND BELHUMEUR, P. 2007. A theory of locally low dimensional light transport. *ACM Trans. Graph. 26*, 3 (July).

MALZBENDER, T., GELB, D., AND WOLTERS, H. 2001. Polynomial texture maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, ACM, New York, NY, USA, SIGGRAPH '01, 519–528.

MARWAH, K., WETZSTEIN, G., BANDO, Y., AND RASKAR, R. 2013. Compressive Light Field Photography using Overcomplete Dictionaries and Optimized Projections. *ACM Trans. Graph. (Proc. SIGGRAPH) 32*, 4, 1–11.

MASSELUS, V., PEERS, P., DUTRÉ, P., AND WILLEMS, Y. D. 2003. Relighting with 4d incident light fields. *ACM Trans. Graph. 22*, 3 (July), 613–620.

MASSELUS, V., PEERS, P., DUTRÉ, P., AND WILLEMSY, Y. D. 2004. Smooth reconstruction and compact representation of reflectance functions for image-based relighting. In *Proceedings of the Fifteenth Eurographics Conference on Rendering Techniques*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, EGSR'04, 287–298.

MATUSIK, W., LOPER, M., AND PFISTER, H. 2004. Progressively-refined reflectance functions from natural illumination. In *Proceedings of EGSR 2004*, 299–308.

NAYAR, S. K. 1989. Sphereo: Determining depth using two specular spheres and a single camera. *International Society for Optics and Photonics*, 245–254.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2003. All-frequency shadows using non-linear wavelet lighting approximation. *ACM Trans. Graph. 22*, 3 (July), 376–381.

NGUYEN, D., AND WIDROW, B. 1990. Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights. In *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, 21–26.

NOWROUZEZAHRAI, D., AND SNYDER, J. 2009. Fast global illumination on dynamic height fields. *Comput. Graph. Forum 28*, 4, 1131–1139.

O'TOOLE, M., AND KUTULAKOS, K. N. 2010. Optical computing for fast light transport analysis. *ACM Trans. Graph. 29*, 6 (Dec.), 164:1–164:12.

O'TOOLE, M., RASKAR, R., AND KUTULAKOS, K. N. 2012. Primal-dual coding to probe light transport. *ACM Trans. Graph. 31*, 4 (July), 39:1–39:11.

OU, J., AND PELLACINI, F. 2011. Lightslice: Matrix slice sampling for the many-lights problem. *ACM Trans. Graph. 30*, 6 (Dec.), 179:1–179:8.

PEERS, P., AND DUTRÉ, P. 2003. Wavelet environment matting. In *Proceedings of EGRW 03*, 157–166.

PEERS, P., AND DUTRÉ, P. 2005. Inferring reflectance functions from wavelet noise. In *Proceedings of the EGSR 2005*, 173–182.

PEERS, P., MAHAJAN, D. K., LAMOND, B., GHOSH, A., MATUSIK, W., RAMAMOORTHI, R., AND DEBEVEC, P. 2009. Compressive light transport sensing. *ACM Trans. Graph. 28*.

RAMAMOORTHI, R. 2009. *Precomputation-Based Rendering*. NOW Publishers Inc.

REDDY, D., RAMAMOORTHI, R., AND CURLESS, B. 2012. Frequency-space decomposition and acquisition of light transport under spatially varying illumination. *European Conference on Computer Vision*.

REN, P., WANG, J., GONG, M., LIN, S., TONG, X., AND GUO, B. 2013. Global illumination with radiance regression functions. *ACM Trans. Graph. 32*, 4 (July), 130:1–130:12.

SEN, P., AND DARABI, S. 2009. Compressive Dual Photography. *Computer Graphics Forum 28*, 2, 609 – 618.

SEN, P., CHEN, B., GARG, G., MARSCHNER, S. R., HOROWITZ, M., LEVOY, M., AND LENSCH, H. P. A. 2005. Dual photography. *ACM Trans. Graph. 24*, 3 (July), 745–755.

SLOAN, P.-P., KAUTZ, J., AND SNYDER, J. 2002. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. *ACM Trans. Graph. 21*, 3 (July), 527–536.

SLOAN, P.-P., HALL, J., HART, J., AND SNYDER, J. 2003. Clustered principal components for precomputed radiance transfer. *ACM Trans. Graph. 22*, 3 (July), 382–391.

TSAI, Y.-T., AND SHIH, Z.-C. 2006. All-frequency precomputed radiance transfer using spherical radial basis functions and clustered tensor approximation. *ACM Trans. Graph. 25*, 3 (July), 967–976.

TURMON, M. J., AND FINE, T. L. 1995. Sample size requirements for feedforward neural networks. In *Advances in Neural Information Processing Systems*, MIT Press, NIPS 7.

VASILESCU, M. A. O., AND TERZOPOULOS, D. 2004. Tensortextures: multilinear image-based rendering. *ACM Trans. Graph. 23*, 3, 336–342.

WALTER, B., FERNANDEZ, S., ARBREE, A., BALA, K., DONIKIAN, M., AND GREENBERG, D. P. 2005. Lightcuts: A scalable approach to illumination. *ACM Trans. Graph. 24*, 3 (July), 1098–1107.

WANG, J., DONG, Y., TONG, X., LIN, Z., AND GUO, B. 2009. Kernel nyström method for light transport. *ACM Trans. Graph. 28*, 3 (July), 29:1–29:10.

WENGER, A., GARDNER, A., TCHOU, C., UNGER, J., HAWKINS, T., AND DEBEVEC, P. 2005. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Trans. Graph. 24*, 3 (July), 756–764.

ZONGKER, D. E., WERNER, D. M., CURLESS, B., AND SALESIN, D. H. 1999. Environment matting and compositing. In *Proceedings of SIGGRAPH 99*, 205–214.

Appendix A: Average Pixel Color as a Neural Network Input

Although the light transport matrix can model arbitrary lighting effects, we consider here for simplicity an opaque surface to show that the average pixel color is highly related to surface appearance and normal direction, while being less related to lighting conditions.

Given a surface point at pixel i with BRDF ρ, its average color $\bar{\mathbf{I}}(i)$ over all the sampled lighting positions $j_{0...N_m}$ can be written as a integration of the BRDF and the averaged incident radiance $\mathbf{L}(i,\mathbf{v})$ over the upper hemisphere Ω defined by normal n:

$$\begin{aligned}\bar{\mathbf{I}}(i) &= \frac{1}{N_m}\sum_{0\le m\le N_m}\int_{\Omega(\mathbf{n}(i))}\rho(i,\mathbf{v})(\mathbf{n}\cdot\mathbf{v})\mathbf{L}(i,j_m,\mathbf{v})\mathrm{d}\mathbf{v}\\ &= \int_{\Omega(\mathbf{n}(i))}\rho(i,\mathbf{v})(\mathbf{n}\cdot\mathbf{v})\frac{1}{N_m}\sum_{0\le m\le N_m}\mathbf{L}(i,j_m,\mathbf{v})\mathrm{d}\mathbf{v}\\ &= \int_{\Omega(\mathbf{n}(i))}\rho(i,\mathbf{v})(\mathbf{n}\cdot\mathbf{v})\bar{\mathbf{L}}(i,\mathbf{v})\mathrm{d}\mathbf{v}. \qquad (12)\end{aligned}$$

By decomposing the BRDF into a diffuse component with coefficient k_d and a specular component with coefficient k_s and specular BRDF ρ_s, the average color can be rewritten as:

$$\begin{aligned}\bar{\mathbf{I}}(i) = k_d(i)&\int_{\Omega(\mathbf{n}(i))}(\mathbf{n}\cdot\mathbf{v})\bar{\mathbf{L}}(i,\mathbf{v})\mathrm{d}\mathbf{v}\\ &+k_s(i)\int_{\Omega(\mathbf{n}(i))}\rho_s(\mathbf{n},\mathbf{v})(\mathbf{n}\cdot\mathbf{v})\bar{\mathbf{L}}(i,\mathbf{v})\mathrm{d}\mathbf{v}. \qquad (13)\end{aligned}$$

Let us consider two neighboring pixels i_a, i_b. Since the two pixels are probably in the same spatial neighborhood, they share similar averaged incident lighting $\bar{\mathbf{L}}(i_a,\mathbf{v}) \approx \bar{\mathbf{L}}(i_b,\mathbf{v})$. Thus the different average colors mainly result from different surface reflectance $\rho(i_a), \rho(i_b)$ or different normal directions $\mathbf{n}(i_a), \mathbf{n}(i_b)$. Also, when

two neighboring pixels with similar reflectance and normal directions are separated by a large depth change, they will have substantial differences in incident lighting. In such cases, the difference in lighting will generally cause a difference in average color.

As seen in Equation 13, it is difficult to separate the reflectance and normal variations in captured images by simply normalizing the image values at each pixel with its average color. As a result, we choose to add the average color as an input parameter of the neural network to provide some physically-based scene information.