Graph Convolution for Intrinsic Decomposition via Large-scale Photorealistic Rendering

Yujie Wang, Qingnan Fan, Kun Li, Dongdong Chen, Jingyu Yang, Jianzhi Lu, Dani Lischinski, Baoquan Chen

Abstract—Intrinsic decomposition is a fundamental problem for many computer vision and graphics applications. It requires the deep understanding of the physics and semantics of the environment, which still exhibits a lot of difficulties for the popular deep learning community. In this paper, we propose the non-local graph convolution network to tackle this problem. We devise the regular image grid as a graph structure, by building the non-local adjacencies whose connection weights are conditioned on the deep feature similarities. We generalize the common convolution operation on this specialized graph-structured data, benefited from which, our algorithm learns better non-local image prior favored by the intrinsic properties. In order to fully explore the potential of our algorithm, we also present a high-quality intrinsic image dataset, composed of more than 20k rendered photorealistic indoor scene images and corresponding pixel-wise annotations for albedo and chromatic shading. Due to the realistic lighting, texture and indoor designs for the 3D scene model, the dataset demonstrates the state-of-the-art quality for the intrinsic decomposition problem and much less domain adaption issue compared to many other competitors. We evaluate our proposed algorithm and dataset on both the popular intrinsic benchmark and a wide range of application scenarios. Experimental results demonstrate suprior performance of our algorithm than the other state-of-the-art approaches.

Index Terms—Intrinsic decomposition, photorealistic rendering, graph convolution

1 Introduction

The physical constituents of an image, such as albedo and shading, are vital for many computer vision and graphics applications. Extracting these key components is an important mid-level vision problem, known as *intrinsic image decomposition*, which is firstly defined by Barrow and Tenebaum in 1978 [2]. In an ideally diffuse environment, the input image can be decomposed into a pixel-wise product of an albedo and a shading image.

Intrinsic decomposition from a single input image is highly ill-posed, since the number of unknowns is twice that of the known values. But due to its huge application potentials, this task has drawn much attention from the research community by leveraging various hand-crafted priors, including the well-known Retinex [21], non-local texture cues [44], global sparsity prior [15], [23], *etc*.

As with many other challenging problems, learning-based approaches have recently been explored to overcome the ill-posedness for intrinsic decomposition. Usually a common encoder-decoder structure is utilized to learn the intrinsics [13], [26], [32], [35], [41]. To fit the piece-wise constancy requirements for intrinsic images, a smooth prior is usually enforced via either post-processed filtering [13], [26], [33], or regularization in the objective functions [26], [27]. However, the aforementioned network structures only are no more than a normal convolution neural network, which is famous for maintaining the spatial locality for filters, but may potentially restrict the performance for intrinsic decomposition that requires the knowledge for the entire image space.

Inspired by the success of graph convolution network for semantic understanding in the shape domain, and the non-local sparsity prior adopted in the classical intrinsic decomposition algorithms [5], [11], [23], [34], [39], [44], we propose a non-local graph convolution network tailored for our task. The design of our non-local graph convolution operation inherits the Graph Convolution Network (GCN) for shape classification [38], but is modified for the 2D image structure. We treat each image point as a vertex in the graph, and build connections between both spatially local and non-local image points, which motivates the graph convolution to learn more global knowledge, favored by the intrinsic property, such as piece-wise constancy.

On the other hand, the learning-based approaches are often sensitive to the dataset used to train their models. This is particularly true for the recently emerging deep learning techniques. Once there's a large discrimination between the training and target domain, the learned network tends to suffer from poor generalization. Indeed, most existing intrinsic image datasets have flaws, for the example of insufficient amount of intrinsic data (MIT [16]), insufficiently realistic rendering techniques (MPI-Sintel [9]), limited object-level images (ShapeNet [35], MIT), or sparse annotations for weak supervision (IIW [4], SAW [19]). These shortcomings prevent us from taking the full advantage of deep learning techniques. Recent efforts [26] have been devoted to rendering realistic scene-level images using the public SUNCG dataset. Despite the improvement obtained by their approach [26], their results still suffers from observable noise and achromatic shading in their rendered data.

To overcome the above limitations, in this paper we propose a new intrinsic image dataset via photorealistic rendering based on the availability of large-scale well-designed 3D indoor scene models, along with the high-quality textures and lightings to emulate the real-world environment. To the best of our knowledge, we are also the first to provide the chromatic shading components for the indoor scene. Experimental results demonstrate that the proposed dataset brings the best rendered image quality over the existing intrinsic dataset, and alleviates the domain gap issues for

Table 1
Comparison between different intrinsic datasets.

Dataset	**MIT** [16]	**ShapeNet** [35]	**MPI-Sintel** [9]	**IIW** [4]	**SAW** [19]	**CGIntrinsics** [26]	**Ours**
Year	2010	2017	2012	2014	2017	2018	2021
Scene	Single object	Single object	Scene	Scene	Scene	Scene	Scene
Type	Captured	Synthetic	Synthetic	Captured	Captured	Synthetic	Synthetic
Images	220	2, 443, 336	890	5K	5K	20K	21K
Albedo Density	Dense	Dense	Dense	Sparse annotations	-	Dense	Dense
Shading Density	Dense	Dense	Dense	-	Sparse annotations	Dense	Dense
Shading Color	Achromatic	Achromatic	Chromatic	-	-	Achromatic	Chromatic
Image Resolution	300×400, etc.	256×256	1024×436	640×480	640×480	640×480	1280×960

'-': The corresponding labels are not available.

the learning-based approaches considerably.

We further provide comprehensive comparisons for decomposition results of the proposed method and the recent state-of-the-art methods. We demonstrate the superior performance of our method on both the mainstream intrinsic image evaluation benchmarks (IIW/SAW), and a variety of downstream application scenarios, which provide more intuitive comparisons.

Our overall contributions can be summarized as follows:

- We propose the first graph convolution network tailored for the intrinsic decomposition problem by incorporating the non-local image prior in the network design.
- We present a new scene-level intrinsic image dataset via large-scale photorealistic rendering. For the first time, we provide chromatic lighting in the indoor environment, which renders much more realistic shading component and provides the foundation for better generalization possibilities of deep networks on real scenes.
- We demonstrate the state-of-the-art performance on various evaluation metrics and application scenarios.

2 Related Work

Dating back to the 1970s, factorizing physical properties from images starts to attract attentions in academic society [2], [21]. The definition of intrinsic images is introduced in [2], after which extensive intrinsic decomposition solutions have been proposed, ranging from the classical algorithms [11], [23], [37], [44] to the recent deep learning approaches [13], [26], [27], [29]. Due to the ill-posedness of single image intrinsic decomposition, additional information is explored to ease such a difficulty, such as image sequences, depth information [11], [43] and user scribbles. In this paper, we focus on intrinsic estimation from single input image, which fits the most common real-world scenario. In this section, we review the recent literature in this field first, and then discuss the previous intrinsic datasets.

2.1 Methods

Traditional approaches. The most influential Retinex theory [21] assumes that the albedo image is piece-wise constant while the shading image varies smoothly and becomes the cornerstone of many classical solutions in this field. For further constraining the problem, many other priors are introduced to construct the intrinsic estimation models, among which the most popular ones are the non-local priors [11], [34], [44] and the sparsity constraints [15], [23]. For instance, Shen *et al.* [34] improve decomposition quality significantly by incorporating non-local correlation that implies that non-adjacent points are still likely to have the same albedo value if they have similar textures. Li *et.al* [23] proposes a global sparsity constraint based on the assumption that natural images usually comprise a small number of colors. Several methods encode global sparsity priors by clustering algorithms to enforce a limited number of different reflectance areas [6], [14], [30]. For instance, [14] clusters pixels into several groups using K-means algorithm in CIELab color space and recent work [30] adopts a much simpler histogram-based clustering solution. Additionally, several models construct pair-wise connections across the image [4], [11] to utilize non-local information. The above works prove that the non-local prior is effective for intrinsic decompositions.

Deep learning based methods. Statistics of real world illumination and geometry is effective for resolving the inherent ambiguity in intrinsic decomposition [1]. In the past few years, with the availability of intrinsic image datasets (MIT, MPI-Sintel, IIW, SAW, *etc*), deep learning techniques are increasingly adopted to estimate intrinsic components [13], [29], [32], [36], [41], [46] due to its superiority in learning statistics from large-scale data, revealed in various computer vision tasks. In order to enforce the piece-wise constancy for the albedo components, some traditional knowledge are explored in the learning approaches, such as post-processing smoothing operations [26], [33], jointly learned guided filter [13] and smoothness loss functions [26]. Most of these methods merely consider local smoothness for albedo without considering its global image structure, which is characterized by intrinsic properties.

Graph convolution networks. In recent years, graph convolution networks (GCNs) bring great improvements for the problems whose data structure are highly irregular, such as point cloud [25], [42]. Based on the spectral graph theory, [8] devises a variant of graph convolution, which is the first prominent research on GCNs. Then GCNs are improved or extended by other spectral methods [18], [22] as well as spatial-based GCNs [17], [31]. Recently, [38] proposes edge conditioned graph convolution, which conducts weighted aggregations over the neighborhood around each point and determine the connection weights based on edge labels. In our context, we build an irregular graph structure for images by treating each image point as a vertex, and considering both spatially close and distant image points as the adjacency for each vertex. We generalize the common convolution operation on this specialized graph structure as a foundation of graph convolution network.

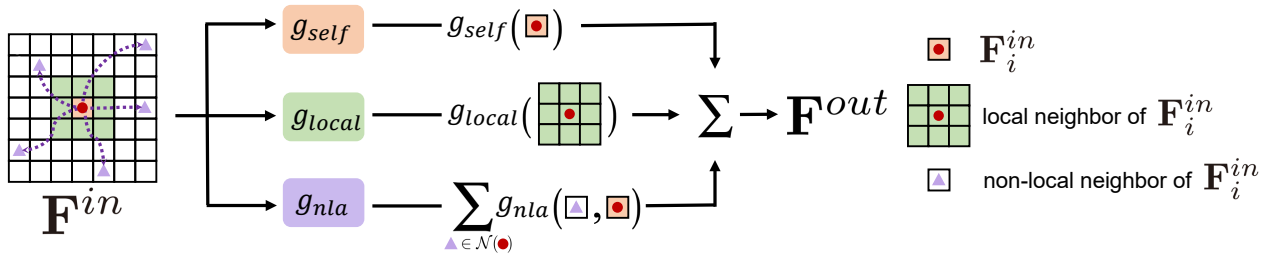


Figure 1. Schematic of the proposed non-local graph convolution layer.

2.2 Dataset

With sufficient diverse training data, deep learning approaches are able to learn a better prior than the hand-crafted one. However, due to the shortcomings in each of the existing intrinsic benchmarks and network design, deep learning techniques are not able to realize their full potential. For example, the MIT intrinsic dataset [16] was created by capturing only a few hundreds of object-level intrinsic images without any background, which is too few to be well suited for deep learning solutions. Bell et al. [4] asked workers from Amazon Mechanical Turk to label pairwise reflectance relationships over a sparse collection of points in real images, giving rise to the famous Intrinsic Images in the Wild (IIW) dataset. Kovacs et al. [20] collected multiple forms of sparse shading annotations via crowdsourcing, known as Shading Annotations in the Wild (SAW). Recently, various large-scale intrinsic datasets have been created via rendering techniques, based on either an open-source 3D animation movie [9], a small number of synthetic outdoor environments [3], a collection of 3D shapes [10], [36] or a large-scale indoor scene models [24], [26].

One of the closest datasets to ours is CGIntrinsics [26], which rendered about 20K indoor scene images from the 3D synthetic dataset SUNCG [40]. Very recently, [24] created InteriorNet, which contains an even larger number (20M) of synthetic indoor images, which however renders 1K continuous frames per scene and thus has similar scene diversity as CGIntrinsics and our dataset. Both of these datasets don't provide rendered shadings, which are instead computed and have many artifacts. Their synthetic images also suffer from limited rendering quality.

3 Approach

In this section, we first describe how a graph convolution layer is constructed in the image domain in Sec. 3.1, then introduce the full network structure in Sec. 3.2, followed by the loss functions in Sec. 3.3.

3.1 Graph convolution layer

As illustrated in Figure 1, the input to the graph convolution layer is a feature map $\mathbf{F}^{in} \in \mathbb{R}^{c \times h \times w}$ with c channels and size of $h \times w$. The graph convolution layer operates in three parallel branches. The first one g_{self} learns to transform the pixel-wise feature vectors. Each input feature vector $\mathbf{F}_i^{in}$ located at point p_i of size c is mapped to a feature vector $\mathbf{F}_i^{self}$. The second branch g_{local} learns to aggregate the local feature information of each point by filtering its surrounding neighbors to generate $\mathbf{F}_i^{local}$. These two operations are implemented via the normal convolution layers of kernels 1×1 and 3×3 separately. They target at extracting the local features, while the third branch g_{nla} learns to aggregate the non-local feature vectors, which is formulated as follows.

For each feature vector $\mathbf{F}_i^{in}$ at point p_i, it selects k random feature vectors across the whole feature map as neighbors, and constructs the connections between $\mathbf{F}_i^{in}$ and each of its neighbors. The feature vectors and corresponding connections serve as vertex and edge to form the undirected graph G.

$$\begin{aligned} &G = \{V, E\}, \\ &V = \{p_i | \forall p_i \in \mathbf{F}^{in}, i \in [1, N]\}, \\ &E = \{e_{i,j} | e_{i,j} =< p_i, p_j >, p_j \in \mathcal{N}(p_i), p_i \in V\}, \end{aligned} \tag{1}$$

where $N = h \times w$ is the number of feature points, and $\mathcal{N}(i)$ is the set of neighboring points of p_i.

Then the output of the non-local aggregation layer $\mathbf{F}_i^{nla}$ is calculated as the weighted combination of the randomly sampled feature vectors around point p_i:

$$\mathbf{F}_i^{nla} = \frac{1}{k} \sum\nolimits_{j \in \mathcal{N}(i)} g_{nla}(\mathbf{F}_i^{in}, \mathbf{F}_j^{in}), \tag{2}$$

where g_{nla} aggregates the feature vector $\mathbf{F}_j{}^{in}$ with a weighting factor $w_{i,j}$ conditioned on the difference between the two points p_i and p_j, computed as:

$$g_{nla}(\mathbf{F}_i^{in}, \mathbf{F}_j^{in}) = \mathbf{w}_{i,j} \odot \mathbf{F}_j^{in}, \tag{3}$$

where $\odot$ represents the element-wise multiplication between two vectors, and the weighting vector $\mathbf{w}_{i,j}$ is calculated by:

$$\mathbf{w}_{i,j} = g_{nla\_mlp}\left(Concat\left(\left(\mathbf{F}_i{}^{in} - \mathbf{F}_j{}^{in}\right), d_{i,j}\right)\right), \tag{4}$$

where g_{nla_mlp} is implemented as the multi-layer perception. $d_{i,j}$ is calculated as the root mean square error (RMSE) between the two point positions:

$$d_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \tag{5}$$

(x_i, y_i) and (x_j, y_j) are the 2D coordinates of p_i and p_j within $\mathbf{F}^{in}$.

Therefore, the overall output of the graph convolution layer is the output summation of the feature transformation layer, local and non-local feature aggregation layers,

$$\mathbf{F}_i^{out} = \mathbf{F}_i{}^{self} + \mathbf{F}_i{}^{local} + \mathbf{F}_i{}^{nla}. \tag{6}$$

3.2 The overall framework with graph convolution

We adopt the common encoder-decoder network structure as the backbone of our framework. Our algorithm takes an image as input, and feeds it to the encoder of 6 convolution layers. The intermediate feature maps are processed by two decoders whose structure is a mirror of the encoder, to generate albedo and

Figure 2. Demonstration of the requirement for global understanding of intrinsic properties. Left: the inter-reflection in blue circle is caused by the pink cabinet in green circle; Right: the inconsistent colors in two blue circles share the same albedo. Both the circles in the same example are spatially distant.

shading separately. We build the skip connection between the corresponding layers in both the encoder and decoder.

Inferring the intrinsic information requires the global understanding of the entire image domain. For the left example in Figure 2, the pink colors on the floor tiles are indirectly reflected from the pink cabinet in green circle. In order to recover the correct shading on the floor, the deep network needs to learn the global information across the whole image; for the right example, the carpet on the floor is made of the same material and share the same albedo, while the part in green circle is in covered under shadows, and estimating its correct albedo requires the learned deep network to retrieve the distant information from the blue circle.

However, the pure encoder-decoder neural network is limited in the size of receptive field, and learns relatively local features. In order to ease such a limitation, we incorporate the non-local cues from the graph convolution layer into our deep neural network. To be specific, we append four graph convolution layers after the encoder, and forwards its output to both albedo and shading decoders.

Post-processed refinement. Many previous learning-based approaches leverage a post-processed filtering module for albedo refinement [13], [26], [33]. Some other work [27], [29] includes a sparsity prior in the constructed supervised loss to generate a clean albedo. Inspired by this type of work, in this paper, we introduce a refinement module, which is instead a fixed filter, but a learned deep network.

Given an input image and an input structure map, the deep network is designed to maintain the important image structure of the input image guided by the input structure map, while eliminating the unimportant details. During the training phase, the refinement module takes the untouched natural image along with a structure map as input, and the filtered natural image using [6] as ground truth label. The filtered image exhibits strictly piece-wise constant regions grouped by superpixels of similar color intensities favored by albedo properties, and the structure map is computed from the filtered image following [13]. Hence the deep network learns to map the input image to the ground truth label based on the guided structure map. During the evaluation phase, it transfers the learned knowledge for albedo refinement by replacing its inputs with the predicted albedo and albedo structure.

Unlike the traditional image filters that follow fixed operations, our alternative learns refinement for the albedo implicitly within a deep neural network, which learns more complex image manipulations and hence achieve better performance as validated in Section 7.1 about ablation study.

Implementation details. The proposed network framework is implemented in PyTorch framework with mini-batch size of 6 and is optimized by Adam algorithm. During the training process, the initial learning rate is set to 0.01, and changed to 0.0005 while finetuning on IIW/SAW dataset. The refinement module is solely trained with learning rate of 0.01 and batch size of 4.

3.3 Loss and Training

In this section, we introduce the loss functions used for supervising our network. In order to render photorealistic images that alleviate the domain gap between the synthetic and real data, besides the pixel-wise annotations for albedo (A) and shading (S), we also render four more visual components, specularity (SP), reflection (RE), refraction (RA) and self-illumination (SI), which however doesn't belong to either albedo or shading. We mask out these regions for supervision,

$$M_i = \begin{cases} 0 & if\ (SP_i + RE_i + RA_i + SI_i) > 0, \\ 1 & \text{otherwise;} \end{cases} \tag{7}$$

where i indicates the pixel index, and M is the computed mask. For remaining regions, the loss function is formulated as:

$$\mathcal{L} = \mathcal{L}_{IE} + \mathcal{L}_{IR}, \tag{8}$$

where $\mathcal{L}_{IE}$ and $\mathcal{L}_{IR}$ are separately for the intrinsic estimation (IE) and refinement (IR) network. The loss functions for the first module are supervised on both the predicted albedo $\widetilde{A}$ and shading $\widetilde{S}$, which are given by

$$\mathcal{L}_{IE} = \mathcal{L}_{MSE} + \lambda_g \mathcal{L}_{grad} + \lambda_r \mathcal{L}_{recon} \tag{9}$$

where λ_g and λ_r are the balance weights. $\mathcal{L}_{MSE}$ and $\mathcal{L}_{grad}$ are the commonly adopted mean squared error (MSE) and image gradient loss:

$$\mathcal{L}_{MSE} = ||\widetilde{A} - A||_2^2 + ||\widetilde{S} - S||_2^2, \tag{10}$$

$$\mathcal{L}_{grad} = ||\nabla\widetilde{A} - \nabla A||_2^2 + ||\nabla\widetilde{S} - \nabla S||_2^2, \tag{11}$$

where $\widetilde{A}$ and $\widetilde{S}$ are predicted albedo and shading. We further append a self-supervised photometric reconstruction loss by synthesizing the image from its estimated counterparts assuming the ideal diffuse environment,

$$\mathcal{L}_{reconstruct} = ||\widetilde{A} \odot \widetilde{S} - I||_2^2. \tag{12}$$

Following the previous work [12] to supervise the refinement module, $\mathcal{L}_{IR}$ is simply defined as,

$$\mathcal{L}_{IR} = ||\widetilde{R} - R||_2^2, \tag{13}$$

where $\widetilde{R}$ is the output of refinement module in the training phase, and R is the image processed by [6]'s approach. Note the major intrinsic estimation network and following refinement network are trained separately with $\mathcal{L}_{IE}$ and $\mathcal{L}_{IR}$, and combined later to form our full evaluation pipeline.

4 Proposed Dataset

We render 21,478 triplets of indoor scene images, including one color image along with its albedo and shading components for each instance. We split the dataset into 18,256 images for training and the others for evaluation. The natural appearance of our data is mainly due to the following factors:

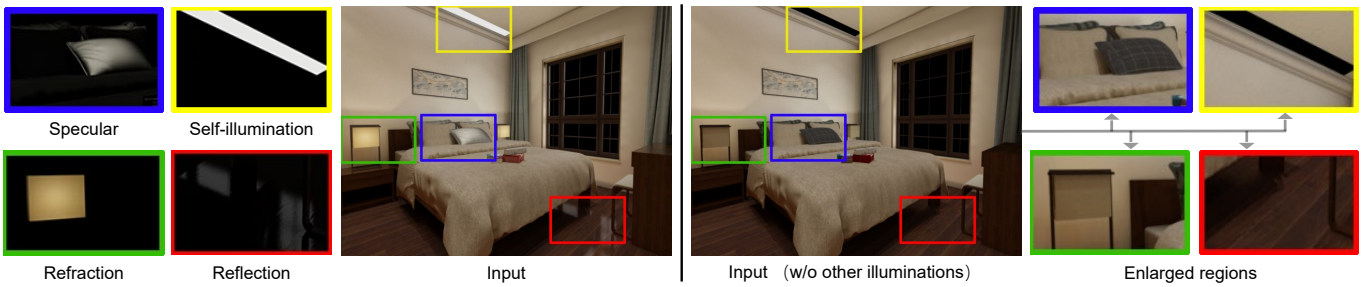


Figure 3. Comparison between rendered images with (left) and without (right) the extra illumination effects (reflection, refraction, self-illumination and specularity).

Figure 4. Rendered intrinsic image comparison between our proposed one (left four) and CGIntrinsics [26] (right three). From top to bottom: input, albedo, shading and magnified region in the input.

Scene layout: In order to build our dataset, we collect 5730 synthetic 3D indoor scene models from a third-party interior design platform[3]. The synthetic scenes are designed for residual buildings, including the combination of living rooms, bedrooms, kitchens, bathrooms, *etc*. They are modeled by hundreds of professional designers/artists, and have been applied in the real-world scenario for home decoration purpose. Hence they highly resemble the natural arrangement of the daily life environment.

Lighting setup: We implement many kinds of light sources in the environment to simulate the real-world visual appearance. Besides the common global illumination and inter-reflections, we also render the reflection, specularity, refraction of the transparent objects, and the self-illumination objects in the scene. We demonstrate one example in Figure 3, the extra more complex illumination effects make the rendered image more photorealistic.

Texture diversity: The 3D scenes are equipped with 800k texture maps, which are randomly sampled to form the synthetic scenes. The large texture diversity can be observed from the rendered images shown in Figure 4 and the supplemental material.

Rendering details: We adopt the render engine from Embree[4] with the Deterministic Monte Carlo (DMC) approach to render the image. The images are rendered under the resolution of 1280×960, with 3228 samples per pixel. Rendering one image of our photorealistic quality in similarly complex indoor environment is time-consuming, which requires hours of computation on a desktop workstation. To accelerate the rendering process, we employ distributed rendering via 32 servers with 32 CPU cores on each, which helps decrease the total rendering time to 537 hours, equivalent to about 22 days. On average, our system only takes 90 seconds to render a 1280×960 image.

We compare various publicly available intrinsic datasets in Table 1. MIT [16] and ShapeNet-intrinsics [35] provide object-level intrinsic images which is either limited by the dataset size or rendering realism. IIW [4] and SAW dataset [19] targets at scene-level real images with only sparse annotations obtained from human judgements by crowdsourcing. On the other hand, MPI [9] also focuses on scene-level images, but it's rendered on animations that lack realism either. The most related dataset to ours is CGIntrinsics [26]. Compared to it, our dataset provides higher image resolution and chromatic lighting in the shading image, which is instead computed in CGIntrinsics.

In Figure 4, we compare the visual images between CGIntrinsics and ours. They exhibit apparent rendered noises, simpler indoor scene layout and only chromatic lighting.

3. https://www.3vjia.com/
4. https://www.embree.org/

5 Experimental Results

In this section, we firstly study the effectiveness of our proposed dataset for improving the intrinsic decomposition performance in Sec. 5.1, and then compare with state-of-the-art approaches on IIW and SAW test sets in Sec. 5.2 and the wild images in Sec. 5.3.

Table 2
Comparison of the effectiveness of different datasets.

Training data	Method	WHDR ↓ (IIW)	AP ↑ (SAW)
CGIntrinsics	Shi *et al.* 2017 [35]	34.55%	92.33%
	Fan *et al.* 2018 [13]	27.13%	93.81%
	Li *et al.* 2018 [26]	19.94%	92.09%
	Ave.	27.21%	92.74%
Ours	Shi *et al.* 2017 [35]	20.53%	95.84%
	Fan *et al.* 2018 [13]	20.89%	92.96%
	Li *et al.* 2018 [26]	18.34%	97.66%
	Ave.	19.92%	95.48%

↓: higher is better. ↑: lower is better.

5.1 Effectiveness of our dataset

We justify our dataset by comparing it with the closest dataset to ours, CGIntrinsics. We train three state-of-the-art learning based intrinsic estimation algorithms on either CGIntrinsics or our dataset, and evaluate their performance on the intrinsic image benchmark, IIW [4] and SAW [20], which are collected on real images with sparse annotations. We remove any post-processed filtering module for a fair comparison of the intrinsic quality. All the methods are trained with their publicly available codes by only changing the training data. As shown in Table 3, we average the results computed among three different methods, and the networks trained on our dataset achieve much superior performance compared to CGIntrinsics. It justifies the effectiveness of our dataset despite of the differences in network structures and loss functions, and our dataset also shows better generalization possibility on real images.

Table 3
Quantitative comparison on IIW/SAW test sets.

Methods	WHDR ↓ (IIW)	AP ↑ (SAW)	
Retinex (color) [16]	26.89%	85.26%	
Garces *et al.* 2012 [14]	25.46%	92.39%	
Zhao *et al.* 2012 [45]	23.20%	89.72%	
Bell *et al.* 2014 [4]	20.64%	92.18%	-
Shi *et al.* 2017 [35]	59.40%	81.30%	
Narihira *et al.* 2015 [41]	37.30%	86.08%	
Zhou *et al.* 2015 [46]	19.95%	86.34%	
Nestmeyer *et al.* 2017 [33]	17.69%	88.64%	
Fan *et al.* 2018 [13]	**14.45%**	85.13%	
Li *et al.* 2018 [26]	14.80%	96.57%	
Ours	17.92%	96.17%	
Ours$^+$	15.10%	**96.65%**	

$^+$: finetuning on IIW and SAW dataset.

5.2 Evaluation on the IIW/SAW dataset

In this section, we provide a more comprehensive comparison between the proposed method and state-of-the-art intrinsic decomposition approaches. Following the previous work [26], we finetune our network on IIW and SAW datasets after training on our proposed dataset from scratch. During the finetuning process, we follow the train/test split in [13], [26] and utilize the ordinal reflectance loss and SAW shading loss presented in [26] for supervision. To make full use of the sparse label provided in the IIW dataset, we also use the augmented labels provided by [26].

The numerical results are reported in Table 3. By solely training on our rendered dataset, we achieve promising performance for albedo (WHDR: 17.92%) and shading (AP: 96.17%), which is better than all the non-learning approaches and most learning based methods. Especially on the SAW benchmark, our approach ranks the second best among all the previous approaches without finetuning. It demonstrates the promising generalization ability of our algorithm to real world images. After finetuning our network, our method achieves comparable performance (WHDR: 15.10%; AP: 96.65%). Instead of training our network from scratch like all the previous competitors [13], [26], [33], our results are obtained only via finetuning.

The corresponding qualitative results are shown in Figure 5. We show three cases from the IIW and SAW test sets, among all of which our approach shows superior ability to separate albedo from shading. Our shading results exhibit chromatic lighting in the scenes, including the self-illuminated light source and inter-reflections bounced between objects, while all of other three methods [13], [26], [33] are limited by the achromatic shading output. On the albedo side, the results from other methods all show somewhat unnatural colors and fail to eliminate the lightings here or there, while our predicted albedo images appear more natural colors.

Moreover, as the shading images for [33] and [13] are computed from the input color and predicted albedo image, the texture in the input image tends to fall into the shading image as long as the albedo is not perfectly predicted. Due to the smoothness regularization enforced in the objective function in [26], their shading outputs are over-smoothed and do not reflect much geometry information of the scene. The aforementioned issues are not observed in our predicted intrinsic images.

Note as pointed out by the previous work [33], [46], the annotations in the IIW dataset is biased on judgements of the same albedo, which occupy 2/3 of all the relative judgements. Hence a heuristic rescaling of the input image from [0,1] to [0.55,1] achieves a comparable WHDR of 25.7 without running intrinsic estimation algorithms, since there's a bigger chance that equal albedo occurs on the relative points in IIW annotations, discussed in [33]. Therefore directly training on the IIW and SAW dataset may overfit the distribution exhibited by their training data. Our framework chooses to train on the synthetic data first and finetune on the IIW and SAW dataset, where our rendered data could present a less-biased intrinsic prior that avoids overfitting in the IIW and SAW data. This could explain why the visual comparison between other results and ours demonstrates larger performance gap, compared to the numerical ones.

5.3 Evaluation on unseen data source

The performance for deep learning based methods is likely to suffer significant degradation when there is a large domain gap between the training data and the test data. As our method is mainly trained on rendered data, it's necessary to evaluate how it performs on images in the wild. Therefore, we compare our model with [13], [26], [33] on some self-collected real world images in Figure 6, where we generate much superior intrinsic decompositions.

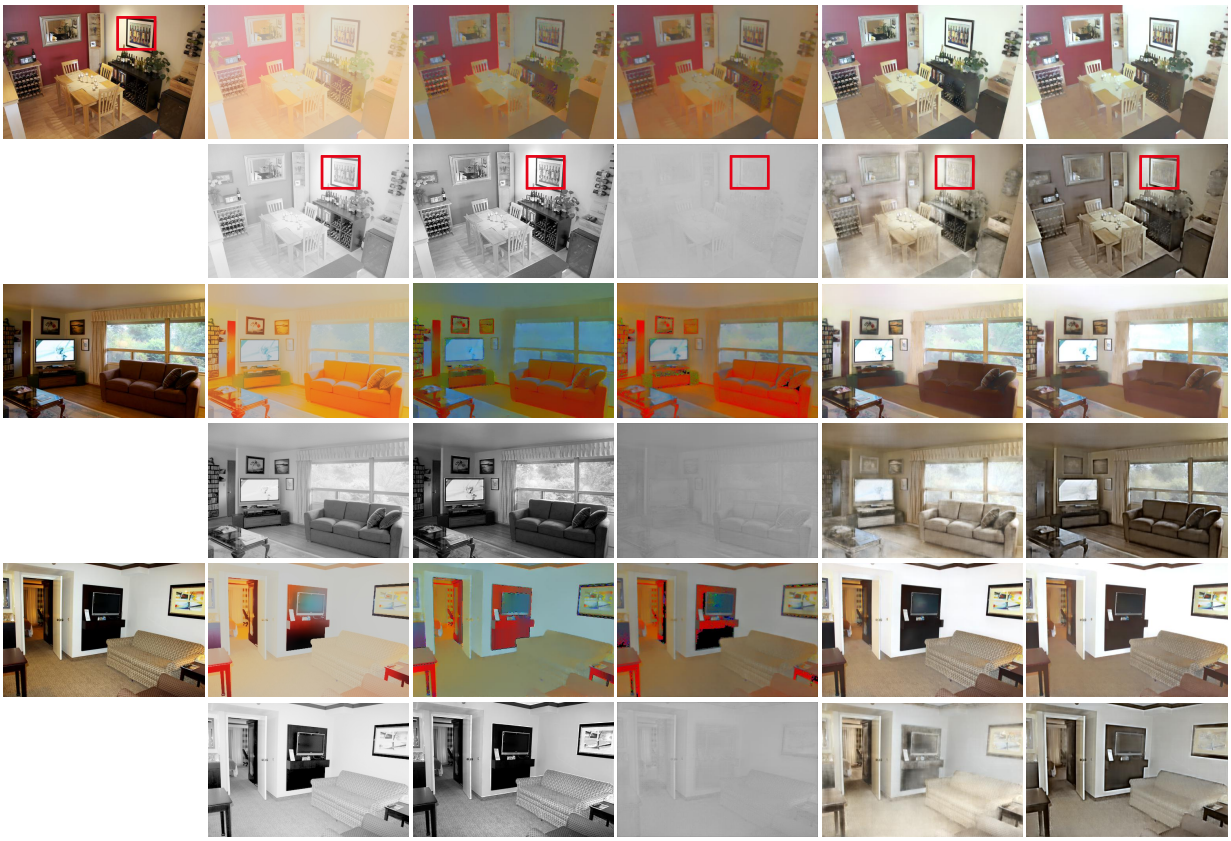


Figure 5. Qualitative comparison on images from IIW/SAW test sets. For each example, the albedo images for different methods are placed in the first row while the corresponding shading images are placed in the second row. Ours with "+" refers to finetuned results.

For the example of the first case, the albedo images recovered by [13], [26], [33] are heavily affected by the light transitions and lose the ceramic textures in the albedo. The second case demonstrates very strong inter-reflections in the environment, while the other approaches fail to separate the chromatic reflections into shading. Finally, we demonstrate an example of outdoor scene, which is not exhibited in the training data. Our approach successfully removes the letters on the boat in the shading image, while others do not.

6 Applications

Intrinsic decomposition is a middle-level vision task and is more valuable only when applied to the downstream applications. In this section, we explore a couple of applied scenarios for intrinsic decomposition and further demonstrate the effectiveness of our algorithm.

6.1 Texture editing

Texture editing is conducted by replacing the estimated albedo with the preferred ones, and recompose the image using untouched shading and modified albedo. This can be used to reveal the quality of recovered shading, which however often suffers from wrongly separated textures and over-smoothing effects.

We demonstrate three examples in Figure 7. In the first case, the eyes and eyebrows of three pigs in the first row are modified in each albedo image, where we adaptively remove the original patterns here and the color of nearby areas remains consistent before and after modification. [13] fails to remove the texture of these areas in the predicted shading image, which makes the original patterns occur in the recomposed image and cause artifacts. The recomposed result of [26] is better than [13]'s, but it still has artifacts that look like shadows around eyes and eyebrows, which are caused by the remaining texture of their shading result. For the second example, we exchange the positions of the two paintings hanging on the wall. Similar phenomenon in the first case is also observed here. The painting at the original poisition remains in the recomposed results both for [13] and [26]. Besides, [13] over-smoothes the albedo, and hence its recomposition lacks painting details.

6.2 Lighting editing

Lighting editing is conducted by modifying the shading image and recomposing with the modified shading and untouched albedo. It can also be used to evaluate the quality of estimated albedo.

We demonstrate two examples in Figure 8. In the first case, we smooth out the highlight area lit by the lamp for the shading image. Since both [13] and [26] fail to remove the lighting in their estimated albedo, their recomposed images still exhibit such lighting, while only our method's output successfully gets rid of the light emitted from the lamp. In the second example, we simply modify the shading component to be grayscale. Obviously, the recomposed images from the other approaches still demonstrate chromatic lighting in the scene, which however doesn't appear in our result. This is mainly due to the fact that their estimated shading is grayscale already, and chromatic lighting are all wrongly separated into the albedo.

Figure 6. Qualitative comparison on images in the wild. For each example, the albedo images for different methods are placed in the first row while the corresponding shading images are placed in the second row.

Following the experiment setting in [46], we take several image sequences from [7] and [28] for the relighting application. Each frame of the same image sequence is taken under different lighting conditions. We demonstrate two examples in Figure 9, for each of which, there are two images taken from the same scene. After recovering the intrinsic images, the recomposition is done by switching each other's shading image and recombining with its own albedo. Visually, our recomposed results are most closest to the input images, while there's always a color shift for other methods.

Table 4
Ablation study results of the proposed method

Methods	WHDR ↓ (IIW)	AP ↑ (SAW)
Ours (w/o h_{nlm})	19.08%	92.19%
Ours (w/o h_r)	20.89%	96.17%
Ours (w/o other components)	**17.65%**	94.94%
Ours (CGIntrinsics)	18.13%	93.10%
Ours	17.92%	**96.17%**

7 Analysis

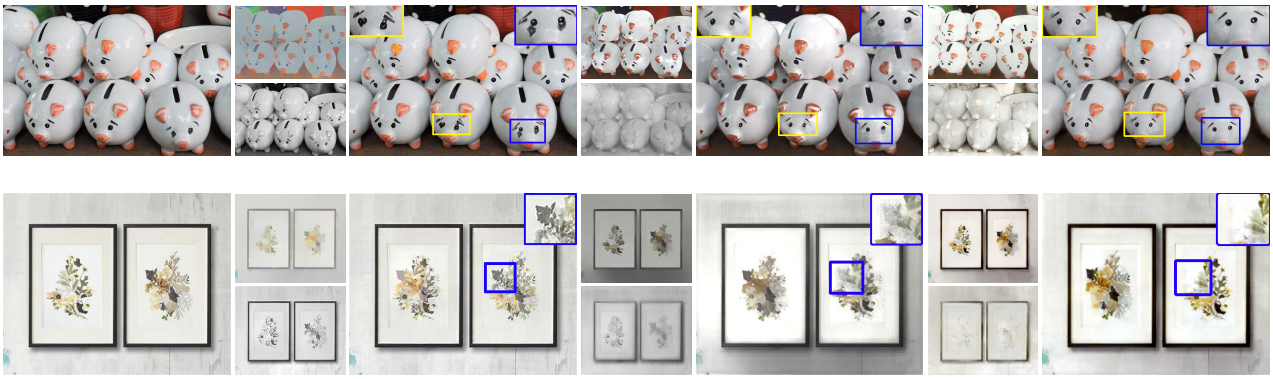
7.1 Ablation Study

In this section, we analyze the contribution of each component in our method including the proposed dataset. The predicted albedo and shading are evaluated on the IIW and SAW dataset without finetuning the network. The numerical results are summarized in Table 6.2.

Effectiveness of the non-local graph module. To investigate the effectiveness of the adopted non-local module h_{nlm}, we train a variant of our network in which h_{nlm} is removed while other settings are unchanged. As the numerical results in the first row of Table 6.2 show, compared with the results of our full model, the performance on both two components suffers noticeable degradations especially for the shading component. This demonstrates that the proposed the non-local module h_{nlm} plays an essential role in our method.

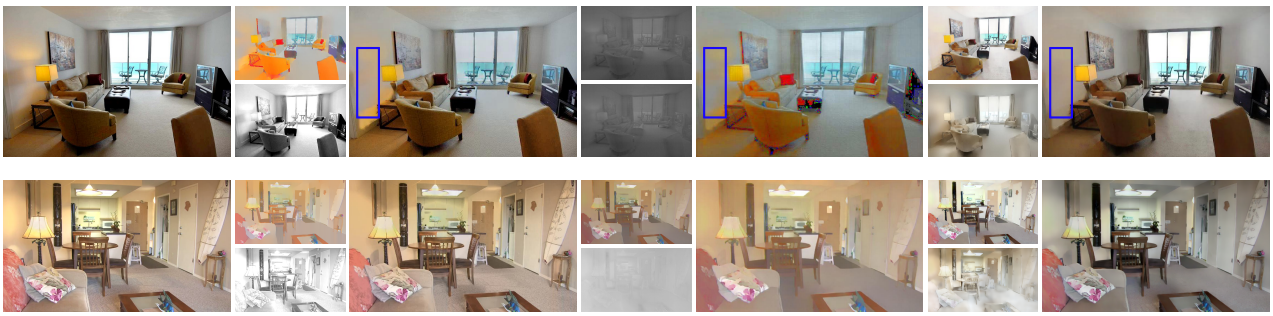
Contribution of the refinement module. As shown in Figure 5 and Figure 6 , the reflectance consistency prior is well represented in our decomposition results while over-smoothing phenomenon is avoided to the largest extent. To reveal the contribution of our refinement module h_r quantitatively, we remove the refinement module from the full model and test the performance and report it in Table 6.2.Without h_r, an apparent degradation is observed on the WHDR error, which increases to 20.89%, while SAW AP result is unchanged as h_r is only applied to the albedo component. This means that the proposed refinement module is able to improve the albedo performance effectively.

Value of other irradiance components. As described in Sec.



Input Fan [13] Li [26] Ours

Figure 7. Two examples of retexturing. In the first example, the eyes and eyebrows for three pigs in the first row are modified adaptively for each albedo image. Then the re-rexturing outputs are produced by the modified albedo images and the unchanged shading images. In the second example, we exchange the positions for two paintings hanging on the wall for all albedo images and then get the outputs.



Input Fan [13] Li [26] Ours

Figure 8. Lighting editing application. Top: for the shading image, we smooth out the reflected lights on the wall lit by the lamp, which is highlighted by the blue box; Down: we adjust the illumination in the environment by changing the estimated shading to grayscale.

4, our dataset not only encodes diffuse material but also non-diffuse materials into the input images, which may influence the estimation of albedo and shading components. To explore how these non-diffuse irradiance components affect the performance, we recompose the input images using ground truth albedo and shading images excluding other components and train our full network on them. The performance for this variant is shown in Table 6.2, which demonstrates a noticeable degradation on SAW AP and a negligible improvement on WHDR error. It indicates that these non-diffuse components exert effective influence on shading prediction.

Role of the proposed dataset. To investigate the contribution of the proposed dataset, we train our network by substituting our dataset with CGIntrinsics dataset [26], which are closest to ours in spirit, and report the evaluation results in Table 6.2. Experimentally, we find both of the numerical results for albedo and shading are degraded, among which the shading performance is much worse than that of our full model trained on the proposed dataset. This further demonstrates our dataset's effectiveness for improving the performance for intrinsic estimation.

8 Conclusion

In this paper, we devise a graph convolutional network for intrinsic decompostion, in which non-local cues are utilized in an explicit manner. In order to overcome the limitations within existing datasets, we render a new photoreslistic dataset of high quality, in which rendered dense labels for albedo and shading are available. The shading labels in our dataset first considers chromatic lighting, which enables our model to better separate material properties and lighting effects especially those introduced by inter-reflections between diffuse surfaces. The effectivess of the proposed method is comprehensively evaluated by conducting a series of comparisons with the competetitors. More interestingly, we apply the decomposition results of both our method and two latest state-of-the-art methods [13], [26] to a range of application scenarios and visually demonstrate the application potentials of each method.

References

[1] J. T. Barron and J. Malik. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 37(8):1670–1687, 2015.

[2] H. Barrow and J. Tenenbaum. Recovering intrinsic scene characteristics. *Comput. Vis. Syst., A Hanson and E. Riseman (Eds.)*, pages 3–26, 1978.

[3] A. S. Baslamisli, T. T. Groenestege, P. Das, H.-A. Le, S. Karaoglu, and T. Gevers. Joint learning of intrinsic images and semantic segmentation. *European Conference on Computer Vision (ECCV)*, 2018.

[4] S. Bell, K. Bala, and N. Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics (TOG)*, 33(4), 2014.

[5] S. Bi, X. Han, and Y. Yu. An ℓ_1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. *ACM Trans. Graphics*, 34(4):78, 2015.

[6] S. Bi, X. Han, and Y. Yu. An L_1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. *ACM Transactions on Graphics (TOG)*, 34(4):78, 2015.

[7] I. Boyadzhiev, S. Paris, and K. Bala. User-assisted image compositing for photographic lighting. *Acm Transactions on Graphics*, 32(4):1–12, 2013.

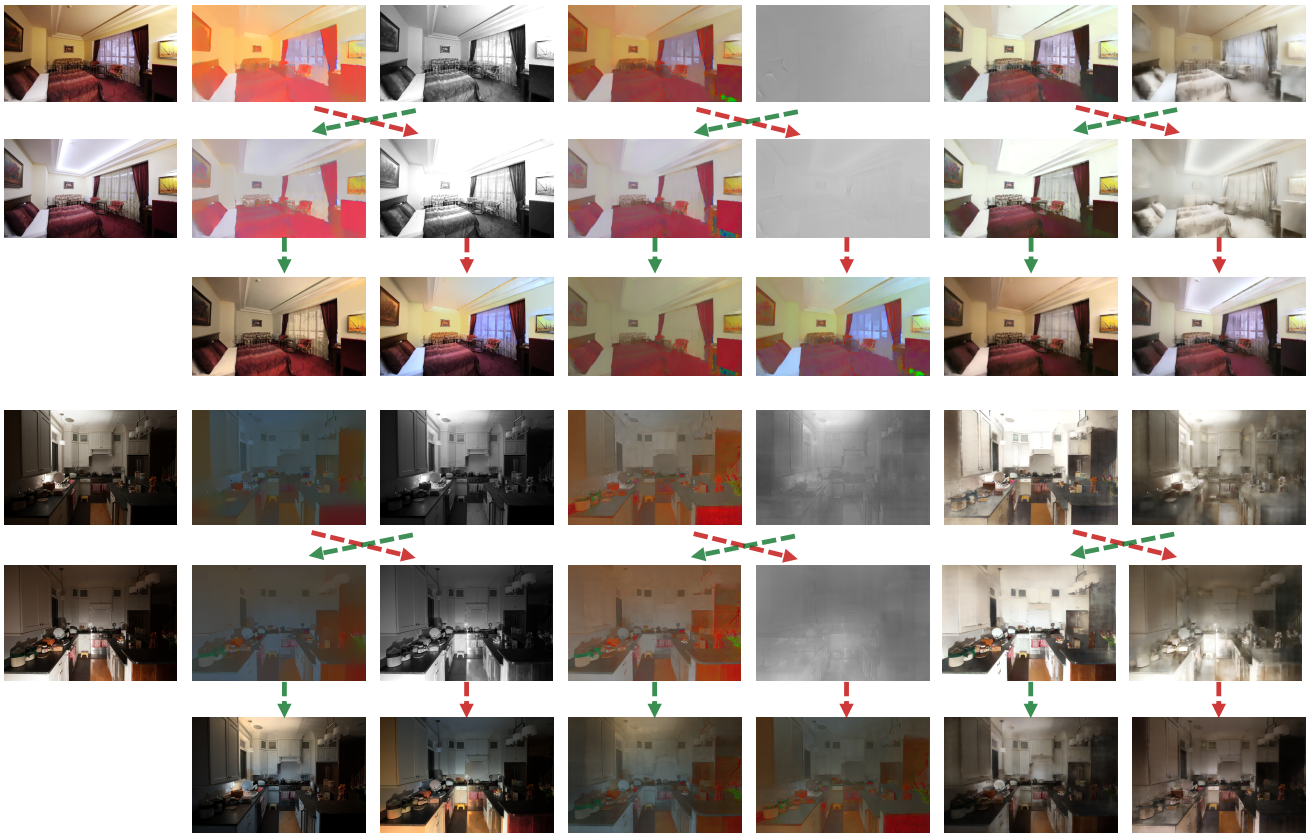


Figure 9. Relighting application for three methods: (a) Fan *et.al* [13] (b) Li *et.al* [26] and (c) ours. For each example, there are two input images captured in the same scene but with different lighting. The recomposed image is generated with its own estimated albedo and the other one's shading.

[8] J. Bruna, W. Zaremba, A. Szlam, and Y. Lecun. Spectral networks and locally connected networks on graphs. 2013.

[9] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *European Conference on Computer Vision (ECCV)*, pages 611–625, 2012.

[10] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. ShapeNet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.

[11] Q. Chen and V. Koltun. A simple model for intrinsic image decomposition with depth cues. In *Proc. ICCV*, pages 241–248, 2013.

[12] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf. A generic deep architecture for single image reflection removal and image smoothing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3238–3247, 2017.

[13] Q. Fan, J. Yang, G. Hua, B. Chen, and D. Wipf. Revisiting deep intrinsic image decompositions. 2018.

[14] E. Garces, A. Munoz, J. Lopez-Moreno, and D. Gutierrez. Intrinsic images by clustering. In *Computer Graphics Forum*, volume 31, pages 1415–1424, 2012.

[15] P. V. Gehler, C. Rother, M. Kiefel, and L. Zhang. Recovering intrinsic images with a global sparsity prior on reflectance. In *Proc. NIPS*, pages 765–773, 2011.

[16] R. Grosse, M. K. Johnson, E. H. Adelson, and W. T. Freeman. Ground truth dataset and baseline evaluations for intrinsic image algorithms. In *International Conference on Computer Vision (ICCV)*, pages 2335–2342, 2009.

[17] W. Hamilton, Z. Ying, and J. Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[18] M. Henaff, J. Bruna, and Y. LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[19] B. Kovacs, S. Bell, N. Snavely, and K. Bala. Shading annotations in the wild. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] B. Kovacs, S. Bell, N. Snavely, and K. Bala. Shading annotations in the wild. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[21] E. H. Land and J. J. McCann. Lightness and retinex theory. *JOSA*, 61(1):1–11, 1971.

[22] R. Li, S. Wang, F. Zhu, and J. Huang. Adaptive graph convolutional neural networks. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[23] S. Li and C. Yeo. Intrinsic images decomposition using a local and global sparse representation of reflectance. In *Proc. CVPR*, pages 697–704, 2011.

[24] W. Li, S. Saeedi, J. McCormac, R. Clark, D. Tzoumanikas, Q. Ye, Y. Huang, R. Tang, and S. Leutenegger. Interiornet: Mega-scale multi-sensor photo-realistic indoor scenes dataset. *arXiv preprint arXiv:1809.00716*, 2018.

[25] Y. Li, S. Hao, X. Guo, and L. Guibas. Syncspeccnn: Synchronized spectral cnn for 3d shape segmentation.

[26] Z. Li and N. Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. *European Conference on Computer Vision (ECCV)*, 2018.

[27] Z. Li and N. Snavely. Learning intrinsic image decomposition from watching the world. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[28] Z. Li and N. Snavely. Learning intrinsic image decomposition from watching the world. In *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[29] W.-C. Ma, H. Chu, B. Zhou, R. Urtasun, and A. Torralba. Single image intrinsic decomposition without a single intrinsic image. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[30] A. Meka, C. Richardt, and C. Theobalt. Live intrinsic video. *Acm Transactions on Graphics*, 35(4):109, 2016.

[31] F. Monti, D. Boscaini, J. Masci, E. Rodola, J. Svoboda, and M. M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5115–5124, 2017.

[32] T. Narihira, M. Maire, and S. X. Yu. Learning lightness from human judgement on relative reflectance. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2965–2973, 2015.

[33] T. Nestmeyer and P. V. Gehler. Reflectance adaptive filtering improves intrinsic image estimation. *IEEE Conference on Computer Vision and*

Pattern Recognition (CVPR), pages 6789–6798, 2017.

[34] L. Shen, P. Tan, and S. Lin. Intrinsic image decomposition with non-local texture cues. In *Proc. CVPR*, pages 1–7, 2008.

[35] J. Shi, Y. Dong, H. Su, and S. X. Yu. Learning non-lambertian object intrinsics across shapenet categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1694, 2017.

[36] J. Shi, Y. Dong, H. Su, and S. X. Yu. Learning non-lambertian object intrinsics across shapenet categories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1685–1694, 2017.

[37] J. Shi, Y. Dong, X. Tong, and Y. Chen. Efficient intrinsic image decomposition for RGBD images. In *Proc. VRST*, pages 17–25, 2015.

[38] M. Simonovsky and N. Komodakis. Dynamic edge-conditioned filters in convolutional neural networks on graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3693–3702, 2017.

[39] P. Sinha and E. Adelson. Recovering reflectance and illumination in a world of painted polyhedra. In *Proc. ICCV*, pages 156–163, 1993.

[40] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 190–198. IEEE, 2017.

[41] M. M. Takuya Narihira and S. X. Yu. Direct intrinsics: Learning albedo-shading decomposition by convolutional regression. In *IEEE International Conference on Computer Vision (CVPR)*, pages 2992–3000, 2015.

[42] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 52–67, 2018.

[43] Y. Wang, K. Li, J. Yang, and X. Ye. Intrinsic decomposition from a single rgb-d image with sparse and non-local priors. In *IEEE International Conference on Multimedia and Expo*, 2017.

[44] Q. Zhao, P. Tan, Q. Dai, L. Shen, E. Wu, and S. Lin. A closed-form solution to retinex with nonlocal texture constraints. *IEEE Trans. PAMI*, 34(7):1437–44, 2012.

[45] Q. Zhao, P. Tan, Q. Dai, L. Shen, E. Wu, and S. Lin. A closed-form solution to retinex with nonlocal texture constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 34(7):1437–1444, 2012.

[46] T. Zhou, P. Krahenbuhl, and A. A. Efros. Learning data-driven reflectance priors for intrinsic image decomposition. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3469–3477, 2015.