Intrinsic Decomposition via Ordinal Shading Supplementary Material

CHRIS CAREAGA AND YAĞIZ AKSOY, Simon Fraser University, Canada

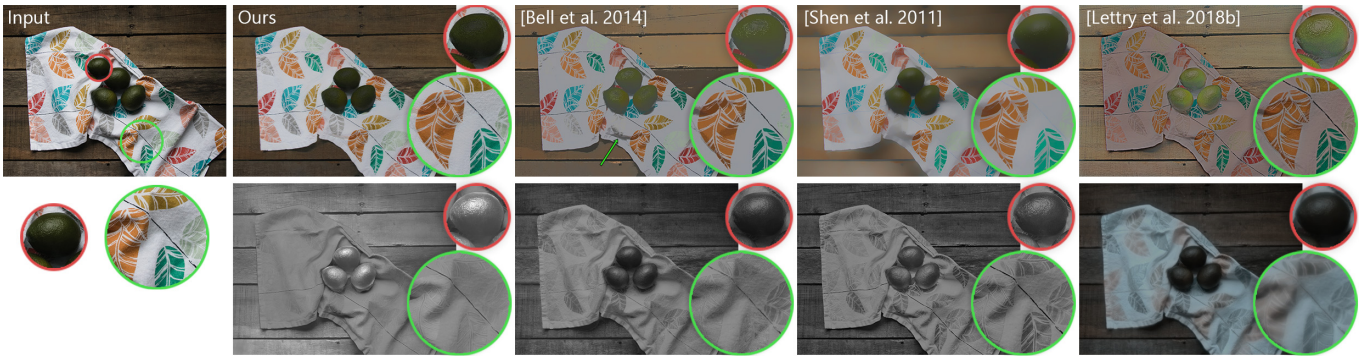

Fig. 1. Extension of Figure 1 in the main paper.

Image from Unsplash by Debby Hudson

ACM Reference Format:
Chris Careaga and Yağız Aksoy. 2023. Intrinsic Decomposition via Ordinal Shading Supplementary Material. *ACM Trans. Graph.* 1, 1 (October 2023), 9 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

In this document, we provide additional information pertaining to the implementation and preprocessing used for each dataset, prior method, and for our proposed pipeline. Additionally, we give an extended qualitative analysis comparing our method to prior works not included in the main paper. We also include alternate versions of various figures from the main paper. Finally, we provide an extended figure and discussion about the efficacy of our proposed multi-illumination training strategy.

A TRAINING DATASETS

We provide details on the datasets used to train our method, as well as the process of preparing the data for training of our networks.

A.1 CGIntrinsics

The CGIntrinsics dataset [Li and Snavely 2018a] is a synthetic dataset containing approximately 20,000 images rendered by the Mitsuba renderer. Each image is (640 x 480) pixels and comes with dense ground truth albedo values saved in PNG files. The lighting effects are mostly Lambertian with some colorful inter-reflections and light sources. The images consist mainly of simple, uncluttered indoor scenes.

A.2 Hypersim

The Hypersim dataset [Roberts et al. 2021] is a synthetic dataset containing over 74,000 high-quality renders of approximately 460 different indoor scenes. Each image is (768 x 1024) pixels and is factored into diffuse illumination, diffuse reflectance, and a non-diffuse residual. The images are diverse and contain realistic lighting effects. The images are provided as HDR images and we use the provided code to tonemap them to LDR without performing gamma compression. We then proportionally scale the albedo, to ensure it is in $[0, 1]$, and save it as a PNG image.

A.3 OpenRooms

The OpenRooms dataset [Li et al. 2021] is a synthetic dataset containing approximately 100,000 rendered images of 1,287 indoor scenes. The rendered images are (480 x 640) pixels and have realism and quality similar to the CGIntrinsics dataset. The dataset is provided with HDR input images that we tonemap and clip using the same technique as the other datasets

A.4 FSVG/GTA

To add more diverse image content, we utilize the dataset provided by [Krahenbuhl 2018]. The dataset consists of over 100,000 captures from the video game "Grand Theft Auto 5". The provided images are (600 x 800) pixels and mainly consist of outdoor scenes with roads, vehicles, buildings, trees, etc.

A.5 Multi-Illumination Dataset

The MIT Multi-Illumination Dataset [Murmann et al. 2019] consists of over 1000 scenes captured under 24 varying illuminations. The scenes are all taken indoors and contain various household objects.

Author's address: Chris Careaga and Yağız Aksoy, Simon Fraser University, Burnaby, BC, Canada.

0730-0301/2023/10-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

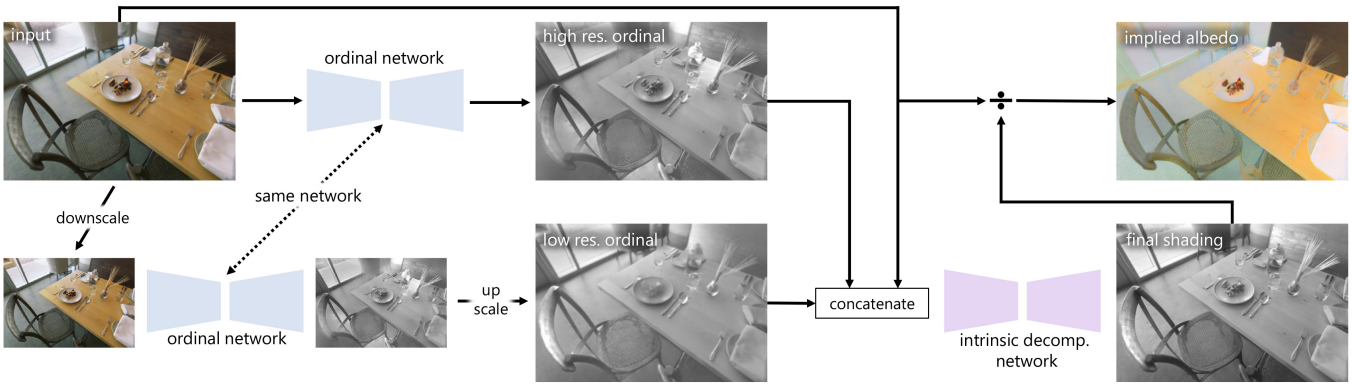

Fig. 2. Our proposed intrinsic decomposition pipeline. Image from Unsplash by Erik Binggeser.

Each image is captured using an automated bounce-flash. The provided images are 1000x1500 in JPEG or EXR format, we tonemap them using the same process as the synthetic datasets. Since each image contains both a specular and diffuse light probe, we also white balance the images using the diffuse light probe.

A.6 Preprocessing

For each dataset, we store the input images and their corresponding albedo. We mask the ground-truth intrinsic components by examining the albedo values at each pixel. If the value is below 0.004 we mark it as invalid since it is likely that the pixel does not have a reliable albedo or shading value because; information was lost during the image compression process, the pixel is on a sky-box, or the pixel is on a highly non-Lambertian surface (such as glass). When training models, we load the input images and albedo and compute the *implied shading*. In other words, given an image I and its albedo component R we can compute the implied shading as $S = I/R$. We use these values to supervise our models.

B EVALUATION DATASETS

We provide specific details about the collection and processing of the datasets used for evaluation of our proposed method.

B.1 ARAP Dataset

For the ARAP dataset, we collect the scenes provided by Bonneel et al. [2017]. We include the scenes given in the supplementary material as well as the extended set of scenes provided on their website. There are 52 scenes in total, each scene may have multiple illuminations resulting in 157 different images. For each data point, the authors provide the image and albedo. To compute the shading we divide the image by the albedo. This results in a 3-channel colorful shading component. For fair comparison, we desaturate the shading and re-multiply by the albedo to generate a white-balanced input image. Additionally, we create a mask that omits pixels with a very low albedo or shading value, in order to avoid evaluating methods on pixels with inaccurate values or lost information.

B.2 IIW Dataset

We utilize the original implementation of the WHDR metric from [Bell et al. 2014] and the test set split from Narihira et al. [2015]. For our pipeline, we resize using the scheme described in Section 4.1 of the main document. This generally results in an upscaled image, even though no information is introduced, we find that the network can produce better details when small images are upscaled. We then send the image through our pipeline and downscale it to its original size before evaluation.

B.3 SAW Dataset

For the SAW dataset, we follow the same resizing scheme used for the IIW dataset. For evaluation, we utilize the implementation provided by Li and Snavely [2018a].

C EVALUATION METRICS

We provide specific details on the parameters and implementation used for each metric used to evaluate ordinal estimations.

C.1 Ordinal

We utilize the pair-wise ordinal metric (Ord.) proposed by Xian et al. [2020] to evaluate the quality of ordinal estimations. This metric randomly samples pairs of points from the image and compares their ordinal relationship to that of the ground truth. We found that if enough points are sampled, this metric varies little from run to run. Nevertheless, we chose to precompute the sampled points and reuse the same set of points for each model tested. For both resolutions, we uniformly sample 10,000 pairs from the valid pixels in each image.

C.2 D3R

We use the implementation of the D3R metric proposed by Miangoleh et al. [2021] for measuring the accuracy of ordinal relationships across local image boundaries. The metric first generates a superpixel segmentation of the image. To determine the ordinal relationship between two pixels, we compute the ratio of their values. For all neighboring pairs of superpixels, we filter out pairs whose ratio is in $[0.9, 1.1]$. This leaves pairs that have a sufficient difference

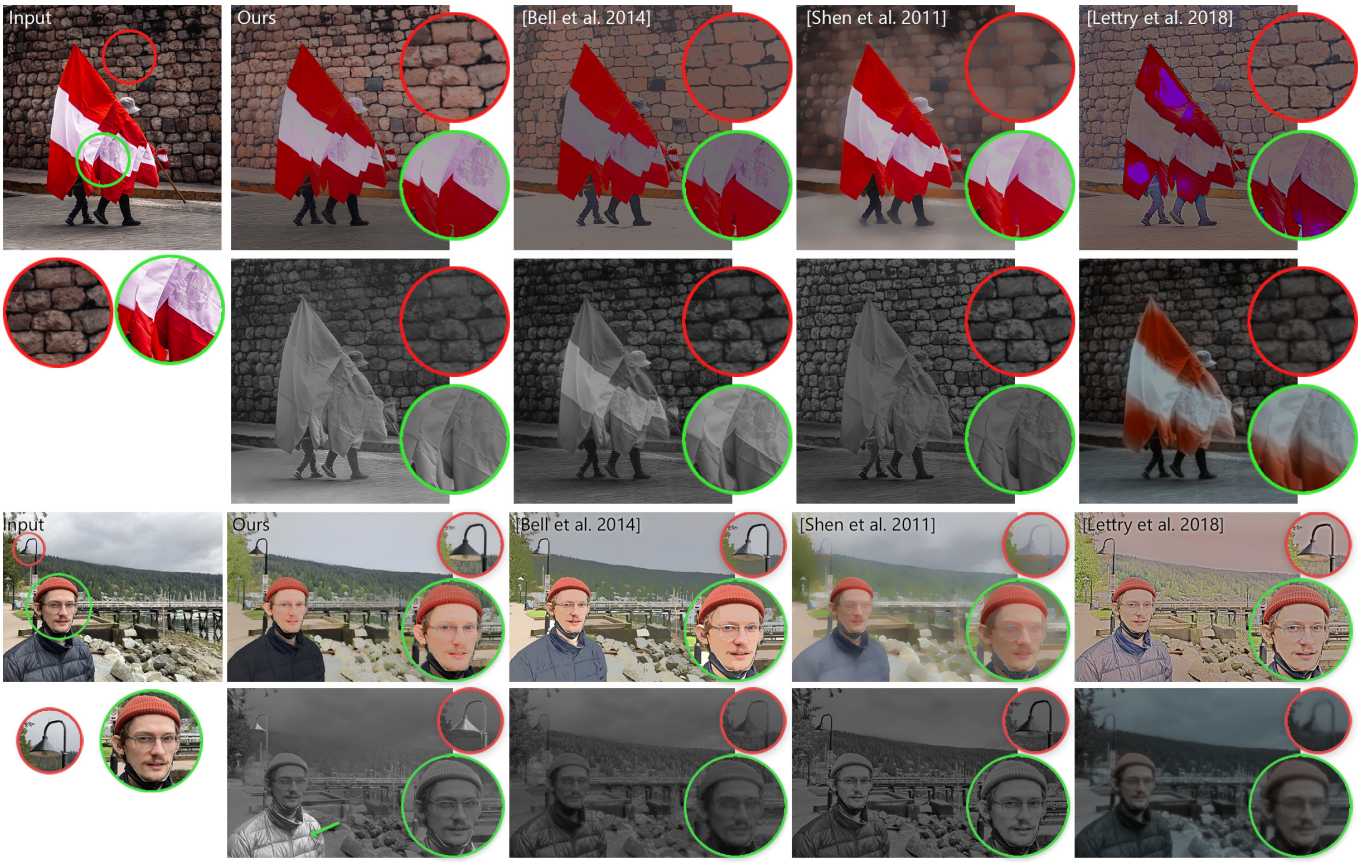

Fig. 3. Extension of Figure 2 in the main paper.

Top image from Unsplash by Mauro Lima

in brightness between them, denoted as $\mathcal{P}$. For each of these pairs, we use the ground-truth shading S to compute the ground-truth ordinal relationships:

$$\bar{r}_{ij} = \begin{cases} 0 & S_i/S_j < 1 \\ 1 & S_i/S_j > 1 \end{cases} \tag{1}$$

The same process is carried out on the estimated ordinal shading to generate r_{ij}. The D3R metric is then computed as:

$$\frac{1}{\mathcal{P}} \sum_{ij \in \mathcal{P}} |\bar{r}_{ij} - r_{ij}| \tag{2}$$

We do not consider any superpixels that are invalid according to the provided mask for each image. We compute superpixels using an implementation of SLIC (specifically SLIC-zero). For the 384 pixel images we compute 1000 superpixels, and for the $\mathcal{R}_0$ resolution images we compute 3000 superpixels. For both, we set the compactness parameter to 0.001. We find these parameters yield uniform segments that still follow image gradients.

D TRAINING IMPLEMENTATION DETAILS

We provide additional details pertaining to the training of both the networks in our proposed pipeline.

Training	CGI	OR	HS	FSVG	MI
Synthetic Only	0.10	0.15	0.30	0.35	–
With Multi-Illumination	0.10	0.10	0.20	0.20	0.40

Table 1. Sampling probabilities for each dataset when training the ordinal network. Values are chosen to reflect each dataset's size and quality.

D.1 Ordinal Training

We perform extensive augmentation on the ground-truth intrinsic components when training the ordinal network. First, we perform random hue and saturation shifting on the provided albedo values. We additionally compute a random scaling of the red and blue channels of the albedo to simulate random white balancing. We combine these altered albedo components with the ground-truth shading to generate a novel input image. To make our predictions more robust to changes in resolution, we also perform random scaling of inputs before randomly cropping a fixed-sized patch the same size as the network's receptive field (384 x 384). This helps the network learn to generate quality predictions at any resolution. Finally, the inputs and ground truth are horizontally flipped with a 50% probability.

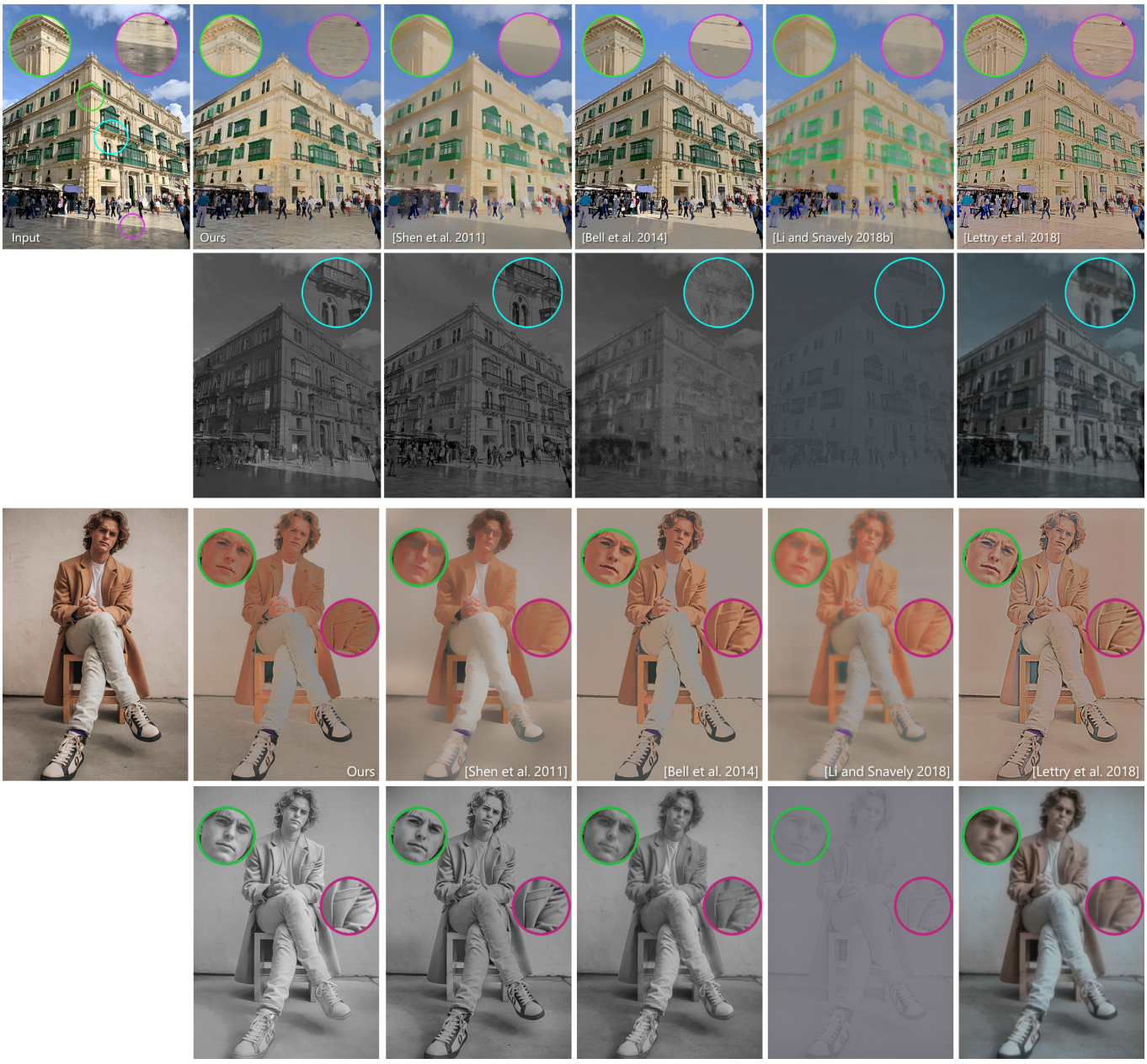

Fig. 4. Extension of Figure 12 in the main paper.

Images from Unsplash by William Jones (left) and Austin Wade.

We train the ordinal network with a batch size of 8. For each batch we randomly sample images from each dataset, the datasets are sampled with the probabilities shown in Table 1. These probabilities are chosen to reflect the size and quality of each dataset. We first train on only synthetic datasets for 700,000 iterations. We then use these weights to generate the multi-illumination data. We continue training for 300,000 iterations with dataset probabilities that are biased toward the multi-illumination data.

Our main signal for ordinal shading estimation is a scale-and-shift invariant (SSI) loss function on inverse shading, where the scale and shift is estimated using least squares during training. As mentioned in Section 5.2 of the main text, the least-squares estimation that is required for scale-invariant losses for full intrinsic decomposition creates instability especially during the early stages of the training. This instability also applies to the SSI loss, although it is less pronounced as the SSI least squares fit is computed on the bounded inverse shading domain. While our inverse shading

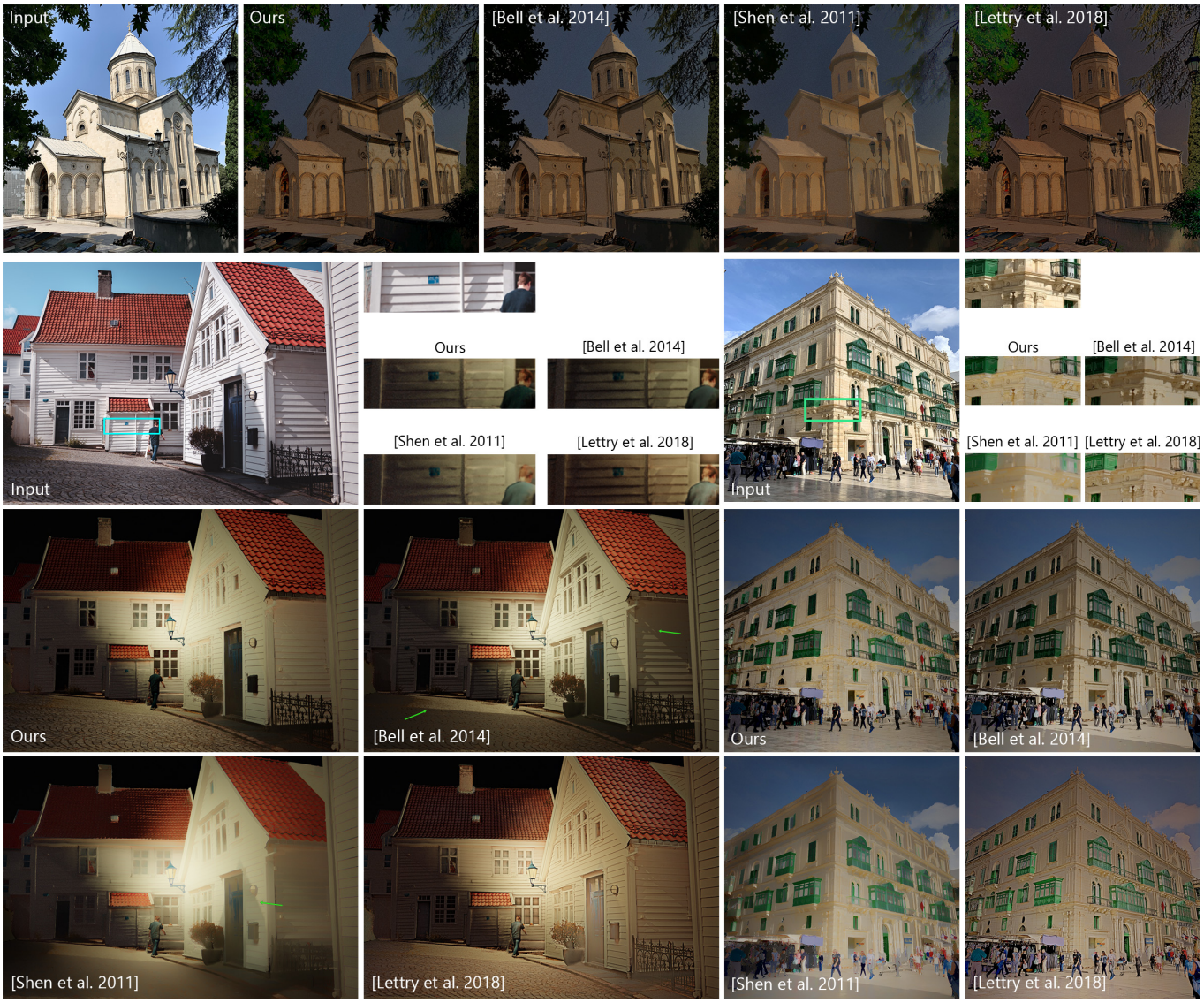

Fig. 5. Extension of Figure 20 in the main paper. All images from Unsplash, bottom left by Lieuwe Terpstra and bottom right by William Jones

formulation does not completely alleviate this issue, we do not observe any issues early on in training. We believe, however, that early training stability can also depend on implementation details such as network architecture and weight initialization. In the case that the training diverges from unstable scale and shift estimation, the ordinal network can be trained with vanilla MSE for a few iterations until network outputs are in a reasonable range. Once the network roughly picks up the task and starts generating somewhat reliable estimates, the SSI loss formulation can be used for the rest of the training.

D.2 Intrinsic Decomposition Training

We train our second network on the high-definition samples from the Hypersim dataset. Since the ordinal network is also trained on the Hypersim dataset, we do not expect the ordinal estimations to exhibit the same artifacts and inconsistencies as real-world images. This results in our high-resolution network learning the trivial solution of simply outputting the high-resolution ordinal input. To mitigate this, we apply augmentation on the high-resolution ordinal estimations by adding low-frequency artifacts that simulate the issues we observe on real images. The network is then forced to learn how to leverage the information given in the low-resolution estimation because it contains the global consistency that the high-resolution estimation lacks.

We begin training the high-resolution network on the Hypersim dataset. We train with a batch size of 8 for 200,000 iterations. We use these weights to generate the multi-illumination data. We then train

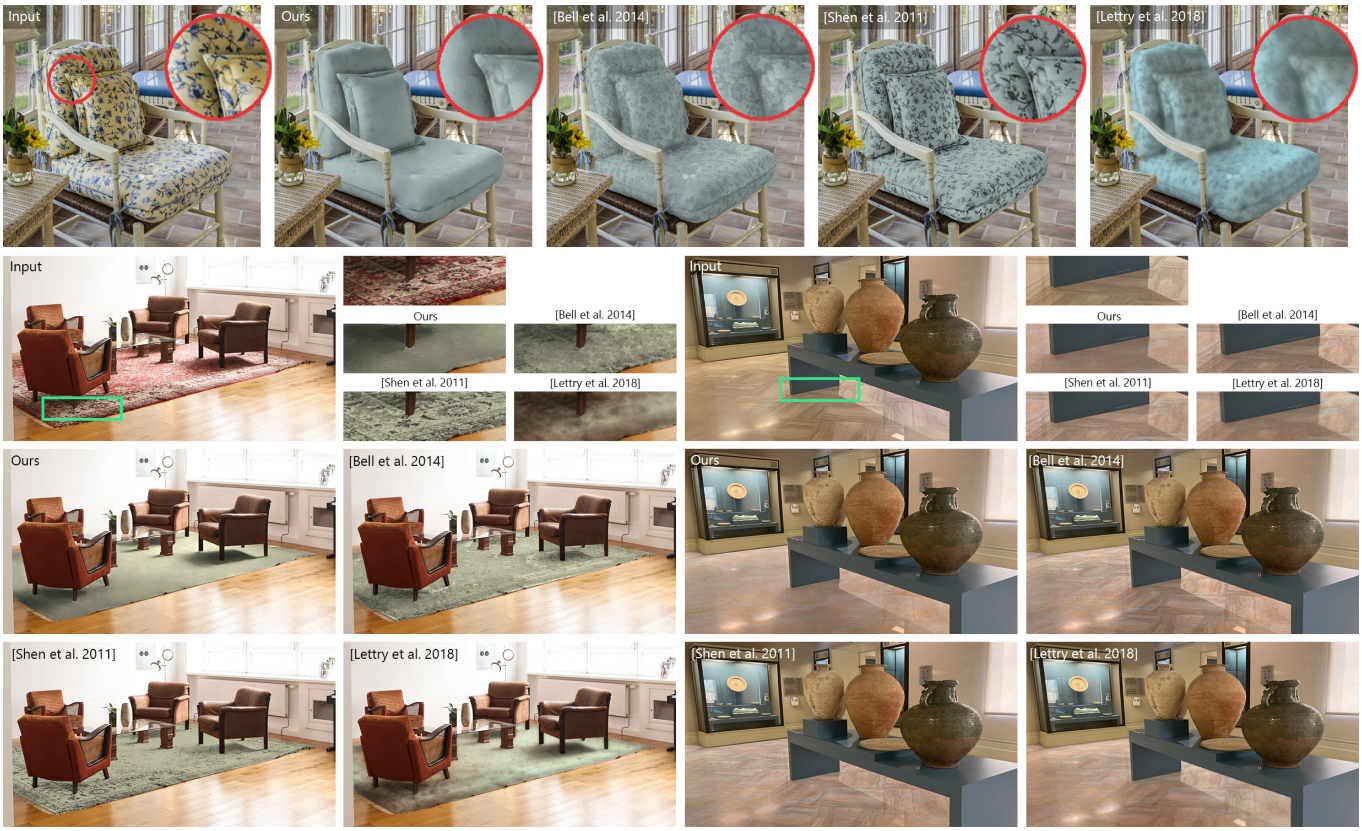

Fig. 6. Extension of Figure 21 in the main paper.

Two images from Unsplash by Francesca Tosolini (top) and Beazy (bottom left).

for 200,000 more iterations with batches consisting of 4 Hypersim images and 4 multi-illumination images.

D.3 Losses

For the multi-scale gradient loss we utilize the code provided by Li and Snavely [2018a]. To compute each scale, the prediction and ground-truth are downscaled to half the size of the previous scale, and the gradient is computed with finite differences. Following Li and Snavely [2018a] we use a pyramid consisting of 4 scales.

D.4 CGIntrinsics-Only Training

For the metrics measured in Table 1 of the main paper, we train a version of our proposed approach with only the CGIntrinsics dataset [Li and Snavely 2018a] to show the relative performance contributions from our formulation and datasets separately. We train both networks in our pipeline, using only the CGIntrinsics dataset which consists of 20,000 synthetic examples. We exclude the extra renderings from the ARAP dataset distributed with the CGIntrinsics dataset. We train the ordinal network for 150 epochs using our proposed ordinal loss formulation. We then train our second network for 150 epochs on square crops of size (480×480), rather than (512×512), due to the smaller image size of the CGIntrinsics dataset. Other than that, all other training settings remain the same.

E COMPETING METHODS IMPLEMENTATION DETAILS

Given that there is no standardized way to perform data pre-processing for intrinsic decomposition, we find that many methods vary in the resizing and linearization process of input images and intrinsic components. The Lambertian formulation in the intrinsic equation assumes a linear image. This means that any aspects of implementation pertaining to the reconstruction of the input image (e.g. reconstruction losses), should be formulated in linear RGB.

Additionally, nearly all data-driven approaches to intrinsic decomposition utilize fully-convolutional neural networks. This means that the networks can process an image at any resolution (assuming its dimensions are divisible by certain values that vary by architecture). Despite this, many works opt to always resize images to a specific resolution in order to generate stable results from their networks.

We find that some of the open-source implementations of prior works are either ambiguous or unclear in terms of these details. We attempt to reconcile the intended usage of each method to ensure fair comparisons.

[Li and Snavely 2018a]. Following the open-source training implementation released by [Li and Snavely 2018a], we assume that their network expects an sRGB image. According to their reconstruction loss, their intrinsic components should combine to create the linear

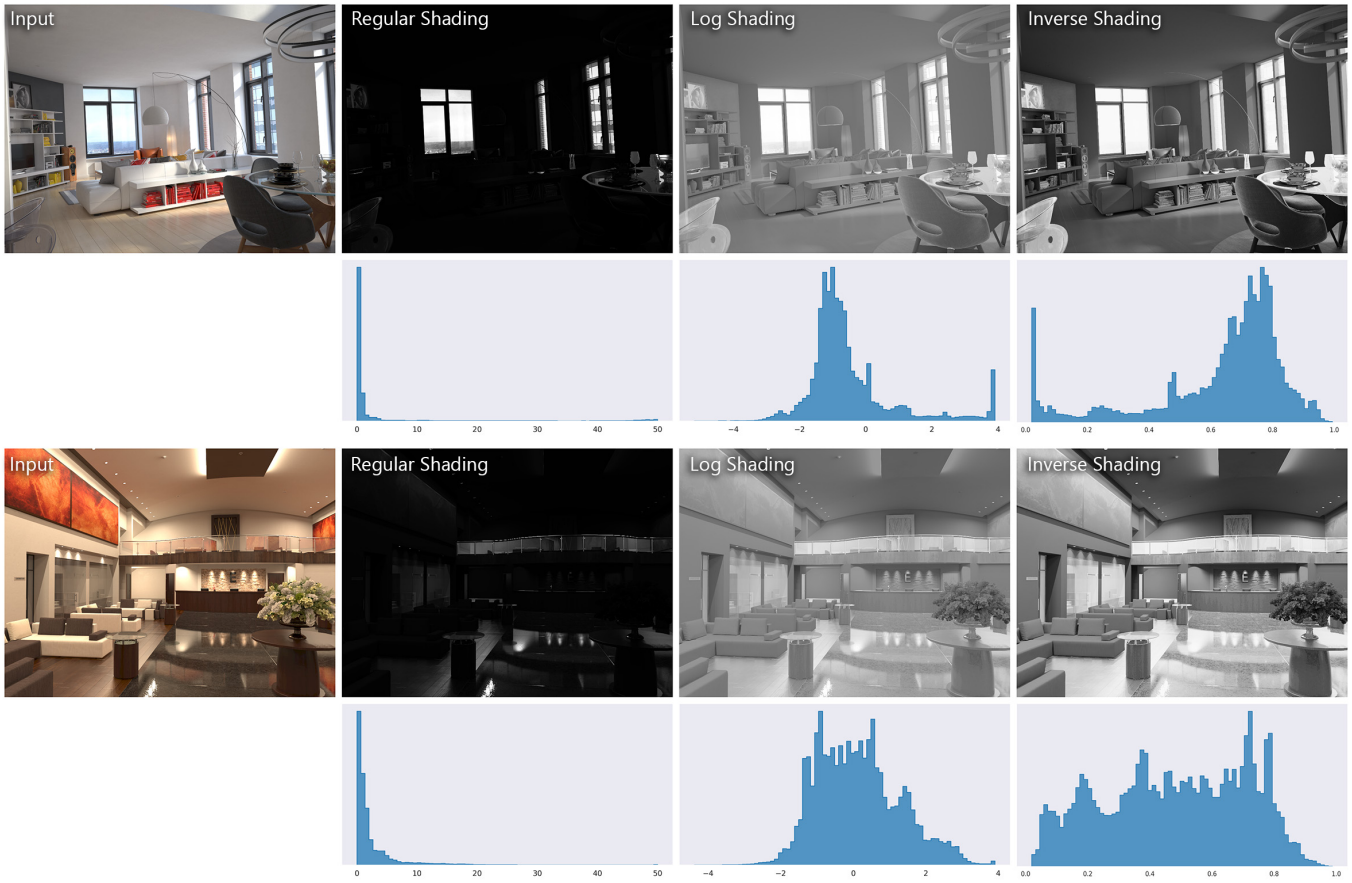

Fig. 7. Extension of Figure 5 in the main paper.

version of the image. Therefore we assume that their network outputs linear albedo and shading. We utilize their provided functions for converting between linear RGB and sRGB. The authors do not provide an inference script, therefore during evaluation, we utilize their resizing function used in their IIW and SAW evaluations that down-scales the image while maintaining the aspect ratio.

[Luo et al. 2020]. The network of Luo et al. [2020] utilizes the CGIntrinsics dataset and therefore uses the same linearization process as Li and Snavely [2018a] (sRGB input, linear outputs). They use a resizing function similar to that of Li and Snavely [2018a] but with smaller sizes.

[Das et al. 2022]. The open-source implementation provided by Das et al. [2022] does not include training code, therefore we follow the pre-processing included in the inference script. The input image is resized to a constant size of (256×256). The network seems to reconstruct whatever image is provided to it, therefore it is assumed that the network should be provided with a linear image to produce linear components. No linearization is included in the inference script, and we observed more artifacts when using linear images as input. For qualitative results, we feed the network sRGB images to avoid artifacts. For quantitative results on the ARAP dataset, the model performs well with the provided linear images, and the reconstruction error is computed against the linear image, therefore we evaluate the method using the linear inputs.

[Baslamisli et al. 2018]. The authors of Baslamisli et al. [2018] release their inference code but do not provide training code. Since the inference script does not linearize the input image, we assume both the input and output are sRGB. When computing reconstruction error we multiply the outputs and compare it with the input image. The authors perform a constant resize of input images to (352×480) which we use for evaluation on the ARAP dataset.

[Lettry et al. 2018]. Following the inference code provided by Lettry et al. [2018], we do not linearize the input image given to their network. Since they define their output in terms of shading and implied albedo like our approach, we assume the output components are not linear (since the input image is not). We slightly alter their code when saving outputs in order to avoid over-saturation of the output shading. To avoid large outliers when normalizing, we divide the shading by the 99^{th} percentile value and then clip between 0 and 1. The authors do not resize the image as part of their inference script, therefore we assume the network can take images of any size.

Fig. 8. Extension of Figure 11 in the main paper. We provide two examples with albedo estimations added. We observe a noticeable improvement in both shading and albedo when training our network using our proposed multi-illumination pseudo-ground-truth. Our predictions become more globally consistent (top row inset), as well as more detailed around image content not seen during training (bottom row inset). Images from Unsplash by Ladislav Stercell (horse) and Philip Myrtorp (apples).

[Bell et al. 2014]. The code provided by Bell et al. [2014] assumes that the input image is linear, and makes sure to convert it before running their algorithm. Since this method is a traditional optimization method, there is no training-dependent resizing and we can therefore feed images in at any resolution.

[Shen et al. 2011]. Although it is not stated, we assume that Shen et al. [2011] expects images in linear RGB. Due to their formulation, the predicted components always reconstruct the input image, and therefore we can conclude that the components are in linear RGB as well. Similar to Bell et al. [2014] we do not perform any fixed resizing since their algorithm accepts images of any size.

F EXTENDED QUALITATIVE ANALYSIS

In the main document, we focus our qualitative evaluation on recent deep learning methods that yield high scores on benchmark datasets. Here, for completeness, we provide further evaluation against more existing methods, including two optimization-based methods and two deep learning methods. Specifically, we evaluate our proposed method against Bell et al. [2014], Shen et al. [2011], Li and Snavely [2018b] and Lettry et al. [2018]. We provide results for these four methods, along with the methods discussed in the main paper for 100 in-the-wild images in our supplementary material.

[Bell et al. 2014]. The authors of Bell et al. [2014] propose an optimization approach to intrinsic decomposition that uses multiple low-level descriptors and corresponding priors to generate plausible decompositions with desirable properties. Their method takes multiple minutes to process a megapixel image. Due to the assumption that albedo is piece-wise sparse, their method tends to incorrectly attribute hard shadows to the albedo component. This can be seen on the hand towel in Figure 1 as well as on the left building facade in Figure 4. This causes residual shadows when using their albedos for relighting, as seen in the house example in Figure 5. We also observe difficulties in regions with high-frequency albedo changes. Their method is unable to separate low-frequency shading changes from high-frequency albedo changes, causing issues when performing recoloring. The effects of this can be seen in the chair and carpet examples in Figure 6. Finally, while their method is able to generate sparse albedos, the corresponding shading often lacks global coherency. For example, the hand towel in Figure 1 has a higher magnitude than the wooden background. Since these two image regions share similar orientations, they should have similar shading magnitudes.

[Shen et al. 2011]. Similar to Bell et al. [2014], the authors of Shen et al. [2008] propose an optimization-based approach to intrinsic decomposition. Their method also requires multiple minutes of runtime for a megapixel image. Their method exhibits similar behavior to Bell et al. [2014] often times attributing hard shadows to the albedo component resulting in inaccurate relighting. Their method also tends to generate over-smoothed albedo components which results in high-frequency information being attributed to shading, this limits the recoloring capabilities of the method as seen in Figure 6.

[Li and Snavely 2018b]. The method proposed by Li and Snavely [2018b] is an unsupervised approach to intrinsic decomposition that leverages multi-illumination data and multiple dense priors to train a CNN. Their method outputs albedo, shading, and a single shading color which is multiplied by the grayscale shading. We show

some examples of their outputs in Figure 4. We observe that their shading generally has very low contrast. Their albedo prediction for the building assigns a very saturated green to the windows and incorrectly contains the shadows on the building and street. Their albedo prediction for the portrait image does not represent the sharp changes in albedo and contains the shadows on the floor.

[Lettry et al. 2018]. The authors of Lettry et al. [2018] also propose an unsupervised approach based on multi-illumination data. Their method predicts colorful shading and ensures reconstruction by generating albedo from their shading prediction and the input image. We observe that their shading often exhibits color leakage from the albedo, we can see this on the leaves of the hand-towel in Figure 1, and on both examples in Figure 4. This leakage of colors into the shading results in shifted colors in the albedo, which subsequently affects the accuracy of the relit examples shown in Figure 5. Their shading predictions are also low-frequency which affects the realism of recoloring effects. We can see this in the edits given in Figure 6

G EXTENDED DISCUSSION AND RESULTS

We provide an extended Figure showing the effectiveness of our multi-illumination as well as discuss the difficulty of qualitative evaluation on real-world images and editing tasks. We also provide in-the-wild comparisons on 100 photographs in the supplementary zip file.

G.1 Multi-Illumination Training Ablation

In Figure 8 we provide an extended version of Figure 11 in the main paper. We show the effect that multi-illumination training has on our pipeline. In the top row, we can see the model fails to predict consistent shading across the apples, this results in an inconsistent albedo intensity as shown in the inset. In the bottom row, we can see that the multi-illumination training helps the model generalize to content unseen during training, such as humans. The inset shows that model can make much more accurate shading estimations on human faces resulting in a more sparse albedo component with fewer leftover shading effects.

Although the performance difference is apparent qualitatively, it is difficult to properly evaluate the improvement numerically. We find that the multi-illumination training allows the model to generalize to real-world scenes. Evaluating this ablation on the ARAP dataset does not measure this improvement as the images are rendered and generally simpler than real scenes. Although real-world datasets exist, they fail to provide meaningful comparisons of performance as discussed in Section 8 of the main paper. For these reasons, we rely on qualitative evaluation to draw conclusions about the performance improvement gained from multi-illumination training.

REFERENCES

A. S. Baslamisli, T. T. Groenestege, P. Das, H. A. Le, S. Karaoglu, and T. Gevers. 2018. Joint Learning of Intrinsic Images and Semantic Segmentation. In *European Conference on Computer Vision*.

Sean Bell, Kavita Bala, and Noah Snavely. 2014. Intrinsic Images in the Wild. *ACM Transactions on Graphics* 33, 4 (jul 2014), 1–12.

Nicolas Bonneel, Balazs Kovacs, Sylvain Paris, and Kavita Bala. 2017. Intrinsic Decompositions for Image Editing. *Computer Graphics Forum (Eurographics State of the Art Reports 2017)* 36, 2 (2017).

Partha Das, Sezer Karaoglu, and Theo Gevers. 2022. PIE-Net: Photometric Invariant Edge Guided Network for Intrinsic Image Decomposition. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR)*.

Philipp Krahenbuhl. 2018. Free Supervision from Video Games. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2955–2964.

L. Lettry, K. Vanhoey, and L. Van Gool. 2018. Unsupervised Deep Single-Image Intrinsic Decomposition using Illumination-Varying Image Sequences. *Computer Graphics Forum* 37, 7 (2018), 409–419.

Zhengqi Li and Noah Snavely. 2018a. CGIntrinsics: Better Intrinsic Image Decomposition through Physically-Based Rendering. In *European Conference on Computer Vision (ECCV)*.

Zhengqi Li and Noah Snavely. 2018b. Learning Intrinsic Image Decomposition from Watching the World. In *Computer Vision and Pattern Recognition (CVPR)*.

Zhengqin Li, Ting Yu, Shen Sang, Sarah Wang, Mengcheng Song, Yuhan Liu, Yu-Ying Yeh, Rui Zhu, Nitesh B. Gundavarapu, Jia Shi, Sai Bi, Hong-Xing Yu, Zexiang Xu, Kalyan Sunkavalli, Miloš Hašan, Ravi Ramamoorthi, and Manmohan Chandraker. 2021. OpenRooms: An Open Framework for Photorealistic Indoor Scene Datasets. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 7186–7195.

Jundan Luo, Zhaoyang Huang, Yijin Li, Xiaowei Zhou, Guofeng Zhang, and Hujun Bao. 2020. NIID-Net: Adapting Surface Normal Knowledge for Intrinsic Image Decomposition in Indoor Scenes. *IEEE Transactions on Visualization and Computer Graphics* (2020).

S Mahdi H Miangoleh, Sebastian Dille, Long Mai, Sylvain Paris, and Yağız Aksoy. 2021. Boosting Monocular Depth Estimation Models to High-Resolution via Content-Adaptive Multi-Resolution Merging. In *Proc. CVPR*.

Lukas Murmann, Michael Gharbi, Miika Aittala, and Fredo Durand. 2019. A Multi-Illumination Dataset of Indoor Object Appearance. In *2019 IEEE International Conference on Computer Vision (ICCV)*.

Takuya Narihira, Michael Maire, and Stella X Yu. 2015. Learning Lightness from Human Judgement on Relative Reflectance. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 07-12-June. 2965–2973.

Mike Roberts, Jason Ramapuram, Anurag Ranjan, Atulit Kumar, Miguel Angel Bautista, Nathan Paczan, Russ Webb, and Joshua M. Susskind. 2021. Hypersim: A Photorealistic Synthetic Dataset for Holistic Indoor Scene Understanding. https://arxiv.org/pdf/2011.02523.pdf

Jianbing Shen, Xiaoshan Yang, Yunde Jia, and Xuelong Li. 2011. Intrinsic images using optimization. In *CVPR 2011*. 3481–3487.

Li Shen, Ping Tan, and Stephen Lin. 2008. Intrinsic image decomposition with non-local texture cues. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*. 1–7. https://doi.org/10.1109/CVPR.2008.4587660

Ke Xian, Jianming Zhang, Oliver Wang, Long Mai, Zhe Lin, and Zhiguo Cao. 2020. Structure-Guided Ranking Loss for Single Image Depth Prediction. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.