Colorful Diffuse Intrinsic Image Decomposition in the Wild Supplementary Material

CHRIS CAREAGA and YAĞIZ AKSOY, Simon Fraser University, Canada

ACM Reference Format:
Chris Careaga and Yağız Aksoy. 2024. Colorful Diffuse Intrinsic Image Decomposition in the Wild Supplementary Material. *ACM Trans. Graph.* 43, 6, Article 178 (December 2024), 6 pages. https://doi.org/10.1145/3687984

1 INTRODUCTION

In this supplementary document we provide additional descriptions of the datasets we used to train our pipeline, implementation details pertaining to the training and evaluation of our pipeline along with competing methods, and finally additional qualitative examples of our results and comparisons to prior intrinsic decomposition methods.

2 IMPLEMENTATION DETAILS

In this section, we first provide additional information on each of the datasets we used to train our method. We then detail the specific training process used for each of our networks, including the augmentation, data loading, and training time. Finally, we explain the preprocessing and evaluation process for each dataset used in our quantitative analysis.

2.1 Datasets

In order to train our pipeline we collect multiple publicly available datasets. Table 1 provides a tabular overview of some of the aspects of each dataset.

Hypersim. Hypersim is a rendered dataset consisting of 461 indoor scenes with approximately 74,000 frames at (1024 x 768) pixel resolution. The scenes are rendered using V-Ray, and various intrinsic components are provided as uncompressed high dynamic range images. This is the only dataset that provides accurate ground-truth diffuse shading, therefore we are able to use it to train our final diffuse shading network.

GTA. The GTA dataset consists of over 200,000 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. Many of the frames are redundant as they come from multiple successive captures, typically from a vehicle. For each RGB frame, a corresponding albedo is provided as an 8-bit PNG.

Authors' address: Chris Careaga; Yağız Aksoy, Simon Fraser University, Burnaby, BC, Canada.

0730-0301/2024/12-ART178 $15.00
https://doi.org/10.1145/3687984

Table 1. Summary of the datasets used to train our method

Dataset	# of Images	Scene Type	Resolution
Hypersim [Roberts et al. 2021]	70838	Indoor	(1024 x 768)
GTA [Krahenbuhl 2018]	223197	Driving	(600 x 800)
Structure3D [Zheng et al. 2020]	78463	Indoor	(1280 x 720)
InteriorVerse [Zhu et al. 2022]	52557	Indoor	(640 x 480)
EDEN [Le et al. 2021]	368663	Gardens	(640 x 480)
Matrix City [Li et al. 2023]	44804	Driving/Aerial	(1000 x 1000)/(1920 x 1080)
Lumos [Yeh et al. 2022]	28319	Humans	(512 x 512)
PRID [Wang et al. 2022]	21475	Indoor	(640 x 480)
MIDIntrinsic [Murmann et al. 2019]	25000	Indoor	(1000 x 1500)

Structure3D. The Structure3D dataset consists of approximately 3,300 rendered indoor scenes resulting in 78,000 frames each with a resolution of (1280 x 720). The frames are rendered using a proprietary ray-tracing engine. Each RGB frame is accompanied by a corresponding albedo stored as a PNG.

InteriorVerse. The InteriorVerse dataset consists of approximately 4,000 rendered indoor scenes resulting in about 52,000 frames each with a resolution of (640 x 480). The images and corresponding albedos are provided as HDR images in the EXR format. We tonemap the images using the simple scheme used by prior works [Careaga et al. 2023; Roberts et al. 2021] without gamma correcting the images to maintain linearity. The dataset also includes roughness parameters and lighting information which we do not utilize.

EDEN. The EDEN dataset consists of approximately 368,000 rendered frames depicting procedurally generated garden scenes. The scenes are rendered using Blender at a resolution of (640 x 480). The dataset provides images and corresponding albedo, as well as various shading layers generated by Blender. We find that the diffuse shading provided is oftentimes noisy and is clipped as the images are stored as PNGs. For this reason, the data is not suitable for training our diffuse shading network.

Matrix City. The Matrix City dataset consists of two large-scale city scenes rendered from many different views using the Unreal Engine. The authors have released albedo for the smaller of the two cities, and data consists of both aerial and street-view subsets. The street-view frames are rendered at a resolution of (1000 x 1000) while the aerial images are rendered at (1920 x 1080). Overall a total of approximately 44,000 frames have corresponding albedo ground truth. The dataset also provides specular and roughness maps which we do not utilize. Although this dataset is a quality source of outdoor data, the rendered images generally have the same overcast weather and therefore lack hard shadows present in real images.

Lumos. The Lumos dataset consists of multiple procedurally generated human character models captured in a virtual lightstage setting. Due to data downloading issues, we are only able to use a subset of the dataset. Specifically, we use 50 of the 500 identities resulting in approximately 28,300 images at a resolution of (512

Fig. 1. Extension of Figure 6 in the main paper.

Image from Unsplash by Slidebean

x 512). The corresponding albedo for each image is provided as a JPEG. The authors also provide various other intrinsic components, including diffuse shading. Since there is not clear description of how the components combine to reconstruct the image, and they are stored in a compressed LDR format, we do not use these components to train our diffuse shading network.

PRID. The "Photo-Realistic Intrinsic Dataset" consists of 21,000 rendered images of indoor scenes. The dataset provides images, albedo, and masks at a resolution of (640 x 480). The dataset also provides what appears to be diffuse shading but it is not discussed exactly what component this is in the datasets description. Additionally it is stored in a clipped form as PNG files, which makes it not suitable for training our diffuse shading network.

MIDIntrinsic. The MIDIntrinsic dataset is an augmented version of the Multi-Illumination Dataset (MID) [Murmann et al. 2019] proposed by Careaga and Aksoy [2023]. The MID provides 1,000 indoor scenes captured under 25 different illumination conditions generated using an automated bounce-flash rig. Each illumination image has a resolution of (1000 x 1500). The MIDIntrinsics dataset consists of the 1,000 original scenes from MID but with pseudo ground truth albedo provided using the method of Careaga and Aksoy [2023]. We use the provided albedo as ground truth to train all of our networks (except for the diffuse shading network). The images are white-balanced using the diffuse gray ball therefore each illumination appears colorless. To create an input image, we sample 1-3 illumination, then shift the color of each illumination by adding a global offset to the a and b channels in the Lab colorspace. By linearly combining these images with a randomly sampled set of alpha values that add up to 1, we end up with a mixed illumination image that we can use as input to our network. The pseudo ground truth albedo is computed from the white-balanced images, therefore we can ensure it represents the underlying color content of the original scene. There can be baked-in shading colors in the pseudo ground truth albedo due to effects like inter-reflections that are not dependent on the main illumination of the scene. We find that this has little effect on our pipeline's final performance and does not outweigh this dataset's importance as it is the sole real-world training dataset.

2.2 Training

We train each of our models with similar settings. Each network has the same architecture and is trained with a batch size of 12. Each image in the batch is (384 x 384). We generate the training images by sampling random-sized square crops and resizing them. We perform horizontal and vertical flipping after generating our crops. In order to speed up training by decreasing I/O, we cache images when loading data. The images are cached in RAM at full

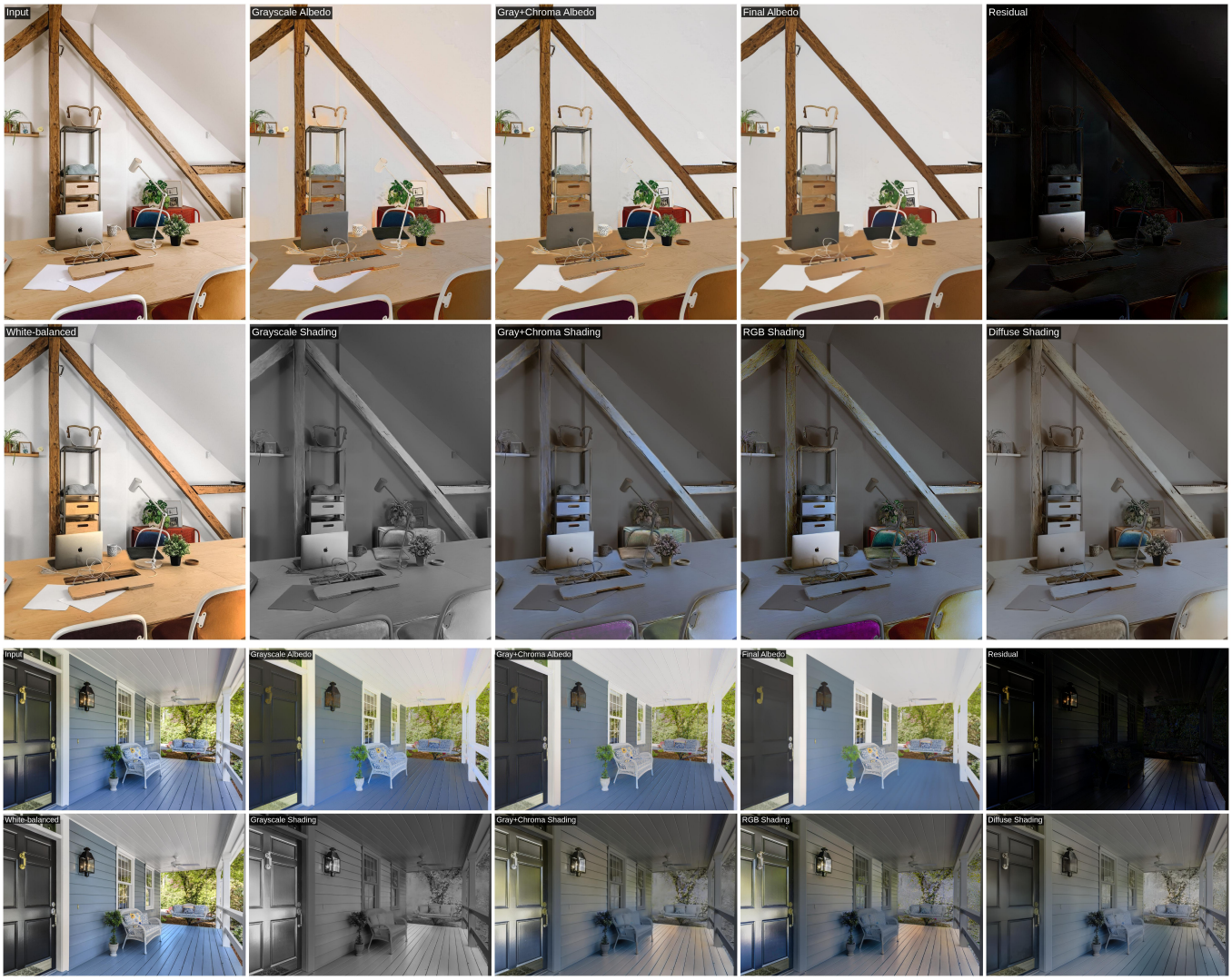

Fig. 2. Examples of each component in our pipeline. Top row: input, albedo from Careaga and Aksoy [2023], albedo generated by chroma network, albedo generated by the albedo network, specular residual. Bottom row: white-balanced image, shading from Careaga and Aksoy [2023], grayscale shading plus estimated chroma, shading from our final albedo, and our diffuse shading. Images from Unsplash by Toa Heftiba (top) and Francesca Tosolini.

resolution before any augmentation. When the dataloader is queried for a new datapoint, either a new datapoint is loaded from disk (and subsequently cached) or an existing cached datapoint is returned. The probability of using a cached example is set to 70% to minimize the large reads that are required to load multiple high-resolution intrinsic components. Note that the data points are cached pre-augmentation, therefore even if the same image is returned in a single batch, it will likely have a completely different augmentation applied to it. For each batch, we sample data points from all datasets with a certain probability which is chosen based on dataset size, diversity, and quality. In all datasets, we use provided masks if they exist, and mask out additional pixels which have very low albedo values as these typically correspond to mirror surfaces, glass, or the sky box.

Ordinal Training. We train the ordinal network starting from the weights provided by [Careaga and Aksoy 2023]. We use the same loss formulation and augmentation from the original paper. We continue training using all of the datasets shown in Table 1 for approximately 1.25 million iterations.

IID Training. We do not continue training the second network from [Careaga and Aksoy 2023]. Since the inputs are still ordinal shading estimations the network does not need to be retrained in order to work with our updated ordinal shading network.

Chroma Training. We train the shading chromaticity network for a total of 500 thousand iterations. We use the updated ordinal network weights when generating the gray-scale input decompositions to ensure that the model generates reasonable decompositions for

Fig. 3. Another example of the intermediate components in the same order as Figure 2 Image from Unsplash by Susmitha Veganosaurus.

all datasets. When training this network we add a global color shift augmentation to simulate inaccurate white balance. The color shift is computed by generating a random RGB vector, normalizing and multiplying it by the input image. This doesn't change the albedo and the shifted shading is computed by dividing the shifted image by the albedo. In the albedo and diffuse shading stages, we cannot easily synthesize the input decompositions when performing this color shift augmentation, therefore we only include this augmentation in the chromaticity estimation stage.

Albedo Training. The albedo estimation network is trained for approximately 600 thousand iterations. We use estimations from the chromaticity network, combined with estimations from the grayscale shading networks as input to this stage, and precompute these decompositions for each dataset.

Diffuse Shading Training. We train the diffuse shading network for approximately 750 thousand iterations. Since the proper diffuse shading component is not typically provided in rendered datasets, we only use Hypersim to train this network.

2.3 Evaluation

For the ARAP dataset, we follow Careaga and Aksoy [2023] using 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. We found that many of the scenes used by Careaga and Aksoy [2023] had duplicate illuminations or were over-represented in the dataset (5+ illuminations of the same scene), so we remove these redundant data points. To add more data, we use some scenes from the MIST Dataset [Hao and Funt 2020] as it is a small-scale dataset and it is zero-shot for all the methods (not used for training) To compute the shading for each render we divide the image by the albedo. This results in a 3-channel colorful shading component. 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. For a given image we run each method using their published code and default resizing functionality. We evaluate the output decomposition at the R_0 resolution following Careaga and Aksoy [2023]. We use a window size of 20 when computing the LMSE metric. For the MAW dataset, we use the public code provided by the authors. For each method, we use the default resizing logic provided and output the albedo to disk. The MAW dataset code handles resizing and computing metrics, we follow the authors scaling when reporting metrics to match the original paper. Since the images are evaluated at a low resolution, we run our method at a resolution of 512 pixels, this does not significantly affect the scores of our method. We use all datasets when training the method of Careaga and Aksoy [2023], the chromaticity estimation network, and the albedo estimation network. We use only the Hypersim dataset when training the diffuse shading network.

3 QUALITATIVE EXAMPLES

In Figure 1 we show an extension of Figure 6 in the main paper. In this example, we can see that the gray-scale methods of Careaga and Aksoy [2023]; Das et al. [2022]; Luo et al. [2020] and Liu et al. [2020] all leave residual lighting effects in the albedo on the ceiling and

Fig. 4. Our pipeline is able to iteratively correct small errors in at each stage. In this example, our pipeline first corrects the colors on the folds of the clothing when estimating shading chromaticity. Then the albedo estimation network is able to remove small errors from the original decomposition by estimating a smooth albedo layer. Images from Unsplash by Austin Wade.

on the floor. In this example, it is obvious that our single-network baseline significantly shifts the colors of the scene which can be seen on the floor and on the blue wall. On the other hand, our method is able to estimate sparse and accurate albedo due to our careful modeling of the lighting chromaticity in the scene.

In Figures 2 and 3 we show each component of our pipeline for 3 different scenes. The top row of each example shows the input image, and three different albedo components (from left to right): the albedo from Careaga and Aksoy [2023] (S_g), our low-res chroma albedo ($\hat{S}_c$) and our final diffuse albedo (S_d). The final image in the top row is the specular residual. The bottom row shows our white-balanced version of the image, the corresponding shading layers from each albedo layer, and finally the diffuse shading layer.

In the first example, we can see how our method is able to correctly represent the inter-reflections from wood beams reflecting onto the ceiling. Additionally, many of the surfaces in the image are slightly specular, our method is able to get the subtle color shift off of the wooden desk and correctly assigns it to the specular residual layer. It is important to note that these colorful lighting effects are incorrectly represented when assigned to the multiplicative shading layer, resulting in odd color shifting (e.g. the slight blue tint on wood surfaces). These color shifts are removed from our diffuse shading as they are correctly placed into the additive specular layer where they belong.

In the second example, we can see that there is a deep blue color behind the furniture from the wall and floor. Our model is able to capture this color shift in the shading layer making the albedo much more uniform. Our method also captures the color specularity on the floor and on the door, first assigning it to the shading layer, then properly representing it in the additive residual layer. The residual layer is also sparse without additional lighting effects.

In the final example, we can see that the plant albedo looks very non-uniform in the decomposition from Careaga and Aksoy [2023]. This is caused by two different lighting effects; the slight specularity from the sun shining on the top of the leaves, as well as the inter-reflections between the leaves creating a deeper green in parts of the plant. We can see that our chroma network correctly estimates the color shift of both effects, assigning them to our initial shading layer. Our albedo network then further smooths the reflectance across the plant and we can see both purple and green lighting effects in the shading layer. Finally, our diffuse shading network correctly assigns the specular light on the top of the leaves to the residual and leaves the diffuse inter-reflections below the leaves.

Although our pipeline relies on an accurate decomposition from the method of Careaga and Aksoy [2023], our multi-stage pipeline is still able to correct small errors in the original decomposition. An example of this behavior is shown in Figure 4. Our model first adjusts the shading chromaticity to remove color from inter-reflections on the clothing. Our albedo estimation network is then able to recognize residual shading effects left in the albedo, correcting them by estimating a sparse albedo layer.

REFERENCES

Nicolas Bonneel, Balazs Kovacs, Sylvain Paris, and Kavita Bala. 2017. Intrinsic Decompositions for Image Editing. *Comput. Graph. Forum* 36, 2 (2017).
Chris Careaga and Yağız Aksoy. 2023. Intrinsic Image Decomposition via Ordinal Shading. *ACM Trans. Graph.* (2023).
Chris Careaga, S. Mahdi H. Miangoleh, and Yağız Aksoy. 2023. Intrinsic Harmonization for Illumination-Aware Compositing. In *Proc. SIGGRAPH Asia.*
Partha Das, Sezer Karaoglu, and Theo Gevers. 2022. PIE-Net: Photometric Invariant Edge Guided Network for Intrinsic Image Decomposition. In *Proc. CVPR.*
Xiangpeng Hao and Brian Funt. 2020. A multi-illuminant synthetic image test set. *Color Research & Application* (2020).
Philipp Krahenbuhl. 2018. Free Supervision from Video Games. In *Proc. CVPR.*
Hoang-An Le, Partha Das, Thomas Mensink, Sezer Karaoglu, and Theo Gevers. 2021. EDEN: Multimodal Synthetic Dataset of Enclosed Garden Scenes. In *Proc. WACV.*
Yixuan Li, Lihan Jiang, Linning Xu, Yuanbo Xiangli, Zhenzhi Wang, Dahua Lin, and Bo Dai. 2023. Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In *Proc. ICCV.*
Yunfei Liu, Yu Li, Shaodi You, and Feng Lu. 2020. Unsupervised Learning for Intrinsic Image Decomposition from a Single Image. In *Proc. CVPR.*
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 Trans. Vis. Comp. Graph.* (2020).
Lukas Murmann, Michael Gharbi, Miika Aittala, and Fredo Durand. 2019. A Multi-Illumination Dataset of Indoor Object Appearance. In *Proc. ICCV.*
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. In *Proc. ICCV.*

Yujie Wang, Qingnan Fan, Kun Li, Dongdong Chen, Jingyu Yang, Jianzhi Lu, Dani Lischinski, and Baoquan Chen. 2022. High quality rendered dataset and non-local graph convolutional network for intrinsic image decomposition. *Journal of Image and Graphics* (2022).

Yu-Ying Yeh, Koki Nagano, Sameh Khamis, Jan Kautz, Ming-Yu Liu, and Ting-Chun Wang. 2022. Learning to Relight Portrait Images via a Virtual Light Stage and Synthetic-to-Real Adaptation. *ACM Trans. Graph.* (2022).

Jia Zheng, Junfei Zhang, Jing Li, Rui Tang, Shenghua Gao, and Zihan Zhou. 2020. Structured3D: A Large Photo-realistic Dataset for Structured 3D Modeling. In *Proc. ECCV*.

Jingsen Zhu, Fujun Luan, Yuchi Huo, Zihao Lin, Zhihua Zhong, Dianbing Xi, Rui Wang, Hujun Bao, Jiaxiang Zheng, and Rui Tang. 2022. Learning-Based Inverse Rendering of Complex Indoor Scenes with Differentiable Monte Carlo Raytracing. In *Proc. SIGGRAPH Asia*.