Point Cloud Color Constancy

Xiaoyan Xing[1,*], Yanlin Qian[2,*], Sibo Feng[2,], Yuhan Dong[1,†], and Jiri Matas[3]
[1]Tsinghua University, [2]Independent Researcher, [3]Czech Technical University

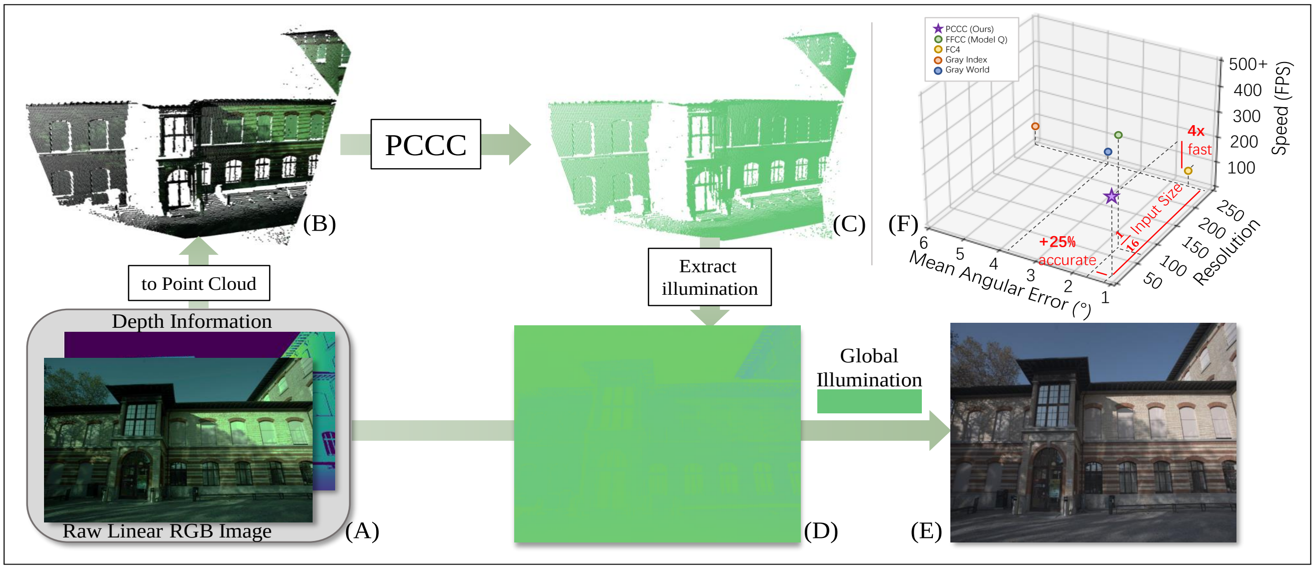


Figure 1. We present *Point Cloud Color Constancy* (PCCC), which is simple, efficient and hardware friendly. (A): Raw RGB-D image. (B): RGB point cloud generated by (A). (C): Point Cloud with illumination rendered from PCCC. (D) 2D illuminant map extracted from (C). (E) Color corrected image. (F): Speed, accuracy and parameter number comparison with state-of-the-art color constancy methods.

Abstract

In this paper, we present Point Cloud Color Constancy, in short PCCC, an illumination chromaticity estimation algorithm exploiting a point cloud. We leverage the depth information captured by the time-of-flight (ToF) sensor mounted rigidly with the RGB sensor, and form a 6D cloud where each point contains the coordinates and RGB intensities, noted as (x,y,z,r,g,b). PCCC applies the PointNet architecture to the color constancy problem, deriving the illumination vector point-wise and then making a global decision about the global illumination chromaticity. On two popular RGB-D datasets, which we extend with illumination information, as well as on a novel benchmark, PCCC obtains lower error than the state-of-the-art algorithms. Our method is simple and fast, requiring merely 16×16*-size input and reaching speed over* 140 *fps (CPU time), including the cost of building the point cloud and net inference.*

1. Introduction

There are over a billion smartphone cameras capturing photographs, where the procedure of perceiving true colors of the real world happens in the ISP pipeline. In general, a good-looking picture relies on a single CMOS sensor with a certain color filter array. On flagship smartphones, there is a trend of leveraging two or more digital sensors to "compute" a higher-quality image. Such cases include {main camera, tele camera} [2], {main camera, near inferred sensor} [48], and {color camera and time-of-flight (ToF) depth sensor} [45]. In this work, we proceed in this direction and use a color camera and a ToF sensor. A ToF depth sensor has been commonly employed for other applications; *e.g.*, auto focus, bokeh effect and 3D face and skeleton scanning. Here we show a novel task based on combing the RGB and ToF sensors – RGB-D or point-cloud illumination estimation.

Illumination estimation refers to the task of estimating the normalized illumination chromaticity given a raw im-

[*] Equal contribution/[†]Corresponding author. Code & Data: https://github.com/xyxingx/Point-Cloud-Color-Constancy

age, which itself is a core task in computational color constancy. Color constancy is an intrinsic property of the human eye, which perceives the world independently of illumination color changes. For technical reasons and engineering benefits, in digital cameras, illumination estimation and a channel-wise rescaling are combined in the so-called auto-white balance (AWB), to realize "color constancy" computationally. An accurate estimation, with proper rescaling factors than render a neutral surface white.

In common the use-case, illumination estimation only relies on a single raw format image. Here we advocate the use of depth sensor which casts the task into a 3D regression problem, which we show delivers more accurate results. We introduce depth based color constancy for the following reasons: 1. image statistics are steered by depth information [30]. More specially, the surface geometry, the texture size and the signal-to-noise ratio varies with depth, which will influence the performance of illumination estimation algorithm; 2. abrupt depth changes correlate with non-smooth illumination distribution. Imagine walking from outside to an indoor room with a tungsten light. In a 2D image, the depth and illumination change dramatically in the area close to the door. As shown in Figure 1, the captured RGB-D images are reprojected to form the colored point cloud, on which the proposed method estimates the illumination for each point. Then these point predictions (Figure 1(C)) are averaged with learnt weights.

In a summary, the contributions of the paper are:

- We formulate the generic problem of illumination estimation in a 3D world, relaxing the over-simplistic assumptions about uniform illumination for a plain 2D image adopted in prior work .
- We present PCCC architecture, derived from PointNet, a novel point cloud net for the color constancy task. We show its superior performance on illumination estimation on three color constancy benchmarks. Side applications, *e.g*., local AWB is reported as well.
- PCCC operates well on the sparse-sampled thumbnail point clouds, which allows over 500 frames per second (include time of building point cloud) on a Nvidia V100 GPU and is hardware friendly for a mobile SoC.
- We annotated the illumination groundtruth labels for three popular RGB-D datasets (NYU-v2 dataset [40], Diode dataset [42] and ETH3D dataset [37]) and collected DepthAWB, a novel RGB-D dataset for the color constancy task.

2. Related Work

The prior art in computational color constancy falls into the learning-based and statistics-based groups. In this section, we consider the candidate method from a different perspective, i.e., the input requirement.

	3D CC	Depth Seg.	CNN based	UV Hist.	Ours
	[30]	[19]	[11, 25, 29, 43]	[8, 9]	PCCC
Semantic info.	×	×	✓	Not explicit	✓
Depth info.	✓	✓	×	×	✓
End to End	×	×	✓	×	✓
Local AWB	×	✓	Some of them	×	✓

Table 1. PCCC - comparision with prior color constancy methods.

Methods relying on a single Image The majority of color constancy methods belong to this category. Firstly, there exist a list of traditional methods, including Gray World [13], White Patch [12], General Gray World [7], Gray Edge [41], Shades-of-Gray [21], LSRS [22], PCA [16] and Grayness Index [35], *etc*. They are easy to be implemented in a ISP chipset and may fail if their assumption is violated. Secondly, more methods are driven by the availability of color constancy datasets which provide carefully annotated illumination groundtruth; [5, 8, 14, 15, 20, 23–26, 39] learn a high-dimensional mapping from the capture raw data to the sought illumination vector after optimization on training set. The learning tools behind includes gamut mapping, least square minimization, random forest, neural networks and so on.

Methods relying on Image(s) and Auxiliary Information There is a research trend of including more information else than a single image in design of color constancy methods, for special purposes. Abdelhamed *et al*. [2] leveraged main camera and tele camera simultaneously in a very lightweight neural network, releasing the discriminative ability of color filter array correlation. Barron *et al*. [9] proposed a Fourier transform-based model (model O in the paper), learning the model weight from the camera metadata like aperture and so on. Qian *et al*. [34, 36] claimed the preceding image sequence benefits the illumination estimation for the shot image. Yoo *et al*. [46] explored the AC light source in a high-speed setting. Ebner *et al*. [19] firstly estimated the illumination chromaticity by combing Gray World with depth map.

Differing from [19], in a end-to-end way, our approach learns how to use depth for referring illumination from a sparsely-sampled point cloud. As shown in Table 1, it also supports semantic info extraction and 3D point-wise AWB.

3. Point Cloud Color Constancy

We describe PCCC in the following order: 1) reformulating the illuminant estimation problem on RGB-D images (Section 3.1); 2) aligning the RGB images and depth maps spatially (Section 3.2); 3) generating point cloud with tristimulus color values (Section 3.3); 4) explaining our point-cloud illuminant estimation network (Section 3.4); and 5) proposing two augmentation tricks based on vibrating cam-

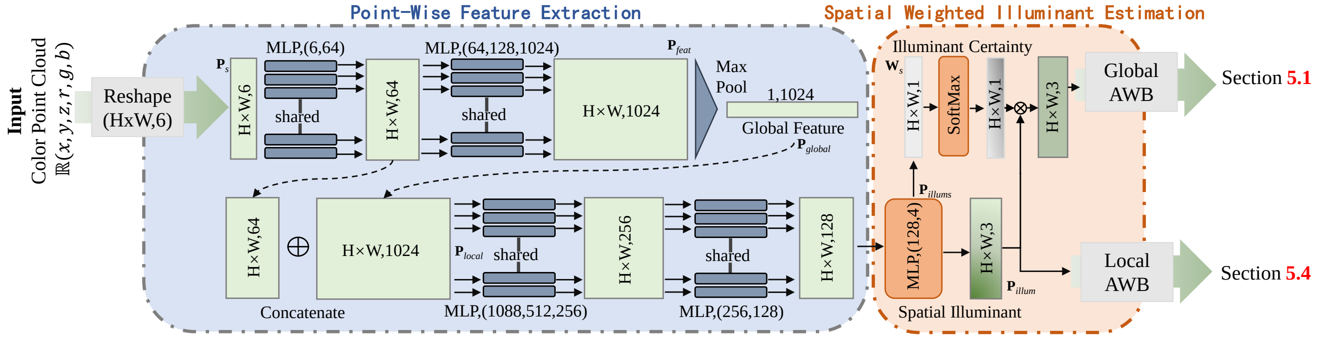


Figure 2. The PCCC architecture consists of a point-wise feature extraction block - a slight modification of PointNet [31], and the spatial weighted illuminant estimation block. Given $H \times W$ points as input, PCCC outputs a spatial weighted illuminant, which benefits global auto white balance, AWB (Section 5.1). With a slight setting change, we can also achieve point-wise illumination estimation (Section 5.4).

era position and illumination chromaticity (Section 3.5).

3.1. Formulation

Given an RGB image $\mathbf{I}$, the image formation can be described as follow:

$$\mathbf{I} = \int_{\lambda} E(\lambda)S(\lambda)R_{RGB}(\lambda)d\lambda, \tag{1}$$

where λ is the visible wavelength. E refers to illuminant spectral power distribution, S is the surface reflectance function, and R_{RGB} refers to the sensor responses. Our aim in this paper is to get illuminant $\mathbf{E} = \int_{\lambda} E(\lambda)R_{RGB}(\lambda)d\lambda$ for the captured environment, or even more ideally, for each point position.

Based on prior works [19, 30] and our observation, with point clouds (or images with depth information) as inputs, the color constancy method benefits from the following: 1) explicitly discriminate the ambiguity illuminant through depth discontinuity; 2) identify the dominant illuminant under different depth; 3) simulate different viewing angle of the same scenes (Section 3.5 and 5.3) ; and 4) estimate global illuminant based on spatial illuminant distribution (Section 3.4, 5.3 and 5.4).

3.2. Alignment between RGB and Depth Map

RGB images and depth maps are captured by different sensors separately, thus alignment is needed. The process of alignment follows as: 1) calibration two cameras' intrinsics and extrisics using Zhang *et al.* [49]; 2) calculate the rigid transformation from coordination of RGB camera to the coordination of depth camera, and 3) colorize the depth map using the corresponding RGB intensities. Since the rigid connection of two sensors holds when mounted, the transformation matrix is pre-computed before dataset collection.

3.3. Point Cloud with Tristimulus Values

Given an RGB image $\mathbf{I} = [\mathbf{I}_r, \mathbf{I}_g, \mathbf{I}_b] \in \mathbb{R}^{U \times V \times 3}$ and its corresponding depth map $\mathrm{D} \in \mathbb{R}^{U \times V \times 1}$, we aim to build a colored point cloud denoted as $\mathbf{P} = \{\mathbf{p}_i | i \in 1,,,n\}$. Each point $\mathbf{p}_i$, as a vector of (x,y,z,r,g,b) values, is computed as:

$$\mathbf{p}_i = (\frac{(u - c_x)d}{f_x}, \frac{(v - c_y)d}{f_y}, d, r, g, b), \tag{2}$$

where u,v are the 2D coordinates in $\mathbf{I}$ for the i_{th} pixel, d is the depth value, $\{f_x, f_y\}$ represent the focal lengths, (c_x, c_y) the principal point. $\{r, g, b\}$ refer to the color channels in $\mathbf{I}$.

Comparing to 2D images, the colored point clouds carry more rich information of inputs such as spatial distance between objects, spatial color transformation, and explicit geometric structure. Unless explicitly stated otherwise, "point cloud" in this paper denotes colored point cloud.

3.4. Illumination Regressor on Point Cloud

Given a dataset of N point clouds, $\mathcal{P} = \{\mathbf{P}_1, \mathbf{P}_2, ..., \mathbf{P}_M\}$ and the corresponding global illumination label $\mathcal{E} = \{\mathbf{E}_1, \mathbf{E}_2, ..., \mathbf{E}_M\}$, we then can train a neural network Given a dataset of N point clouds, $f_\theta : \mathcal{P} \to \mathcal{E}$

$$\hat{\mathbf{E}_i} = f_\theta(\mathbf{P}_i), \tag{3}$$

where $\hat{\mathbf{E}}$ is the predicted illumination vector.

To estimate the dominant illuminant $\mathbf{E}$, we 1) extract the lighting feature out of the point set; 2) obtain the approximately illuminant distribution, and 3) estimate the dominant illuminant from the spacial distribution.

Feature extraction Point clouds are spatial sparse and disorderly. Traditional CNN architectures are designed for 2D images (which are in regular format) and may not fit the task. We employ PointNet [31] as our feature extraction module, which is designed for the point cloud-based classification task. PointNet subtly overcomes the unordered issue with an architecture that consists of several multi-layer perceptrons (MLP) and max pooling as its basic layers.

To feed MLP, $\mathbf{P}$ is converted to an unordered sequential of N points, $\mathbf{P}_s \in \mathbb{R}^{N\times 6}$, where 6 channels represent the camera spatial coordinate $\{X, Y, Z\}$ and corresponding color values $\{R, G, B\}$ for each point. Firstly, MLP transforms the $\mathbf{P}_s$ of 6 channels into $\mathbf{P}_{feat}$ of 1024 channels, while N remains the same. Then, max pooling aggregates $\mathbf{P}_{feat}$ with N points into one single global feature $\mathbf{P}_{global}$.

Spatial weighted illuminant estimation Technically, the MLP can decrease the channel of global feature $\mathbf{P}_{global}$ from 1024 to 3, yielding the final prediction, which is simple but loses local estimates. Inspired by [25], we devise a spatially weighted estimation module, by introducing the point-cloud weight matrix $\mathbf{W}_s$ in stage of network inferring.

To include more local information, feature extraction is boosted by aggregating the global feature $\mathbf{P}_{global}$ and the intermediate output from first MLP group to be $\mathbf{P}_{local} \in \mathbb{R}^{N\times 1088}$. Then $\mathbf{P}_{local}$ is reduced to $\mathbf{P}_{illums} = [\mathbf{P}_{illum}, \mathbf{W}_s] \in \mathbb{R}^{N\times 4}$ by MLP, where $\mathbf{P}_{illum} \in \mathbb{R}^{N\times 3}$ represents the illumination estimation of each point, and spatial illuminant matrix $\mathbf{W}_s \in \mathbb{R}^{N\times 1}$. Mathematically global illuminant $\hat{\mathbf{E}}_{global} \in \mathbb{R}^{1\times 3}$ can be achieved by :

$$\hat{\mathbf{E}}_{global} = \sum_{R,G,B} (\text{softmax}(\mathbf{W}_s) \cdot \mathbf{P}_{illum}), \tag{4}$$

where softmax normalizes $\mathbf{W}_s$ into $\{0,1\}$ as a probability mask. The global illuminant is employed for global color constancy. The probability mask and local illuminant can later be leveraged for the trial in local AWB.

Loss Function Similar to the prior arts, we use the recovery angular loss $\mathcal{L}_{ang}$ as our loss function:

$$\mathcal{L}_{ang}(\hat{\mathbf{E}}, \mathbf{E}) = \arccos(\frac{\hat{\mathbf{E}} \cdot \mathbf{E}}{||\hat{\mathbf{E}}|| \cdot ||\mathbf{E}||}), \tag{5}$$

where $\hat{\mathbf{E}}$ is the predicted illumination vector and $\mathbf{E}$ is the ground truth illumination label. With the loss, our net converges in an end-to-end way.

Discussion Theoretically, any network that design for point clouds [28, 33] or for RGB-D input can be employed to replace the PointNet-based net here. However, due to the nature of the color constancy problem, we argue that an ideal point cloud color constancy network should 1) have a simple but effective structure that keeps good trade-off between accuracy and running speed; 2) be not sensitive to the noise and sparse sampling rate and 3) in default working with thumbnail-size input, which is usually provide by specialized ISP hardware node. Quantitative experiments to the mentioned properties of our design are in Section 5.3.

3.5. Point Cloud Augmentation

Point cloud allows us to augment data from a 3D prospective. We propose two augmentation methods based on our observation that illumination on a point is roughly invariant to mild *camera pose* jittering and *light intensity* changes. These methods rely on separating a point cloud into two parts 1) coordination value $\mathbf{P}_{xyz}$ and 2) color value $\mathbf{P}_{RGB}$, considering the order of point cloud is fixed during our augmentation period. We can augment the camera position and light intensity separately as following.

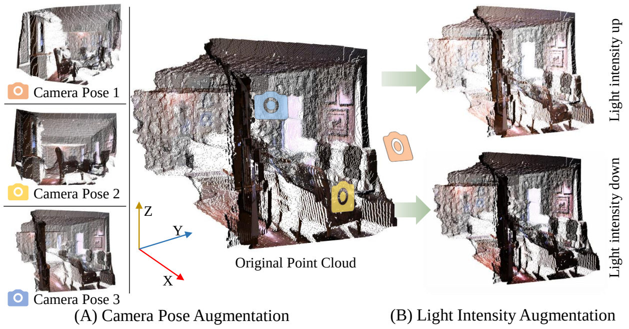


Figure 3. We use two augmentation approaches: (A) camera pose augmentation simulates images capture from a different viewing angle. (B) light intensity augmentation generates new pixel-wise illumination field under the same illuminant.

Camera Pose Augmentation is inspired by that small rotation does not affect the decision of the dominant illumination color. As shown in Figure 3, we sample a rotation matrix $\mathbf{R} \in SO(3)$ to simulate the scenarios where we take images from different perspectives.

Light Intensity Augmentation is inspired by luminance can be changed when the relative distance of the illuminated objects and the light source changes. Based on this, we use an intensity scaling factor L to realize luminance-based augmentation.

Considering both augmentation ways, the new point cloud $\mathbf{P}^{'}$ is:

$$\mathbf{P}^{'} = \mathbf{R}\mathbf{P}_{xyz} \oplus L\mathbf{P}_{RGB}, \tag{6}$$

where $\mathbf{R}$ is the sampled rotation matrix, L is a scalar from a normal distribution $\mathcal{N}(1, 0.15^2)$, and $\oplus$ is channel-wise concatenation.

4. Dataset Preparation

To train PCCC, the pairs of the point sets (from RGB-D images) and the corresponding illumination labels are needed. However, to our knowledge, there are no such datasets publicly available. To evaluate our method, we first present in Section 4.1 reannotating the commonly used depth datasets NYU-v2 [40] and Diode [42]. Then in Section 4.2 we relabel the ETH3D dataset [37], which includes sensor raw files. Finally and the most importantly, we show the collection of our point cloud illumination dataset using a single lens digital camera and an accompanied Intel laser depth sensor.

4.1. Annotating NYU-v2 & Diode Depth Dataset

NYU-v2 [40] is a classical RGB-D dataset, with hundreds of indoor scenes captured by Microsoft Kinect. Diode [42] is a latest RGB-D dataset released by Toyota Technological Institute at Chicago, this dataset contains more than 20 scenes include indoor and outdoor scenes. Affected by the illuminant and position, there are numbers of AWB-biased images in these datasets (See supplement material). We found these images are suitable for us to label the correct illumination according to the neutral surfaces and validate the color constancy algorithm.

Images from these two datasets are in sRGB. To label the ground truth illuminant, we process images by the following steps: 1) switch sRGB to LRGB using inverse gamma operation [6, 18]; 2) remove the tuning operation using a 3×3 matrix as suggested in [3]; and 3) label out the neutral surface area (which is deemed as neutral without doubt, like white printed paper) and calculate the illumination vector (Details in supplement materials). The steps can be formulated as:

$$\mathbf{E}_{label} = \Gamma(\mathbf{M}^{-1} \times \mathbf{I}_{sRGB}^{1/\gamma}), \tag{7}$$

where $\Gamma(\cdot)$ represents the area selection and illuminant calculation. $\mathbf{E}_{label}$ is the labeled ground truth illumination, $\mathbf{M}$ is the tuning matrix of images.

After the preprocessing steps, we obtain linearization images with labels from two datasets. These images contain indoor and outdoor scenes, with diverse label distribution.

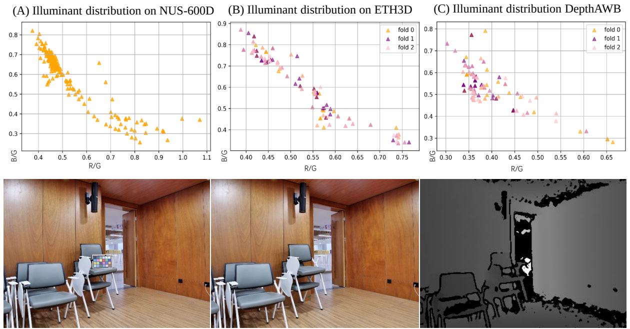
Discussion We acknowledge that relabeled illumination on sRGB images may not be as accurate as that in raw LRGB images, and our transformation can not really transform the sRGB to LRGB, which is a rough operation and may bring nosie to the images. Since all color constancy methods will surfer the same bias on the datasets, we argue that these data are of experimental value for us to test our method (See Section 5 for more experimental details).

4.2. Annotating ETH3D Depth Dataset

ETH3D dataset [37] is a stereo RGB-D dataset with 13 different scenes with indoor and outdoor, the RGB images are captured by a Nikon D3 camera, and the depth maps are from a radar. The dataset provides the raw format of images. We can process the images using LibRaw, and obtain the raw linear RGB images, which makes them closer to the images we used in camera auto white balance pipeline.

We use the similar label method as Section 4.1 to obtain the illuminant. Considering we already have raw DSLR images $\mathbf{I}_{Raw}$, the labeled illuminant $\mathbf{E}_{label}$ can be obtained by:

$$\mathbf{E}_{label} = \Gamma(\mathbf{I}_{Raw}). \tag{8}$$



(D) Image w/ Color Checker (E) Image w/o Color Checker (F) Depth Map

Figure 4. (A)-(C): Illumination distribution of NUS-600D dataset, our relabel ETH3D dataset and propose dataset. (D), (E) a pair of images we captured sequentially. We use (D) with color checker for labelling and (E) for training and testing. (F) ToF depth map.

4.3. DepthAWB – Real Color Constancy Dataset with Depth

Despite we have the annotated dataset on raw images, still the ground truth illuminants have biases, since we can not really know the accurate color. Therefore, an RGB-D dataset with raw images and color checker labeled illuminants is needed. Following prior works on RGB-D datasets collection [40] and color constancy dataset collection [23], we collected our own depthAWB dataset, which to our knowledge is the first RGB-D dataset with labeled illuminant. Our dataset preparation steps are as following:

Data collection: Since RGB-D camera like Kinect-V2 does not provide the raw images, therefore, following [37], we use LUMIX GH5S camera [1] to capture the DSLR RGB images and Intel Resense L515 to capture the depth maps. Our data capture setup is presented in supplement material.

Data calibration: Calibration is a fundamental operation, as introduced in Section 3.2, we first capture several images from RGB camera and depth camera, by tilting the a checker board with checker size 30 millimeter vertically and horizontally; then we calculate the transformation matrix using the Caltech camera calibration toolbox [1].

Illumint annotation: Since we obtain the raw images in data collection, we demosaic the raw images similar as Section 4.2, and we directly annotated the ground truth on the xrite color checkerboard.

4.4. Illumination distribution

In Figure 4 (A)-(C), we compare the illumination distribution of our labeled ETH3D dataset, our collected Depth AWB dataset to the commonly used NUS dataset [16] Cannon 600D subset (NUS-600D). Our datasets cover a wide range of illuminant values, and share the same trend in illuminant distribution as NUS-600D.

[1] We fixed focal length and set camera aperture as $f/8.0$, while extend the shutter speed, details discussed in supplement.

Method	Angular Error (NYUv2&Diode)					Angular Error (ETH3d RAW)					Angular Error (DepthAWB)					Training	Test
	Mean	Med.	Tri.	B25%	W25%	Mean	Med.	Tri.	B25%	W25%	Mean	Med.	Tri.	B25%	W25%	time (s)[1]	time (ms)
GrayWorld [13]	5.39	3.83	4.37	1.23	11.81	3.18	2.87	2.95	1.00	5.70	6.03	5.23	5.40	3.48	9.75	-	10.00
WhitePatch [12]	6.37	5.53	5.59	3.77	10.46	6.28	6.41	6.49	1.20	13.09	11.77	12.51	12.23	3.74	18.80	-	7.83
1^{st}GrayEdge [41]	4.02	2.71	2.96	0.82	9.43	7.81	5.44	6.49	1.06	17.09	6.56	6.13	6.06	1.52	12.83	-	34.00
2^{nd}GrayEdge [41]	4.15	2.90	3.10	0.78	9.81	8.02	5.30	6.38	1.21	18.20	7.12	6.91	6.97	1.87	13.07	-	42.50
Shade of Gray [21]	3.69	2.68	2.90	0.61	8.59	7.25	5.02	6.23	0.95	15.59	5.84	4.59	5.11	1.22	11.82	-	320.00
APAP (GW) [4]	3.47	2.31	2.61	0.67	7.78	2.71	2.53	2.54	0.82	4.69	3.69	2.69	3.08	1.70	6.50	16	16.00
Shen et al. [38]	5.55	3.61	3.88	1.03	13.39	3.90	3.12	3.31	1.08	7.54	5.05	3.19	3.80	0.77	11.85	-	37.00
GI [35]	4.28	3.01	3.27	0.85	9.81	3.26	2.02	2.19	0.49	7.35	3.91	2.09	2.58	0.49	10.23	-	109.00
Bayesian [23]	9.64	10.01	9.59	3.61	15.68	7.32	5.31	6.02	1.70	15.42	5.97	5.09	5.24	1.91	11.20	35	2134.42
Cheng et al. [17]	3.11	2.02	2.27	0.59	7.39	3.44	1.71	2.06	0.44	9.43	1.89	1.15	1.24	0.22	4.71	63	331.20
Quasi U CC [10]	3.76	2.20	2.54	0.45	9.28	3.90	3.29	3.43	0.85	7.58	4.45	3.80	3.90	1.04	9.03	-	345.13
FC4 [25]	2.49	1.61	1.79	0.47	6.03	1.19	0.90	0.98	0.32	2.46	1.34	0.98	1.04	0.24	3.08	13000	38.21
FFCC (Model J) [9]	2.82	1.87	2.00	0.51	6.82	1.16	0.91	0.99	0.31	2.43	1.32	0.76	0.88	0.27	3.21	67	29.00
FFCC (Model Q) [9]	2.75	1.90	2.01	0.48	6.63	1.08	0.88	0.91	0.26	2.26	1.40	0.81	0.93	0.29	3.40	51	1.10
Ours (16 Points, w/o depth)	2.96	1.82	2.12	0.63	6.91	1.58	0.98	1.09	0.22	3.82	2.60	2.11	2.20	0.50	5.72	1700	4.00
Ours (16 Points)	2.23	1.67	1.74	0.40	5.16	1.12	0.75	0.78	0.14	2.79	1.64	0.95	1.05	0.17	4.30	1700	4.00
Ours (256 Points, w/o depth)	2.80	1.89	2.22	0.57	6.39	1.42	1.02	1.08	0.33	3.11	2.39	1.75	1.90	0.42	5.50	3120	7.03
Ours (256 Points)	2.20	1.42	1.58	0.34	5.30	0.88	0.53	0.62	0.17	2.15	1.05	0.37	0.55	0.09	3.02	3120	7.03
Ours (4096 Points)	2.18	1.30	1.54	0.33	5.39	0.78	0.42	0.51	0.11	1.97	0.99	0.28	0.50	0.08	2.94	5700	47.12

[1] Since we have multi-scale datasets, the train time is for the DepthAWB dataset.

[2] The profiled time for PCCC includes the time slot for aligning RGB image and depth image, building point cloud and also network inferring.

Table 2. Quantivate comparison of our reannotated NYU-v2 [40], Diode [42], ETH3D [37] datasets and our collected dataset, respectively. We highlight top two performers in each metric with pink and yellow background, respectively. All test time are CPU time.

BackBone	Image	Train	FPS	NYUv2&Diode				ETH3d RAW				DepthAWB			
	Scale	time		Mean	Med.	Tri.	W25%	Mean	Med.	Tri.	W25%	Mean	Med.	Tri.	W25%
PointCNN [28]	16*16	∼ 2	∼ 20	2.41	1.68	1.83	5.58	1.55	1.33	1.35	3.09	2.59	1.78	1.95	5.92
PointCNN [28]	64*64	> 24	∼ 5	2.19	1.55	1.69	5.02	1.30	1.02	1.11	2.72	1.31	0.91	0.97	3.00
PointNet++ [33]	16*16	∼ 3	∼ 100	2.69	2.87	3.76	3.59	1.44	1.50	1.52	1.93	1.61	1.56	1.57	2.39
PointNet++ [33]	64*64	> 24	∼ 3	2.98	1.98	2.20	7.04	1.66	1.25	1.34	3.39	1.91	2.02	1.99	2.57
Ours	16*16	∼ 0.5	∼ 530	2.35	1.58	1.80	5.37	1.08	0.78	0.87	2.43	1.05	0.37	0.55	3.02
Ours	64*64	∼ 1.5	∼ 495	2.20	1.30	1.57	5.53	0.87	0.52	0.61	2.13	0.99	0.28	0.50	2.94

Table 3. Quantitative results of point cloud based backbones. We test three different backbones on three datasets, with two different input scales respectively. Training time in hours. FPS is from GPU. Color setting as in Table 2.

5. Experiments

We evaluate PCCC and a large list of prior methods on three public RGB-D datasets (relabeled by us, Section 4.1 and 4.2) and a newly-collected RGB-D dataset (Section 4.3). The dataset overview is as follows:

- **NYU-v2 & Diode Dataset**: 324 linearized RGB images and their corresponding depth map from NYU-v2 [40] and Diode datasets [42].
- **ETH3D Dataset**: 195 reprocessed Raw RGB images and their corresponding depth map from ETH-3D datasets [37].
- **DepthAWB Dataset**: 185 DSLR RGB images captured by LUMIX GH5S and depth maps by Intel Realsense L515.

Training: We use Pytorch to build and train our method on NVIDIA V100 (Tesla) GPU, Adam [27] is employed as the optimizer. The learning rate is 0.0003. The epoch is set to 10k.

Metrics: Similar as prior color constancy tasks [43, 44, 47], we report the mean, median (Med.), trimean (Tri.), best (B) and worst (W) 25% of angular error (Eqution (5)). To validate our method's time efficiency, we also present the time cost during training (for the learning-based methods) and testing (for all).

5.1. Quantitative Results

Table 2 presents the quantitative results of PCCC and a large selection of prior arts on three RGB-D color constancy datasets. In overall, the proposed PCCC (with 4096 points) leads three leaderboards, obtaining the lowest error in almost all statistics. Shrinking the input size to 256 points[2], PCCC also performs steadily better than the state-of-the-art learning based methods [9, 25] by a large margin, again showing the advantage of PCCC. To our a bit surprise, the PCCC model with 16-points input still delivers meaningful results, demonstrating the net learns the illumination from the 96 float values w.r.t. image.

From a multiview point of view, 3D point cloud can be rendered to 2D images, as done in [32], which forms a projection-based augmentation boosting the task performance.

It worth mentioning that all PCCC variants provide an over-realtime inference time, nearly 7 mini-seconds (equals

[2]This equals to a 16 × 16 RGB image plus a same-size depth map.

Module			Angular Error (16x16)				Angular Error (64x64)			
CPA	LIA	SW	Mean	Med	Tri.	W25%	Mean	Med	Tri.	W25%
-	-	-	1.32	0.81	0.90	3.35	1.33	0.75	0.90	3.61
-	-	✓	1.23	0.79	0.88	3.14	1.18	0.53	0.70	3.19
-	✓	✓	1.14	0.57	0.72	3.13	1.04	0.43	0.60	2.95
✓	-	✓	1.19	0.50	0.72	3.32	1.10	0.42	0.63	3.16
✓	✓	✓	1.05	0.37	0.55	3.02	0.99	0.28	0.50	2.94

Table 4. PCCC ablation study. On the DepthAWB dataset, for we two resolutions of the point cloud – 16x16 (left) and 64x64 (right). CPA - camera pose augmentation, LIA - light intensity augmentation; both described in Section 3.5. SW - refers to the spatial weighting, see Section 3.4. Color setting as in Table2.

to 140 frames per second) on CPU. Considering the tiny input size, we believe it is a viable solution for smartphone photography.

Figure 6 shows a qualitative comparison of PCCC and our selected well-performing methods. Among all choices, the proposed method obtains the lowest error and delivers the most neural appearance, for both outdoor and indoor scenarios. Please see supplement for more visual results.

Notice All learning-based methods and their hyper-parameters are from their official implementations or carefully tuned. All test time are the CPU time.

5.2. What Depth Brings

In Table 2, we also report the results when PCCC is not given depth (in practice, it is done by initializing the depth as 1 uniformly). We observe that, for 16-points and 256-points models, the accuracy degenerates by a noticeable gap, proving the contribution of the depth. On ETH3d and DepthAWB datasets, the PCCC variant without depth still performs over the majority of learning-based methods, showing that it learns to infer illumination chromaticity well from a "flatten" point cloud. Empirically, we deduce that depth itself contains color information but provides geometric clue for PCCC to tell how the illumination varies from one location to another in a 3D space.

Figure 5 visualizes the illumination certainty map to evaluate the importance of depth information. With depth, PCCC can aggregate feature more from the area which is of more "achromatic" surface and meaningful depth.

5.3. Ablation Study

Point Cloud Processing Backbone As discussed in Section 3.4, there are obviously many selections for point cloud based net that can be employed as the backbone of the proposed method. We replace the PointNet backbone by different point cloud nets (e.g., PonitCNN [28], PointNet++ [33]) and test their performance in our task.

Table 3 proves that given the input of same size, PointNet backbone gives better mean angular error compared to PointCNN and PointNet++ backbones, while costs much less time ($\sim$ 1/80 times). As PointNet backbone achieves a good trade-off between accuracy and efficiency, we adopt PointNet as our default backbone. For fair comparison, only the backbone is switched while other modules are the same.

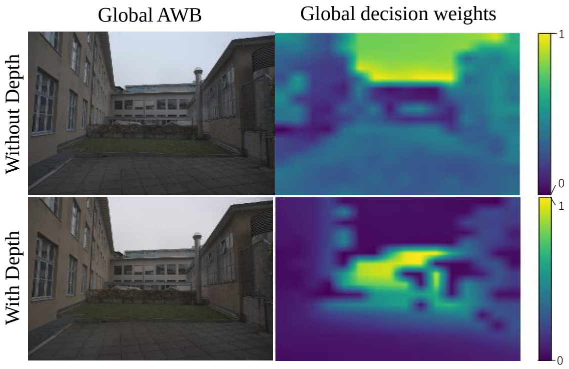


Figure 5. Illustration of how the depth affects the final decision. Depth information helps PCCC to estimate illumination from more meaningful information (building wall) instead of noisy information (bright sky with disturbing color).

Point Set Size and Augmentation We investigate the influence of three proposed modules: camera pose augmentation (CPA), light intensity augmentation (LIA), and spatial weight distribution (SW), using two different size point clouds 16×16 and 64×64 respectively.

Results are shown in Table 4. Compared to using a very sparsely-sampled point cloud (256 points), the 4096-point point cloud obtains better results on each metric. By deactivating each devised module (CPA, LIA, SW) one by one, we notice a gradual performance drop, which validates the benefit of each module.

5.4. Beyond Global Illumination Estimation

Considering that in PCCC illumination-related feature is deduced point-wisely, it is straight-forward to explore its potential application in point-wise AWB (or referred as local AWB).

As introduced in Section 3.4, we can achieve point-wise illumination distribution (later weighted by a weight matrix and fused into the global illuminant). If we take this intermediate feature as our final output, we obtain a $N \times 3$ illumination tensor, where N is the point number. The illumination tensor can be visulized as a colored point cloud (Figure 7, 2^{nd} row), and, if projected to pixel coordinate system, as a 2D illumination map (Figure 7, 3^{rd} row).

With 2D illumination map, getting the reciprocals for each point's vector yields its AWB gain. We apply the AWB gain pixelwisely and visualize in Figure 8. Comparing the image intensities recovered by both solutions, we can find the local one more accurately estimate the dual illumination.

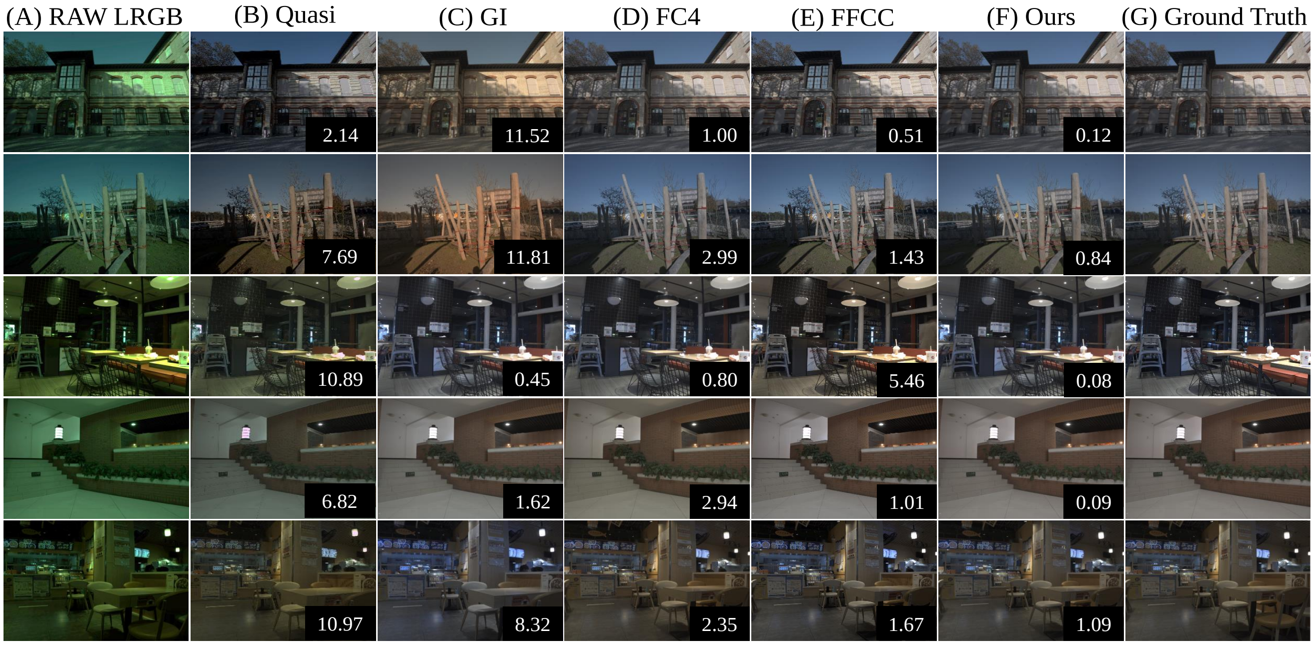


Figure 6. Quantitative global illuminant estimation results on the relabeled ETH3d dataset [37] and the new DepthAWB dataset. (A) and (G) are Raw linear RGB images and images corrected by ground truth illuminants, respectively. From (B) to (F): images with estimated illuminants from Quasi [10], Gray Index [35], FC4 [25], FFCC [9], and PCCC, respectively.

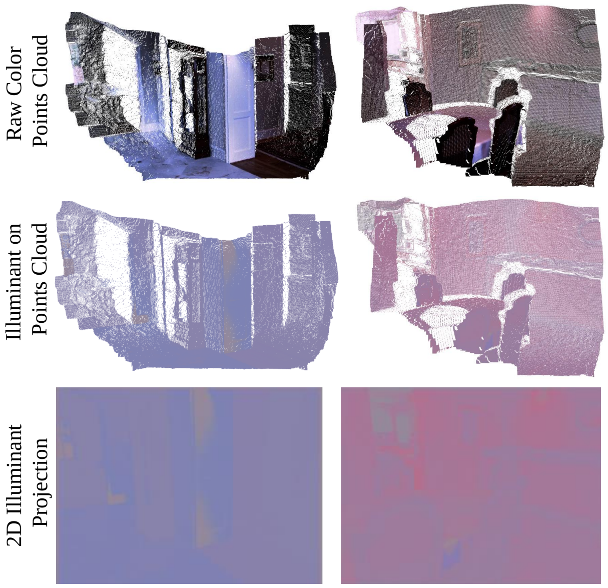


Figure 7. Illustration of illumination distribution over 3D point clouds and the corresponding 2D illumination maps.

6. Conclusion

We develop a point set-based regression net, PCCC, for the color constancy task. By leveraging the geometry information and extensive point-wise augmentation, it deduces the global illumination chromaticity accurately, even in challenging mixed-lighting environments. On two popular RGB-D datasets which we added illumination information to, and on the novel DepthAWB benchmark, PCCC obtains lower error than over the state-of-the-art, reaching the mean angular error equals of 0.99 degree on DepthAWB. Its side applications like local AWB is also discussed. The PCCC method is efficient, using just 16×16-size input, and fast, reaching ~ 140 fps on CPU. In future work, we would like to study more specific point cloud-based operator for color processing task and its interpretability.

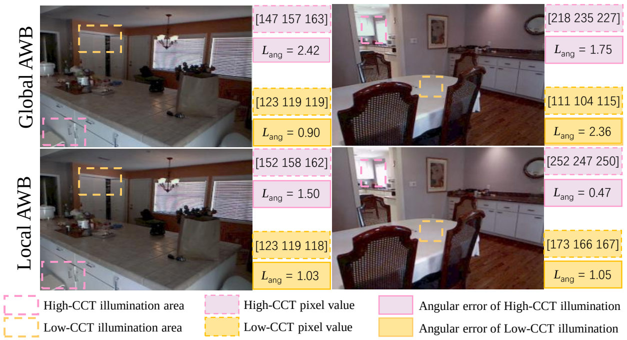


Figure 8. Visualization of Global AWB result and Local AWB result. We use yellow and cyan bonding box to indicate the lower and higher correlated color temperature (CCT), respectively.

Limitations: The quality and quantity of datasets can be further improved, including add Local AWB labels.

References

[1] Camera calibration toolbox. http://www.vision.caltech.edu/bouguetj/calib_doc/, Accessible at 30/10/2021. 5

[2] A. Abdelhamed, A. Punnappurath, and M. S. Brown. Leveraging the availability of two cameras for illuminant estimation. In *CVPR*, pages 6637–6646, 2021. 1, 2

[3] M. Afifi, B. Price, S. Cohen, and M. S. Brown. When color constancy goes wrong: Correcting improperly white-balanced images. In *CVPR*, pages 1535–1544, 2019. 5

[4] M. Afifi, A. Punnappurath, G. Finlayson, and M. S. Brown. As-projective-as-possible bias correction for illumination estimation algorithms. *JOSA A*, 36(1):71–78, 2019. 6

[5] A.Gijsenij, T. Gevers, and J. Van De Weijer. Generalized gamut mapping using image derivative structures for color constancy. *Springer IJCV*, 86(2-3), 2010. 2

[6] M. Anderson, R. Motta, S. Chandrasekar, and M. Stokes. Proposal for a standard default color space for the internet—srgb. In *CIC*, volume 1996, pages 238–245. Society for Imaging Science and Technology, 1996. 5

[7] K. Barnard, V. Cardei, and B. Funt. A comparison of computational color constancy algorithms. i: Methodology and experiments with synthesized data. *TIP*, 11(9), 2002. 2

[8] J. T. Barron. Convolutional color constancy. In *ICCV*, 2015. 2

[9] J. T. Barron and Y.-T. Tsai. Fast fourier color constancy. In *CVPR*, 2017. 2, 6, 8

[10] S. Bianco and C. Cusano. Quasi-unsupervised color constancy. In *CVPR*, pages 12212–12221, 2019. 6, 8

[11] S. Bianco, C. Cusano, and R. Schettini. Color constancy using cnns. In *CVPR workshop*, 2015. 2

[12] D. H. Brainard and B. A. Wandell. Analysis of the retinex theory of color vision. *JOSA A*, 3(10), 1986. 2, 6

[13] G. Buchsbaum. A spatial processor model for object colour perception. *Journal of the Franklin Institute*, 310(1), 1980. 2, 6

[14] A. Chakrabarti. Color constancy by learning to predict chromaticity from luminance. In *NIPS*, 2015. 2

[15] A. Chakrabarti, K. Hirakawa, and T. Zickler. Color constancy with spatio-spectral statistics. *IEEE TPAMI*, 34(8), 2012. 2

[16] D. Cheng, D. K. Prasad, and M. S. Brown. Illuminant estimation for color constancy: Why spatial-domain methods work and the role of the color distribution. *JOSA A*, 31(5):1049–1058, 2014. 2, 5

[17] D. Cheng, B. Price, S. Cohen, and M. S. Brown. Effective learning-based illuminant estimation using simple features. In *CVPR*, 2015. 6

[18] M. Ebner. *Color constancy*, volume 7. John Wiley & Sons, 2007. 5

[19] M. Ebner and J. Hansen. Improving color constancy in the presence of multiple illuminants using depth information. In *International Conference on Bio-inspired Systems and Signal Processing*, volume 2, pages 133–140. SCITEPRESS, 2014. 2, 3

[20] G. D. Finlayson. Corrected-moment illuminant estimation. In *ICCV*, 2013. 2

[21] G. D. Finlayson and E. Trezzi. Shades of gray and colour constancy. In *Color Imaging Conference (CIC)*, 2004. 2, 6

[22] S. Gao, W. Han, K. Yang, C. Li, and Y. Li. Efficient color constancy with local surface reflectance statistics. In *ECCV*, 2014. 2

[23] P. V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. Bayesian color constancy revisited. In *CVPR*, 2008. 2, 5, 6

[24] A. Gijsenij and T. Gevers. Color constancy using natural image statistics and scene semantics. *IEEE TPAMI*, 33(4), 2011. 2

[25] Y. Hu, B. Wang, and S. Lin. FC4: Fully convolutional color constancy with confidence-weighted pooling. In *CVPR*, 2017. 2, 4, 6, 8

[26] H. R. V. Joze and M. S. Drew. Exemplar-based color constancy and multiple illumination. *IEEE TPAMI*, 36(5), 2014. 2

[27] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[28] Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen. Pointcnn: Convolution on x-transformed points. *Advances in neural information processing systems*, 31:820–830, 2018. 4, 6, 7

[29] Y.-C. Lo, C.-C. Chang, H.-C. Chiu, Y.-H. Huang, C.-P. Chen, Y.-L. Chang, and K. Jou. Clcc: Contrastive learning for color constancy. In *CVPR*, pages 8053–8063, 2021. 2

[30] R. Lu, A. Gijsenij, T. Gevers, V. Nedović, D. Xu, and J.-M. Geusebroek. Color constancy using 3d scene geometry. In *ICCV*, pages 1749–1756, 2009. 2, 3

[31] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660, 2017. 3

[32] C. R. Qi, H. Su, M. Nießner, A. Dai, M. Yan, and L. J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *CVPR*, pages 5648–5656, 2016. 6

[33] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017. 4, 6, 7

[34] Y. Qian, K. Chen, J. Kämäräinen, J. Nikkanen, and J. Matas. Recurrent color constancy. In *ICCV*, 2017. 2

[35] Y. Qian, J.-K. Kamarainen, J. Nikkanen, and J. Matas. On finding gray pixels. In *CVPR*, 2019. 2, 6, 8

[36] Y. Qian, J. Käpylä, J.-K. Kämäräinen, S. Koskinen, and J. Matas. A benchmark for burst color constancy. In *ECCVW*, 2020. 2

[37] T. Schps, J. L. Schnberger, S. Galliani, T. Sattler, and A. Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *CVPR*, 2017. 2, 4, 5, 6, 8

[38] H.-L. Shen and Q.-Y. Cai. Simple and efficient method for specularity removal in an image. *Applied optics*, 48(14):2711–2719, 2009. 6

[39] W. Shi, C. C. Loy, and X. Tang. Deep specialized network for illumination estimation. In *ECCV*, 2016. 2

[40] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, pages 746–760. Springer, 2012. 2, 4, 5, 6

[41] J. Van De Weijer, T. Gevers, and A. Gijsenij. Edge-based color constancy. *IEEE TIP*, 16(9), 2007. 2, 6

[42] I. Vasiljevic, N. Kolkin, S. Zhang, R. Luo, H. Wang, F. Z. Dai, A. F. Daniele, M. Mostajabi, S. Basart, M. R. Walter, et al. Diode: A dense indoor and outdoor depth dataset. *arXiv preprint arXiv:1908.00463*, 2019. 2, 4, 5, 6

[43] J. Xiao, S. Gu, and L. Zhang. Multi-domain learning for accurate and few-shot color constancy. In *CVPR*, 2020. 2, 6

[44] B. Xu, J. Liu, X. Hou, B. Liu, and G. Qiu. End-to-end illuminant estimation based on deep metric learning. In *CVPR*, 2020. 6

[45] S. Yan, J. Yang, J. Kapyla, F. Zheng, A. Leonardis, and J.-K. Kamarainen. Depthtrack: Unveiling the power of rgbd tracking. In *ICCV*, pages 10725–10733, 2021. 1

[46] J.-S. Yoo and J.-O. Kim. Dichromatic model based temporal color constancy for ac light sources. In *CVPR*, 2019. 2

[47] H. Yu, K. Chen, K. Wang, Y. Qian, Z. Zhang, and K. Jia. Cascading convolutional color constancy. In *AAAI*, 2020. 6

[48] X. Zhang, T. Sim, and X. Miao. Enhancing photographs with near infrared images. In *CVPR*. IEEE, 2008. 1

[49] Z. Zhang. A flexible new technique for camera calibration. *IEEE TPAMI*, 22(11):1330–1334, 2000. 3