REVERSE IMAGING PIPELINE FOR RAW RGB IMAGE AUGMENTATION

Samu Koskinen

Huawei Technologies Oy (Finland) Co. Ltd
Tampere, Finland

Dan Yang, Joni-Kristian Kämäräinen

Vision Group
Tampere University

ABSTRACT

We propose a reverse camera imaging pipeline to convert arbitrary images to raw RGB responses of a specific camera. The pipeline requires only that the camera's RGB responses are characterized. The reversed pipeline helps camera developers to generate camera specific raw images and use them to train learning-based imaging pipeline algorithms. In our experiments, three recent deep color constancy architectures achieve superior results in the cross-dataset setting using generated images.

***Index Terms*—** Data augmentation, imaging pipeline

1. INTRODUCTION

Cameras are light-measuring devices, but a core component in the camera chips is the camera imaging pipeline that performs important processing steps in a sequence to obtain the final RGB output [1, 2]. The typical processing steps are, for example, demosaicing, white-balance, color space mapping and noise reduction and many of them are well-known research topics in their own right. In the recent literature, learning-based methods have shown superior performance in the above processing steps and, in particular, this holds for the deep learning based methods [3, 4, 5].

The main difficulty in using deep architectures is that many imaging pipeline steps are camera specific and therefore require capturing a large amount of raw RGB and high quality ground truth images. In this work, we propose a reverse imaging pipeline where the ground truth images can be arbitrary, for example, downloaded from the Web, and the corresponding raw RGB images are generated by reversing the imaging pipeline. The only requirement is that spectral characterisation of the target camera is available. Our main contributions are:

- A reverse camera pipeline to generate raw RGB images from high-quality ground truth images.
- Experiments on the three recent color constancy (CC) network architectures which are trained using generated images and achieve superior results in the challenging cross-dataset setting.

2. RELATED WORK

Deep learning based methods have recently been proposed for the image pipeline processing steps, e.g., demosaicking [3], white-balance (color constancy) [4], noise reduction [5] and deblurring [6]. However, these deep models were trained using available datasets making them camera-independent and unsuitable for camera-specific processing. Karaimer and Brown [7] recently proposed a *forward* imaging pipeline which can be used to study effects of the processing steps to final images. Their pipeline can be used in data augmentation if the raw RGB images are available, but in our case we reverse the pipeline for arbitrary images.

RGB to spectral image conversion is an essential step in generating raw RGB images from standard RGB images. Several methods have been proposed for this task [8, 9, 10, 11, 12]. The methods in [8, 11, 12] assume that the spectral characterization of each image is available, but this is an invalid assumption in our case. In [9] spectral color palette and RGB values are matched in the L*a*b* space, but the problem is addressed separately from the imaging pipeline. [10] proposes a hybrid method to combine a low resolution hyperspectral image together with a high resolution RBG image to create a high resolution spectral image. [13] addresses camera processing models that can be used for reverse processing, but they do not consider the case of arbitrary cameras.

3. REVERSE IMAGING PIPELINE

Formation of a raw RGB image I of a scene R with the camera C of known spectral sensitivities $S_{i=R,G,B}$ and under a global illumination L can be expressed as [14]

$$I_i(x,y) = \int L(\lambda)S_i(x,y,\lambda)R(x,y,\lambda)d\lambda, i \in \{\text{R,G,B}\} \ . \tag{1}$$

Using the notation in [15] this can be written in the matrix form

$$\boldsymbol{\Phi_{cam}} = \boldsymbol{C_{cam}}\text{diag}(\boldsymbol{l})\boldsymbol{R} = \boldsymbol{C_{cam}}\boldsymbol{L}\boldsymbol{R} \ , \tag{2}$$

where $\boldsymbol{l}$ is a $1 \times N$ illumination spectrum and diag$(\cdot)$ creates a $N \times N$ diagonal matrix, where N is the number of discrete

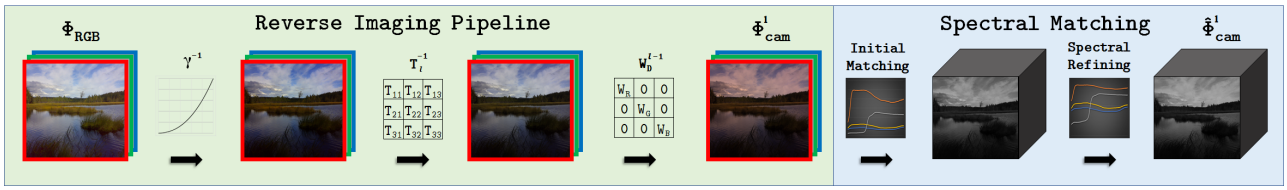

Fig. 1. The reverse imaging pipeline from the perceptual color space $\boldsymbol{\Phi_{RGB}}$ to camera's raw RGB space $\boldsymbol{\Phi_{cam}}$ and spectral image $\hat{\boldsymbol{\Phi}}_{\boldsymbol{cam}}$.

wavelength samples. $\boldsymbol{C_{cam}}$ is a row-wise and $\boldsymbol{R}$ a column-wise spatial reflectance matrix of sensor response and scene colors, respectively. The goal of the camera pipeline is to convert $\boldsymbol{\Phi_{cam}}$ to a perceptual color space. Unlike in [15] we use a color space specific standard illuminant $\boldsymbol{L_{ref}}$ in the definition resulting to

$$\boldsymbol{\Phi_{XYZ}} = \boldsymbol{C_{XYZ}} \boldsymbol{L_{ref}} \boldsymbol{R} \ , \tag{3}$$

where $\boldsymbol{C_{XYZ}}$ uses the CIE 1931 XYZ matching functions [16] and $\boldsymbol{L_{ref}}$ is the $D65$ illuminant in the case of an sRGB color space. While this is the *forward* imaging pipeline the goal in our work is to estimate $\boldsymbol{\Phi_{cam}}$ given $\boldsymbol{\Phi_{XYZ}}$, i.e. the *reverse* imaging pipeline (Figure 1).

3.1. RGB Linearization

The most popular color coding is the gamma corrected standard RGB (sRGB) space (CIE 1931 RGB) and therefore we adopt $\boldsymbol{\Phi_{RGB}}$ instead of the XYZ version. Since futher processing is more convenient in the linear color space the gamma correction is removed using the following equation [17]:

$$C = \begin{cases} \frac{C'}{12.92}, & C' \leq 0.04045 \\ \left(\frac{C'+0.055}{1+0.055}\right)^{2.4}, & C' > 0.04045 \end{cases} \ , \tag{4}$$

where C' are the R, G and B values of $\boldsymbol{\Phi_{RGB}}$ and C is the corresponding linear value.

3.2. Color Space Transform (CST)

The purpose of a camera specific color space transform $\boldsymbol{T_l}$ is to transform camera observed colors to the standard observer (CIE 1931 RGB) color space. The transformation is a factory computed photometric calibration using (adapted from [15])

$$\boldsymbol{T_l} = \arg\min_{\boldsymbol{T_l}} ||\boldsymbol{C_{RGB}} \text{diag}(\boldsymbol{l}) \boldsymbol{R_{ref}} - \boldsymbol{T_l} \boldsymbol{W_D^l} \boldsymbol{\Phi_{cam}^l}||^2 \ . \tag{5}$$

The optimization typically requires a set of captured calibration images $\boldsymbol{\Phi_{cam}^l}$ from references with known spectra $\boldsymbol{R_{ref}}$ or CIE XYZ/RGB values. $\boldsymbol{T_l}$ is illumination specific as seen in (5). We selected a number of pre-defined illuminants for our method. Therefore we can optimize the transforms for each case separately. $\boldsymbol{W_D^l}$ is a diagonal white-balance transform discussed further in Section 3.3.

Illuminant $R^k(\lambda)$ *Type*	*# of*	Color Temps (K)
Daylight	70	2500-9400
LED	13	2300-5800
Tungsten	9	2200-3250
Fluorescent	8	2500-4250

Table 1. Database of typical illuminants used in the proposed data augmentation.

The process can be simplified by utilizing spectral information. In $\boldsymbol{R_{ref}}$ we use the known color spectra of the 24 colors in the Classic X-Rite color chart. We also have a characterized camera with measured spectral responses for each color channel $\boldsymbol{C_{cam}}$. The only component that needs a special attention is the illumination defined by $\boldsymbol{l}$. This could be estimated from the input image, but we adopted a more straightforward solution. The illuminant $\boldsymbol{l}$ is randomly picked from a set of natural light sources with known spectrum. We are not actually interested about the true illuminant, but how the raw image would appear under some illuminant. A set of 100 natural illuminants were selected to cover typical indoor and outdoor use cases (Table 1).

A CST matrix for each image is computed from (5) as

$$\boldsymbol{T_l} = \arg\min_{\boldsymbol{T_l}} ||\boldsymbol{C_{RGB}} \text{diag}(\boldsymbol{l}) \boldsymbol{R_{ref}} - \boldsymbol{T_l} \boldsymbol{W_D^l} \boldsymbol{C_{cam}} \text{diag}(l) \boldsymbol{R_{ref}}||^2 \ . \tag{6}$$

$\boldsymbol{T_l}$ is optimized using gradient descent on each non-diagonal coefficient separately. The coefficient-wise optimization is iterated until the coefficients do not change anymore. All three diagonal values were re-calculated after each step in the coefficient optimization to ensure that $\sum t_{i,*} = 1$, i.e. to maintain white-balance. ${\boldsymbol{T_l}}^{-1}$ and ${\boldsymbol{W_D^l}}^{-1}$ were used to transform the linear RGB image $\boldsymbol{\Phi_{RGB}}$ to camera's RGB space $\boldsymbol{\Phi_{cam}^l}$.

3.3. White-balance

The white-balance matrix can be computed using (2) by setting a flat (white) spectrum $\boldsymbol{R} = \boldsymbol{1}$ and using the spectrum of the selected illuminant $\boldsymbol{l}$ and the camera sensitivity values $\boldsymbol{C_{cam}}$ as

$${\boldsymbol{W_D^l}}^{-1} = \text{diag}(\boldsymbol{C_{cam}} \boldsymbol{l}^T) \ . \tag{7}$$

3.4. Spectral Matching

The previous sections provide the essential tools to construct a raw RGB image from an arbitrary input image using

$$\boldsymbol{\Phi}^{\boldsymbol{l}}_{\boldsymbol{cam}} = {\boldsymbol{W}_{\boldsymbol{D}}^{\boldsymbol{l}}}^{-1} {\boldsymbol{T}_{\boldsymbol{l}}}^{-1} \gamma^{-1} \left(\boldsymbol{\Phi}_{\boldsymbol{RGB}} \right) \ , \tag{8}$$

where γ^{-1} is the inverse gamma correction (linearization). The final *spectral matching* step is needed for enabling accurate spatial simulations for all color processing steps in the imaging pipeline, such as the color shading. For spectral matching, the large Munsell Glossy database was selected [18] to represent natural spectra.

Initial Matching is conducted in the $\mathrm{L}^\star\mathrm{a}^\star\mathrm{b}^\star$ color space where color differences are accurately represented by the Euclidean distance [19] and the luminance $\mathrm{L}^\star$ can be omitted. At first, k nearest neighbors are selected and the raw RGB values are replaced by the weighted sums of k Munsell spectra

$$\begin{aligned} \hat{\boldsymbol{\Phi}}^{\boldsymbol{l}}_{\boldsymbol{cam}}(x,y) &= \sum_k w_k \boldsymbol{L}\boldsymbol{R}^k_{Munsell} \\ \{w_k\} &= \underset{\{w_k\}}{\arg\min} \, || \boldsymbol{\Phi}^{\boldsymbol{l}}_{\boldsymbol{cam}}(x,y) - \sum_k w_k \boldsymbol{L}\boldsymbol{R}^k_{Munsell} ||^2_{a,b} \end{aligned} \ , \tag{9}$$

where $\hat{\boldsymbol{\Phi}}$ is the spectral estimate of $\boldsymbol{\Phi}$ in the RGB space. Note that this is performed separately for each pixel (x, y).

Spectral Refinement is needed to compensate the interpolation errors generated by a finite number of spectra in the Munsell database. It is noteworthy that by increasing k and optimizing (9) a perfect match can be achieved, but that may lead to unnatural and peaky spectra. Instead, we propose to perform refinement in the RGB space and to iteratively modify the initial spectrum with the weighted CIE sensitivity curves

$$\hat{\boldsymbol{\Phi}}^{\boldsymbol{l(t+1)}}_{\boldsymbol{cam}} = \hat{\boldsymbol{\Phi}}^{\boldsymbol{l(t)}}_{\boldsymbol{cam}} + \mathrm{diag}(\boldsymbol{c}) \boldsymbol{C}_{\boldsymbol{RGB}} \hat{\boldsymbol{\Phi}}^{\boldsymbol{l(t)}}_{\boldsymbol{cam}} \ , \tag{10}$$

where the channel-wise estimation coefficients are ($\hat{i}$ is the estimate and i the target value)

$$\boldsymbol{c} = (c_R, c_G, c_B)^T, \ \ c_i = \frac{i + \epsilon}{\hat{i}} - 1 \ . \tag{11}$$

The iteration is continued until the channel-wise errors are $|i - \hat{i}| < 10^{-5}$ and ϵ makes sure that the final spectrum is non-zero everywhere (ϵ was set to 10^{-6}).

4. EXPERIMENTS

We experimented the proposed reverse pipeline in the problem of *auto white-balance* (AWB) or *color constancy* which is an essential step in imaging pipelines. The goal of color constancy is to estimate the illumination color and remove its effect from the raw RGB ($\boldsymbol{W}^{\boldsymbol{l}}_{\boldsymbol{D}}$ in Section 3.3).

4.1. Datasets, Methods and Settings

The datasets used in the experiments were *Shi-Gehler* [20] (568 images, 2 cameras) and *NUS* [21] (1,853 images, 9 cameras). Results are reported only for the cross-dataset experiment which is the most challenging setup. Since the NUS dataset is considered as the most difficult it was selected as the test data.

The selected methods were three recent deep learning models for color constancy: CNN-CC [22], VGG-CC [23] and FC4 [4]. CNN-CC consists of five convolutional layers that are trained from the scratch while VGG-CC and FC4 adopt VGG16 [24] as the feature extraction network. VGG-CC is a straightforward extension with a regression layer and FC4 uses a spatial confidence map and a tailored loss function for better regularization.

Image augmentation was done using the proposed reverse pipeline in Section 3. Images were randomly selected from the Google OpenImages dataset [25, 26] and more technical details are given in Section 4.2.

Performance measure is the angular distance between a ground truth and an estimated color correction vector. We also report the proportion of images for which the error is below 3.0° which considered as sufficient accuracy.

4.2. Camera Data Augmentation

Camera Characterization was done using the Labsphere's QES-1000 device to measure spectral response of a Huawei P20 mobile camera in the the range from 380nm to 1100nm with 1nm steps. We averaged the spectra of each non-overlapping 50×50 pixel region for each color channel providing a spatial spectral characterization grid.

Image generation was conducted using the reverse pipeline in Section 3 (Figure 2). To increase randomness for data augmentation purposes the spectrums were combined by a simple rule to construct a random illuminant $\boldsymbol{l}$ as $\boldsymbol{l} = 0.95 \cdot \boldsymbol{l} + 0.05 \cdot \boldsymbol{l}'$ where $\boldsymbol{l}'$ is a randomly selected daylight illuminant. Moreover, we introduced a random multiplier between $[0.6, 1.4]$ to generated images in order to simulate under and over exposures with clipped highlights.

4.3. Results

The results for all three methods and for different amount of generated images are shown in Table 2. There are two important findings: i) all methods achieve better results with the generated data than with the real data (Shi-Gehler), ii) the results gradually improve with increasing number of generated data. CNN-CC achieves the best results with only 500 generated images, but also the other amounts provide very similar results. We believe that this can be explained by the fact that CNN-CC network is relatively small (designed for Shi-Gehler

Fig. 2. Reverse pipeline generated raw RGB images (left) and their forward pipeline generated ground truth images (right). Gamma was added for the raw RGB images for visualization.

with only 568 images), patch-based (no full image) and cannot therefore benefit from more than 500 images. VGG-CC is a huge network and clearly benefits from increasing amount of training images with relative improvement more than 40% from 500 to 10k generated images in $\leq 3.0°$. The best results were achieved with FC4 which uses spatial confidence maps to regularize the VGGNet features. FC4 performs relatively well with only Shi-Gehler images, but with 10k generated images it achieves 12% improvement.

4.4. Backward-forward Verification

To validate the proposed reverse pipeline we conducted a verification check with 1,000 random images from Google OpenImages. The images were first reverse processed to spectral images and then the images were forward processed with the same pipeline. This backward-forward experiment revealed how much $\Delta C^* = \sqrt{(a_1^* - a_2^*)^2 + (b_1^* - b_2^*)^2}$ error is introduced by the pipeline itself. In literature errors below 1.0 are considered not noticeable to humans. The results are in Table 3 and examples in Figure 3.

5. CONCLUSIONS

In this paper, we proposed a reverse imaging pipeline that can be used to augment existing large image datasets to simulate different color processing stages in the imaging pipeline. This enables large quantities of camera specific training data where the size is only limited by the computation time and not with the amount of laborious data gathering. The experiments with the latest CNN based color constancy algorithms prove that CNN training benefits from generated data and superior accuracy was achieved in the challenging cross-dataset setting.

Method	*Tr. Data*	*Error*		
		Mean	Med.	$\% \leq 3.0°$
CNN-CC [22]	Shi-Gehler	**4.03**	3.66	35.5
	Gen. 500	4.48	**3.12**	**46.8**
	1k	4.37	3.20	44.1
	2k	4.77	3.26	45.1
	10k	4.42	3.21	45.2
VGG-CC [23]	Shi-Gehler	6.39	5.68	14.3
	Gen. 500	6.38	5.53	22.1
	1k	5.88	4.83	27.4
	2k	5.82	4.96	26.4
	10k	**5.54**	**4.46**	**31.5**
FC4 [4]	Shi-Gehler	**3.71**	3.15	46.1
	Gen. 500	4.43	3.41	42.6
	1k	4.15	3.47	41.3
	2k	4.09	3.22	46.2
	10k	3.99	**2.92**	**52.0**

Table 2. Cross-dataset color constancy results for the NUS dataset [21] (9 cameras and 1,853 images): mean error, median error and proportion of images with error less than $3.0°$. The methods are trained either with the Shi-Gehler [20] real data or increasing amounts of generated data using the proposed reverse imaging pipeline.

ΔC^*	Mean	Median	Max	Top-99%
	$1.6 \cdot 10^{-3}$	$3.0 \cdot 10^{-3}$	1.7	$1.98 \cdot 10^{-2}$

Table 3. Backward-forward verification of the proposed pipeline. Results are for 1,000 random Google OpenImages.

Fig. 3. Pipeline verification example: the original image (left) and the backward-forward processed image (right). For this example the average ΔC^* was $9.0 \cdot 10^{-3}$ and the maximum 0.1261.

The results also show that the performance improves with additional data. Therefore we can easily generate even larger datasets to further improve the results.

6. REFERENCES

[1] logicBRICKS, *logiISP Image Signal Processing (ISP) Pipeline, Datasheet v2.0*, 2018.

[2] Qualcomm Technologies Inc., *Breakthrough Mobile Imaging Experiences - How Qualcomm Snapdragon Processors Are Transforming Photography*, 2014.

[3] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," in *SIGGRAPH Asia*, 2016.

[4] Yuanming Hu, Baoyuan Wang, and Stephen Lin, "FC4: Fully convolutional color constancy with confidence-weighted pooling," in *CVPR*, 2017.

[5] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2noise: Learning image restoration without clean data," in *ICML*, 2018.

[6] J. Mustaniemi, J. Kannala, J. Matas, S. Särkkä, and J. Heikkilä, "LSD2 - joint denoising and deblurring of short and long exposure images with convolutional neural networks," *CoRR*, vol. abs/1811.09485, 2018.

[7] H.C. Karaimer and M.S. Brown, "A software platform for manipulating the camera imaging pipeline," in *ECCV*, 2016.

[8] A.S. Glassner, "How to derive a spectrum from an RBG triple," *IEEE Computer Graphics and Applications*, vol. 9, no. 4, 1989.

[9] H. Kotera, "RGB to spectral image conversion using spectral pallete and compression by SVD," in *ICIP*, 2003.

[10] R. Kawakami, Y. Matsushita, J. Wright, M. Ben-Ezra, Y.-W. Tai, and K. Ikeuchi, "High-resolution hyperspectral imaging via matrix factorization," in *CVPR*, 2011.

[11] B. Arad and O. Ben-Shahar, "High-resolution hyperspectral imaging via matrix factorization," in *ECCV*, 2016.

[12] Y. Jia, Y. Zheng, L. Gu, A. Subpa-Asa, A. Lam, Y. Sato, and I. Sato, "From RGB to spectrum for natural scenes via manifold-based mapping," in *ICCV*, 2017.

[13] Ayan Chakrabarti, Daniel Scharstein, and Todd Zickler, "An empirical camera model for internet color vision," in *BMVC*, 2009.

[14] J. von Kries, "Influence of adaptation on the effects produced by luminous stimuli," *Source of Color Science*, pp. 109–119, 1970.

[15] Hakki Can Karaimer and Michael S. Brown, "Improving color reproduction accuracy on cameras," in *CVPR*, 2018.

[16] J. Guild, "The colorimetric properties of the spectrum," *Philosophical Transactions of the Royal Society of London*, vol. 26, no. 1, 1932.

[17] International Electrotechnical Commission IEC, *Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space - sRGB*, 1999.

[18] J. Orava, "The reflectance spectra of 1600 glossy munsell color chips," Accessed: 2018-12-27.

[19] International Organization for Standardization, *ISO 11664-4 Standard - Colorimetry Part 4: CIE 1976 L*a*b* Colour Space*, 2008.

[20] Lilong Shi and Brian Funt, "Re-processed version of the gehler color constancy dataset of 568 images," 2008, Accessed: 2018-12-26.

[21] Dongliang Cheng, Dilip K. Prasad, and Michael S. Brown, "Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution," *J. Opt. Soc. Am. A*, vol. 31, no. 5, pp. 1049–1058, May 2014.

[22] Simone Bianco, Claudio Cusano, and Raimondo Schettini, "Color constancy using cnns," in *CVPR Workshop*, 2015.

[23] Y. Qian, K. Chen, J.-K. Kämäräinen, J. Nikkanen, and J. Matas, "Deep structured-output regression learning for computational color constancy," in *Int. Conf. on Pattern Recognition (ICPR2016)*, 2016.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[25] A. Kuznetsova, H. Rom, N. Alldrin, J. Uijlings, I. Krasin, J. Pont-Tuset, S. Kamali, S. Popov, M. Malloci, T. Duerig, and V. Ferrari, "The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale," *arXiv:1811.00982*, 2018.

[26] I. Krasin, T. Duerig, and N. Alldrin et al., "OpenImages: A public dataset for large-scale multi-label and multi-class image classification.," *Dataset available from https://storage.googleapis.com/openimages/web/index.html*, 2017.