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 characterization 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]

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

(1)

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

$$\Phi_{cam} = C_{cam}\text{diag}(l)R = C_{cam}LR ,$$  

(2)

where $l$ is a $1 \times N$ illumination spectrum and diag$(\cdot)$ creates a $N \times N$ diagonal matrix, where $N$ is the number of discrete
wavelength samples. \( C_{cam} \) is a row-wise and \( R \) a column-wise spatial reflectance matrix of sensor response and scene colors, respectively. The goal of the camera pipeline is to convert \( \Phi_{cam} \) to a perceptual color space. Unlike in [15] we use a color space specific standard illuminant \( L_{ref} \) in the definition resulting to

\[
\Phi_{XYZ} = C_{XYZ} L_{ref} R ,
\]

where \( C_{XYZ} \) uses the CIE 1931 XYZ matching functions [16] and \( L_{ref} \) 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 \( \Phi_{cam} \) given \( \Phi_{XYZ} \), 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 \( \Phi_{RGB} \) instead of the XYZ version. Since further processing is more convenient in the linear color space the gamma correction is removed using the following equation [17]:

\[
C = \begin{cases} 
\frac{C'}{12.92}, & C' \leq 0.04045 \\
(\frac{C'+0.055}{1.055})^{2.4}, & C' > 0.04045
\end{cases}
\]

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

### 3.2. Color Space Transform (CST)

The purpose of a camera specific color space transform \( T_{l} \) 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])

\[
T_{l} = \arg \min_{T_{l}} ||C_{RGB} \text{diag}(l) R_{ref} - T_{l} W_{D}^{l} \Phi_{cam}^{l}||^2 .
\]

The optimization typically requires a set of captured calibration images \( \Phi_{cam}^{l} \) from references with known spectra \( R_{ref} \) or CIE XYZ/RGB values. \( T_{l} \) 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. \( W_{D}^{l} \) is a diagonal white-balance transform discussed further in Section 3.3.

The process can be simplified by utilizing spectral information. In \( R_{ref} \) 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 \( C_{cam} \). The only component that needs a special attention is the illumination defined by \( l \). This could be estimated from the input image, but we adopted a more straightforward solution. The illuminant \( l \) 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).

<table>
<thead>
<tr>
<th>Illuminant</th>
<th>Type</th>
<th># of</th>
<th>Color Temps (K)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Daylight</td>
<td>70</td>
<td>2500-9400</td>
<td></td>
</tr>
<tr>
<td>LED</td>
<td>13</td>
<td>2300-5800</td>
<td></td>
</tr>
<tr>
<td>Tungsten</td>
<td>9</td>
<td>2200-3250</td>
<td></td>
</tr>
<tr>
<td>Fluorescent</td>
<td>8</td>
<td>2500-4250</td>
<td></td>
</tr>
</tbody>
</table>

### 3.3. White-balance

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

\[
W_{D}^{l-1} = \text{diag}(C_{cam}^{lT}) .
\]
3.4. Spectral Matching

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

$$\Phi_{\text{cam}}(x, y) = \sum_k w_k L R^k_{\text{Munsell}}$$

where $$\Phi_{\text{cam}}$$ is the spectral estimate of $$\Phi$$ 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 the initial spectrum with the weighted CIE sensitivity curves

$$\hat{\Phi}_{\text{cam}}^{l(t+1)} = \hat{\Phi}_{\text{cam}}^{l(t)} + \text{diag}(c) C_{\text{RGB}} \hat{\Phi}_{\text{cam}}^{l(t)}$$

where the channel-wise estimation coefficients are $$(i$$ is the estimate and $$t$$ the target value)

$$c = (c_R, c_G, c_B)^T, \quad c_i = \frac{i + \epsilon}{i} - 1$$

The iteration is continued until the channel-wise errors are $$(i - \hat{i}) < 10^{-5}$$ and $$\epsilon$$ makes sure that the final spectrum is non-zero everywhere ($$\epsilon$$ 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 ($$W^L_I$$ in Section 3.3).
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 $\leq 3.0^\circ$. 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.

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

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.

<table>
<thead>
<tr>
<th>$\Delta C^*$</th>
<th>Mean</th>
<th>Median</th>
<th>Max</th>
<th>Top-99%</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$1.6 \cdot 10^{-3}$</td>
<td>$3.0 \cdot 10^{-3}$</td>
<td>1.7</td>
<td>$1.98 \cdot 10^{-2}$</td>
</tr>
</tbody>
</table>

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


