SWBNet: A Stable White Balance Network for sRGB Images

Chunxiao Li, Xuejing Kang, Zhifeng Zhang, Anlong Ming

School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications
{chunxiaol,kangxuejing,zhangzhifeng,mal}@bupt.edu.cn

Abstract

The white balance methods for sRGB images (sRGB-WB) aim to directly remove their color temperature shifts. Despite achieving promising white balance (WB) performance, the existing methods suffer from WB instability, i.e., their results are inconsistent for images with different color temperatures.

We propose a stable white balance network (SWBNet) to alleviate this problem. It learns the color temperature-insensitive features to generate white-balanced images, resulting in consistent WB results. Specifically, the color temperature-insensitive features are learned by implicitly suppressing low-frequency information sensitive to color temperatures. Then, a color temperature contrastive loss is introduced to facilitate the most information shared among features of the same scene and different color temperatures. This way, features from the same scene are more insensitive to color temperatures regardless of the inputs. We also present a color temperature sensitivity-oriented transformer that globally perceives multiple color temperature shifts within an image and corrects them by different weights. It helps to improve the accuracy of stabilized SWBNet, especially for multi-illumination sRGB images. Experiments indicate that our SWBNet achieves stable and remarkable WB performance.

Introduction

White balance for sRGB images (sRGB-WB) aims at removing the color temperature shifts caused by the improper or personalized white balance (WB) settings in image signal processing (ISP) (Afifi et al. 2019a). It is a burgeoning research direction and has a great impact on some computer vision tasks, such as semantic segmentation and image classification (Afifi and Brown 2019).

Since 2019, several sRGB-WB methods have been proposed, which can be divided into exemplar-based methods and deep neural network (DNN)-based methods. Exemplar-based methods, such as KNN-WB (Afifi et al. 2019a) and MixedWB (Afifi, Brubaker, and Brown 2022), compute non-linear mappings for different color temperatures and then correct the input images according to them. DNN-based method, i.e., DWB (Afifi and Brown 2020a), builds a unified mapping for different color temperatures with the DNN architecture trained in an end-to-end manner. Although achieving promising WB accuracy, both categories of methods have inconsistent results for the images with different color temperatures. That is, they suffer from WB instability, which limits the performances when applied to downstream tasks, such as TV broadcasting and image editing (Vazquez-Corral and Bertalmío 2014). Fig.1 shows some examples. Three representative sRGB-WB methods (KNN-WB, MixedWB, DWB) are used to correct the sRGB images rendered with four typical color temperatures. Ideally, regardless of the inputs, an effective sRGB-WB method should generate white-balanced images with consistent neutral color temperature. However, for the single-illumination inputs in Fig.1(a) (first four rows), the white-balanced images generated by three methods have inconsistent color temperatures even if the scenes are fixed. This is reflected in the visualizations and varying $\Delta E_{2000}$ values (Sharma, Wu, and Dalal 2005) in Fig.1(b)-(d). Worse yet, for the multi-illumination inputs in Fig.1(a) (last two rows), WB instability is severe, and WB accuracy is greatly degraded.

In this paper, we propose a stable white balance network
called SWBNet to mitigate WB instability. Our SWBNet learns the color temperature-insensitive features to generate white-balanced sRGB images, thus effectively reducing their color temperature inconsistency. Specifically, we propose a color temperature-insensitive feature (CTIF) extractor based on low-frequency information’s color temperature-sensitive property. By suppressing this information, the CTIF extractor can extract the color temperature-insensitive features. Then, these features are encouraged to become more insensitive by the color temperature contrastive (CT-contrastive) loss. It explicitly maximizes the shared information among features with the same scene and different color temperatures. We also propose a color temperature sensitivity-oriented (CTS-oriented) transformer to further improve WB accuracy, especially for multi-illumination sRGB images. Benefiting from the global receptive field, the CTS-oriented transformer can perceive different color temperature shifts within an image and thus corrects them differently. In summary, the contributions are:

1. We propose a CTIF extractor that gathers color temperature-sensitive information from extracted features and then adaptively suppresses it, thus implicitly learning color temperature-insensitive features.
2. We propose a CT-contrastive loss to enhance the insensitivity by promoting features with the same scenes and different color temperatures to share the most information.
3. We present a CTS-oriented transformer to further improve WB accuracy by correcting multiple color temperature shifts differently.
4. Our SWBNet achieves stable and state-of-the-art WB performance on commonly used datasets.

Related Work

In the WB module of ISP, the classical WB methods (Raw-WB) for raw images are used to remove their color shift caused by the scene illumination (Gijsenij and Gevers 2007; Foster 2011; Hu, Wang, and Lin 2017; Li, Fu, and Heidrich 2021; Kim et al. 2021; Zhao et al. 2022). Most digital cameras offer an option to adjust these methods according to the desired color temperature. However, once the WB is personalized or incorrectly set, the final sRGB images still have color temperature shifts. Further, due to non-linear renderings in ISP, the shifts within an sRGB image is inconsistent (Afifi et al. 2019a). Raw-WB methods cannot correct them, which motivated the development of sRGB-WB methods.

In 2019, (Afifi et al. 2019a) proposed the first exemplar-based sRGB-WB method called KNN-WB. It modeled the nonlinear mapping as a polynomial kernel function (Hong, Luo, and Rhodes 2001) by extracting the high-order color matrix. Such a color temperature-sensitive feature makes KNN-WB affected by color temperatures. Followed by KNN-WB, some exemplar-based methods were also proposed. For example, CTT (Afifi et al. 2019b) embedded mappings computed by KNN-WB in the JPEG annotations. When the color temperature shift appeared in an sRGB image, CTT fetched these mappings to correct it. Interactive-WB (Afifi and Brown 2020b) allowed users to select a reference point in an sRGB image and then use KNN-WB to correct it or reset its color temperature. Recently, MixedWB (Afifi, Brubaker, and Brown 2022) applied the mappings in KN...
\[ \hat{I}_{\text{wb}} = G(F(I_{ct})), \]  
\[ \alpha \text{ arg min } \theta f, \theta g \sum_{ct_{i}} \sum_{s_{j}} \| I_{\text{wb},i} - G_{\theta g}(F_{\theta f}(I_{ct_{i}})) \|_{1}, \]  
where \( \theta f \) and \( \theta g \) are the parameters of encoder \( F_{\theta f}(\cdot) \) and decoder \( G_{\theta g}(\cdot) \), \( s_{j} \) is the \( j \)-th scene. Eq.4 aims to reconstruct image content perfectly and lacks consideration of different color temperature influences. Such constraint makes the trained \( G_{\theta g}(F_{\theta f}(\cdot)) \) affected by color temperatures. As a result, it would have inconsistent WB accuracy when correcting different color temperature inputs.

Next, we verify the above analysis. In Fig.3(a), the input sRGB images differ only in color temperatures and have the same GT. Affected by color temperatures, the existing sRGB-WB methods (e.g. KNN-WB, DWB) learn the color temperature-sensitive features for white-balanced sRGB image generation. As in Fig.3(b) and (e), there are clear non-overlaps among these features. Inevitably, in Fig.3(c), (d), (f), and (e), the generated white-balanced sRGB images have inconsistent color temperatures. Experiments in Rendered Cube and Mixedscene datasets accord with the above phenomena (Supplementary materials). Thus, we can conclude that existing sRGB-WB methods are affected by color temperatures and thus suffer from WB instability.

**Stable White Balance Network**

In this section, we introduce the SWBNet, which aims to alleviate WB instability. The whole framework is shown in Fig.4. We first introduce the CTIF extractor and CT-contrastive loss (Fig.4A). They collaborate to promote the SWBNet to learn the color temperature-insensitive features so as to alleviate WB instability. Then, we introduce the CTS-oriented transformer (Fig.4B). It helps to improve the performance of stabilized SWBNet by correcting multiple color temperature shifts within an sRGB image differently.

### Color Temperature-insensitive Feature Extractor

We propose the CTIF extractor to learn the color temperature-insensitive features. As shown in Fig.4A, it consists of a color temperature-sensitive frequency separation (CTFS) block and a color temperature sensitivity reduction (CTSR) block in each downsampling (DS) module. The CTFS block gathers color temperature-sensitive information from features. Then, the CTSR block suppresses this information, which makes features insensitive to color temperature. In the following, we introduce two blocks in detail.

#### Color Temperature-sensitive Frequency Separation Block

Inspired by (El Helou, Zhou, and Süssstrunk 2020; Xu et al. 2020), as in Fig.5, we observe that the coefficients sensitive to color temperatures are concentrated in the lowest continuous indexes (\( t \leq 4 \)) in the discrete cosine transform (DCT) domain. Based on this, we calculate the
Figure 4: The framework of our SWBNet: A. The CTIF extractor and CT-contrastive loss work together to learn the color temperature-insensitive features for achieving stable WB performance. B. The CTS-oriented transformer corrects multiple color temperature shifts differently to improve WB accuracy, especially for the multi-illumination sRGB images.

color temperature-sensitive DCT coefficients for the \(c^{th}\) channel of the deep feature \(X_{cti}\) as:

\[
m^{(c,t)} = T_t(X^{(c)}) = w_{t_1,t_2} \ast X^{(c)},
\]

where \(t_1, t_2 = \text{zig}^{-1}(t), \ t \in [0, T]\) is the index range (we set \(T = 4\) according to Fig.5), \(w_{t_1,t_2}\) is the DCT basis in index \(t\), \(\text{zig}^{-1}(\cdot)\) is the inverse zig-zag function (Guo, Mousavi, and Monga 2019), \(\ast\) is the convolution operation. The color temperature-sensitive information \(X^{(c,t)}_{cti}\) can be computed by inverse DCT (IDCT):

\[
X^{(c,t)}_{cti} = T_t^{-1}(\{m^{(c,t)}\}) = \sum_{0}^{T} w_{t_1,t_2} \ast g_s(m^{(c,t)}),
\]

where \(g_s(\cdot)\) is the S zero-padding function (Guo, Mousavi, and Monga 2019). In Fig.5, the remaining high-frequency information is insensitive to color temperatures, which can be calculated as \(X^{(c)}_{cti} = X^{(c,t)}_{cti} - X^{(c,l)}_{cti}\).

The attributions and structure of the CTSFS block are in Table.1 and Fig.4A (orange block). Due to the linear property of DCT and IDCT, we make the CTSFS block participate in the training of the CTIF extractor by modeling Eq.5 and Eq.6 as group convolutions. This way, when applying the CTSFS block to the features with different color temperatures (Fig.6(a)), there is a clear difference in the gathered color temperature-sensitive parts (Fig.6(b)), while the color temperature-insensitive parts are similar (Fig.6(c)). Further, such ability provides a centralized object for color temperature sensitivity reduction in the CTSR block.

**Color Temperature Sensitivity Reduction Block** Since the color temperature sensitivity is concentrated in the information gathered by our CTSFS block, the whole feature will be color temperature-insensitive if suppressing this portion. Our CTSR block achieves the goal by implicitly learning the color temperature shifts from its intensity description:

\[
\hat{X}^{l}_{cti} = \text{Conv}(X^{l}_{cti} \cdot \gamma) = \text{Conv}(X^{l}_{cti} \cdot (Fc(Avg(X^{l}_{cti})) + Fc(Var(X^{l}_{cti}))))
\]

where \(\cdot\) is the dot product, \(\text{Conv}(\cdot)\) is the convolution layer. Based on the fact that the color temperatures mainly affect feature intensities (Foster 2011) (e.g. Fig.6(b)), the comprehensive descriptions of these influences can be obtained...
by computing the average and variance in each channel by the adaptive average pooling $Avg(\cdot)$ and variance computation $Var(\cdot)$. Then, we apply the fully connected layers $Fc(\cdot)$ to learn the color temperature shift parameter $\gamma$ from these descriptions. Color temperature-insensitive information $X_{ct}^{l}$ can be obtained by multiplying $X_{ct}^{l}$ and $\gamma$. This effectively suppresses the color temperature sensitivity (Fig.6(d)). Thus, the deep feature $X_{ct_{i}}$ that reconstructed by $X_{ct}^{l}$ and $X_{ct_{i}}^{h}$ is color temperature-insensitive (Fig.6(f)).

**Color Temperature Contrastive Loss**

Since features from the same scene have identical GT, they become increasingly insensitive to color temperatures as sharing more information. This can further promote the model to learn color temperature-insensitive features. Inspired by (He et al. 2020), we introduce an explicit constraint called CT-contrastive loss to facilitate information sharing:

$$L_{ct} = -\log \frac{\exp\left(\frac{\|Avg(X_{ct_{1}})\|}{\tau}\right)}{\sum_{q} \exp\left(\frac{\|Avg(X_{ct_{1}})\|}{\tau}\right)}$$

where $Avg(\cdot)$ is the adaptive average pooling that extracts the global color temperature influence representation in each channel, $\|\cdot\|_2$ is the $l_2$ distance computation, $\tau$ is a temperature hyper-parameter ($\tau = 0.7$) (Peng et al. 2019), $\epsilon$ is a constant. From Eq.8, we target the feature $X_{ct_{1}}$, reducing the distance between it and the positive feature $X_{ct_{1}}^{p}$ (differ from $X_{ct}$ only in color temperatures) encourages them to share the same information. Simultaneously, increasing the distance between $X_{ct}$ and the negative feature $X_{ct_{1}}^{n}$ (differ from $X_{ct}$ in scenes) preserves as much shared information as possible. Consequently, with our CT-contrastive loss, the features from the same scene (Fig.6(a)) become color temperature-insensitive (Fig.6(f)).

In Fig.4A, we use a momentum-updated student model to provide the positive features (He et al. 2020). Also, since the multi-level features extracted by the CTIF extractor are critical to the image generation (Ronneberger, Fischer, and Brox 2015), we apply the CT-contrastive loss on them after the skip connections. This way, the CTIF extractor and CT-contrastive loss work together to learn color temperature-insensitive features via implicit color temperature sensitivity suppression and explicit information-sharing promotion.

**Color Temperature Sensitivity-oriented Transformer**

Due to the nonlinear renderings in ISP (Eq.1), the color temperature shifts within an sRGB image are inconsistent. Existing methods hardly capture these shifts globally because of the convolution’s limited receptive field (Afifi and Brown 2020a; Afifi, Brubaker, and Brown 2022) or the mapping’s limited color temperature representation (Afifi et al. 2019a). This limits WB accuracy except for instability, especially for the multi-illumination sRGB images (Fig.1, Fig.2(b)). The transformer can effectively capture the multiple color temperature shifts based on its global receptive field (Xie et al. 2021; Hermann, Chen, and Kornblith 2020). Benefiting from it, we propose a CTS-oriented transformer, which adapts the CTSFS block to extract color temperature-sensitive features for patch embedding. This provides dense and direct clues for exploring the global color temperature shifts.

As shown in Fig.4B, there are three parts in the framework: CTIF extractor, decoder, and CTS-oriented transformer. The input is first passed to the first two for pre-
liminary correction. Then, it is forwarded to the patch embedding module, which adopts two CTSFS blocks to obtain the color temperature-sensitive patches and then projects them as embeddings. To capture correlations, these embeddings are added with 1D position encoding and then fed to the transformer encoder with a multi-headed self-attention (MSA) computing inter-similarities. A linear decoder is further applied to learn the global color temperature shifts by linear and convolution layers. In Fig.7(b) and (c), the learned mask assigns higher correction weights to the areas with larger color temperature shifts. Thus, correcting the inputs with these weights can achieve superior WB accuracy, especially for the multi-illumination sRGB images.

**Experiments**

**Experiments Settings**

**Datasets** Following (Afifi and Brown 2020a), we randomly selected 12000 sRGB images from Rendered WB dataset (Afifi et al. 2019a) for training and used Rendered Cube and Mixedscene datasets for evaluation.

**Error Metric** We used the same error metrics as the recent works (Afifi et al. 2019a; Afifi, Brubaker, and Brown 2022; Afifi and Brown 2020a): mean square error (MSE), mean angle error (MAE), and $\Delta E_{2000}$. We reported the metrics’ mean, first quantile (Q1), second quantile (Q2), and third quantile (Q3) for evaluation.

**Loss Function** We firstly applied the CT-contrastive loss $\mathcal{L}_{ct}$ (Eq.8) and the $l_1$ loss $\mathcal{L}_p$ (Eq.4) to train the CTIF extractor and decoder (Fig.4A). Secondly, we trained the whole model under the supervision of loss $\mathcal{L}_p$ (Fig.4B).

**Implementation Details** We implement the SWBNet on Pytorch with CUDA support. To train the CTIF extractor and decoder, we use the Adam (Kingma and Ba 2015) with $\beta_1 = 0.9$. We set the learning rate as $1 \times 10^{-4}$ and then decay it to $1 \times 10^{-5}$ after 150 epochs. To train the whole model, we use the AdamW optimizer (Loshchilov and Hutter 2019) with weight-decay $10^{-2}$. We set batch size as 64 and train all modules for 200 epochs in two phases. Following (Afifi and Brown 2020a), we randomly select four 128×128 patches from each image and their corresponding GTs for training and apply geometric rotation and flipping as data augmentations to avoid overfitting.

**Comparison with State-of-the-art Methods** We compare the proposed SWBNet with several state-of-the-art sRGB-WB methods, including KNN-WB (Afifi et al. 2019a), DWB (Afifi and Brown 2020a), Interactive WB (Afifi and Brown 2020b) and MixedWB (Afifi, Brubaker, 2022). We used the same error metrics as the recent works (Afifi et al. 2019a; Afifi, Brubaker, and Brown 2022; Afifi and Brown 2020a): mean square error (MSE), mean angle error (MAE), and $\Delta E_{2000}$. We reported the metrics’ mean, first quantile (Q1), second quantile (Q2), and third quantile (Q3) for evaluation.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Rendered Cube dataset ($\Delta E_{2000}$)</th>
<th>Mixedscene dataset ($\Delta E_{2000}$)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2850K</td>
<td>3800K</td>
</tr>
<tr>
<td>KNN-WB</td>
<td>6.44</td>
<td>5.64</td>
</tr>
<tr>
<td>DWB</td>
<td>6.35</td>
<td>5.74</td>
</tr>
<tr>
<td>MixedWB</td>
<td>6.24</td>
<td>5.45</td>
</tr>
<tr>
<td>SWBNet</td>
<td>5.21</td>
<td>4.70</td>
</tr>
</tbody>
</table>

Table 2: Quantitative comparison for WB stability on the Rendered Cube and Mixedscene datasets. The top results are in bold. We discarded the sRGB images rendered by As Shot (AS) and Auto AWB (AU) due to the unfixed color temperatures.

We conducted an extensive experiment to evaluate the performance of the proposed SWBNet on the Rendered Cube and Mixedscene datasets. The results are summarized in Table 2. The top results are in bold.

![Figure 8: Qualitative comparison for WB stability. The $\Delta E_{2000}$ value is marked in the bottom right of the image.](image-url)
<table>
<thead>
<tr>
<th>Methods</th>
<th>Mean square error (MSE)</th>
<th>Mean angle error (MAE)</th>
<th>$\triangle E$ 2000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mean</td>
<td>Q1</td>
<td>Q2</td>
</tr>
<tr>
<td>Mixed scene dataset</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Interactive WB</td>
<td>1059.88</td>
<td>616.24</td>
<td>896.90</td>
</tr>
<tr>
<td>KNN-WB</td>
<td>1226.57</td>
<td>680.65</td>
<td>1062.64</td>
</tr>
<tr>
<td>DWB</td>
<td>1130.59</td>
<td>621.00</td>
<td>886.32</td>
</tr>
<tr>
<td>MixedWB</td>
<td>819.47</td>
<td>655.88</td>
<td>845.79</td>
</tr>
<tr>
<td>SWBNet</td>
<td>816.00</td>
<td>426.01</td>
<td>554.07</td>
</tr>
</tbody>
</table>

Table 3: Quantitative comparison for WB accuracy on the Rendered Cube and Mixedscene datasets. The top results are in bold.

and Brown 2022). The comparison results are in Table 2, Fig. 2 and Table 3 respectively.

**WB Stability** From Table 2 and Fig. 2, WB instability appears for KNN-WB, DWB, and MixedWB, reflected by the significant differences among the average $\Delta E_{2000}$ values for different color temperatures in Rendered Cube and Mixedscene datasets. In contrast, our SWBNet can reduce such differences significantly. In addition, from Fig. 8, the color temperatures of the white-balance sRGB images generated by SWBNet are more consistent. These results indicate that our SWBNet can effectively alleviate WB instability by learning the color temperature-insensitive features.

**WB Accuracy** From Table 3, our SWBNet achieves the best accuracy on two public datasets. In the Mixedscene dataset, our SWBNet outperforms the previous methods by 13.38%, 25.58%, 22.85%, 12.36% in the mean, Q1, Q2, and Q3 of $\Delta E_{2000}$ respectively. The improvements in the MSE and MAE are also significant. Meanwhile, in the Rendered Cube dataset, our SWBNet achieves the improvements of 6.75%, 7.96% in mean and Q2 of $\Delta E_{2000}$. These results confirm that our SWBNet can achieve better WB accuracy, especially for the multi-illumination sRGB images.

**Ablation Analysis**

We carried out an ablation study for the SWBNet on the Rendered Cube and Mixedscene datasets. We trained six combinations with 8000 sRGB images from Rendered Cube dataset and use the remaining images (Rendered Cube test) and the Mixedscene dataset for evaluation (testing images are resized to 128 × 128). The results are in Fig. 9. Compared with the vanilla U-Net (C1), replacing the encoder with the CTIF extractor (C2) or applying CT-contrastive loss (C3) can greatly reduce the differences of average $\Delta E_{2000}$ values for various color temperatures. Further, as C4, the CT-contrastive loss and CTIF extractor cooperate in achieving smaller differences of average $\Delta E_{2000}$ values. This indicates that learning color temperature-insensitive features is an effective way to alleviate WB instability. In addition, benefiting from the global view, the CTS-oriented transformer reduces the average $\Delta E_{2000}$ values for the most color temperatures (comparing C5 with C1), especially for the multi-illumination sRGB images. According to the above, the proposed components work together to alleviate WB instability and improve WB accuracy, reflected in the closest and lowest average $\Delta E_{2000}$ values in C6.

**Conclusion**

In this paper, we propose a SWBNet to alleviate WB instability of the existing sRGB-WB methods. We achieve this by learning the color temperature-insensitive features via implicit color temperature influence reduction and explicit information-sharing promotion. Further, we correct multiple color temperature shifts differently based on the transformer’s global receptive field. This helps to improve WB accuracy of stabilized SWBNet, especially for the multi-illumination sRGB images. Extensive experiments indicate that our SWBNet effectively learns the color temperature-insensitive features and corrects multiple color temperature shifts, thereby achieving stable and remarkable results. In future work, we plan to extend the SWBNet to improve the stability of temporal white balance, which is beneficial for mobile terminal filming.
Acknowledgments

This work was supported by the national key R & D program intergovernmental international science and technology innovation cooperation project (2021YFE0101600).

References


Vazquez-Corral, J.; and Bertalmio, M. 2014. Color stabilization along time and across shots of the same scene, for one or several cameras of unknown specifications. IEEE Transactions on Image Processing, 23(10): 4564–4575.
