Integral Fast Fourier Color Constancy

Supplementary Material

1. Data Augmentation

During training, the most straightforward input is the log-chroma histogram of an image. However, this approach overlooks spatial information and does not utilize additional sources of information, such as edges or spatial neighborhood relationships. In the main paper, we estimate the most accurate illumination L by filtering a set of histograms N constructed from the log-chroma values and local absolute deviation metrics of the pixels in the image I.

Beyond this, the model can also process a set of histograms $\{N_j\}$ derived from a group of "augmented" images $\{I'_j\}$. The filtered responses of these histograms are aggregated and passed through a softmax function to compute probabilities. These augmented images incorporate edge and spatial statistical information from I, enabling the model to leverage multiple information sources.

It is important to note that the chroma histograms must maintain a precise mapping between channel scaling in the input image and shifts in histogram space. As a result, augmented images must be non-negative and preserve intensity scaling. We experimented with four types of augmented images:

$$
\begin{aligned}
I_1 &= I \\
I_2 &= \max\left(0, I * \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}\right) \\
I_3 &= b(I^3, 9)^{1/3} \\
I_4 &= \sqrt{b(I^2, 3) - b(I, 3)^2}, \\
I_5 &= \frac{1}{8} \sum_{i=-1}^{1} \sum_{j=-1}^{1} |I(m, n) - I(m+i, n+j)|
\end{aligned}
\tag{1}
$$

where I_1 is the original image I, I_2 is the high-pass filtered version enhancing edges with a Laplacian-like filter and ensuring non-negative values via $\max(0, \cdot)$, I_3 is the cube root of a blurred cubic version of I with $b(I^3, 9)$ representing a 9-pixel neighborhood blur, I_4 is the local standard deviation computed as the square root of the difference between the blurred squared intensity $b(I^2, 3)$ and the square of the blurred intensity $b(I, 3)^2$, and I_5 is the average absolute deviation of a pixel from its 3×3 neighbors, calculated as the mean of absolute differences.

The experimental results, presented in Table 1, indicate that when the model is provided with histograms from four different types of augmented images, there is a noticeable improvement in performance. However, this enhancement comes at the cost of reduced processing speed. After balancing both performance and efficiency, the combination of $I + I_5$ emerges as the most optimal approach.

	Mean	Med.	Best 25%	Worst 25%	Time(ms)
I	2.31	2.03	0.57	6.14	-
I+I2	2.04	1.84	0.46	5.18	**4.9**
I+I3	2.07	1.88	0.48	5.29	11.3
I+I4	2.09	1.92	0.49	5.31	13.8
I+I5	1.98	1.79	0.42	4.91	6.2
I+I2+I3+I4+I5	**1.90**	**1.74**	**0.40**	**4.79**	35.7

Table 1. The effect of different data augmentations on model performance. Experiments were conducted on the LSMI Galaxy dataset, with window and overlap sizes set to 128 and 64, respectively.

2. Histogram Features

In the main paper, we set the number of bins to 64. Next, we will investigate the impact of varying bin sizes on the experimental results. The experiments are conducted on the LSMI Galaxy dataset, using two histograms as input: (1) the pixel intensity distribution in log-chroma space (I_1), and (2) the gradient intensity histogram I_5. To ensure that the data distribution within the dataset is fully contained within the defined u and v bounds, the range of u and v was set to $[-2.525, 2.525]$.

The experimental results, as shown in the Table2, demonstrate that when the number of bins n is too small, the color distribution is overly simplified, leading to a reduction in the amount of color detail captured by the model. This, in turn, hampers the model's ability to accurately recover true colors under diverse lighting conditions. Conversely, when n is too large, model performance tends to degrade. The increased number of bins may capture more noise than meaningful information. Furthermore, an excessive number of bins can lead to data sparsity, especially in smaller datasets, where each bin contains less information, thereby affecting the model's feature extraction capabilities.

	Mean	Med.	Best 25%	Worst 25%
n=16	2.79 ± 0.01	2.39 ± 0.01	0.90 ± 0.01	5.47 ± 0.01
n=32	2.17 ± 0.01	1.84 ± 0.01	0.51 ± 0.01	4.98 ± 0.01
n=64	1.98 ± 0.01	1.79 ± 0.01	0.42 ± 0.01	4.91 ± 0.01
n=128	2.19 ± 0.01	1.81 ± 0.01	0.41 ± 0.01	5.05 ± 0.01

Table 2. Impact of varying bin numbers n on model performance.

3. Extensions of IFFCC

In the main paper, we learned a set of weights that determine the shape of the filters used in frequency-domain convolution, as well as the gain and bias of those filters. Here, we extend IFFCC by learning a mapping from each training sample x_i to a set of weights w_i, rather than learning a single model. We optimize the weights in a small neural network with a ReLU activation function. Similar to other experiments, we use L-BFGS for training; however, the 'pretraining phase' from previous experiments is removed, and we train for 64 iterations only. The network weights are initialized with random Gaussian noise, and the weights w are indirectly regularized during the training process. This network-based methodology enhances the model's capability to infer information beyond simple pixel and edge log-chroma histograms, incorporating higher-level features such as semantic information and camera metadata.

To obtain additional information from images (metadata), we conducted experiments on the Shadow dataset, which includes metadata for each image captured by different cameras, such as exposure time and aperture size. When the network is aware that the images come from two different sensors, it can better handle the mapping relationships. Exposure time and similar metadata provide useful cues for distinguishing between ambient light and artificial light sources. Experimental results demonstrate that these metadata features are highly informative, significantly improving model performance and reducing error. We encode the additional information as a feature vector:

$$\begin{aligned} E &= \text{vec}\left([\log(A); \log(B); \log(C); 1; 1]\right) \times \mathbf{e}_i \\ \mathbf{e}_i &= [0, \ldots, 0, 1, 0, \ldots, 0] \in \mathbb{R}^N \end{aligned} \tag{2}$$

where A represents the shutter speed, B represents the image's f-number, and C represents the ISO. The i-th element of $\mathbf{e}_i$ is 1, with the remaining elements being 0, representing the name of the i-th camera.

We performed experiments using the nighttime scene from the Shadow dataset, which contains images captured by five distinct cameras. As demonstrated in the table, incorporating additional metadata (such as shutter speed, ISO, aperture size, and camera name) into the neural network significantly enhanced the model's performance. This improvement can be attributed to the fact that the sensor characteristics and exposure settings of different cameras have a substantial effect on the color and brightness of the images. By integrating this metadata, the model is better equipped to comprehend and adapt to the shooting conditions of various cameras, resulting in more accurate color restoration, particularly under diverse lighting conditions. Furthermore, the metadata offers valuable contextual information regarding the camera and shooting environment, thereby facilitating more effective color correction and color recovery. As a result, the inclusion of this metadata substantially boosts the model's overall performance. However, when the network depth was increased, we observed a slight decrease in performance, which may be attributed to overfitting.

		Mean	Med.	Best 25%	Worst 25%
-	base	2.85	1.53	0.54	7.43
L=2	deep+A	2.69	1.50	0.54	6.93
L=2	deep+B	2.71	1.52	0.53	6.94
L=2	deep+C	2.66	1.51	0.53	6.86
L=2	deep+A+B+C	**2.56**	**1.47**	**0.52**	**6.52**
L=4	deep+A+B+C	2.65	1.50	0.56	6.82
L=6	deep+A+B+C	2.69	1.53	0.57	6.88

Table 3. Variants of IFFCC. 'deep' indicates the use of a network to learn a set of parameters, where 'L' represents the number of layers, and 'A', 'B', 'C' correspond to different types of encoded additional information. The number of hidden neurons in each layer of the network is set to 4.

4. Temporal Color Constancy

Our proposed IFFCC extends FFCC by introducing an integral histogram, which significantly enhances performance in multi-illuminant scenarios. FFCC constructs a temporally coherent illuminant estimate using a probabilistic per-frame model, which generates a posterior distribution over illuminants parameterized as a bivariate Gaussian. IFFCC further refines this process by leveraging the integral histogram to capture spatially varying illuminants more effectively. This addition allows IFFCC to aggregate illumination information over regions of interest, improving robustness against complex lighting conditions. By integrating the Kalman filter with the integral histogram, IFFCC achieves superior temporal consistency and adaptability, making it particularly effective in dynamic, multi-light-source environments. We evaluated the performance of IFFCC on the temporal color constancy dataset [1], and the results are shown in Fig. 1.

5. More Visual

To further highlight the effectiveness of our method, we present additional visual comparison results in the supplementary materials. These results cover a range of challenging scenarios, including complex lighting conditions, diverse color distributions, and intricate spatial arrangements, as illustrated in Fig. 2, Fig. 3, and Fig. 4.

References

[1] Yanlin Qian, Jani Käpylä, Joni-Kristian Kämäräinen, Samu Koskinen, and Jiri Matas. A benchmark for burst color constancy. In *ECCVW*, 2020. 2

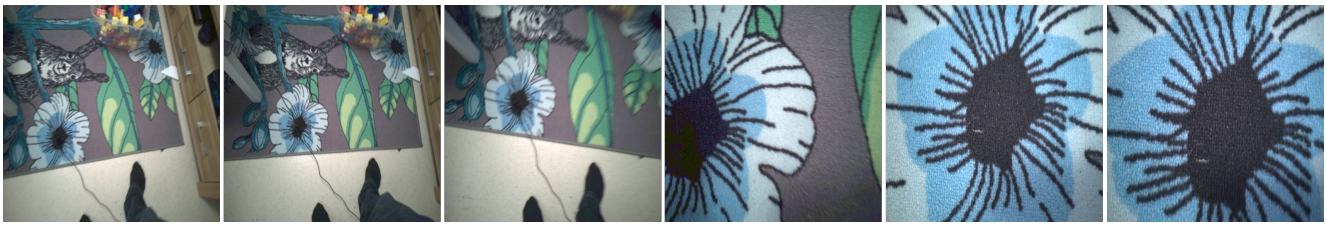

Figure 1. Performance of IFFCC on the Temporal Color Constancy (TCC) dataset. The images from left to right correspond to different frames from the same video sequence, demonstrating the model's ability to maintain temporal consistency in color constancy estimation.

Figure 2. More visual results on the LSMI dataset. (a) represents the input RAW image, (b) shows the illumination map and white-balanced image predicted by our model (both visualized after color correction and gamma correction for better clarity), and (c) depicts the ground-truth illumination map and white-balanced image (also visualized after color correction and gamma correction). The window size and overlap size for IFFCC are set to 32 and 16, respectively.

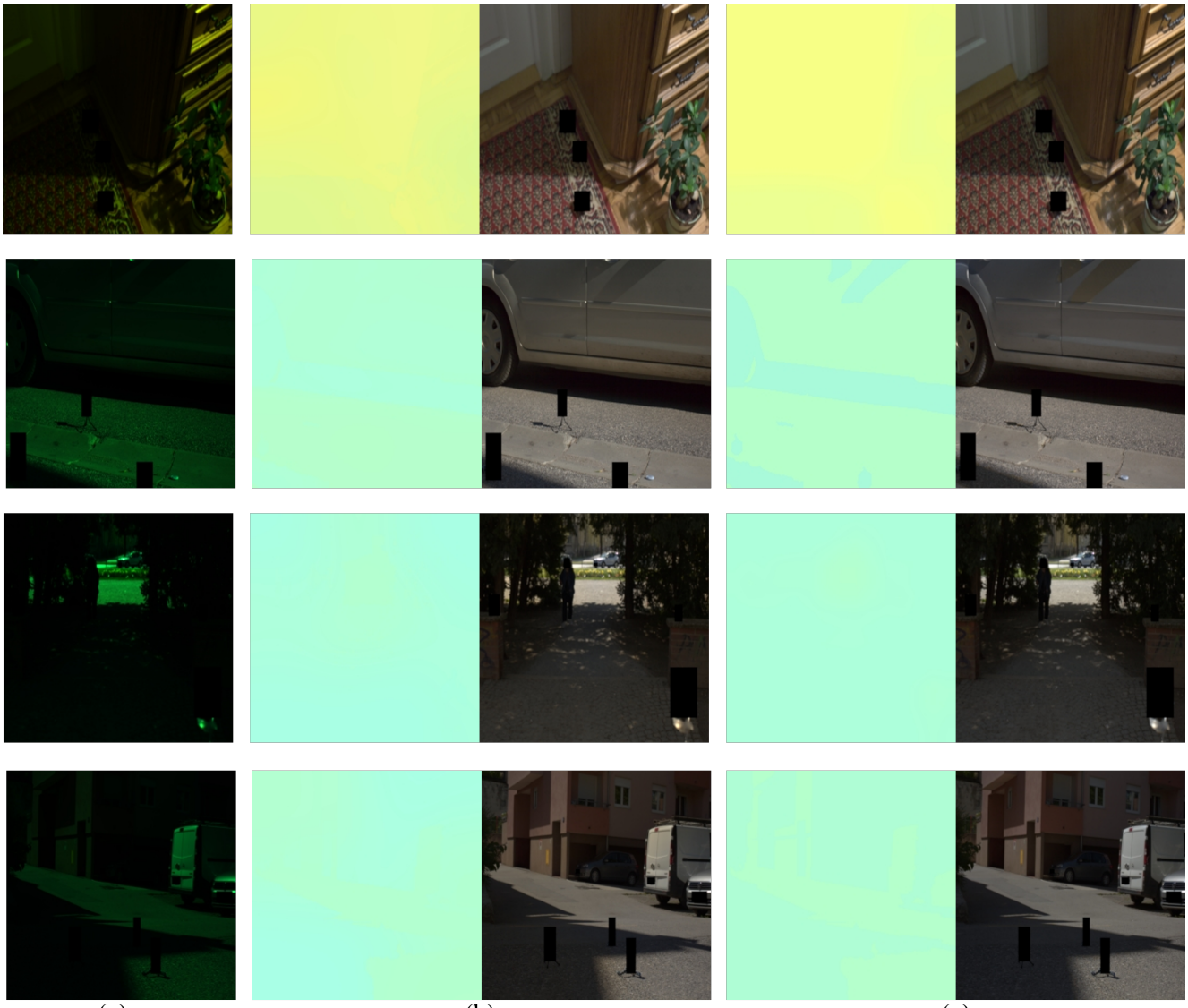

Figure 3. More visual results on the Shadow dataset. (a) represents the input RAW image, (b) shows the illumination map and white-balanced image predicted by our model (both visualized after color correction and gamma correction for better clarity), and (c) depicts the ground-truth illumination map and white-balanced image (also visualized after color correction and gamma correction). The window size and overlap size for IFFCC are set to 32 and 16, respectively.

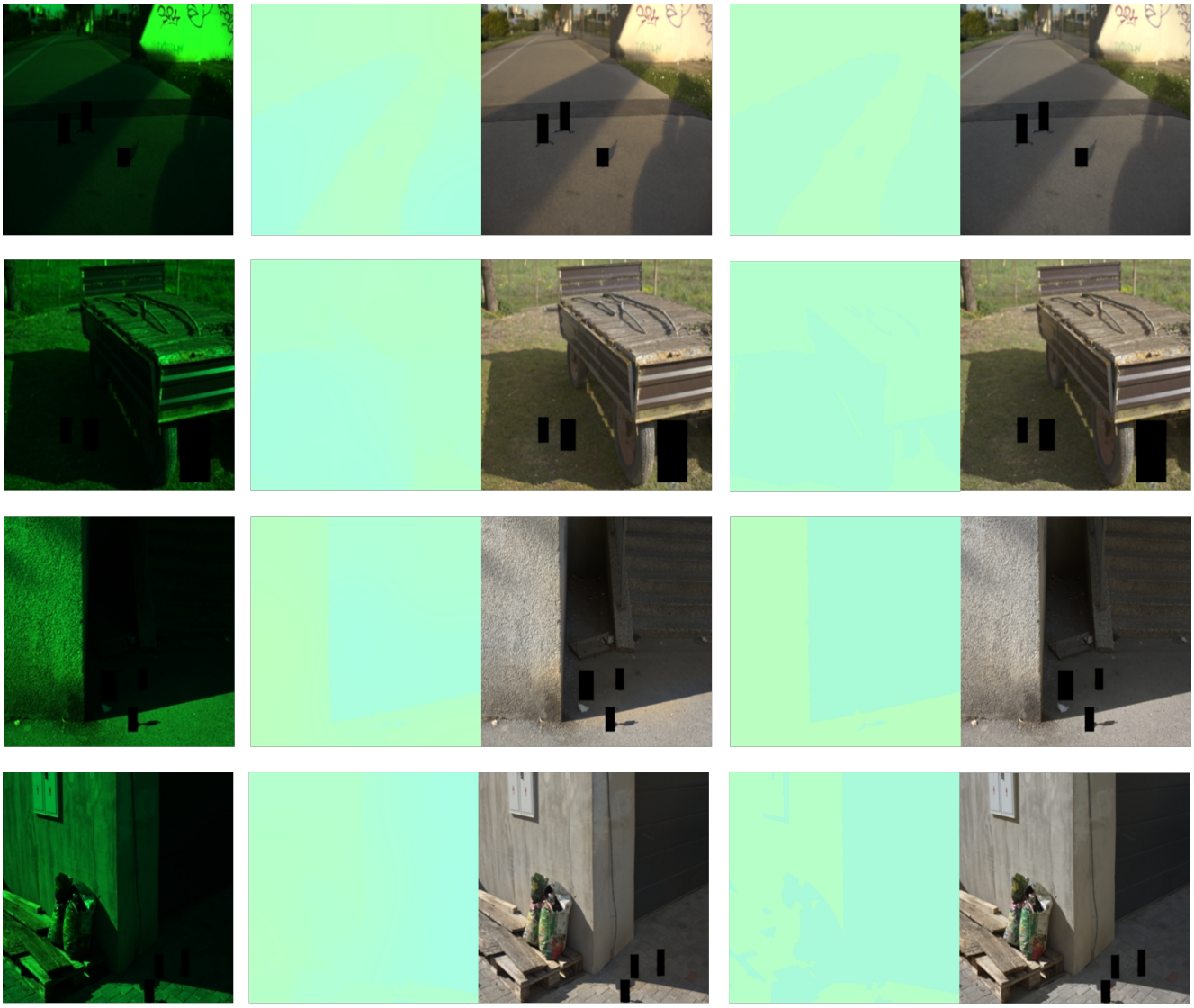

Figure 4. More visual results on the Shadow dataset. (a) represents the input RAW image, (b) shows the illumination map and white-balanced image predicted by our model (both visualized after color correction and gamma correction for better clarity), and (c) depicts the ground-truth illumination map and white-balanced image (also visualized after color correction and gamma correction). The window size and overlap size for IFFCC are set to 32 and 16, respectively.