Spatially Aware Metadata for Raw Reconstruction

Abhijith Punnappurath   Michael S. Brown

Samsung AI Center – Toronto

{abhijith.p,michael.b1}@samsung.com

Abstract

A camera sensor captures a raw-RGB image that is then processed to a standard RGB (sRGB) image through a series of onboard operations performed by the camera's image signal processor (ISP). Among these processing steps, local tone mapping is one of the most important operations used to enhance the overall appearance of the final rendered sRGB image. For certain applications, it is often desirable to de-render or unprocess the sRGB image back to its original raw-RGB values. This "raw reconstruction" is a challenging task because many of the operations performed by the ISP, including local tone mapping, are nonlinear and difficult to invert. Existing raw reconstruction methods that store specialized metadata at capture time to enable raw recovery ignore local tone mapping and assume that a global transformation exists between the raw-RGB and sRGB color spaces. In this work, we advocate a spatially aware *metadata-based raw reconstruction method that is robust to local tone mapping, and yields significantly higher raw reconstruction accuracy (6 dB average PSNR improvement) compared to existing raw reconstruction methods. Our method requires only 0.2% samples of the full-sized image as metadata, has negligible computational overhead at capture time, and can be easily integrated into modern ISPs.*

1. Introduction

Cameras employ dedicated hardware in the form of an image signal processor (ISP) to convert the scene-referred raw-RGB response recorded by the sensor to the final display-referred output of the camera, which is typically a standard RGB (sRGB) image. The ISP applies a series of operations that include white-balance, chromatic adaptation, and photo-finishing color manipulations, on the sensor's linear raw-RGB image to produce the non-linear sRGB image. One of the key photo-finishing routines applied by the ISP is a local tone mapping operation. Unlike a global tone mapping operator that modifies a pixel's intensity the same regardless of its spatial location, local tone
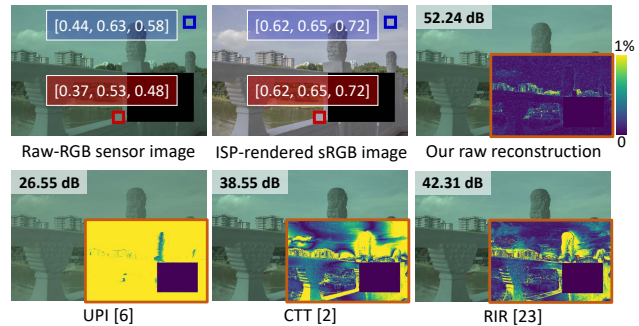

Figure 1. Two spatially separated pixels (marked red and blue) having very different raw-RGB sensor responses are mapped to the same output sRGB intensity value due to the local tone mapping operation applied by the camera's ISP. Existing raw reconstruction methods, such as [6, 2, 23], assume a global transformation from sRGB to raw-RGB, and do not model local tone mapping. Our proposed raw reconstruction approach considers not just the pixel's intensity value but also its spatial position, and thereby produces a more accurate result. PSNR (dB) values of the raw reconstruction, and the corresponding error maps are shown as insets.

operators dynamically modify the tone mapping operator to enhance contrast in a spatially varying manner. Local tone mapping has a significant impact on the rendered sRGB image's final appearance. Local tone mapping operators may even vary per scene-category (e.g., food, indoor, portrait) to impart a particular aesthetic to the output sRGB image.

Most modern cameras provide users the ability to shoot in the raw-RGB format. However, raw images are not suited for sharing and viewing – raw files are typically 4–6 times larger than sRGB encoded images, and they need to be rendered to a display-referred color space before viewing. Therefore, the vast majority of captured images are saved in a display-referred format, such as sRGB. However, many computer vision problems, such as image deblurring, color constancy, and photometric stereo, that assume a linear relationship between scene radiance and pixel intensity are more effective when applied to raw-RGB data [24]. Recent deep learning methods for image denoising [29] and

high dynamic range reconstruction [20] have also demonstrated the advantages of raw data and incorporating knowledge of the camera ISP stages into their training framework. Additionally, the raw-RGB format is very useful for photographic applications. Using photo-editing software such as Adobe Photoshop or Luminar, users can render the raw-RGB image to their own preferred rendering styles. As a result, methods for recovering the raw-RGB sensor image from the rendered sRGB image are an active area of research [6, 17, 28, 23, 2].

De-rendering an sRGB image to its original raw-RGB values is a challenging task. This is because many ISP operations, including local tone mapping, are non-linear and hard to invert. Moreover, most ISPs are 'blackboxes' to the end user, and as such, proprietary color manipulations applied by these ISPs cannot be accurately captured by generic models, such as in [6, 17]. A few works [28, 23, 2] have advocated saving specialized metadata at capture time along with the sRGB file to facilitate more accurate raw recovery. However, these existing approaches have several limitations. The image upsampling method of [28] adds considerable overhead in the form of a 1.5–6 MB downsampled raw file saved as metadata along with the sRGB file. The algorithm of [23] imposes a significant computational burden at capture time from the need to estimate a complex mapping function using a full-sized raw-sRGB image pair. The approach of [2] entails substantial changes to ISP hardware since a single capture requires multiple passes through the ISP. More importantly, methods such as [6, 23, 2] assume a global mapping from sRGB to raw-RGB color spaces, and ignore the effects of local tone mapping. In this paper, we propose a metadata-based raw reconstruction method that addresses these above drawbacks.

Contributions We propose an approach to sparsely sample the demosaiced raw-RGB image at capture time on a known uniform grid and store these raw-RGB samples as metadata along with the sRGB image. To undo the ISP processing post capture, we sample the output sRGB image on the same spatial grid, extract the raw-RGB samples from the metadata, and use these corresponding raw-sRGB samples to compute a mapping function from sRGB to raw-RGB that explicitly accounts for the spatial location of the samples. Our *spatially aware* sRGB to raw-RGB mapping is robust to local tone mapping and produces significantly higher raw reconstruction accuracy compared to existing raw reconstruction methods. An example is shown in Fig. 1. Our raw-RGB metadata, at 0.2% samples of the full-sized image, adds only a nominal storage overhead. The additional computational cost at capture time is minimal since no mapping function needs to be estimated; the raw samples themselves constitute the metadata. Moreover, our approach can be easily integrated into current ISP hardware since only a straightforward sampling operation on the raw-RGB grid is

involved. We also demonstrate the utility of our approach for various computer vision and photographic editing tasks, such as white balance manipulation, exposure correction, and motion deblurring.

2. Related work

We discuss works on raw reconstruction as well as the closely connected topic of radiometric calibration. Raw reconstruction algorithms can be classified into those that leverage specialized metadata embedded at capture time inside the sRGB file, and blind approaches that require no additional metadata. The metadata-based approaches are more closely connected to our work, and we examine them in greater detail below.

While most present-day cameras provide users the flexibility to save the raw-RGB sensor image, access was restricted to only the ISP-rendered sRGB image on older-generation devices. Thus, preliminary work was targeted simply at linearizing the sRGB data, and not on recovering the actual raw-RGB image. This process of linearization, called 'radiometric calibration', employs a camera response function to model the relation between the output pixel intensities and the amount of light falling on the sensor. Traditional radiometric calibration methods typically employ a single response function per color channel [10, 14, 21], which is inadequate to model the many non-linear stages of the camera ISP.

With modern cameras providing access to the raw sensor image, the goal shifted from radiometric calibration to actual raw reconstruction i.e., recovering the original raw-RGB sensor data instead of merely linearizing the sRGB values. However, just as with radiometric calibration, raw reconstruction methods [8, 7, 16, 13] also usually require cumbersome calibration procedures to be repeated per camera and even per camera setting. Recent deep learning methods for raw reconstruction (e.g., [22, 20]) too suffer from similar drawbacks since the trained models are camera specific, and sufficient training data has to be available for each camera. Methods such as [6, 17] that assume a standard set of ISP operations cannot model proprietary routines applied by individual ISPs. Moreover, these techniques assume that the target camera parameters (e.g., white balance and color correction matrices) are known a priori.

Raw reconstruction with metadata A few recent works [28, 23, 2] have advocated storing extra metadata along with the sRGB image at capture time. This additional information facilitates more accurate raw reconstruction than the blind approaches discussed thus far. Yuan and Sun [28] store a quarter- or half-sized small raw image along with the full-resolution sRGB image, and perform a guided upsampling of this lower-resolution raw image to recover the full-resolution raw image. However, the small raw file is still 1.5–6 MB, limiting the method's practical utility.

Raw-RGB sensor image

ISP hardware

ISP-rendered sRGB image

Sample the raw-RGB sensor image on a sparse uniform grid

96 KB metadata

Save these raw-RGB samples as additional metadata along with the sRGB image

Sample the sRGB image on the same grid

From the correspondence between the sRGB and raw-RGB values, a de-rendering function $f_{\{R,G,B,X,Y\}}$ can be estimated

Extract the metadata saved in the sRGB image to obtain the raw-RGB samples

$f_{\{R,G,B,X,Y\}}$

Applying the function $f_{\{R,G,B,X,Y\}}$ to the image un-processes the sRGB values back to the raw-RGB values

sRGB image
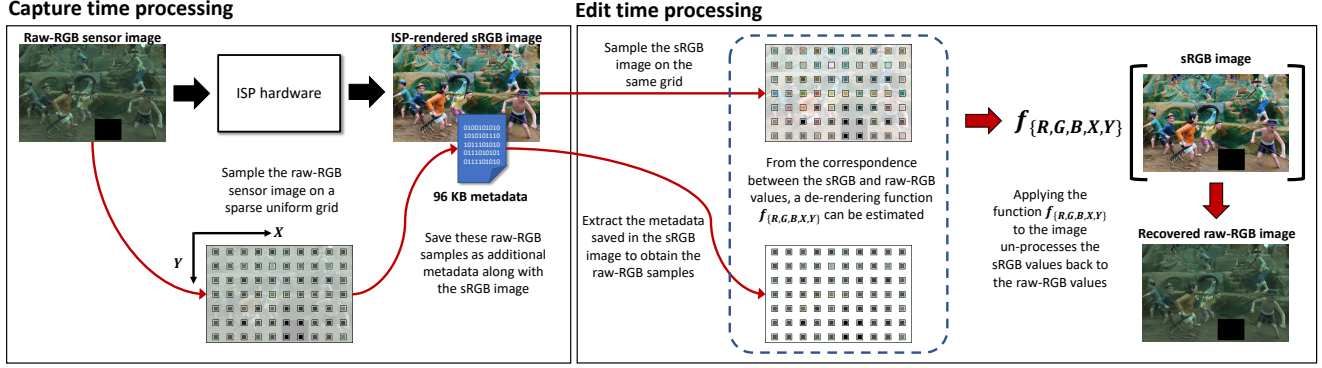
Recovered raw-RGB image

Figure 2. An overview of our proposed method. At capture time, we sample the demosaiced raw-RGB sensor image on a sparse uniform grid. The size of the pixels is exaggerated in the figure for visualization purposes. These raw-RGB samples are then stored as additional metadata along with the ISP-rendered sRGB image. To edit post-capture, we first recover the raw-RGB samples from the saved metadata. We also sample the sRGB image at the same spatial locations. From these corresponding raw-sRGB samples, we estimate a de-rendering function $\mathbf{f}_{\{R,G,B,X,Y\}}$ that considers the RGB intensities as well as their spatial locations (X, Y). This lends our method robustness to the local tone mapping operation applied by the ISP. Applying $\mathbf{f}_{\{R,G,B,X,Y\}}$ on the sRGB image produces our estimated raw-RGB image.

Nguyen and Brown [23, 24] estimate and store a set of raw reconstruction parameters at capture time from the given raw-sRGB image pair. These parameters model the typical operations on an ISP, such as white-balance, color space transformation, global tone mapping, and gamut mapping. But their optimization algorithm applied on full-resolution images has high computational cost and is hard to implement on-device in real time.

Recently, Afifi et al. [2] have proposed a multi-pass imaging framework where a downsampled demosaiced raw-RGB image is passed through the ISP multiple times with different ISP settings. Mapping functions are then computed between each different rendering and a down-sampled version of the output sRGB image. The authors demonstrate that these mapping function parameters, stored as metadata at capture time, allow accurate raw recovery or post-capture image manipulations. However, most ISPs do not support multiple passes, and their proposed pipeline demands substantial changes to existing design.

It is important to note that [23, 2] assume a global mapping from sRGB to raw-RGB and ignore the effects of local tone mapping. On the other hand, we employ a *spatially aware* de-rendering function that makes our method robust to local tone mapping. And in contrast to [28, 23, 2], our proposed method requires only 0.2% samples of the full-sized image as metadata, adds minimal computational overhead at capture time, and is easily integrable to existing ISP design, as explained next.

3. Method

On a typical ISP, demosaicing is one of the very first operations after the raw sensor response is digitized and the black light and lens distortions have been compensated for.

The demosaicing step interpolates the single-channel Bayer raw to three full-resolution RGB channels. We propose a simple modification to the ISP whereby this demosaiced raw-RGB image is sampled sparsely (0.2% samples of the full-sized image) on a uniform grid, and these raw-RGB samples are held in memory. The demosaiced raw-RGB image then passes through the rest of the pipeline, and undergoes the usual steps of white balance, chromatic adaptation, and photo-finishing color manipulations to produce the display-referred sRGB output of the camera. The raw-RGB samples are now saved as metadata along with the sRGB image. This extra metadata adds only a minimal overhead of less than 96 KB to the full-resolution sRGB image, which on modern cameras is usually 12–20 megapixels and several MB in size. Note also that the additional computational cost at capture time is nominal since only a straightforward sampling operation is involved.

To recover the raw sensor image post capture, we first extract the metadata saved along with the sRGB image to obtain the raw-RGB samples. Next, we sample the sRGB image on the same grid. We then use these corresponding raw-sRGB samples to estimate a de-rendering function. We use a scatter point interpolant as our mapping function from sRGB to raw-RGB. We construct the interpolant to consider the RGB intensity value as well as the spatial location (X, Y) i.e., five dimensions. Radial basis functions (RBF) can be used to interpolate scattered data in higher dimensions [5]. Usually, the interpolation comprises a sum of weighted radial interactions and a polynomial correction. The weights of the radial function and the coefficients of the polynomial, which constitute the unknowns, can be computed by solving a linear system built from known samples. Once these unknown parameters are determined, we

can query the interpolating function, which is expressed as the weighted summation of the radial contribution of each source sample and of the polynomial term. We discuss this approach formally below.

Let $\mathbf{s}_i = (s_{R_i}, s_{G_i}, s_{B_i}, s_{X_i}, s_{Y_i})$ with $1 \leq i \leq N$ denote the ith sRGB sample, where the first three coordinates represent the RGB pixel intensity values and the last two coordinates represent the X and Y spatial position. Here N denotes the total number of samples. Let us introduce

$$\mathbf{M} = [M_{ij}]_{N \times N} = \varphi\left(\|\mathbf{s}_i - \mathbf{s}_j\|_{\mathbb{R}^5}\right), \qquad (1)$$

where φ denotes the radial function. Let us also define

$$\mathbf{P} = \begin{pmatrix} 1 & s_{R_1} & s_{G_1} & s_{B_1} & s_{X_1} & s_{Y_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & s_{R_i} & s_{G_i} & s_{B_i} & s_{X_i} & s_{Y_i} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & s_{R_N} & s_{G_N} & s_{B_N} & s_{X_N} & s_{Y_N} \end{pmatrix}, \qquad (2)$$

$$\boldsymbol{\gamma}_C = \begin{pmatrix} \gamma_{C_1} \\ \vdots \\ \gamma_{C_i} \\ \vdots \\ \gamma_{C_N} \end{pmatrix}, \quad \mathbf{r}_C = \begin{pmatrix} r_{C_1} \\ \vdots \\ r_{C_i} \\ \vdots \\ r_{C_N} \end{pmatrix}, \quad \boldsymbol{\beta}_C = \begin{pmatrix} 1 \\ \beta_{C_R} \\ \beta_{C_G} \\ \beta_{C_B} \\ \beta_{C_X} \\ \beta_{C_Y} \end{pmatrix}, \quad (3)$$

where C can be one of $\{R, G, B\}$ color channels, γ represents the weights of the radial function φ, $\mathbf{r}$ denotes the raw-RGB samples stored as metadata, and β corresponds to the coefficients of the linear polynomial function.

Solving the following linear system,

$$\begin{pmatrix} \mathbf{M} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\gamma}_C \\ \boldsymbol{\beta}_C \end{pmatrix} = \begin{pmatrix} \mathbf{r}_C \\ \mathbf{0} \end{pmatrix}, \qquad (4)$$

gives us our required interpolating function $\mathbf{f}_{C_{\{R,G,B,X,Y\}}} = \begin{pmatrix} \boldsymbol{\gamma}_C \\ \boldsymbol{\beta}_C \end{pmatrix}$. Given a set of Q query points $\mathbf{s}_i^* = (s_{R_i}^*, s_{G_i}^*, s_{B_i}^*, s_{X_i}^*, s_{Y_i}^*)$ with $1 \leq i \leq Q$, which in our case is the full-sized output sRGB image, we first construct the matrices

$$\mathbf{M}^* = [M_{ij}^*]_{Q \times N} = \varphi\left(\|\mathbf{s}_i^* - \mathbf{s}_j\|_{\mathbb{R}^5}\right), \qquad (5)$$

and

$$\mathbf{P}^* = \begin{pmatrix} 1 & s_{R_1}^* & s_{G_1}^* & s_{B_1}^* & s_{X_1}^* & s_{Y_1}^* \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & s_{R_i}^* & s_{G_i}^* & s_{B_i}^* & s_{X_i}^* & s_{Y_i}^* \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & s_{R_Q}^* & s_{G_Q}^* & s_{B_Q}^* & s_{X_Q}^* & s_{Y_Q}^* \end{pmatrix}, \quad (6)$$

and finally perform the matrix-vector product

$$\mathbf{r}_C^* = \begin{pmatrix} \mathbf{M}^* & \mathbf{P}^* \end{pmatrix} \begin{pmatrix} \boldsymbol{\gamma}_C \\ \boldsymbol{\beta}_C \end{pmatrix}, \qquad (7)$$

to obtain the reconstructed raw-RGB values $\mathbf{r}_C^*$. An overview of our method is provided in Fig. 2.

One disadvantage of RBF interpolation is that the method is slow if N increases beyond, say, a few hundred samples [3]. To overcome this limitation, we adopt a patch-wise approach. This has the advantage of eliminating radial interactions outside a local spatial neighbourhood. Furthermore, it makes our method amenable to parallelization (say, on a GPU) since each patch is independent. In particular, we query patches of size 100×100 pixels, and use samples drawn only from a 500×500 pixel neighbourhood around this patch while computing the interpolation function. We use a linear RBF $\varphi(z) = \epsilon z$, with $\epsilon = \frac{1}{2.5}$. Please see the supplementary material for ablation studies on type of RBF and number of samples.

4. Experiments

The first set of experiments, in Section 4.1, is designed to evaluate our proposed method's raw reconstruction performance. In Section 4.2, we assess the accuracy of our approach in sRGB space by re-rendering our estimated raw-RGB image through an ISP. The remaining set of experiments are designed to demonstrate the utility of our approach when applied to various computer vision and photographic editing tasks, such as white balance manipulation (Section 4.3), exposure correction (Section 4.4), and motion deblurring (Section 4.5). Ablation studies are presented in Section 4.6. We use images from the NUS dataset [9] for all our experiments.

4.1. Raw reconstruction

We first evaluate our method's effectiveness in recovering the raw-RGB image. We test on seven cameras from the NUS dataset [9] – Canon 1Ds MkIII, Canon 600D, Nikon D40, Nikon D5200, Olympus E-PL6, Panasonic GX1, and Sony SLT-A57 – for a total of 1455 images. The NUS dataset includes both sRGB images and their corresponding raw files. Note that the sRGB JPEG-compressed images in the dataset have been produced by each individual camera's hardware ISP. To evaluate our method, we also require the demosaiced raw-RGB image corresponding to each sRGB image. As already mentioned, the hardware ISPs of these cameras are proprietary and inaccessible. Therefore, to obtain the demosaiced raw-RGB image, we use the software ISP platform of Karaimer and Brown [15], which allows access to the various stages of the camera pipeline. In particular, we render the raw files from the NUS dataset through their pipeline, and use the images obtained after the demosaicing stage of their software ISP as the ground truth raw-RGB images, for all our experiments.

We apply our proposed framework of Section 3 to compute our raw reconstruction. Before applying our method, the color charts are masked out in both the raw-RGB im-
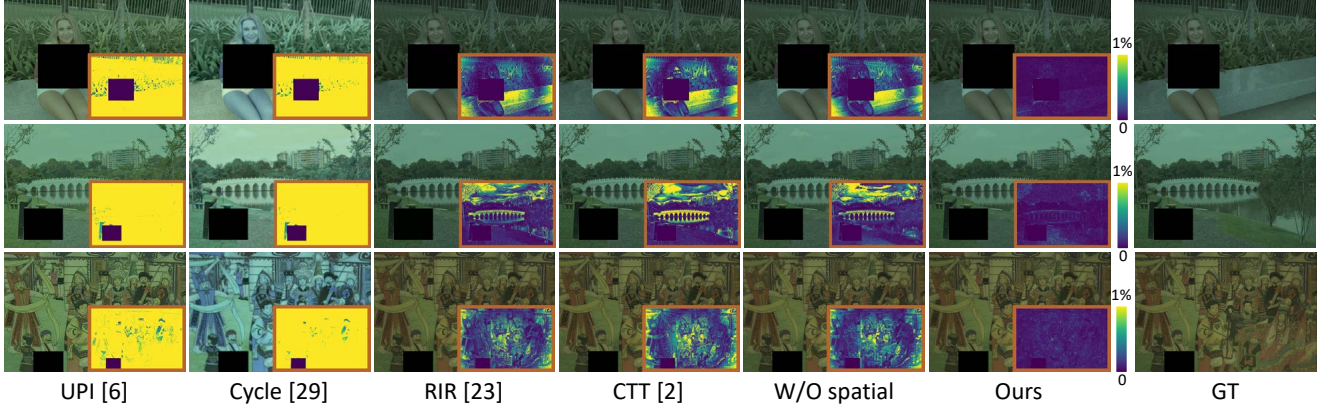
Figure 3. Raw reconstruction. Results of the competing approaches of [6, 29, 23, 2] are presented. A variant of our method, denoted 'W/O spatial', that constructs a global mapping function by considering only pixel RGB values and ignoring spatial information, is also provided for comparison. Our results and the ground truth (GT) are shown in the last two columns, respectively. The corresponding per-pixel error maps are shown as insets. Note that a gamma function has been applied to all raw-RGB images for better visualization.

age and the sRGB image. The PSNR (dB) values between our reconstructed raw image and the ground-truth (GT) raw image is presented in Table 1. We provide comparisons against the blind (i.e., without metadata) raw reconstruction methods UPI [6] and Cycle [29]. We also compare against the metadata-based raw recovery approaches RIR [23] and CTT [2]. We also include a variation of our framework, denoted 'W/O spatial', where a global mapping function is computed from the raw-sRGB samples using only the pixel's intensity values, and spatial information is discarded. Note that the deep learning approach Cycle [29] cannot process the full-resolution image. Hence, we provided downsampled images as input to this method. For fairness of comparison, the PSNR values of all competing algorithms, including our own method, are also computed on this downsampled resolution. It can be observed that our proposed approach outperforms our closest competitor CTT [2] by a substantial margin, approximately 6 dB mean PSNR. Our proposed spatially aware algorithm also performs significantly better than the W/O spatial variant (see, in particular, the worst 25% results). Another observation is that the metadata-based approaches, as expected,

perform better than their blind counterparts. A few representative examples of our raw reconstruction, along with comparisons, are provided in Fig. 3. The corresponding per-pixel errors maps are also displayed in the figure.

4.2. Re-rendering to sRGB

As discussed in the introduction, the raw-RGB format is not suitable for viewing, and it is the display-referred image that is used in most consumer applications. Therefore, we evaluate our method's performance by rendering our raw reconstruction through an ISP, and comparing our re-rendered result with the ground truth in the sRGB space.

We test on three cameras – Samsung NX2000, Olympus E-PL6, and Sony SLT-A57 – from the NUS dataset [9]. The total number of images used in this experiment is 678. The demosaiced raw-RGB ground truth images are obtained following the same procedure as in Section 4.1. Our objective is to process our raw reconstruction through the ISP to the sRGB space; however, the hardware ISPs corresponding to these cameras are not accessible. Therefore, we render our estimated raw-RGB image to sRGB using a software ISP. In particular, we selected Adobe Photoshop as the software ISP for this experiment. To ensure that the ISP operations are consistent between our re-rendered sRGB output and the ground truth sRGB images used for evaluation and as input, we render the raw files from the NUS dataset to sRGB using Photoshop. We treat these sRGB outputs as the ground truth, and not the camera-rendered sRGB images as in Section 4.1.

We now apply our proposed framework on the demosaiced raw-RGB and sRGB pair to obtain our raw reconstruction. We then save our estimated raw reconstruction as a raw-DNG file, and render it to sRGB using Photoshop. The PSNR and structural similarity index mea-

Table 1. Quantitative raw recovery results on seven cameras from the NUS dataset [9]. The PSNR (dB) of the estimated raw-RGB result is shown. Best results are provided in bold.

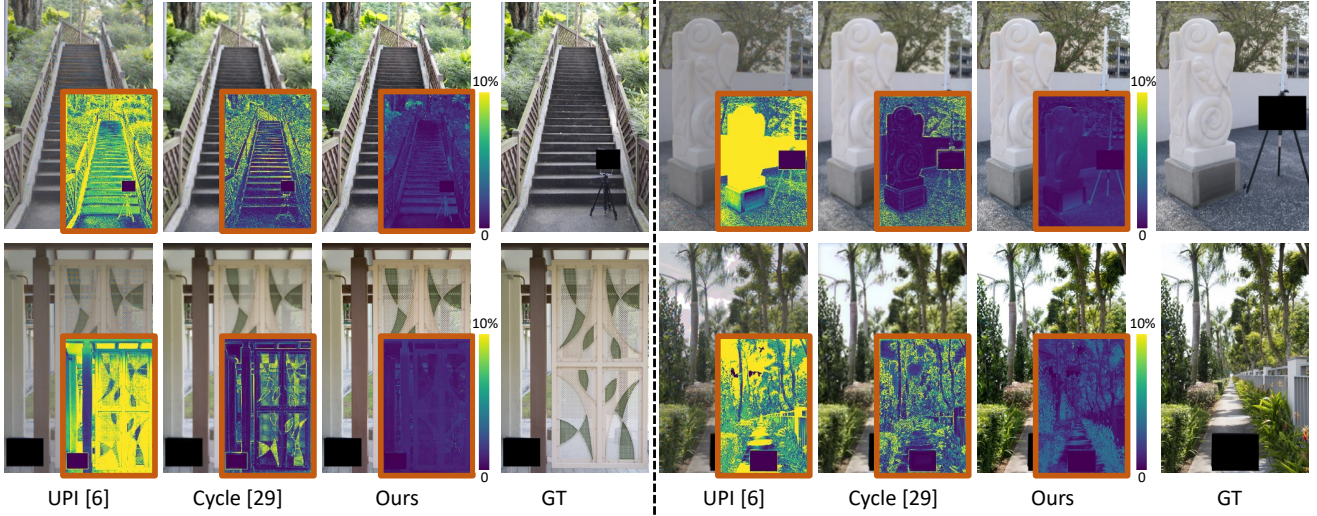| Method	PSNR (dB)			
	Mean	Median	Worst 25%	Best 25%
UPI [6]	26.00	25.83	21.33	31.03
Cycle [29]	17.33	17.06	13.66	21.41
RIR [23]	44.14	45.33	27.93	58.86
CTT [2]	45.04	44.39	33.58	57.79
W/O spatial	47.87	48.28	34.87	60.14
Ours	**51.23**	**51.03**	**42.60**	**60.47**

Figure 4. Re-rendering to sRGB. Results of [6, 29] and our method are provided. The ground truth is presented in the last column. The corresponding per-pixel error maps are shown as insets.

Table 2. Quantitative sRGB re-rendering results on three cameras from the NUS dataset [9]. The PSNR (dB) and SSIM of the estimated sRGB result are shown.

Method	PSNR (dB) / SSIM			
	Mean	Median	Worst 25%	Best 25%
UPI [6]	20.62	20.69	17.96	23.25
	0.856	0.856	0.797	0.913
Cycle [29]	27.91	27.61	24.57	31.81
	0.867	0.870	0.787	0.939
Ours	**31.12**	**31.03**	**27.48**	**35.08**
	0.973	**0.986**	**0.928**	**0.995**

sure (SSIM) [27] values computed between our re-rendered sRGB results and the ground-truth sRGB images are shown in Table 2. Comparisons have been provided against the methods of [6, 29] because only these two methods model the ISP in both the forward and reverse directions. It can be observed that our method offers significantly more accurate results. A few qualitative examples of our re-rendered sRGB results, along with comparisons, are shown in Fig. 4.

4.3. White balance manipulation

Next, we apply our proposed method to the task of white balance (WB) manipulation. We use the same three cameras from the NUS dataset as in our earlier experiment in Section 4.2. However, instead of rendering the raw files to sRGB with the "as shot" parameters, we render out to a different WB setting. Specifically, we use Photoshop to generate sRGB images under Tungsten WB (2850 K) and Shade WB (7500 K). We now apply our method to the demosaiced raw-RGB images and these differently white-balanced sRGB images. As before, we save our estimated

raw-RGB image as a raw-DNG file. The target is to evaluate our method's accuracy in re-rendering to a different WB from our raw reconstruction. Towards this goal, we render our estimated raw-DNG to the Daylight WB (5500 K) setting using Photoshop. The ground truth is obtained by rendering the original raw files in the dataset to Daylight WB using Photoshop. To quantitatively evaluate performance, we adopt four commonly used error metrics: (i) mean squared error (MSE), (ii) mean angular error (MAE), (iii) Δ E 2000 [25], and (iv) Δ E 76; the results are provided in Table 3. We compare our results against a classic diagonal WB manipulation performed on the sRGB image using an *exact achromatic* reference point obtained from the color chart placed in the scene. The diagonal WB manipulation is performed both with and without the commonly used pre-linearization step using a 2.2 gamma [4, 11]. We also compare against the CTT method of [2] that renders out multiple tiny images through the ISP under different WB settings to enable post-capture WB manipulation in the sRGB space. The target WB value of 5500 K is one of their pre-selected color temperatures for which a direct mapping function was estimated at capture time. It can be observed from the results in Table 3 that our method outperforms both diagonal WB manipulation and CTT [2] on all four metrics.

In Fig. 5, we show results of auto WB correction. Photoshop's auto WB algorithm is used in all cases, except for the comparison with the CCW method of Afifi and Brown [1] shown in the third column. In the second column, Photoshop's WB correction is applied directly on the sRGB image, in the fourth column, on our reconstructed raw-DNG, and in the last column, on the ground-truth raw image from the NUS dataset. It can be observed that our rendered sRGB

Table 3. Quantitative WB manipulation results on three cameras from the NUS dataset [9]. Comparisons are provided against diagonal (Diag) WB, applied directly in sRGB (using the exact achromatic reference point), and on "linearized" sRGB [4, 11]. The result of the recent CTT [2] method is also shown. The terms MSE and MAE stand for mean squared error and mean angular error, respectively.

| Method	MSE				MAE				Δ E 2000				Δ E 76			
	Mean	Med	B25%	W25%	Mean	Med	B25%	W25%	Mean	Med	B25%	W25%	Mean	Med	B25%	W25%
Diag WB	0.541	0.194	0.033	1.503	5.653	3.883	1.716	11.794	7.057	4.142	2.182	14.088	10.575	6.975	3.376	20.821
Diag WB w linearization	0.475	0.174	0.030	1.325	5.444	3.844	1.652	11.399	6.686	3.949	2.086	13.419	9.935	6.677	3.214	19.443
CTT [2]	0.114	0.085	0.032	0.247	1.744	1.602	0.859	2.875	2.965	2.811	1.731	4.458	4.140	3.830	2.336	6.456
Ours	**0.014**	**0.007**	**0.003**	**0.039**	**1.304**	**0.876**	**0.493**	2.847	**1.419**	**1.057**	**0.805**	**2.718**	**1.824**	**1.235**	**0.938**	**3.754**
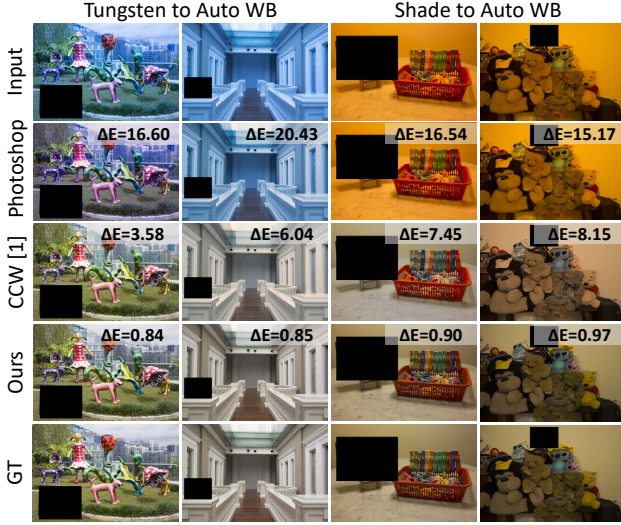
Figure 5. Auto white balance. We perform auto WB on images captured under Tungsten (2850 K) and Shade WB (7500 K). In the second row, Adobe Photoshop's auto correction of the input sRGB image of the first row is shown. The third row shows the result of the auto WB algorithm CCW of [1] applied on the sRGB input. The fourth row shows Photoshop's auto WB algorithm applied to our recovered raw-DNG. The ground truth shown in the last row is obtained by running Photoshop's auto WB routine on the ground-truth raw image from the NUS dataset [9]. ΔE 2000 values are provided in the insets.

output is visually a close match to the ground-truth sRGB. This is also verified quantitatively by the ΔE 2000 values provided in the figure insets.

4.4. Exposure correction

Next, we demonstrate our method's usefulness for exposure correction. The exposure setting used by the ISP at capture time directly affects the overall brightness of the rendered sRGB image. Exposure settings are represented by exposure values (EV). Exposure errors are a frequent occurrence in consumer photography and lead to dark image regions (underexposure error) or bright washed-out regions (overexposure error). Both types of errors reduce the contrast and visual appeal of the rendered image, and thus exposure correction is an important problem.

For this experiment, we choose the 202 images from the

Samsung NX2000 camera of the NUS dataset [9]. The NUS dataset was captured by a professional photographer with carefully selected exposure settings. Thus, we use the original raw files rendered through Photoshop with zero-gain relative EV as ground truth. This is equivalent to the original correct exposure setting applied onboard the camera at capture time. Next, we render the ground truth raw files with two different *incorrect* EV values using Photoshop – -2 EV to mimic underexposure errors and +1 EV to simulate overexposure errors. We apply our method to these incorrectly exposed input images, and render our reconstructed raw-DNG to sRGB under the zero-gain EV setting. We compare our results against the 'Auto tone' feature in Photoshop that allows automatic exposure correction. We also compare with the recent Retinex-based exposure correction method DIEC [30] that is designed to handle both underexposed and overexposed images. Both comparison methods work directly in the sRGB space. An example is shown in Fig. 6. It can be seen that our result more closely matches the ground truth for both under- and overexposure errors. Quantitative results averaged over all test images are provided in the supplementary material.
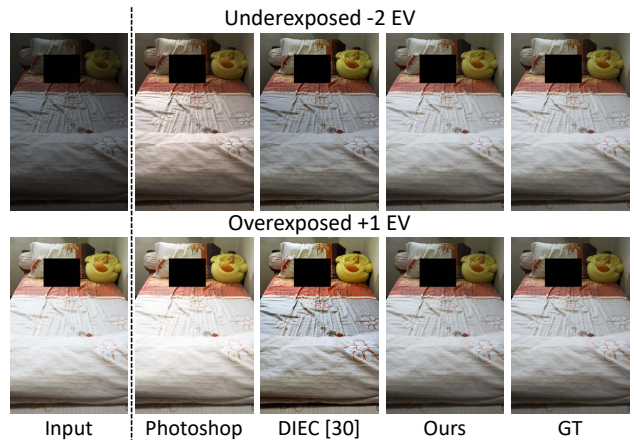

Figure 6. Exposure correction. The first row shows an example of underexposure error (-2 EV), while the second row illustrates an overexposure error (+1 EV). The results of Photoshop's 'Auto tone' algorithm and the retinex-based exposure correction method DIEC [30] are provided for comparison.
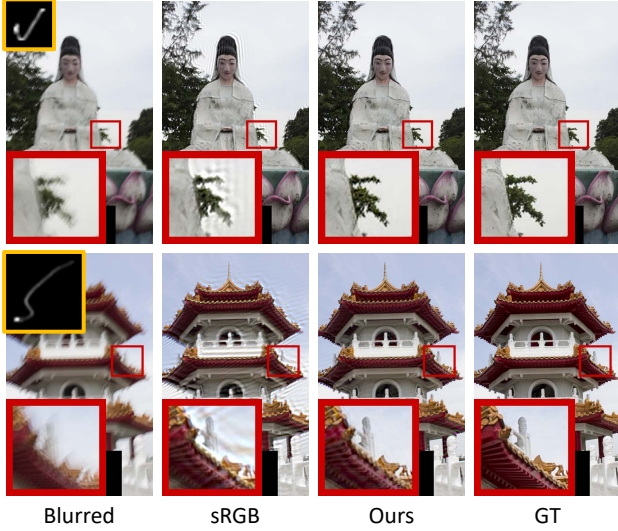
| Blurred | sRGB | Ours | GT |

Figure 7. Motion deblurring. The blurred input image and the corresponding motion blur kernel are shown in the first column. The second and third columns show results obtained by performing deblurring in sRGB space, and in raw-RGB space by applying the raw reconstruction procedure, respectively. The non-blind motion deblurring algorithm of Krishnan and Fergus [18] was used. The last column shows the ground-truth sharp image.

4.5. Motion deblurring

Traditionally, motion blur is formulated using a linear image formation model. The blurred image is expressed as the convolution between an underlying sharp image and a motion blur kernel [12]. However, this relationship does not hold for blurred sRGB images since linearity is broken by the non-linear processing stages of the ISP. Therefore, it is advantageous to deblur the image in a linear scene-referred space [26]. To demonstrate our method's usefulness for the task of motion deblurring, we select the statistics-based non-blind motion deblurring method of Krishnan and Fergus [18] and compare its performance in sRGB space, and in raw-RGB space by applying our raw reconstruction procedure. We select the Samsung NX2000 images used previously in Section 4.4 for this experiment too. To generate motion blurred sRGB images, we blur each ground-truth demosaiced raw-RGB image with a kernel selected randomly from one of the four kernels in the widely used motion blur benchmark dataset of Lai et al. [19]. The blurred raw-RGB images are then rendered to sRGB. We run our raw reconstruction algorithm on this blurred data, perform deblurring on our recovered raw image, and then render this deblurred raw estimate to sRGB. Fig. 7 shows two representative deblurring results. It can be clearly observed from the zoomed-in regions that directly deblurring the sRGB image produces a lot of ringing artifacts. In comparison, our results are sharper and less prone to ringing. Deblurring
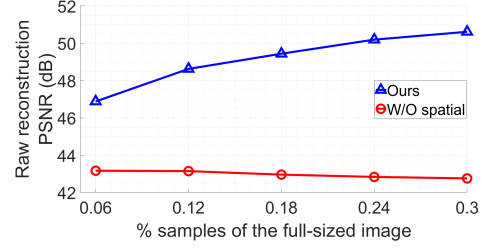

Figure 8. Sampling density versus raw reconstruction accuracy.

in the linear raw-RGB space produced an average PSNR of 27.10 dB, whereas sRGB yielded only 24.77 dB, which clearly demonstrates our proposed method's utility for the problem of motion deblurring. Additional quantitative results are provided in the supplementary material.

4.6. Ablation studies

The plot of Fig. 8 presents an ablation study of sampling density versus raw reconstruction accuracy. The 200 images from the Canon 600D camera were used for this experiment. It can be observed that beyond roughly 0.2% sampling density, the improvement in PSNR is marginal. For all our experiments in this paper, we used less than approximately 0.2% samples for an overhead of less than 96 KB, as a compromise between accuracy and metadata size. Observe that the performance of the W/O spatial variant does not improve with more samples.

We also experimented with different RBFs, such as a cubic spline $\varphi(z) = (\epsilon z)^3$ and a thin plate spline $\varphi(z) = (\epsilon z)^2 \log(\epsilon z)$. On the Canon 600D test set, we obtained average raw reconstruction PSNR values of 46.96 dB using a cubic spline, 48.64 dB with a thin plate spline, and 49.44 dB using a linear RBF, which indicates that a linear RBF works well for the task at hand. All results reported in this work are using a linear RBF $\varphi(z) = \epsilon z$.

5. Conclusion

We presented a spatially aware raw reconstruction method that can be easily integrated into existing ISPs with minimal hardware modifications. We demonstrated how our simple framework is more accurate than existing raw reconstruction methods, and also showcased its utility on a number of computer vision and photographic applications. Our method's significantly improved performance over prior methods is attributed to a more robust modeling of the local tone mapping operation performed onboard the ISP. Incorporating spatial information allows us to circumvent the one-to-many ambiguity faced by existing methods that arises when estimating the reverse mapping from sRGB to raw-RGB. Our approach also has the advantage of low computational overhead at capture time, and requires less than 96 KB of additional metadata.

References

[1] Mahmoud Afifi, Brian Price, Scott Cohen, and Michael S. Brown. When color constancy goes wrong: Correcting improperly white-balanced images. In *Computer Vision and Pattern Recognition*, 2019.

[2] Mahmoud Afifi, Abhijith Punnappurath, Abdelrahman Abdelhamed, Hakki Can Karaimer, Abdullah Abuolaim, , and Michael S. Brown. Color temperature tuning: Allowing accurate post-capture white-balance editing. In *Color Imaging Conference*, 2019.

[3] Isaac Amidror. Scattered data interpolation methods for electronic imaging systems: A survey. *Journal of Electronic Imaging*, 11(2):157–176, 2002.

[4] Matthew Anderson, Ricardo Motta, Srinivasan Chandrasekar, and Michael Stokes. Proposal for a standard default color space for the internet - sRGB. In *Color Imaging Conference*, 1996.

[5] Marco Evangelos Biancolini. *Fast Radial Basis Functions for Engineering Applications*. Springer International Publishing, 2018.

[6] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T. Barron. Unprocessing images for learned raw denoising. In *Computer Vision and Pattern Recognition*, 2019.

[7] Ayan Chakrabarti, Daniel Scharstein, and Todd E. Zickler. An empirical camera model for internet color vision. In *British Machine Vision Conference*, 2009.

[8] Ayan Chakrabarti, Ying Xiong, Baochen Sun, Trevor Darrell, Daniel Scharstein, Todd Zickler, and Kate Saenko. Modeling radiometric uncertainty for vision with tone-mapped color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2185–2198, 2014.

[9] 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. *Journal of the Optical Society of America A*, 31(5):1049–1058, 2014.

[10] Paul E. Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH*, 2008.

[11] Marc Ebner. *Color Constancy*. Wiley Publishing, 1st edition, 2007.

[12] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph. In *ACM SIGGRAPH*, 2006.

[13] Han Gong, Graham D Finlayson, Maryam M Darrodi, and Robert B Fisher. Rank-based radiometric calibration. In *Color Imaging Conference*, 2018.

[14] Michael D. Grossberg and Shree K. Nayar. Determining the camera response from images: What is knowable? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1455–1467, 2003.

[15] Hakki Can Karaimer and Michael S. Brown. A software platform for manipulating the camera imaging pipeline. In *European Conference on Computer Vision*, 2016.

[16] Seon Joo Kim, Hai Ting Lin, Zheng Lu, Sabine Süsstrunk, Stephen Lin, and Michael S. Brown. A new in-camera imaging model for color computer vision and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(12):2289–2302, 2012.

[17] Samu Koskinen, Dan Yang, and Joni-Kristian Kämäräinen. Reverse imaging pipeline for raw RGB image augmentation. In *International Conference on Image Processing*, 2019.

[18] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-Laplacian priors. In *Advances in neural information processing systems*, 2009.

[19] Wei-Sheng Lai, Jia-Bin Huang, Zhe Hu, Narendra Ahuja, and Ming-Hsuan Yang. A comparative study for single image blind deblurring. In *Computer Vision and Pattern Recognition*, 2016.

[20] Yu-Lun Liu, Wei-Sheng Lai, Yu Sheng Chen, Yi-Lung Kao, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Single-image HDR reconstruction by learning to reverse the camera pipeline. In *Computer Vision and Pattern Recognition*, 2020.

[21] Tomoo Mitsunaga and Shree K. Nayar. Radiometric self calibration. In *Computer Vision and Pattern Recognition*, 1999.

[22] Seonghyeon Nam and Seon Joo Kim. Modelling the scene dependent imaging in cameras with a deep neural network. In *Internation Conference on Computer Vision*, 2017.

[23] Rang M. H. Nguyen and Michael S. Brown. RAW image reconstruction using a self-contained sRGB-JPEG image with only 64 KB overhead. In *Computer Vision and Pattern Recognition*, 2016.

[24] Rang M. H. Nguyen and Michael S. Brown. RAW image reconstruction using a self-contained sRGB–JPEG image with small memory overhead. *International Journal of Computer Vision*, 126(6):637–650, 2018.

[25] Gaurav Sharma, Wencheng Wu, and Edul N. Dalal. The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations. *Color Research & Application*, 30(1):21–30, 2005.

[26] Yu-Wing Tai, Xiaogang Chen, Sunyeong Kim, Seon Joo Kim, Feng Li, Jie Yang, Jingyi Yu, Yasuyuki Matsushita, and Michael S. Brown. Nonlinear camera response functions and image deblurring: Theoretical analysis and practice. *IEEE transactions on pattern analysis and machine intelligence*, 35(10):2498–2512, 2013.

[27] Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[28] Lu Yuan and Jian Sun. High quality image reconstruction from RAW and JPEG image pair. In *International Conference on Computer Vision*, 2011.

[29] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. CycleISP: Real image restoration via improved data synthesis. In *Computer Vision and Pattern Recognition*, 2020.

[30] Qing Zhang, Yongwei Nie, and Wei-Shi Zheng. Dual illumination estimation for robust exposure correction. *Computer Graphics Forum*, 38(7):243–252, 2019.