https://doi.org/10.2352/issn.2169-2629.2020.28.31

Camera ISP Modification to Enable Image De-rendering

Abhijith Punnappurath and Michael S. Brown
Samsung Research, AI Center, Toronto, Canada

Abstract

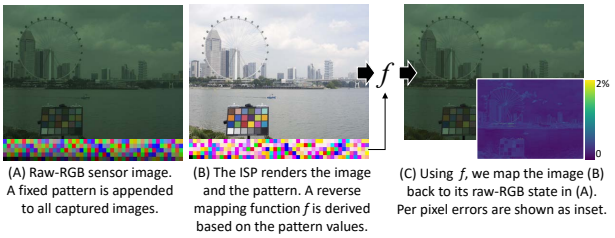
A camera's image signal processor (ISP) is dedicated hardware that performs a series of processing steps to render a captured raw sensor image to its final display-referred output suitable for viewing and sharing. It is often desirable to be able to revert – or de-render – the ISP-processed image back to the original raw sensor image. Undoing the ISP rendering, however, is not an easy task. This is because ISPs perform many nonlinear routines in the rendering pipeline that are difficult to invert. Moreover, modern cameras often apply scene-specific image processing, resulting in a wide range of possible ISP parameters. In this paper, we propose a modification to the ISP that allows the ISP-rendered image to be reverted back to a raw image. Our approach works by appending a fixed-sampling of the raw sensor values to all captured images. The appended raw samples comprise no more than 8 rows of pixels in the full-sized image and represent a negligible overhead given that 12–16 MP sensors typically have 3000 rows of pixels or more. The appended pixels are rendered along with the captured image to the final output. From these rendered raw samples, a reverse mapping function can be computed to undo the ISP processing. We demonstrate that this method performs almost on par with competing state-of-the-art approaches for ISP de-rendering while offering a practical solution that is integrable to current camera ISP hardware.

Introduction

A camera's image signal processor (ISP) is dedicated hardware that renders the raw sensor image to its final output image encoded to a display-referred format. In this paper, we refer to the unprocessed sensor image as a raw-RGB image and the ISP-rendered display-referred image as a standard RGB (sRGB) image. The ISP hardware performs a number of routines to the sensor's raw-RGB image. These routines include steps such as white-balance, chromatic adaptation, and photo-finishing color manipulation to impart a particular aesthetic to the captured image. On modern cameras, the ISP parameters can vary per scene-category (e.g., person, food, outdoors), allowing for a wide range of rendering styles.

Professional photographers often prefer to shoot in an unprocessed raw-RGB image format. Shooting in raw-RGB provides users the ability to render the raw-RGB image to user controlled styles via photo-editing software, such as Adobe PhotoShop or Luminar. In addition, certain image processing tasks, such as image deblurring, color constancy, and photometric stereo, are more effective when applied to images in a raw-RGB format [18]. While raw-RGB offers advantages over display-referred images, most consumers still do not shoot in a raw format. One reason is because raw images are typically 4–6 times larger than sRGB encoded images. Moreover, raw-format images are not easily shared as they require rendering to a display-referred color space before viewing. Due to these drawbacks, the vast majority of captured images are saved in a display-referred format. This paper examines a method to de-render an display-referred image back to its raw-RGB state.

Reverting an sRGB image back to its original raw sensor state is a challenging problem. This is because many of the operations applied by the ISP are nonlinear and proprietary – the ISP is essentially a 'black box' to the end user. A few recent works [1, 17, 22] have explored the possibility of saving additional metadata along with the sRGB image at capture time so as to enable reversing the sRGB image back to its raw state. However, these methods have several drawbacks. The approach of [22] requires 1.5–6 MB of additional metadata in the form of a small raw file to be saved along with the sRGB image. The framework of [1] requires hardware modifications to the camera pipeline to downsample the raw image and make multiple passes through the ISP. Such downsampling may not be supported by all ISP hardware. The method proposed by [17] is computationally intensive, requiring the estimation of a complex inversion function at capture time from the full-resolution image. This latter method is not conducive to real-time performance on-device.



(A) Raw-RGB sensor image. A fixed pattern is appended to all captured images.
(B) The ISP renders the image and the pattern. A reverse mapping function *f* is derived based on the pattern values.
(C) Using *f*, we map the image (B) back to its raw-RGB state in (A). Per pixel errors are shown as inset.

Figure 1. *(A) Our approach modifies the camera ISP by appending a* fixed *pattern of raw-RGB samples to the captured sensor image. (B) The camera ISP renders this raw-RGB image, including the appended pattern, to the final sRGB image. A reverse mapping function, f, can be estimated based on the correspondence between the raw-RGB and sRGB samples in the appended pattern. (C) This image shows the result of reverting the image (B) back to (A) using f. The inset shows a per-pixel error map of this de-rendered result.*

Contributions We propose a practical mechanism for raw recovery that is integrable to current ISP hardware. Our approach is to append a pattern of fixed samples of raw-RGB values to the captured sensor image and render it through the ISP to sRGB. The appended raw-RGB values are rendered based on the ISP settings specific to that image. Based on these corresponding raw-RGB and rendered sRGB values, we can construct a reverse mapping that can be applied on the output sRGB image to revert to its raw sensor image as shown in Fig. 1. Our framework offers many advantages over existing techniques. Our fixed pattern requires only around eight rows of pixels (assuming a minimum sensor width of 4000 pixels, which is common on modern consumer cameras), and equals just 96 KB of additional data. Our method entails minimal modifications to the existing ISP design since we merely append a few additional rows; no downsampling or multi-pass operations on the ISP are required. The computational overhead at capture time is nominal since the rendered sRGB samples themselves constitute the metadata, and no mapping function parameters need to be estimated. We demonstrate that our method performs almost on par with existing metadata approaches for raw recovery when tested on the NUS dataset [7].

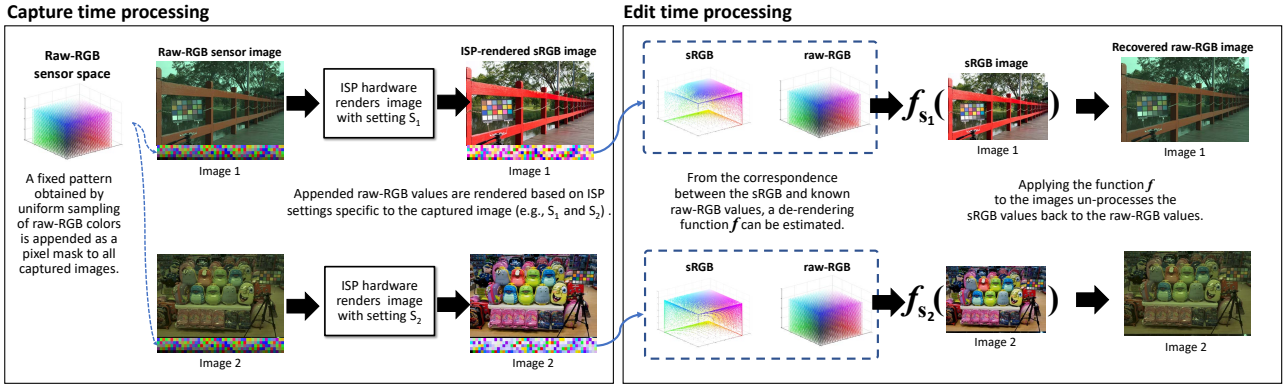


Figure 2. *An overview of our proposed method. A fixed pattern of raw-RGB colors is generated by uniformly sampling the raw-RGB color space. This pattern is appended to all captured images and rendered by the ISP. The size of the pattern in the figure is exaggerated for visualization purposes. Shown are two different images rendered with different ISP settings, s_1 and s_2, to the display-referred sRGB image state. The raw-RGB pattern is also rendered by the ISP. Based on the correspondence between the sRGB pattern values and the a priori known raw-RGB pattern values, a de-rendering function, f_{s_i}, can be estimated. Note that the function is specific to the ISP settings used for that image. Using f we can de-render the sRGB image to recover the original raw-RGB values.*

Related Work

We survey below works on radiometric calibration and raw reconstruction. Raw reconstruction algorithms can be categorized into those that leverage additional metadata stored with the sRGB image at capture time, and blind methods that require no extra information. We examine the former category in greater detail since these are more closely connected to our work.

Before camera manufacturers began allowing consumers access to the raw sensor image, work on raw reconstruction was targeted mainly towards what was termed 'radiometric calibration'. The goal of radiometric calibration was simply to undo the nonlinear processing applied onboard the camera, rather than recover the actual raw-RGB sensor image. Conventional radiometric calibration algorithms compute a camera response function that models the relationship between the output pixel intensities and the amount of light falling on the sensor. These methods typically employ simplistic models with a single response function per color channel [8, 11, 15].

As cameras started to allow capture in raw-RGB, the objective shifted from linearizing the sRGB values to fully recovering the original raw-RGB sensor image. Most noteworthy among these raw recovery methods was work by Chakrabarti et al. [5, 6], Kim et al. [12], and Gong et al. [10], who developed more complex models for the onboard processing on cameras. However, the drawback of these methods [5,6,10,12] is the cumbersome calibration required per camera and even per camera setting. Recent work by Brooks et al. [4] attempts to invert the camera ISP, step by step, to recover the raw-RGB sensor image. However, their approach requires a priori knowledge about the target camera (e.g., white-balance and color-correction matrices), which limits its practical utility. More importantly, several operations on the ISP are proprietary and it can be difficult to reverse-engineer such black boxes. In a similar manner, the imaging pipeline reversal technique of [13] also assumes the target camera parameters are known. Recently, deep learning methods (e.g., [14, 16]) that attempt to undo the ISP processing have also been proposed. However, the trained models are specific to the given camera and ISP setting. Punnappurath and Brown [20] advocated a deep learning strategy for sRGB image compression that simultaneously favours accurate raw reconstruction from the compressed data.

Raw reconstruction with metadata A few recent works [1, 17, 22] have explored the use of specialized metadata stored along with the sRGB image at capture time to enable more accurate raw recovery than the blind approaches discussed thus far. Yuan and Sun [22] proposed storing a down-sampled raw image as additional metadata along with the full-resolution sRGB image. This low-resolution raw image can then be upsampled to full resolution with the sRGB image acting as a guide for the upsampling operation. However, the quarter- or half-sized small raw images used in this work still contribute 1.5–6 MB of additional metadata that needs to be stored.

Nguyen and Brown [17, 18] estimate the parameters necessary to reconstruct the raw image from the sRGB image by modelling the typical operations performed by the camera ISP. In particular, the parameters modelled are a white-balance matrix, a color-correction matrix, a tone map operator, and a gamut mapping function. These parameters are estimated from the sRGB-raw image pair at capture time and stored as metadata. However, their inverse tone map computation and modified octree-based gamut mapping estimation are computationally intensive and hard to implement on-device in real time.

Recent work by Afifi et al. [1] proposes a modified ISP pipeline where the raw image is downsampled to a tiny image and rendered through the ISP multiple times with different camera settings. Mapping functions computed at capture time between these tiny re-rendered images and a downsampled version of the output sRGB image are stored as metadata. These mapping functions can be applied to edit the sRGB image post-capture, or recover the raw sensor image. However, this multi-pass framework requires substantial changes to the conventional ISP.

In contrast to [1, 17, 22], our proposed method requires only 96 KB of metadata, adds minimal computational overhead at capture time, and is easily integrable to existing ISP design, as explained next.

Proposed Method

The two works most closely related to ours are the methods of [1] and [17]. Both methods work by constructing a reverse mapping from the sRGB color space to the raw-RGB color space. The mapping is computed using the demosaiced raw-RGB sensor image and its camera-rendered sRGB version. 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. The mapping computed between this demosaiced raw-RGB and sRGB image pair is specific to that image and needs to be estimated at capture time. However, mapping function estimation is computationally intensive and hard to practically achieve on-device in real time. The parameters of this mapping function are then stored as metadata. Note that applying the mapping function on the sRGB image de-renders it to the demosaiced raw-RGB image, and not the Bayer raw.

Akin to [1, 17], our method also uses raw-sRGB correspondences to build a reverse mapping from sRGB to raw-RGB. However, unlike the methods in [1] and [17] that use samples drawn from the image itself, we use a small number of *fixed* raw samples that are *independent* of the scene being captured. The corresponding rendered sRGB samples are saved as metadata. This eliminates the need to compute the mapping function at capture time, thereby greatly reducing the computational overhead.

The raw samples themselves are selected by uniformly sampling the full space of raw-RGB colors that the camera sensor is capable of producing. For example, consider a camera where the raw-RGB image is recorded by the sensor with a bit depth of 10. The total number of unique raw-RGB color values that the sensor can record is $1024 \times 1024 \times 1024$. We bin this 3D RGB volume uniformly using $32 \times 32 \times 32$ samples. This fixed raw-RGB pattern is appended to each captured image after the demosaicing step and rendered through the ISP with the full frame. The resulting $32 \times 32 \times 32$ camera-rendered sRGB samples constitute our metadata. These additional pixels represent only 96 KB of overhead to the full-size image. We can compute a reverse mapping, f, from sRGB to raw-RGB from these rendered sRGB samples and their corresponding a priori known fixed raw-RGB samples. This mapping can be computed post-capture when the user edits the image. Since we only need to append a few rows of pixels, hardware changes required to the ISP are minimal. An overview of our proposed framework is shown in Fig. 2. Two images processed with different ISP settings, denoted as s_1 and s_2, are shown. Note that the reverse mapping function f is specific to each image (i.e., f_{s_1} and f_{s_2}) because the associated sRGB pattern values for each image are different due to the different ISP settings used to render the image.

Our mapping function f is implemented via a standard scatter point interpolation framework [2]. The interpolation process involves constructing a Delaunay tetrahedrization [19] of the given scattered point set, determining which tetrahedron each query point belongs to (for query points outside the convex hull, linear extrapolation is performed), and applying a trivariate linear interpolation within each tetrahedron. An overview of these steps can be found in [2], pp. 159–160.

Experiments

We first evaluate our method's effectiveness in recovering the raw-RGB image. We test on three cameras – a Samsung NX2000, an Olympus E-PL6, and a Sony SLT-A57 – from the NUS dataset [7]. The total number of images used in our experiments is 678. In our implementation, we convert the raw images available as part of the dataset to the standard raw-DNG (Digital Negative) format. We then modify the raw-DNG files to contain our fixed pattern of raw-RGB samples. We render this modified raw-DNG image to sRGB using a software camera ISP. In our experiments, we selected Adobe Photoshop as the software ISP to closely mimic the processing onboard the camera hardware. The rendered image additionally contains the sRGB samples that we use to construct our reverse mapping to the raw space.

Method	PSNR (dB)			
	Avg.	Q1	Q2	Q3
[1]	45.18	38.85	44.76	51.80
[17]	45.65	39.40	47.82	53.36
Ours	43.32	38.71	44.28	48.71

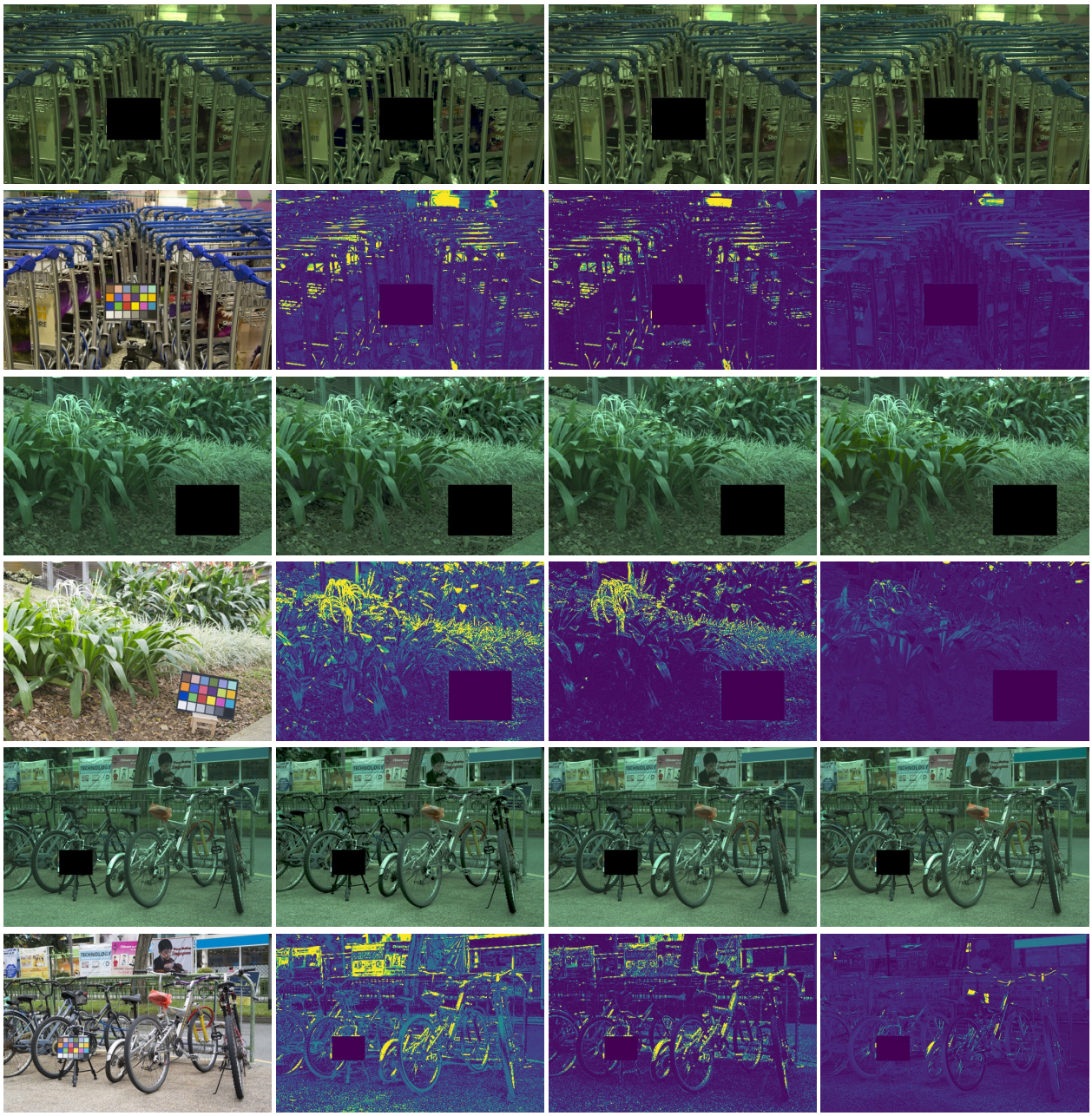
Table 1. **Quantitative raw recovery results on three cameras from the NUS dataset [7]. The PSNR (dB) of the estimated raw-RGB result is shown. All three methods use metadata saved at capture time to reconstruct the raw-RGB image. The terms Q1, Q2, and Q3 denote the first, second (median), and third quantile, respectively.**

The average peak signal-to-noise ratio (PSNR) between our reconstructed raw image and the ground-truth (GT) raw image is provided in Table 1. We also compare against the metadata-based raw recovery approaches of [1] and [17]. The terms Q1, Q2, and Q3 refer to the first, second (median), and third quantile, respectively, of the PSNR values obtained by each method. Our accuracy is slightly lower than [1,17] because these two methods build their mapping functions at capture time using the raw image itself, whereas our raw samples are generic and not tailor-made to the scene. As already mentioned, our approach has a clear upperhand in terms of ease of integration into existing ISP design and computational efficiency at capture time. A few representative examples of our raw reconstruction, along with comparisons, are provided in Fig. 3. Note that the color charts are masked for the purpose of comparing against [1,17], who draw samples from the raw image itself. We mask out this region in our result too for fairness of comparison, although our method itself is unaffected by the presence/absence of a color chart in the scene.

Next, we apply our proposed method to the task of white-balance (WB) manipulation. We use the same set of modified DNGs, but instead of rendering to sRGB with the "as shot" camera parameters as in our earlier experiment, we render out using a different WB setting. In particular, we generate sRGB images under Tungsten WB (2850 K) and Shade WB (7500 K). Note that along with the image, our embedded raw samples are also rendered to sRGB under these two different WB settings. We now reconstruct the raw image using our proposed framework. In this experiment, our objective is to evaluate the accuracy of re-rendering to a different WB using our raw reconstruction. To this end, we save our raw reconstruction as a raw-DNG file. Note that

Method	Diag WB	Linear Diag WB	[1]	Ours
MSE	0.54	0.48	0.11	0.20
MAE	5.65	5.44	1.74	4.41
Δ E 2000	7.06	6.69	2.96	4.27
Δ E 76	10.58	9.94	4.14	5.66
Δ L*	2.77	2.48	2.80	2.56
Δ C*	8.53	7.91	3.81	5.23
Δ H*	70.81	69.93	40.16	52.96

Table 2. **Quantitative results for white-balance (WB) manipulation on three cameras from the NUS dataset [7]. We compare our results against diagonal WB manipulation, denoted as Diag, using an *exact achromatic* reference point obtained from the color chart in the scene. The diagonal manipulation is applied directly on the sRGB images, and on the "linearized" sRGB [3,9]. We also compare against the metadata approach of [1]. To apply the WB, both our method and [1] also use a diagonal matrix. The terms MSE and MAE stand for mean squared error and mean angular error, respectively.**



(A) GT
Raw-RGB / sRGB

(B) [1]
Raw-RGB / Error map

(C) [17]
Raw-RGB / Error map

(D) Ours
Raw-RGB / Error map

Figure 3. *Raw reconstruction results. (A) Ground-truth (GT) raw-RGB sensor image and camera-rendered sRGB output image. (B–D) Estimated raw-RGB results and the corresponding errors maps using the methods of [1], [17], and our proposed algorithm, respectively. Note that a 2.2 gamma has been applied to all raw-RGB images for better visualization. The error maps depict absolute pixel value from scale 0 to 3%.*

our reconstruction is a demosaiced raw-RGB image, and so while saving to a raw-DNG format, we set the appropriate DNG tags to indicate that the raw image has already been debayered. We then render our estimated raw-DNG to the Daylight WB (5500 K) setting using Photoshop. The ground truth is obtained by rendering the original raw-DNG images 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 [21], and (iv) Δ E 76; the results are presented in Table 2. In addition to the consolidated Δ E 76 error, the Δ L*, Δ C*, and Δ H* values have also been provided. 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 [3, 9]. We also compare against the method of [1] that renders out multiple tiny images through the ISP under different WB settings to enable post-capture WB manipulation. 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 2 that our method outperforms diagonal WB manipulation on all four metrics while being only marginally less accurate than [1].

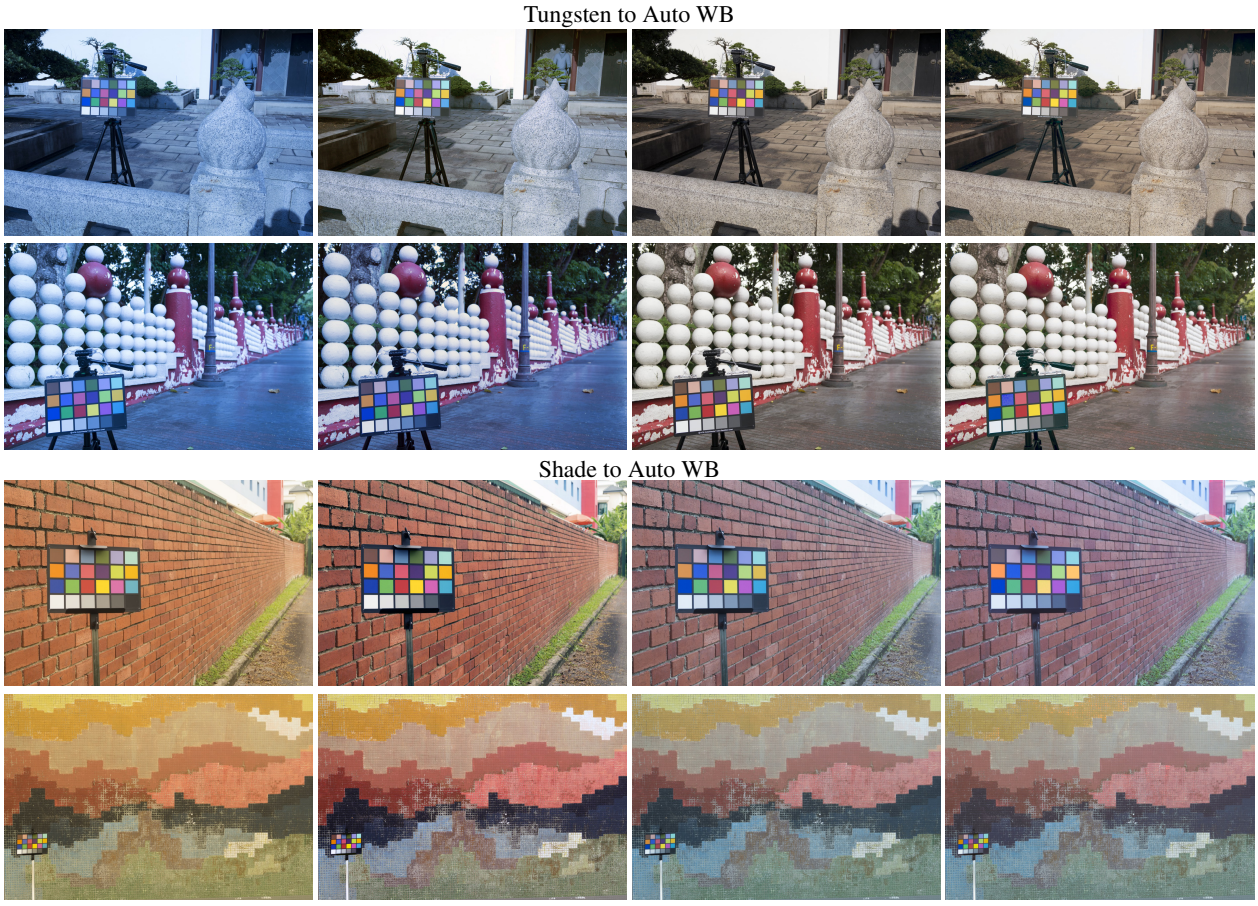


Figure 4. *White-balance correction. We perform auto WB on images captured under Tungsten and Shade WB. In the second column, Adobe Photoshop's auto correction of the input sRGB image of the first column is shown. The ground truth shown in the third column is obtained by running Photoshop's auto WB routine on the ground-truth raw-DNG image. The last column shows Photoshop's auto WB algorithm applied to our recovered raw-DNG.*

In Fig. 4, we show qualitative results of auto WB correction. Photoshop's auto WB algorithm is used in all cases – in the second column, it is applied directly on the sRGB image, in the third column, on the ground-truth raw-DNG image, and in the last column, on our reconstructed raw-DNG. It can be observed that our rendered sRGB output is visually a close match to the ground-truth sRGB.

Our method is also useful for general photographic applications and image enhancement tasks. Two low-contrast images enhanced using our raw reconstruction are shown in Fig. 5.

The plot of Fig. 6 presents an ablation study of sampling density versus raw reconstruction accuracy. The images from the Samsung NX2000 camera were used for this experiment. It can be observed that the improvement in PSNR is marginal beyond 32 bins. On the other hand, a finer sampling of the raw space requires more rows to store the pattern – approximately 28 rows for 48 bins and 65 rows for 64 bins, assuming a 4000-pixel-wide sensor. We used 32 bins in all our experiments since at eight rows, it offered the best compromise between accuracy and size.

Concluding Remarks

We have presented a method to de-render a processed display-referred image back to a version of its initial raw state. Our approach requires a modification to the conventional ISP in the form of a fixed set of samples that are appended to the demosaiced raw image. These fixed samples span the raw-RGB

Figure 5. *Our proposed method can be applied to image enhancement tasks. The sRGB image "as shot" by the camera is input to our raw-recovery algorithm. Enhanced re-rendered sRGB results obtained using our estimated raw-DNG outputs are shown.*

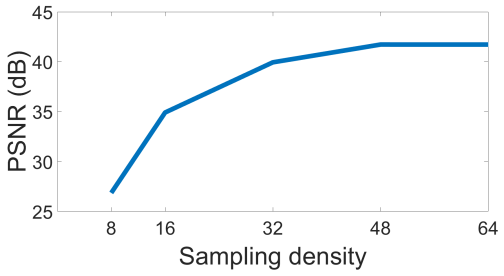


Figure 6. *Sampling density versus raw reconstruction accuracy. We use* $32\times32\times32$ *bins since it offers the best trade-off between metadata size and accuracy.*

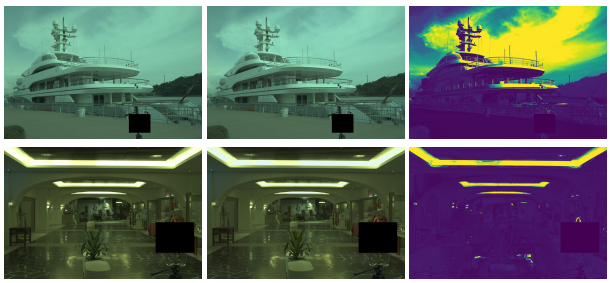


Figure 7. *Failure cases of our method arising from saturation and local tone mapping. The error maps are absolute pixel value from scale 0 to 5%.*

sensor gamut and are rendered with the full-frame sensor image to the final sRGB display-referred output. From these rendered appended samples, a mapping can be estimated to revert sRGB image values to the raw-RGB sensor values, effectively de-rendering the image back to a raw image state. Our approach requires only eight additional rows of pixels and provides de-rendering quality almost on par with other state-of-the-art methods that rely on constructing capture time mapping functions.

Like prior methods [1] and [17], our method does not currently handle the one-to-many ambiguity that may arise in the reverse mapping from sRGB to raw-RGB. Moreover, our method does not account for saturation or local tone mapping that may have occurred (see Fig. 7), or multi-frame fusion, such as in high dynamic range imaging. In such cases, our approach reverts back to a plausible raw image that when rendered forward would produce the same sRGB image. This proxy raw is still suitable for high-quality photo-editing for photographic purposes.

References

[1] M. Afifi, A. Punnappurath, A. Abdelhamed, H. C. Karaimer, A. Abuolaim, , and M. S. Brown. Color temperature tuning: Allowing accurate post-capture white-balance editing. In *Color Imaging Conference*, 2019.

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

[3] M. Anderson, R. Motta, S. Chandrasekar, and M. Stokes. Proposal for a standard default color space for the internet - sRGB. In *Color Imaging Conference*, 1996.

[4] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron. Unprocessing images for learned raw denoising. In *Computer Vision and Pattern Recognition*, 2019.

[5] A. Chakrabarti, D. Scharstein, and T. E. Zickler. An empirical camera model for internet color vision. In *British Machine Vision Conference*, 2009.

[6] A. Chakrabarti, Y. Xiong, B. Sun, T. Darrell, D. Scharstein, T. Zickler, and K. Saenko. Modeling radiometric uncertainty for vision with tone-mapped color images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(11):2185–2198, 2014.

[7] D. Cheng, D. K. Prasad, and M. 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.

[8] P. E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *SIGGRAPH*, 2008.

[9] M. Ebner. *Color Constancy*. Wiley Publishing, 1st edition, 2007.

[10] H. Gong, G. Finlayson, M. Darrodi, and R. Fisher. Rank-based radiometric calibration. In *Color Imaging Conference*, 2018.

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

[12] S. J. Kim, H. T. Lin, Z. Lu, S. Süsstrunk, S. Lin, and M. 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.

[13] S. Koskinen, D. Yang, and J. Kämäräinen. Reverse imaging pipeline for raw RGB image augmentation. In *International Conference on Image Processing*, pages 2896–2900, 2019.

[14] Y.-L. Liu, W.-S. Lai, Y.-S. Chen, Y.-L. Kao, M.-H. Yang, Y.-Y. Chuang, and J.-B. Huang. Single-image HDR reconstruction by learning to reverse the camera pipeline. In *Computer Vision and Pattern Recognition*, June 2020.

[15] T. Mitsunaga and S. K. Nayar. Radiometric self calibration. In *Computer Vision and Pattern Recognition*, 1999.

[16] S. Nam and S. J. Kim. Modelling the scene dependent imaging in cameras with a deep neural network. In *Internation Conference on Computer Vision*, pages 1726–1734, 2017.

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

[18] R. M. Nguyen and M. 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.

[19] G. Nielson, H. Hagen, and H. Müller. Scientific visualization. IEEE, 1997.

[20] A. Punnappurath and M. S. Brown. Learning raw image reconstruction-aware deep image compressors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(4):1013–1019, 2019.

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

[22] L. Yuan and J. Sun. High quality image reconstruction from RAW and JPEG image pair. In *International Conference on Computer Vision*, 2011.