https://doi.org/10.2352/issn.2169-2629.2020.28.30

Improving Color Space Conversion for Camera-Captured Images via Wide-Gamut Metadata

Hoang Le, Mahmoud Afifi, Michael S. Brown; York University, Toronto, Canada

Abstract

Color space conversion is the process of converting color values in an image from one color space to another. Color space conversion is challenging because different color spaces have different sized gamuts. For example, when converting an image encoded in a medium-sized color gamut (e.g., AdobeRGB or Display-P3) to a small color gamut (e.g., sRGB), color values may need to be compressed in a many-to-one manner (i.e., multiple colors in the source gamut will map to a single color in the target gamut). If we try to convert this sRGB-encoded image back to a wider gamut color encoding, it can be challenging to recover the original colors due to the color fidelity loss. We propose a method to address this problem by embedding wide-gamut metadata inside saved images captured by a camera. Our key insight is that in the camera hardware, a captured image is converted to an intermediate wide-gamut color space (i.e., ProPhoto) as part of the processing pipeline. This wide-gamut image representation is then saved to a display color space and saved in an image format such as JPEG or HEIC. Our method proposes to include a small sub-sampling of the color values from the ProPhoto image state in the camera to the final saved JPEG/HEIC image. We demonstrate that having this additional wide-gamut metadata available during color space conversion greatly assists in constructing a color mapping function to convert between color spaces. Our experiments show our metadata-assisted color mapping method provides a notable improvement (up to 60% in terms of $\triangle E$) over conventional color space methods using perceptual rendering intent. In addition, we show how to extend our approach to perform adaptive color space conversion based spatially over the image for additional improvements.

1. Introduction and Motivation

Images captured on cameras are almost exclusively encoded in a display-referred color space that is intended for viewing on a display device. The most common color space is the standard RGB (sRGB) color space, which was defined over two decades ago [1]. The sRGB color space encodes a relatively small color gamut and was designed primarily for 1990s cathoray tube technology. Adobe introduced the Adobe RGB color space [2] which provides a wider color gamut (almost 50% more colors than sRGB) supported by many cameras. Recently, Apple started encoding its camera images in the Display-P3 color space (also referred to as Apple P3 or DCI-P3), which provides a similarly sized wider color gamut to Adobe RGB.

This mismatch in color gamut of different color spaces makes color management challenging. To address this, ICC profiles are used to describe the specifications of the characteristics of working color spaces (e.g., sRGB, Adobe RGB, Display-P3) or a device's color space characteristics (actual hardware) [3]. A profile connection space (PCS), such as CIE XYZ or CIE L*a*b*, is used to link different color spaces, by mapping the source first to the PCS and then from the PCS to the target color space. The problem lies in the different gamut ranges among the

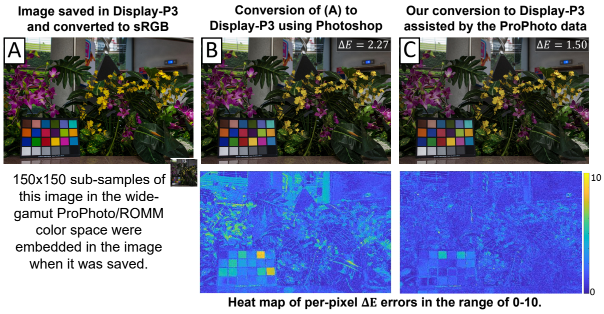

Figure 1. *(A) This image was encoded in Display-P3 and then converted to the sRGB color space. The image also has a small amount of wide-gamut ProPhoto RGB values (150×150 pixels) recorded from the camera hardware at capture time and saved in the image. (B) shows the result of performing conventional color space conversion back to Display-P3. (C) shows the result of using our method that exploits the available wide-gamut metadata to construct a global color mapping function between sRGB and Display-P3. Per-pixel CIEDE2000 errors are also shown.*

different color spaces. In the case of mapping from a larger to smaller gamut, out-of-gamut color values in the source must be compressed or clipped in the target space. There are several *rendering intents* (e.g., absolute colorimetric, relative colorimetric, and perceptual) that propose how such colors should be handled when mapping between different color gamuts. While the various rendering intents each have their own pros and cons in regards to minimizing color errors, it is not possible to avoid some errors due to the color gamut sizes. This is easiest observed when mapping from a wider gamut color space to a smaller gamut color space and then back. Fig. 1 shows an example—the image was originally encoded in Display-P3 and then converted to sRGB using Photoshop's conversion with a perceptual rendering intent. The image is then converted back to Display-P3 using a perceptual rendering intent. The corresponding error map in CIEDE2000 (ΔE) shows that the colors are not accurately restored. This is because they were clipped or compressed on the initial color space conversion from Display-P3 to sRGB. We seek to ameliorate this problem by exploiting data readily available on the camera at capture time, but ignored when encoding the image into its final file format, such as JPEG or HEIC.

Contribution We propose a method to assist the color space conversion problem that involves embedding wide-gamut metadata in the initial captured image. Our observation comes from the fact that most camera hardware applies a conversion of the original sensor-RGB image to an intermediate image state in the wide-gamut ProPhoto color space when rendering the image to the final display-referred color space. We show that by storing a very small number of these wide-gamut ProPhote samples only (e.g., 150×150 samples for a $\sim$20MB image), we are able to construct global color mapping functions that are highly effective for color space conversion. We also describe how to extend this idea to apply local color mapping for further improvements.

2. Related Work

Color management systems [4] are instrumental in allowing images and media to be shared among different color encodings and display devices. The need to handle the mismatch among different color spaces' and devices' gamuts is a well-studied topic (e.g., [5–12]). While improved color encodings have been proposed (e.g., [13] and Rec. 2020 [14]) that facilitate improved color space conversions and very wide gamuts, for images captured by cameras the dominant color spaces remain sRGB, Adobe RGB, and Display-P3. Our method focuses on these color spaces as our approach targets images generated by consumer cameras. In particular, we are interested in converting between these formats, such as AdobeRGB to sRGB and back.

When converting between color spaces, early work (e.g., [15]) introduced the notion of "rendering intent" versus strict colorimetric reproduction. The *perceptual rendering intent* is generally preferred for color space conversions for photographic images as it strives to minimize perceptual errors [16]. Color space conversion methods can be performed in a global or local manner (i.e., factoring in the neighborhood information about a pixel); however, the color mappings are generally applied agnostically of any additional information about the image. The exception is methods that exploit object-level semantics, such as skin-tone [17], or objects labeled by deep neural networks [18].

Our approach is unique in that it assists color space conversion by embedding auxiliary information in the saved image in the form of wide-gamut color samples from the captured image that is available on the camera hardware. Specifically, the onboard camera rendering hardware includes several steps that convert the RAW sensor image to its final display-referred output. One of these steps involves the conversion of the scene-referred RAW image to a wide-gamut reference output medium metric (ROMM) color space [19,20] commonly referred to as ProPhoto. ProPhoto contains 90% of the colors in CIE L*a*b, which is used as a profile connection space. Inside the camera pipeline, photo-finishing routines are applied to the ProPhoto image to improve its photographic qualities. The final stage in the camera pipeline is to convert the ProPhoto/ROMM image to a display-referred color space (e.g., sRGB, AdobeRGB, Display-P3). Related work in [21] proposed to embed metadata to allow the image to be reverted from a display-referred color space (i.e., sRGB) back to its scene-referred RAW image state. However, reverting back to RAW would require a forward rendering to convert the RAW into a output-referred color space. The necessary data to re-render the RAW image correctly is rarely available. Our ProPhoto encoding does not suffer from this problem as it directly represents the rendered image's colors. We also note recent work in [22] that embedded metadata regarding the image's color rendering options (i.e., white-balance) to allow post-capture white-balance manipulation. Similar to [22], we found that only a small amount of color samples is required to assist in our gamut mapping.

3. Proposed Method

Our method involves embedding a small amount of wide-gamut image data gleaned from the camera hardware in the final outputted image saved in a medium- or small-gamut color space. This wide-gamut data is used in generating color space conversion mappings for converting between different color spaces. An overview of the proposed method is shown in Fig. 2.

3.1 ProPhoto Sub-Sampling

The camera image processing hardware applies a series of processing steps to convert a captured RAW sensor image to its final rendered output. One of the steps performed is mapping the image color values to the ProPhoto/ROMM color space (see [20] for an overview of in-camera processing). This ProPhoto color space provides a wide-gamut color encoding suitable for photo-finishing routines, such as general color enhancement, selective color rendering for skin, and local and global tone manipulation. When the photo-finishing is complete, the last step applied by the hardware is to convert the image to its final display-referred color space and then save the file in a compressed format (i.e., JPEG or HEIC). DSLR cameras often allow the user to change the display-referred output color space (e.g., sRGB or Abobe RGB), while smartphone cameras typically use a fixed output color space (e.g., Display-P3 or sRGB). Our key insight is that before the image is saved, the image has pixel values encoded in wide-gamut color space. It is this wide-gamut ProPhoto data—available on the hardware for every captured image but later discarded—that we exploit. As mentioned in Sec. 2, the idea of gleaning information from the camera hardware at rendering time and embedding it in the saved image is inspired by prior work [21, 22] that targeted different applications.

To this end, we propose a minimal modification to existing camera hardware to be able to extract the ProPhoto data as shown in Fig. 2-(A). Specifically, we uniformly sub-sample the full-frame image spatially to produce a *tiny image*, termed $\mathbf{S}_{\mathrm{ProP}}$. This sampling is performed on the intermediate ProPhoto image after photo-finishing processing has been applied, but before the ProPhoto image is converted to its output color space encoding. In our experiments, we sample 150×150 pixels from the full-resolution 20-megapixel intermediate ProPhoto image, $\mathbf{I}_{\mathrm{ProP}}$. This sub-sampling represents less than 0.1% of the pixels in the full-frame image. We then store the ProPhoto samples, $\mathbf{S}_{\mathrm{ProP}}$, as auxiliary data in the output image, $\mathbf{I}_{\mathrm{out}}$, where $\mathbf{I}_{\mathrm{out}}$ is encoded in one of the three prevailing display-referred color spaces—sRGB, Adobe RGB, or Display P3. Using a 12-bit encoding per each R/G/B color channel, the auxiliary ProPhoto data of 150×150 can be stored as an uncompressed comment field in an JPEG or HEIC file using less than 100KB.

3.2 Color Space Conversion with Wide-Gamut Metadata

Given a camera image $\mathbf{I}_{\mathrm{out}}$ with the stored metadata $\mathbf{S}_{\mathrm{ProP}}$, the goal is to map our source image's colors in a target color space's color gamut. We refer to the target color gamut as T and denote the target as image $\mathbf{I}_{\mathrm{T}}$. For evaluation purposes, we assume that we have access to ground truth image $\mathbf{I}_{T(\mathrm{GT})}$ in the target color gamut T. We describe two strategies—global and local—for this color space conversion in the following. Note that we assume images are represented in an $3 \times n$ matrix for format, where n refers to the total number of pixels in the image.

Conventional methods based on ICC profile specifications and using the profile connection space (PCS) would first convert the source image $\mathbf{I}_{\mathrm{out}}$ to the PCS and then convert from the PCS to the target color space $\mathbf{I}_{\mathrm{T}}$. In our scenario, the wide-gamut metadata represents color values for our source image that are already in a space that is very similar to the current PCS, CIE L*a*b and CIE XYZ—recall, ProPhoto spans 90% of the CIE L*a*b color space. As a result, our metadata encodes values that essentially serve as a proxy for the PCS mappings. We can directly use these values in our color mapping functions.

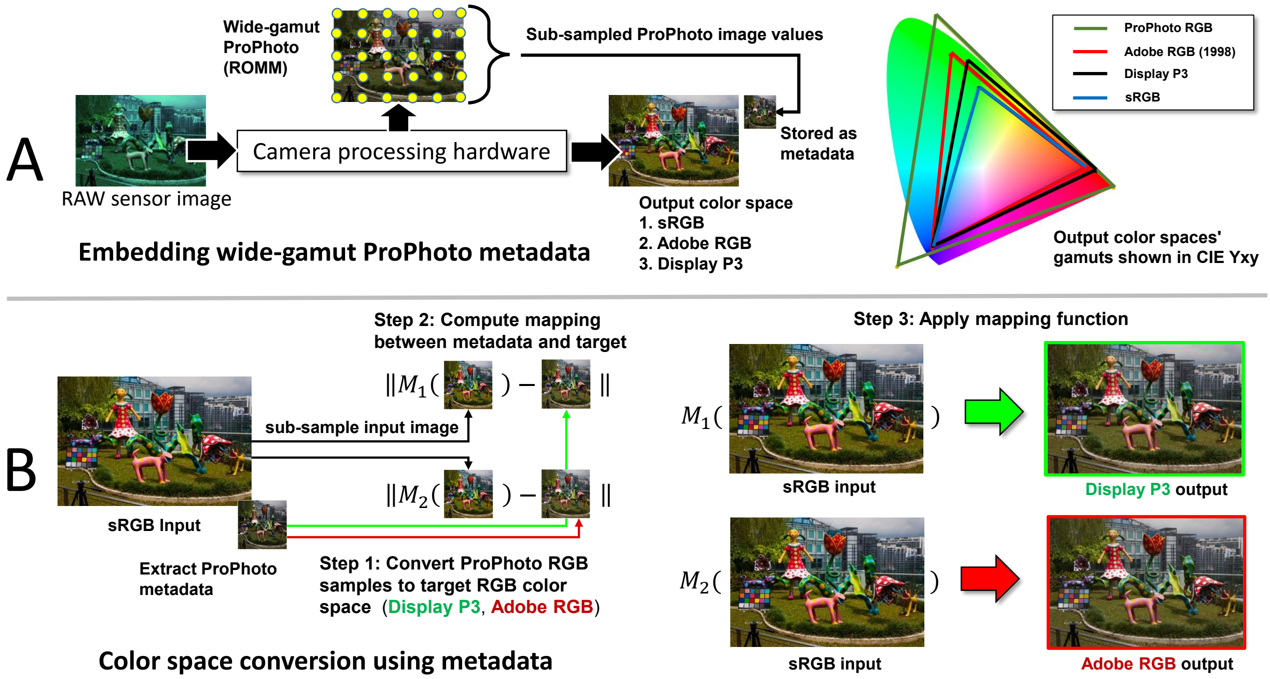

Figure 2. Overview of the proposed method. (A) The camera hardware converts the initial RAW sensor image into an intermediate wide-gamut color space (e.g., ProPhoto color space) format for photographic rendering purposes. Our method extracts a small sub-sampling of the ProPhoto image (150×150 samples) and stores this information as a metadata in the final saved standard color space (e.g., Adobe RGB (1998), Display P3, or sRGB). (B) To apply gamut mapping to any other color space, we first retrieve our sampled wide-gamut image from the metadata stored in the final camera-rendered image. Then, we convert the samples to the target gamut. Finally, we apply a locally adaptive nonlinear mapping to map from the final output color space gamut to the target color space gamut.

Global Mapping Our approach follows the strategy proposed in [22] of applying a global fitting to map our camera-rendered image $\mathbf{I}_{\text{out}}$, in its full-size, to the target color space T. For this procedure, we first convert the ProPhoto samples, $\mathbf{S}_{\text{ProP}}$, to the target color space—assumed to be either sRGB, Display-P3, or Adobe RGB. We denote these converted samples as $\mathbf{S}_T$. Because the ProPhoto samples are in a color gamut larger than sRGB, Display-P3, and Adobe RGB, it is suitable for all rendering intents. We will assume that a perceptual rendering intent is applied.

Next we followed the same sampling mechanism used in Sec. 3.1 to sample a tiny version of the $\mathbf{I}_{\text{out}}$ source image. Here, we refer to this sampled image as $\mathbf{S}_{\text{out}}$. Our global color conversion function can then be computed in a closed form as follows:

$$\arg\min_{\mathbf{M}_{\text{global}}} \left\| \mathbf{M}_{\text{global}} \, \Phi\left(\mathbf{S}_{\text{out}}\right) - \mathbf{S}_T \right\|_{\text{F}}, \tag{1}$$

where $\|.\|_{\text{F}}$ is the Frobenius norm and Φ is a polynomial function that maps the RGB triplets to a higher-dimensional space. In our implementation we used the polynomial mapping function used in [23]. Specifically, this polynomial function is represented as $\Phi : \Phi\left([\text{R, G, B}]^\top\right) \rightarrow [\text{R, G, B, RG, RB, GB, R}^2, \text{G}^2, \text{B}^2, \text{RGB}, 1]^\top$. Once the global fitting matrix $\mathbf{M}_{\text{global}} \in \mathbb{R}^{3\times 11}$ is computed, we can generate the image $\mathbf{I}_T$ as follows:

$$\mathbf{I}_T = \mathbf{M}_{\text{global}} \, \Phi\left(\mathbf{I}_{\text{out}}\right). \tag{2}$$

This procedure is shown in Fig. 2-(B). In that example, the source image is saved in sRGB color space that has the smallest gamut. We then convert the ProPhoto metadata samples to both Display-P3 and Adobe-RGB and convert the image based on the respective mappings $\mathbf{M}_1$ and $\mathbf{M}_2$.

Local Mapping The global mapping described above achieves a noticeable improvement compared to conventional color space conversion methods; however, there is still an ambiguity in the one-to-many mapping when the target gamut has a wider color gamut (e.g., from the sRGB color gamut to the Display-P3 color gamut). In such cases, a single color in the source gamut may have multiple corresponding colors in the target gamut. To minimize errors, the global mapping will map these source colors towards the mean of the corresponding target colors.

To overcome this, we can use a local mapping function that incorporates the spatial location (x,y) of a pixel in the mapping function. We perform a simple procedure to break the image into local regions and compute local mappings for each regions in a weighted least squares manner. First, we convert our ProPhoto "tiny image" samples to the desired target $\mathbf{S}_T$ as done in the global mapping procedure. The corresponding tiny source image, $\mathbf{S}_{\text{out}}$, is segmented into superpixels using the simple iterative clustering (SLIC) algorithm [24] with the number of superpixels, k, set to 100. For each of these 100 superpixels, p, we build a binary mask $\mathbf{L}_{(p)}$ where all pixels in $\mathbf{S}_{\text{out}}$ that belong to p are set to 1, and 0 otherwise. This masks is dilated with a morphological operator and blurred to produce a weight map denoted as $\mathbf{L}_p$. This weight map is normalized so all of its weights sum to 1.

We then compute a mapping function per superpixel, p, as follows:

$$\arg\min_{\mathbf{M}_p} \left\| \left(\mathbf{S}_T - \mathbf{M}_p \, \Phi\left(\mathbf{S}_{\text{out}}\right)\right)^\top \mathbf{W} \left(\mathbf{S}_T - \mathbf{M}_p \, \Phi\left(\mathbf{S}_{\text{out}}\right)\right) \right\|_{\text{F}}, \tag{3}$$

where $\mathbf{W}$ is a diagonal weight matrix where each entry is the weight from the weight map. Eq. 3 represents a weighted least squares fit between the source $\mathbf{S}_{\text{out}}$ and $\mathbf{S}_{\text{T}}$ for those pixels belonging to superpixel p.

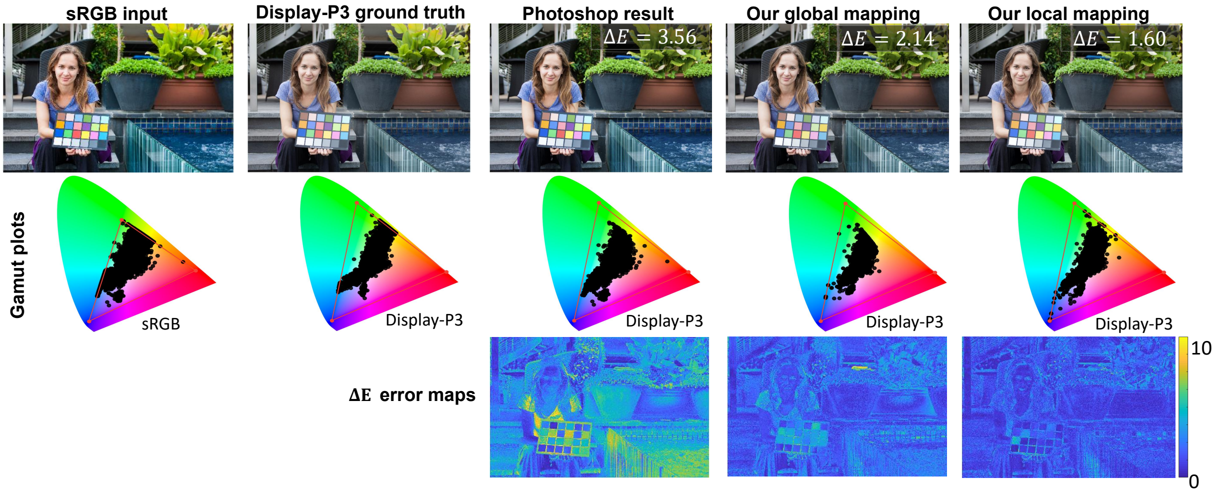

Figure 3. *An example of converting from the sRGB color space to the Display-P3 color space using global mapping and local mapping. (A) An input image in sRGB. (B) shows the ground truth image and its color gamut in the Display-P3 color space. (C) shows the color gamut of global mapping's result and the color heatmap between the global mapping's result and the ground truth image. (D) shows the color gamut of the local mapping's result and the heatmap between the local mapping's result and the ground truth image. Per-pixel CIEDE2000 errors are also shown on the heatmaps.*

The superpixel segmentation mask is upsampled to the size of the input image $\mathbf{I}_{\text{out}}$. Each pixel in $\mathbf{I}_{\text{out}}$ belonging to the upsampled superpixel p is converted to $\mathbf{I}_{\text{T}}$ using the mapping function $\mathbf{M}_p$ as described in Eq. 2. We find this simple local mapping procedure gives improved results as shown in Fig. 3. The entire process runs in $\sim$1.5 minutes using unoptimized Matlab code for a full-frame 20MB image.

4. Experimental Results

We evaluated our method quantitatively and qualitatively. We first describe the data used in our evaluation in Sec. 4.1. Then, we show the quantitative results in Sec. 4.2, followed by the qualitative results in Sec. 4.3.

4.1 Data

To evaluate our method, we randomly chose 100 RAW images from the NUS white-balance dataset [25]. Given the difficulty in accessing the intermediate image states on actual camera hardware, we instead used Adobe Camera RAW SDK to render each RAW image into the different color spaces: ProPhoto, sRGB, Adobe RGB, and Display-P3. The Adobe Camera RAW SDK mimics the camera pipeline rendering. We note that similar functionality can be obtained using other software camera pipelines (e.g., [20]). In our experiments, we selected the sRGB images to represent the display-referred output images, the ProPhoto images were used to create our sampled metadata, and the other color spaces (i.e., Display P3 and Adobe RGB) were used as target color spaces. This dataset and code will be made publicly available online upon acceptance.

4.2 Quantitative Results

In our evaluation, we used the small sRGB color gamut as our input and the corresponding Display-P3 and Adobe RGB color spaces as our target color spaces. We compared our results against a common method that converts between color profiles using Adobe Photoshop. Specifically, we used the perceptual rendering intent color conversion provided in Adobe Photoshop to convert our sRGB images into the Display-P3 and Adobe RGB color spaces. The perceptual intent aims to preserve the overall color appearance and is favored for such images that contain many out-of-gamut colors. Adobe Photoshop conversions are compared with our global fitting and local fitting results.

We adopted four widely used evaluation metrics: ΔE 2000, ΔE 76, mean angular error (MAE), and peak signal-to-noise ratio (PSNR). For each image, we computed the error/PSNR values between the converted image and the corresponding ground truth image obtained directly from the RAW rendering to the respective color space. We reported the mean, lower quartile (Q1), median (Q2), and the upper quartile (Q3) of the values of each evaluation metric over the 100 test images. Table 1 shows results of converting sRGB output images to Adobe RGB and Display-P3 color spaces, respectively. As can be seen, our proposed method achieves the best results over all evaluation metrics.

4.3 Qualitative Results

In addition to the quantitative evaluation, we evaluate our method qualitatively. Fig. 4 provides qualitative examples of our color space conversions for both Adobe RGB and Display-P3. Also shown are the results obtained by Photoshop's conversion using perceptual rendering intent. As indicated by the error maps, our method provides a significant improvement due to the availability of the ProPhoto metadata.

5. Concluding Remarks

We have presented a method to keep a sub-sampling of wide-gamut ProPhoto color encodings of a camera-captured image as auxiliary metadata to assist in color space conversion of the image. Our insight is that this wide-gamut ProPhoto data is readily available on modern camera hardware and can easily be embedded into existing saved images with minimal memory overhead (e.g., less than 100KB). While our examples focused on sRGB, Adobe RGB, and Display-P3, given their current prevalence as preferred color space encodings used by commodity cameras, our basic framework is suitable for any color space, including Rec2020. Our framework can be further improved by considering more recent vision models for wide-gamut imagery [7]. Finally, we acknowledge the support of the Canada First Research Excellence Fund for the Vision: Science to Applications (VISTA) programme and an NSERC Discovery Grant.

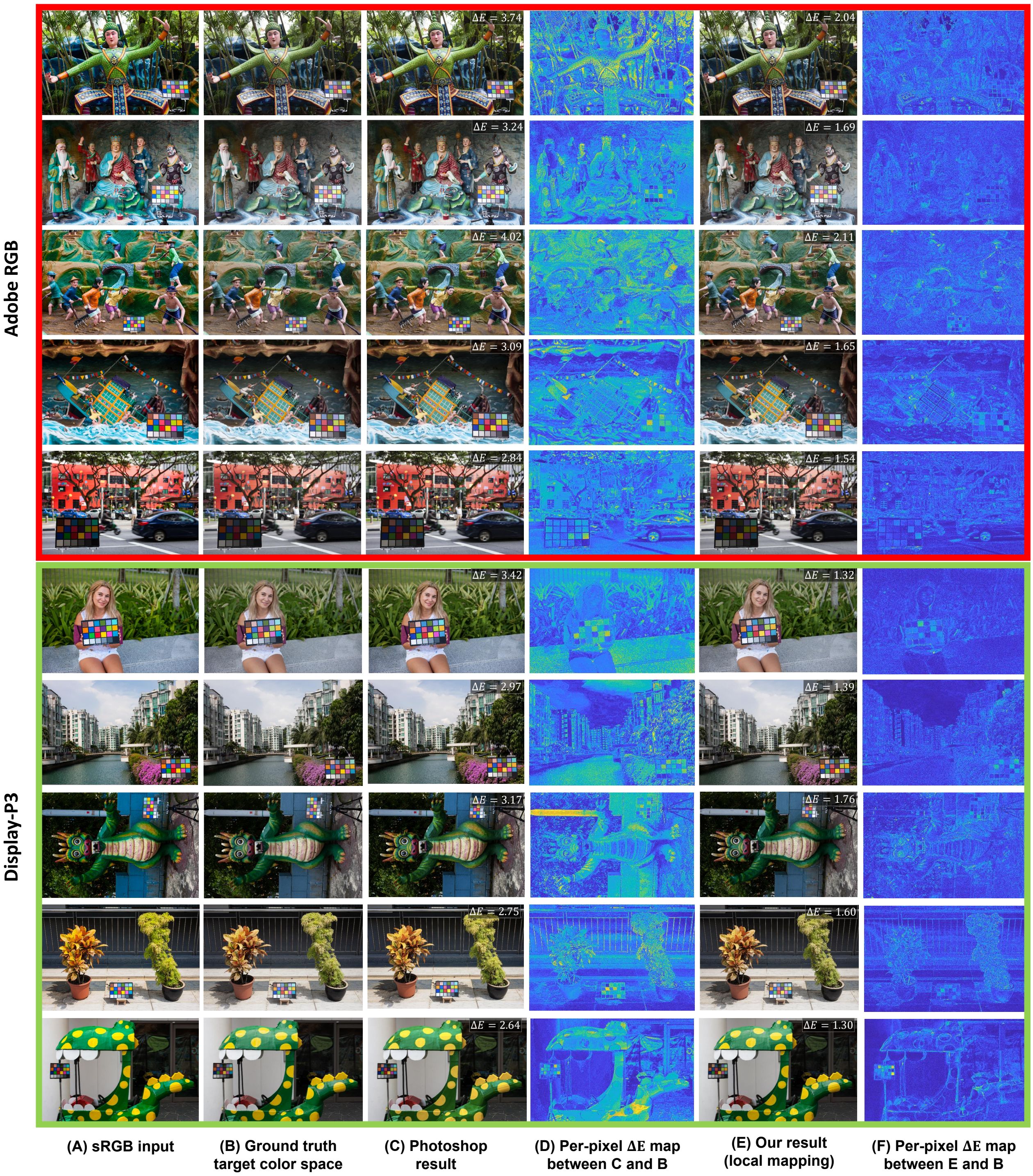

Figure 4. *Comparisons between our proposed method and Photoshop for converting from the small-gamut sRGB color space to the medium-gamut Adobe RGB and Display-P3 color spaces. (A) Input image in sRGB. (B) Ground truth images in target color spaces. (C) Result from Adobe Photoshop using perceptual rendering intent. (D) Heatmap of per-pixel ΔE 2000 error between Adobe Photoshop's results and corresponding ground truth images. (F) Our color space conversion results using the ProPhoto metadata using the local mapping method. (G) Heatmap of per-pixel ΔE 2000 between our results and corresponding ground truth images. The scale ranges between ΔE values 0 and 10.*

Table 1: Results of converting from the camera-rendered sRGB color space to the Adobe RGB and the Display-P3 color spaces. The shown results were obtained using 100 testing images randomly taken from the NUS dataset [25]. For each evaluation metric, we show the mean and the first (Q1), second (Q2), and third (Q3) quantile values. For the △E 2000 [26], △E 76, and the mean angular error, lower numbers are better, while for the peak signal-to-noise ratio (PSNR), higher numbers are better. The best results are highlighted in yellow and boldface.

Method	△E 2000 [26] ↓				△E 76 ↓				Mean Angular Error ↓				PSNR ↑			
	Mean	Q1	Q2	Q3	Mean	Q1	Q2	Q3	Mean	Q1	Q2	Q3	Mean	Q1	Q2	Q3
From camera-rendered sRGB to Adobe RGB																
Photoshop	3.07	2.65	2.99	3.47	4.96	3.90	4.53	6.32	2.81°	1.96°	2.63°	3.61°	30.72	29.51	30.84	32.07
Global Mapping (Ours)	1.85	1.64	1.87	2.05	2.51	2.05	2.40	2.89	1.81°	1.07°	1.57°	2.36°	34.40	33.22	34.23	35.64
Local Mapping (Ours)	**1.49**	**1.29**	**1.47**	**1.70**	**2.01**	**1.55**	**1.92**	**2.40**	**1.47°**	**0.85°**	**1.29°**	**1.74°**	**36.24**	**35.15**	**36.26**	**37.33**
From camera-rendered sRGB to Display-P3																
Photoshop	2.99	2.61	2.91	3.50	4.83	3.84	4.41	6.02	3.03°	2.23°	3.01°	3.60°	30.65	29.26	30.81	31.98
Global Mapping (Ours)	1.75	1.55	1.76	1.93	2.43	1.98	2.31	2.78	1.96°	1.31°	1.70°	2.51°	34.27	33.33	34.24	35.33
Local Mapping (Ours)	**1.42**	**1.23**	**1.40**	**1.60**	**1.94**	**1.51**	**1.89**	**2.31**	**1.58°**	**1.02°**	**1.37°**	**1.90°**	**36.13**	**35.25**	**36.06**	**36.95**

References

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

[2] *Adobe® RGB (1998) Color Image Encoding*, 1998 (accessed: 04.25.2020). `https://www.adobe.com/digitalimag/pdfs/AdobeRGB1998.pdf`.

[3] Sabine Süsstrunk, Robert Buckley, and Steve Swen. Standard RGB color spaces. In *Color and Imaging Conference*, 1999.

[4] Ján Morovič and Johan Lammens. Color management. *Colorimetry: Understanding the CIE system*, pages 159–206, 2007.

[5] J. Morovic and M. R. Luo. Evaluating gamut mapping algorithms for universal applicability. *Color Research & Application*, 26:85 – 102, 12 2000.

[6] Syed Waqas Zamir, Javier Vazquez-Corral, and Marcelo Bertalmío. Gamut extension for cinema. *IEEE Transactions on Image Processing*, 26(4):1595–1606, 2017.

[7] Syed Waqas Zamir, Javier Vazquez-Corral, and Marcelo Bertalmío. Vision models for wide color gamut imaging in cinema. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[8] John J McCann. Color gamut mapping using spatial comparisons. In *Color Imaging: Device-Independent Color, Color Hardcopy, and Graphic Arts VI*, 2000.

[9] Chae-Soo Lee, Yang-Woo Park, Seok-Je Cho, and Yeong-Ho Ha. Gamut mapping algorithm using lightness mapping and multiple anchor points for linear tone and maximum chroma reproduction. *Journal of imaging Science and Technology*, 45(3):209–223, 2001.

[10] Ján Morovič. *Color gamut mapping*, volume 10. John Wiley & Sons, 2008.

[11] Arne Bakke, Ivar Farup, and Jon Hardeberg. Evaluation of algorithms for the determination of color gamut boundaries. *Journal of Imaging Science and Technology*, 54(5):50502–1, 2010.

[12] Jens Preiss, Mark D Fairchild, James A Ferwerda, and Philipp Urban. Gamut mapping in a high-dynamic-range color space. In *Color Imaging XIX: Displaying, Processing, Hardcopy, and Applications*, 2014.

[13] Jan Froehlich, Timo Kunkel, Robin Atkins, Jaclyn Pytlarz, Scott Daly, Andreas Schilling, and Bernhard Eberhardt. Encoding color difference signals for high dynamic range and wide gamut imagery. *Color and Imaging Conference*, 2015.

[14] ITU-R Recommendation BT.2020. 2012.

[15] A. S. Viggiano and N. M. Moroney. Color reproduction algorithms and intent. In *Color Imaging Conference*, 1995.

[16] Kristyn Falkenstern, Nicolas Bonnier, Marius Pedersen, Hans Brettel, and Françoise Viénot. Using metrics to assess the ICC perceptual rendering intent. In *Image Quality and System Performance VIII*, 2011.

[17] Yeong-Kang Lai and Shu-Ming Lee. Wide color-gamut improvement with skin protection using content-based analysis for display systems. *Journal of Display Technology*, 9(3):146–153, 2012.

[18] Fayez Lahoud, Bin Jin, Maria Segovia, and Sabine Susstrunk. Keyword-based image color re-rendering with semantic segmentation. In *ICIP*, 2017.

[19] Kevin E Spaulding, Geoffrey J Woolfe, and Edward J Giorgianni. Reference Input/Output Medium Metric RGB Color Encodings (RIMM/ROMM RGB). *Proc. IS&T PICS*, 2000.

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

[21] Rang MH 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.

[22] 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.

[23] Guowei Hong, M Ronnier Luo, and Peter A Rhodes. A study of digital camera colorimetric characterization based on polynomial modeling. *Color Research & Application*, 26(1):76–84, 2001.

[24] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, 2012.

[25] 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.

[26] 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.