

Multi-Camera Color Correction via Hybrid Histogram Matching

Chunqiu Ding and Zhan Ma, *Senior Member, IEEE*

Abstract—With the explosive popularity of multi-camera systems, a challenging problem is how to guarantee the inter-camera color consistency. In this paper, we have developed a *hybrid histogram matching* (HHM) algorithm that leverages the cumulative color histogram to fulfill this purpose, in which it mainly includes the *global color mapping*, and *local color straightening* for uniform color presentation across cameras. Our approach is applicable to any camera pair with sufficient viewing angle overlap, and can be easily extended to support multi-camera setup by pairwisely color map transfer. This HHM method has provided the superior efficiency for color correction when compared with the state-of-the-art algorithms, both objectively and subjectively, via extensive validations on different multi-camera systems. Additional ablation studies have been performed to further examine the capacity of our algorithm in practical applications. We would like to make all materials publicly accessible at <https://njuvision.github.io/HHM/> for reproducible research.

Index Terms—Multi-camera color correction, histogram matching, global color mapping, local straightening

I. INTRODUCTION

Recently, we have witnessed the explosive growth of imaging techniques, including the ultra high-definition photography with spatial resolution from million-pixel to gigapixel [1], high dynamic range snapshot [2], high-speed video acquisition [3], light-field imaging [4], [5], etc. A fundamental architecture that enables these advanced imaging processes is the *multi-camera* design, by which images or videos captured by each microcamera will be synthesized (e.g., stitched) computationally for final reconstruction with extended granularity. Multi-camera systems, however, often face the inconsistent color responses across sensors, leading to unexpected processing errors (e.g., misalignment in stitching) or inferior performance (e.g., non-uniform color rendering with unpleasant quality of experience (QoE)) in subsequent steps. Thus, color correction is critical for ensuring the consistent light radiance response across cameras. Note that color correction is also referred to as the *color transfer* or *alignment* in literature [6]. It can be utilized in many computer vision, and image processing tasks including the stylistic transfer, tone mapping, stitching, etc, for any pair of images without constraints. This work, more specifically, puts the focus on corrections of color images from the physical multi-camera system used in practices.

Copyright © 20xx IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending an email to pubs-permissions@ieee.org.

This paper is supported in part by National Natural Science Foundation of China (62022038). (Corresponding Author: Z. Ma)

C. Ding and Z. Ma are with the Nanjing University, Nanjing, Jiangsu 210093 China. E-mails: MF1823085@mail.nju.edu.cn, mazhan@nju.edu.cn

A. Observations

The key step of multi-camera (e.g., number of cameras $n \geq 2$) color correction is how to effectively align colors between two of them for uniform and appealing presentation. A reasonable assumption is often made to let the camera pair have sufficient overlap of their respective viewing coverage (or viewport), by which we can leverage the signal statistics of overlapped areas from both camera snapshots (e.g., image, video) to perform the color correction. Recent off-the-shelf multi-camera systems, such as the dual-camera, or triple-camera smartphones, etc, well support aforementioned hypothesis. We then assume the cameras are aligned [7] with paired instantaneous snapshots, leaving the emphasis of this work solely on the analysis and development of color correction algorithm.

We refer one camera to as the *source* or *reference* camera, and the other one to as the *target* or *test* camera, by which respective snapshots are noted as *source* \mathbf{I}_s and *target* \mathbf{I}_t images. Overlapped image regions or areas are then noted as \mathbf{O}_s and \mathbf{O}_t accordingly. Ideally, we wish to find an accurate mapping or projection function to correct the target image using the source statistics (e.g., color distribution). Following the conventions, images are presented in popular RGB color space, and mapping functions are produced for each channel c separately.

Common practices may apply the color correction or transfer matrix M that can be derived using the means (e.g., first-order moments) and/or variances (e.g., second-order moments) [8]–[10], or using the least-squared-error fitting [11]–[13], of the pixels from both \mathbf{O}_s and \mathbf{O}_t (or even \mathbf{I}_s and \mathbf{I}_t for some application¹). Other explorations may devise the mapping function fitted using splines between matched feature points (e.g., speeded up robust features - SURF points [14]) [15]). However, it is impossible to have pixel-to-pixel alignment in overlapped area, leading to inevitable color errors presented in noticeable block segments (e.g., color banding) [6], [16], especially for the image with saturated pixels. This would severely degrade the image QoE after color correction, and may also lead to incorrect inter-camera stitching. Recently, cumulative histogram matching-based algorithms [17], [18] have emerged and shown their advantages to leverage the full spectrum of pixel distributions with the tolerance of pixel drifting (e.g., geometric location, intensity, etc) to some extent.

¹In this work, we mainly use overlapped areas for subsequent color correction in multi-camera systems.

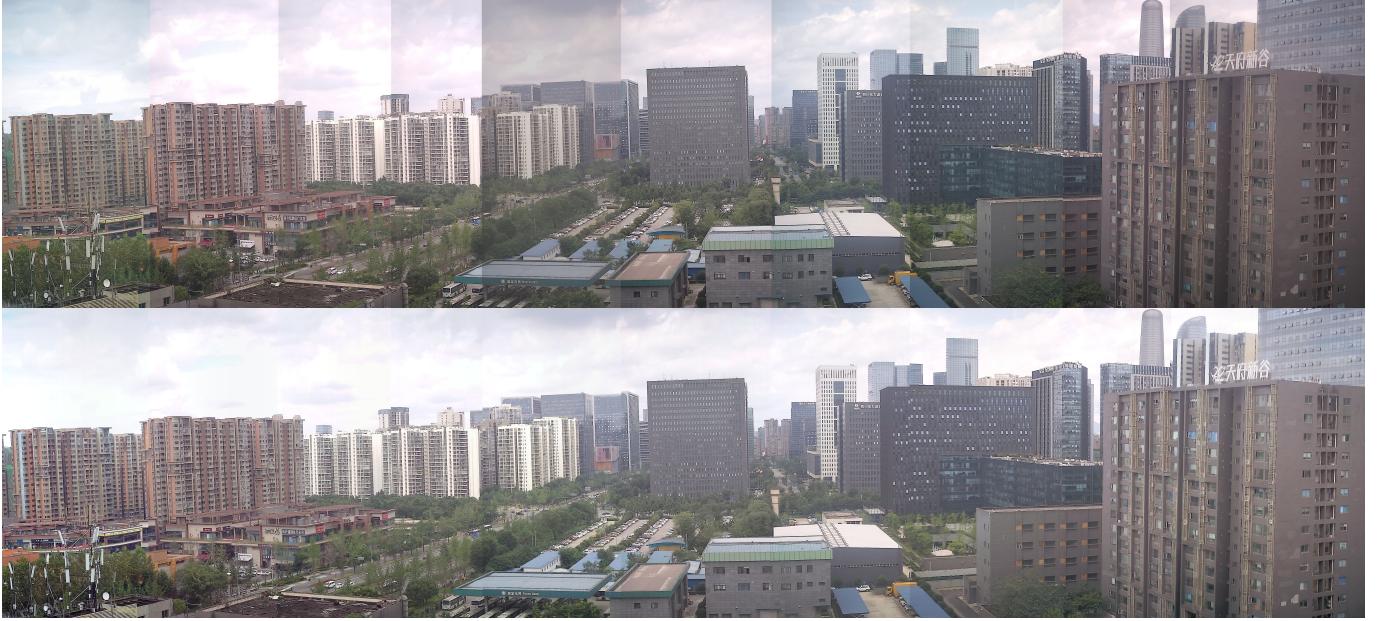


Fig. 1. **Color-corrected Snapshot.** An example of panoramic image with and without color correlation. *Top:* stitched image before color correction; *Bottom:* stitched image after color correction;

B. Our Approach

In this work, we have attempted to revisit this problem, and propose the *hybrid histogram matching* (HHM) to generate more accurate color mapping function for better reconstruction, both objectively and subjectively. Our HHM model is basically an example of cumulative histogram matching-based approaches. It includes hybrid *global mapping* and *local straightening* to better characterize the color statistics for overall efficiency improvement.

More specifically, we first utilize the cumulative histograms of both \mathbf{O}_s and \mathbf{O}_t to derive a global color mapping function f_{map} . Note that we generate individual f_{map} for separate color channels (e.g., R, G, B)². Given the limited bit precision used in practice, e.g., 8-bit or [0, 255] of integer pixels, we can facilitate the f_{map} using a lookup table where each table element is referred to as the projection or mapping pair between respective target and source pixel and corresponds to an specific 2D point sampled from the f_{map} curve.

Recalling that it typically presents imaging variations across cameras, such as the sensor difference, illumination change, etc, mapping errors are often encountered. A typical case is the staircase phenomena that would produce the color banding blocks, leading to unpleasant visual quality sensation. Thus, we suggest to remove these staircase related noisy mapping pairs in default f_{map} , and then directly connect neighbor points using straight lines if there are vacancy in between. We simply call this local straightening. In the end, we apply an averaging-based smoothing filter via sliding window for final mapping function or lookup table derivation.

We can easily extend such color correction from a dual-camera setup to multi-camera system, where we can manually

define the *source* label to transfer all images captured by different cameras for consistent color presentation. This *source* label can be from a specific camera, or even an averaged mapping function from all possible camera pairs. Figure 1 visualizes the snapshots after color correction where averaged mapping function is applied for this 10-camera panoramic imaging system.

Our HHM is evaluated extensively using practical multi-camera system. It has demonstrated significant performance gains, when compared with existing color correction algorithms in [8], [15], [17], [19], [20]. Objectively, the HHM is superior to the state-of-the-art color correction methods in terms of multiple popular measurement metrics. On the other hand, thanks to the hybrid global and local color mapping schemes, proposed HHM can effectively resolve mapping errors for improved visual presentation. Additionally, we have performed the ablation studies to investigate a variety of aspects of our algorithm, e.g., overlapping ratio, complexity, etc, for understanding the capacity in practical multi-camera systems.

The rest of this paper proceeds as follows: In Sec. II, we give a short review of relevant works on color correction methods and evaluation metrics; Sec. III details our proposed method, followed by the comparative experiments in Sec. IV. Additional ablation studies are performed in Sec. V to further examine the capabilities of our proposed method; Final conclusion is drawn in Sec. VI.

II. RELATED WORK

This section reviews the advancements in color correction approaches, as well as the efficiency evaluation methodologies.

²Though it is more accurate to use f_{map}^c as the color mapping function of channel c , we, for the sake of simplicity, sometimes directly utilize the f_{map} instead.

TABLE I
A BRIEF SUMMARY OF COLOR CORRECTION ALGORITHMS

Algorithm	Color Space	Description
R [8]	$l\alpha\beta$	Use the mean and variance of two images for color mapping matrix
AM [12]	RGB	Apply least squared minimization on input two images for color mapping matrix
HM [17]	RGB	Utilize the cumulative histograms of overlapped areas from two images for color mapping function
G [19]	RGB	Approximate the color mapping function assuming the Gaussian distributed histogram of two images
3MS [6]	RGB	Derive 6-node spline model for color mapping using paired SIFT points from overlapped areas
GPS [15]	RGB	Generate 4-knot spline model with gradient preserving feature from overlapped areas for color mapping
Proposed HHM	RGB	1) Use cumulative histograms from overlapped areas for global mapping function; 2) Remove noisy mapping pair via re-ordered cumulative histogram; 3) Directly connect neighbor points if there are vacancies in between for local straightening

A. State-of-the-art Color Correction Algorithms

In principle, color correction algorithm can be divided into two major computational modules consecutively: one is probing and estimating the mapping function, and the other one is applying the mapping function for correction. Popular color correction methodologies can be categorized depending on how to effectively generate the function for color mapping. A great amount of details can be found in [15], where a thorough study is extensively conducted by introducing a compositional framework to dissect and reassemble computational modules in color correction algorithm. Therefore, we give a brief glance in this paper instead, especially for those algorithms utilized for later experimental comparison.

For example, Reinhard *et. al* [8] suggested to use the mean and variance of paired images for color transfer, which is noted as “R” subsequently. Ilie [11] utilized the linear least-squared-error minimization to determine the transfer matrix coefficients as the refinement to its hardware-based color calibration. Both methods used a very limited statistical information (mean, variance as exemplified) of pixel colors.

It is expected to have better mapping efficiency when exploring more statistics. Thus, Fecker *et. al* [17] proposed a histogram matching method to fully utilize the image color information, where the mapping function was generated by comparing cumulative histogram of source and target images. It is referred to as the “HM” scheme which is a global color mapping strategy. As will be revealed in experimental studies, such “HM” method has presented noticeable color banding distortions due to imperfect pixel-to-pixel matching.

Later then, Oliveira *et. al* [19] proposed to approximate the histogram using the Gaussian mixture model by which both mean and variance were estimated statistically to generate histogram matching function for color transfer. This is noted as “G” algorithm for further experiments.

Moreover, color mapping function could be represented using splines that were estimated from each channel independently. One example was given in [20], which was regarded as “3MS” method. Spline-based mapping was then improved by introducing the gradient preserving feature for better performance as reported in [15] (noted as “GPS”). Such “GPS” scheme was accelerated by coarse-to-fine spline search, leading to a fast implementation - “FGPS”. Both “GPS” and “FGPS” were reportedly having the state-of-the-art efficiency by examining hundreds of compositional combinations in [15].

Table I gives a brief technical summary. More comprehensive comparisons of different color correction or transfer

TABLE II
NOTATIONS

item	description
$\mathbf{I}_s, \mathbf{I}_t, \mathbf{I}_{cc}$	source, target, & color corrected images
c	color channel indicator, $c \in \{R, G, B\}$
i	pixel intensity, $i \in \{0, 1, 2, \dots, 255\}$ for 8-bit precision
\mathbf{O}_s	overlapped region in source image
\mathbf{O}_t	overlapped region in target image
\mathbf{H}_s^c	histogram of \mathbf{O}_s at c channel
\mathbf{H}_t^c	histogram of \mathbf{O}_t at c channel
\mathbf{CH}_s^c	cumulative histogram of \mathbf{O}_s at c channel
\mathbf{CH}_t^c	cumulative histogram of \mathbf{O}_t at c channel
f_{map}^c	color correction mapping function
η	predefined percentile for removing noisy points on f_{map}^c
n_w	number of points for sliding window

algorithms could be referred to [15], [16], [20].

B. Performance Evaluation Metrics

There are several metrics widely adopted to quantitatively evaluate the efficiency of color correction algorithms, such as the PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity) [21], FSIM (Feature Similarity Index) [22], and more recent iCID (improved Color Image Difference) [23]. The higher PSNR, SSIM, and FSIM come with the better performance; while the lower iCID offers the better efficiency.

Following the suggestions in [15], we apply the averaged indices to measure the performance of different color correction methods, e.g.,

$$M_g = \frac{1}{2}(M_s + M_t), \quad (1)$$

with M as one of four referred metrics above. Taking PSNR as an example, M_s is the PSNR measured using the pixels from the overlapped area of source image and its color corrected sample; and M_t is the corresponding PSNR for the overlapped area in target image and its corresponding color corrected representation. Averaging the measurement from both source and target samples in (1) is also used to guarantee the color consistency across inter-cameras.

Despite these objective metrics, the visual quality is still an important factor for measuring the efficiency of color correction algorithm. In this study, thus, we also offer the subjective evaluation of the proposed algorithm by visualizing the reconstructions.

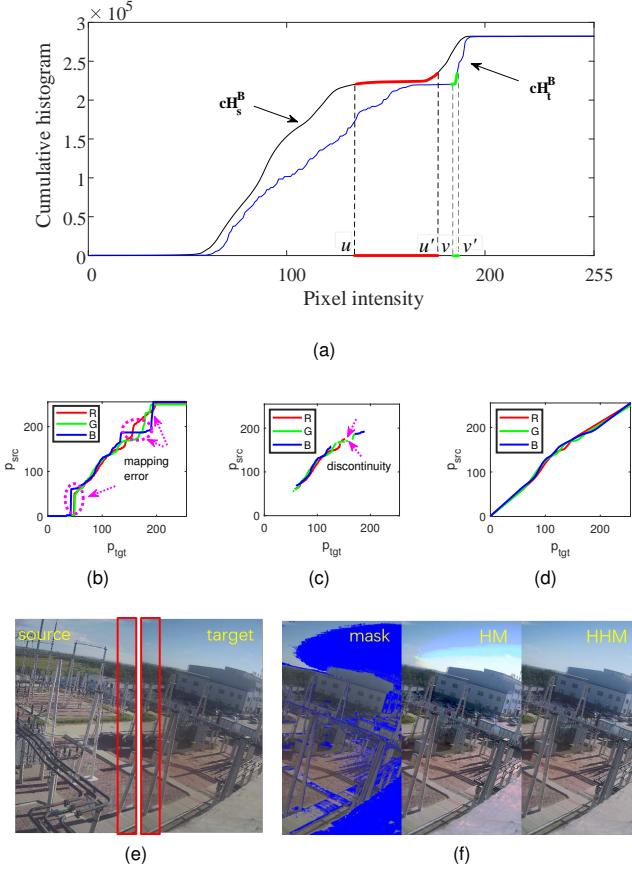


Fig. 2. Hybrid Histogram Matching. (a) cumulative histograms \mathbf{cH}_s^B and \mathbf{cH}_t^B of respective overlapped areas \mathbf{O}_s and \mathbf{O}_t from source and target images in (e); Only Blue channel is visualized for clear presentation; (b) global color mapping function f_{map}^c with staircase mapping errors; (c) noisy mapping pair removal using re-ordered cumulative histograms; (d) local straightening by directly connecting neighbor points if there are vacancies in between; (e) source and target images with highlighted overlapping areas; (f) a masked image to underline the region in the image corresponding to the problematic color bins, and the color corrected images using respective HM [17] and proposed HHM methods. All steps can be implemented using a lookup table. In subplots (b), (c) and (d), we simply use P_{tgt} and P_{src} to respectively represent the pixel intensity in target and source images.

III. HYBRID HISTOGRAM MATCHING: GLOBAL MAPPING AND LOCAL STRAIGHTENING

Our algorithm consists of the *global color mapping* and *local straightening*. Both steps utilize the cumulative histogram of overlapped image areas from paired cameras.

General procedure is briefly presented in Algorithm 1. Note that in this work, we perform the color correction separately for each channel c , e.g., R, G, B as exemplified, and we begin with the dual-camera example.

A. Dual-Camera Color Correlation

First, we resort for feature-point-based algorithms to determine the \mathbf{O}_s^c and \mathbf{O}_t^c from respective source and target images. Popular SURF [14] is used in this paper, and other approaches would have the similar performance. Pixels in \mathbf{O}_s^c and \mathbf{O}_t^c are utilized to derive cumulative histograms \mathbf{cH}_s^c and \mathbf{cH}_t^c that will be used to generate the global mapping function f_{map}^c subsequently shown from line #3 to #13 in Algorithm 1.

Similar cumulative histogram-based color correction has been utilized in [17] as well for improving the multiview video compression. Ideally, overlapped areas \mathbf{O}_s^c and \mathbf{O}_t^c should exhibit almost the same color distribution because the same view coverage is captured. However, due to the imaging variations (e.g., sensor difference, illumination changes, etc), it is difficult or even impossible to have exact correspondences in overlapped areas. Thus, instead of applying the exact pixel-to-pixel mapping, we turn to the cumulative histogram that characterizes the full spectrum of pixel color intensity, by which data statistics are learned and mapped from the source image to the target one. Later then, for any pixel v from the target image, it can be mapped to $u = f_{\text{map}}^c(v)$ to satisfy the source image statistics.

Both theoretical analysis and experimental simulation have revealed that a small percentage of pixel drifting in either location, or intensity, would yield staircase phenomena in global mapping function f_{map}^c , leading to inevitable mapping errors (e.g., color banding blocks). We call these “noisy mapping pairs” for simplicity. For example, as visualized in Fig. 2a, pixels from v to v' in target image, are enlarged noticeably to the range of $[u, u']$, yielding staircase errors shown in Fig. 2b. Such phenomenon is often presented because of the sensor differences, illumination variations, across paired snapshots for color correction. We propose to devise *local straightening* scheme to resolve the staircase mapping issue (see line #15 in Algorithm 1 and corresponding detailed functional steps in Algorithm 2). Such a linear interpolation-based straightening approach is simple, yet efficient, as reported by our extensive experiments.

As demonstrated in our experiments, we have observed that aforementioned staircase mapping errors (see Fig. 2b) are often introduced when such pixels having the specific intensities occupy a small percentile in either target or source image areas (e.g., pixels in $[v, v']$ mapped to $[u, u']$ of Fig. 2a). Therefore, as illustrated from line #3 to #6 in Algorithm 2, we re-order the original histogram \mathbf{H}^c from the pixel bin having the least occurrences, to the one having the most outcomes, e.g., $\tilde{\mathbf{H}}^c[i_0] \leq \tilde{\mathbf{H}}^c[i_1] \leq \dots \leq \tilde{\mathbf{H}}^c[i_{255}]$, and record the total number of pixels as m^c . We then can provide the cumulative histogram upon the re-ordered $\tilde{\mathbf{H}}^c$ as \mathbf{cH}^c . In this work, we apply the simple thresholding with a parameter η , by which we can determine the pixels that present cumulative histograms less than predefined level, i.e., $\eta \cdot m^c$. These pixels are marked as the noisy bins that are removed from the mapped function f_{map}^c , together with its mapping correspondence. η is set to 5% in this paper according to our extensive simulations. Note that the same procedure is applied for histograms of both source and target images, as shown in Algorithm 2 from line #7 to #14. Figure 2c illustrates the mapping function after noisy pair removal. We then directly connect neighbors with straight line to have the straightened mapping function \tilde{f}_{map}^c .

To ensure the smooth color mapping, we also apply the smoothing operations by averaging n_w points in a sliding window to finally produce the \hat{f}_{map}^c as depicted in Fig. 2d. Such window smoothing is widely utilized in image processing algorithms. $n_w = 15$ for the well trade-off between efficiency and complexity. In order to understand the performance impact

Algorithm 1 Hybrid Histogram Matching-based Dual-Camera Color Correction HHM()

Input: $\mathbf{O}_t, \mathbf{O}_s$; //Overlapped region of respective target and source images. Note that following steps will be iterated consecutively for R, G, B channels.

Output: $f_{\text{map}}^c, c \in \{\text{R}, \text{G}, \text{B}\}$.

- 1: **for** $c \in \{\text{R}, \text{G}, \text{B}\}$
- 2: //global color mapping.
- 3: $\mathbf{O}_t^c \rightarrow \mathbf{H}_t^c; \mathbf{O}_s^c \rightarrow \mathbf{H}_s^c$; //derive histogram
- 4: $\mathbf{H}_t^c \rightarrow \mathbf{cH}_t^c; \mathbf{H}_s^c \rightarrow \mathbf{cH}_s^c$; //drive cumulative histogram
- 5: **for** $0 \leq v \leq 255$ //8-bit pixel precision assumption
- 6: **while** $u \leq 255$
- 7: **if** $\mathbf{cH}_s^c(u) \leq \mathbf{cH}_t^c(v) < \mathbf{cH}_s^c(u + 1)$
- 8: $f_{\text{map}}^c(v) = u$;
- 9: **break**;
- 10: **end if**
- 11: $u++$;
- 12: **end while**
- 13: **end for**
- 14: //local noisy mapping pair straightening.
- 15: $\tilde{f}_{\text{map}}^c = \text{LocalStraightening}(f_{\text{map}}^c, \mathbf{O}_t, \mathbf{O}_s)$;
- 16: //mapping function smoothing
- 17: $\hat{f}_{\text{map}}^c = \text{WindowSmoothing}(\tilde{f}_{\text{map}}^c, n_w)$; // n_w :# of points in smoothing window.
- 18: **end for**

of smoothing process, we have separately tested the HHM algorithm without window smoothing, which is quoted as HHM_NS below.

In practical implementation, given the integer representation of pixel intensity, e.g., from 0 to 255 for 8-bit precision, we can utilize the look-up tables to facilitate the mapping functions for different color channels, where the length of table is determined by the pixel bit-width.

B. Multi-camera Extension

Our HHM can be easily extended from aforementioned dual-camera setup to any multi-camera systems. There are many possible geometric layouts to configure multiple cameras. In principle, we can label any camera as the source, and derive the mapping function starting from its closest neighbor to the farthest one in a predefined order. For any target camera, it would be associated with a particular mapping function, given a specific source camera. It then can lead to multiple mapping functions, by setting different source labels. For example, if we have ten micro-cameras in total, we could have ten mapping functions for each one by iteratively setting each individual camera as the source. These ten mapping functions can be averaged as well for color corrections. For the scenario current camera itself is applied as the source, its mapping function is $y = x$. Applying either a specific color mapping function with a particular source camera or an averaged function, can be fully user oriented in our design. Figure 1 shows the reconstruction quality after color correction using the averaged mapping function, while the following Fig. 8 offers different image rendering results using various color mapping functions.

Algorithm 2 Local Noisy Mapping Pair Straightening Local-Straightening()

Input: $f_{\text{map}}^c, \mathbf{O}_s, \mathbf{O}_t$

Output: $f_{\text{map}}^c, c \in \{\text{R}, \text{G}, \text{B}\}$. //local straightened color mapping function.

- 1: **for** $c \in \{\text{R}, \text{G}, \text{B}\}$
- 2: //noisy mapping pair derivation
- 3: $\mathbf{O}_t^c \rightarrow \mathbf{H}_t^c; \mathbf{O}_s^c \rightarrow \mathbf{H}_s^c$; //derive histogram
- 4: $\mathbf{H}_t^c \rightarrow m_t^c; \mathbf{H}_s^c \rightarrow m_s^c$; //derive the total number of pixels
- 5: $\mathbf{H}_t^c \rightarrow \tilde{\mathbf{H}}_t^c; \mathbf{H}_s^c \rightarrow \tilde{\mathbf{H}}_s^c$; //re-order the pixel bins from the default $\{0, 1, \dots, 255\}$ to $\{i_0, i_1, i_2, \dots, i_{255}\}$ with their histogram from the least to the most one. $\tilde{\mathbf{H}}^c[i_0] \leq \tilde{\mathbf{H}}^c[i_1] \leq \dots \leq \tilde{\mathbf{H}}^c[i_{255}]$
- 6: $\tilde{\mathbf{H}}_t^c \rightarrow \mathbf{c}\tilde{\mathbf{H}}_t^c; \tilde{\mathbf{H}}_s^c \rightarrow \mathbf{c}\tilde{\mathbf{H}}_s^c$; //derive cumulative histogram
- 7: **for** $k = 0, 1, \dots, 255$ //derive noisy pixel bins for removal.
- 8: **if** $\mathbf{c}\tilde{\mathbf{H}}_t^c(k) < \eta \cdot m_t^c$ // η noisy level in percentile.
- 9: $\vec{i}^t+ = i_k^t$;
- 10: **end if**
- 11: **if** $\mathbf{c}\tilde{\mathbf{H}}_s^c(k) < \eta \cdot m_s^c$
- 12: $\vec{i}^s+ = i_k^s$;
- 13: **end if**
- 14: **end for**
- 15: $f_{\text{map}}^c \rightarrow \tilde{f}_{\text{map}}^c$; //remove \vec{i}^s and \vec{i}^t associated pairs and connect neighbors with straight line if there are missing points in between.
- 16: **end for**

IV. EXPERIMENTAL COMPARISON

Multi-camera Prototypes. In practice, it is difficult to have two completely identical imaging sensors. Multi-camera system often faces the inconsistent color responses across sensors. Therefore, in order to ensure the generalization of our proposed HHM algorithm, we have built and deployed seven multi-camera systems at seven different places in Southern, Northern and Western China. Due to the viewing range coverage variations in field, the total numbers of microcameras are slightly different (e.g., one 6-camera, four 7-camera, two 10-camera systems), but the sensors are using the same Sony IMX385 chipset with the 1080p 60Hz video acquisition support. The overlapping area is about 10% for neighbor microcameras. We shoot nine scenes by using these seven systems. Additional two scenes are captured using a 7-camera and a 10-camera system respectively at another time instance with different ambient light illumination.

Dataset. Taking “Xingu2” scene as an example, let us pickup two adjacent images (a.k.a., image pair), such as #8 and #9 subplots below. The process of generating image pairs is stepwisely illustrated in Fig. 3 as follows: 1) For each image pair, we first apply the SURF detector [14] to find feature point pairs for homography-based image alignment; 2) The aligned pair of images can be utilized as source and target scene respectively. For exemplified “Xingu2”, it has nine native image pairs, leading to 18 pairs in total for color correction since each individual image can be either source

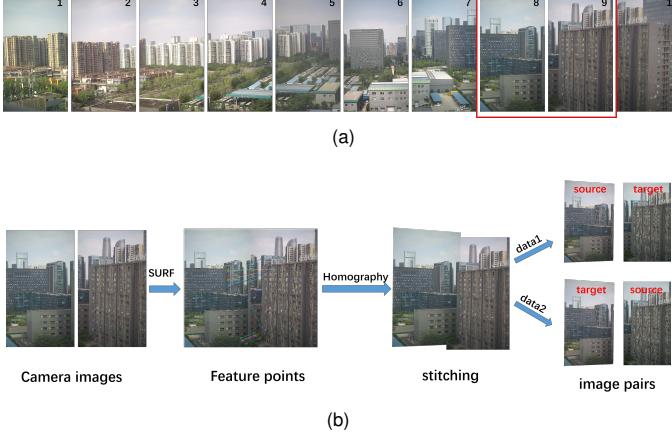


Fig. 3. **Dataset Example Using “Xingu2” scene.** (a) native image pairs in a 10-camera system; (b) stepwise processing for test dataset generation.

or target scene. We can have 124 pairs (e.g., $124 = 3 \times 18 + 5 \times 12 + 10$) from all seven multi-camera systems. However, for some neighboring images with less texture details, the SURF detector fails to extract sufficient feature points for subsequent processing. We then remove these images, and finally have generated 110 effective pairs as a dataset for color correction test. In principle, this dataset covers the characteristics from different cameras, illumination conditions and local details for well justifying the efficiency of various color correction algorithms.

Efficiency. We first report the efficiency of our methods in comparison to other popular color correction algorithms, e.g., R [8], AM [12], HM [17], G [19], 3MS [20], FGPS and GPS [15]. For each method, we apply the same and simple global mapping (GL) scheme as the color map Prober and Aggregator (PA) following the suggestions in [15]. It will be an interesting topic to explore other PA methods in future study. Table III lists the objective performance measured by PSNR, SSIM, FSIM, as well as the iCID for aforementioned algorithms, on average. Overall, our HHM has demonstrated the state-of-the-art efficiency, with noticeable performance margin against all these popular methods. By excluding the HHM, the HHM_NS is still ranked at the first place. In addition, we have visualized sample snapshots of color corrected reconstructions in Fig. 4. Other scenes have the similar presentation. Our HHM has provided significant perceptual improvements to other methods, with uniform color presentation. Particularly, color banding can be effectively mitigated when comparing subplots Fig. 4(b), (g) and (h).

Though HM method in [17] offers the better objective scores than FGPS and GPS in [15], as shown in Table III, it has obvious color banding artifacts in Fig. 4(g) with inferior visual quality to Fig. 4(e) and (f) of respective FGPS and GPS algorithms. On the contrary, our HHM offers the consistent quality both objectively and subjectively.

V. ABLATION STUDIES

This part will give more details by examining different aspects of our proposed HHM algorithm.

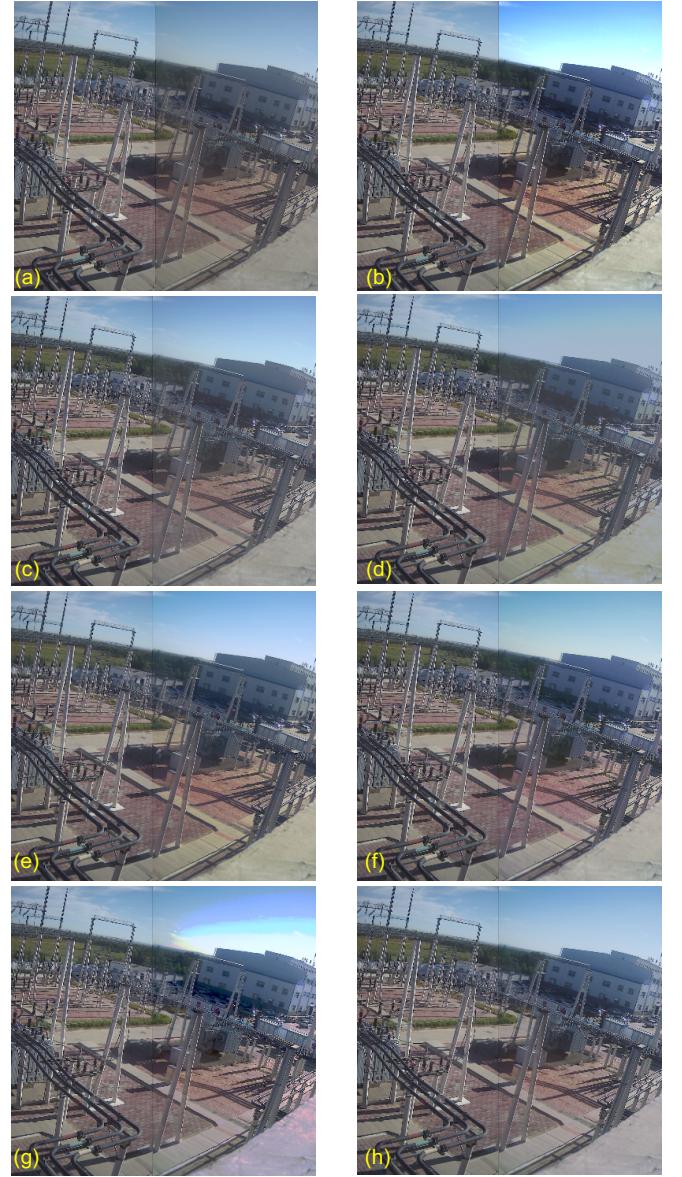


Fig. 4. **Subjective Comparison.** (a) native image pair without color correction (left: source; right: target); and color corrected targets (right part) using (b) R [8]; (c) 3MS [20]; (d) G [19]; (e) GPS [15]; (f) FGPS [15]; (g) HM [17]; (h) proposed HHM. In subplots (b) and (g), color banding issue is clearly observed; while (c) and (d) fail to correct the grayish color in native target of (a).

A. Hybrid Global and Local Pipeline

The proposed HHM algorithm includes the both global color map derivation and local color straightening. Such hybrid processing pipeline could greatly improve the reconstruction quality by actively learning the accurate mapping function. Figure 5 illustrates the side-by-side comparison for three different scenarios, where first row is the reconstruction with only global color mapping-based color correction, and second row is the reconstruction with both global mapping and local straightening operations. Zoomed areas are also included for better visualization of local details. It clearly shows that our HHM can effectively remove color banding (see zoomed

TABLE III

OBJECTIVE EVALUATION OF DIFFERENT COLOR CORRECTION METHODS
(THE BEST PERFORMANCE IS IN RED; AND THE SECOND BEST IS IN BLUE)

Method	PSNR↑	SSIM(%)↑	FSIM(%)↑	iCID(%)↓
R/GL [8]	24.37	85.78	92.22	24.31
AM/GL [12]	24.11	81.71	89.06	26.09
HM/GL [17]	26.67	86.88	93.16	20.36
G/GL [19]	24.32	83.38	89.92	25.39
3MS/GL [20]	24.41	82.74	89.55	26.90
FGPS/GL [15]	25.60	86.56	92.74	24.15
GPS/GL [15]	25.92	86.85	93.09	21.84
Prop. HHM_NS/GL	26.90	86.99	93.25	20.14
Prop. HHM/GL	26.94	87.07	93.32	19.97

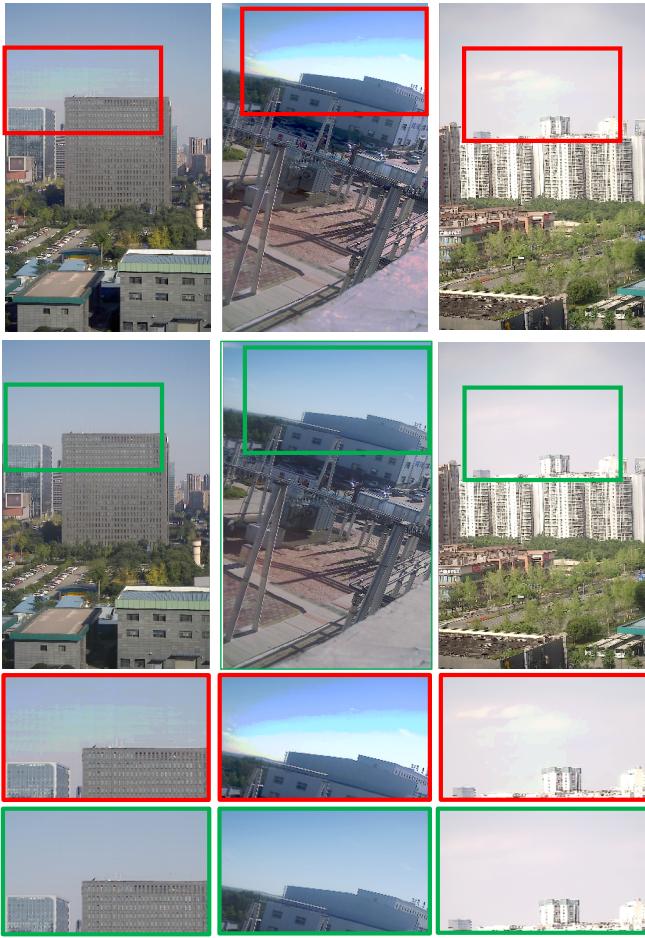


Fig. 5. **Impact of global mapping & local straightening in HHM.** Visualization of color corrected reconstructions with global-mapping-only (First row and zoomed area in third row), and with both global mapping and local straightening (second row and fourth row). Color banding can be clearly observed in reconstructions using global-mapping-only algorithm.

snapshots in Fig. 5) with better and more realistic presentation. Note that the HM method suggested by Fecker *et. al* in [17] is an example of global mapping-based color correction. Thus, this part further shows the superior performance of our HHM to the HM scheme [17].

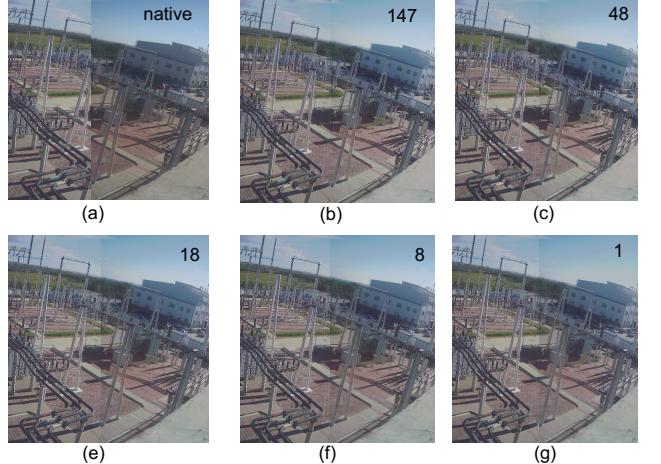


Fig. 6. **Impact of overlapped pixel-column.** (a) native image pair before color correction; (b)-(g) color corrected reconstructions when having different overlapping areas. Overlapping area is measured by the number of columns in pixel width (a.k.a., pixel-column). For example, pixel-column = 8 means that 8 columns of source and target images are overlapped.

B. Overlapped Pixels

The HHM relies on the overlapped areas to derive the mapping function for subsequent color transfer. We wish to accurately learn the color mapping with overlapped pixels as less as we can. Without losing generality, we can measure the overlapping percentile using the number of columns in pixel width, a.k.a., pixel-column. For panoramic imaging with multi-cameras, the smaller pixel-column, the less overlapped area.

We visualize the color correction performance in Fig. 6 where Fig. 6(a) is the native presentation of source and target pair without color correction, and subplots from Fig. 6(b) to Fig. 6(g) provide color corrected reconstructions when having different overlapping percentiles measured via pixel-column. For example, “pixel-column = 147” is for the case having 147 columns overlapped for image snapshots captured by neighbor cameras. An extreme scenario is also provided when pixel-column = 1.

Along with the decreasing of pixel-column, we can see that the color consistency after correction is gradually degraded. But it is still acceptable even when pixel-column = 1, in comparison to the native presentation before color correction in Fig. 6(a). In this example, original snapshot from each camera is with the 1920×1080 (1080p) resolution. For pixel-column = 147, overlapped area occupies < 15% pixels to the original image, while pixel-column = 48 only requires < 5% pixels in overlapped area. In practices, users can manually adapt the pixel-column for application driven optimization.

C. Ambient Light Consistency

Ambient light changes from time to time. We have sampled the snapshots at five different time instants from the morning to the dusk, e.g., from t_1 to t_5 , as visualized in Fig. 7. In this example, we are deriving the color mapping function at t_1 , and proposing to apply this function to all other time instants. As we can see, even with very different ambient light conditions,

a single mapping function can still offer the color correction capability to some extent. However, the reconstruction quality may suffer from poor appearance, especially for the large time lapse, such as the color distortions in buildings at t_5 . It therefore suggests the time-dependent mapping function. The complexity of color correction deployment is closely related to its granularity in time domain. Our experiments have then shown that a single mapping function can be applied for a couple of hours, with satisfying image quality presentation. For a 24-hour duration, it just requires less than ten individual maps. This can be easily facilitated by updating the mapping function or lookup table on-the-fly.

D. Multi-label Consistency

In a multi-camera system, we have the flexibility to choose different source labels for color correction. Figure 8 depicts the native snapshots (before color correction), and color corrected illustrations from a 10-camera imaging device prototype. We set different labels for correction as shown in Fig. 8(b) and (c), where bluish effect of #1 camera and grayish color tone of #5 are learned and consistently transferred to the final reconstruction. Oftentimes, we can utilize averaged mapping table for color transfer, as shown in Fig. 8 and Fig. 1. Such averaging across different source labels could balance the image quality globally. Nevertheless, these experiments have also shown that for any specific source label, our HHM has presented consistent efficiency to provide uniform quality. In practical applications, users can choose their personalized preference by adapting different labels.

E. Multi-Camera Imaging with Reference

There are mainly two types of multi-camera structures. One is leveraging the neighboring overlaps to perform the stitching. Our 10-camera prototype is one example in this category; The Insta 360 Pro³ is another example. Alternatively, a reference wide-view camera can be utilized to help the projections of individual narrow-view micro-cameras for a ultra-high-definition videography [1]. The Mantis 70 is an example belonging to this class, where 18 4K narrow-view micro-cameras (e.g., 3 rows and 6 ones in each row) and one 4K wide-view camera (in the middle) are used, producing 102 million-pixels video at 30 Hz⁴. We simply call later one as the multi-camera imaging with reference.

For such multi-camera system with reference, we can transfer the color of middle reference camera to all other narrow-view micro-cameras for final uniform color response. Figure 9 illustrates the color corrected reconstruction, where we can mostly observe the uniform color across cameras, revealing the generalization of our algorithm to different multi-camera imaging structures. Those artifacts slightly around the edges of local images are mainly caused by camera vignetting effects that is not the focus of this work. It will be an interesting problem for future study.

³<https://www.insta360.com/product/insta360-pro/>

⁴www.aqueti.com



Fig. 7. **Impact of ambient light.** (a) native image pair before color correction; (b) color corrected rendering; Sampled snapshots from t_1 to t_5 , ranging from the morning illumination to the dusk. Color mapping function is generated using snapshots captured at t_1 , and applied to all time instants. Red rectangles are extracted for final display.

F. Complexity

We originally implement the HHM algorithm using the Matlab R2018b on a off-the-shelf laptop with Intel Core i7-8700K CPU 3.70 GHz, and 16.0 GB memory. For the 10-camera setup, it takes about 2.474 seconds to produce the averaged color mapping function for each individual camera. As aforementioned, it first derives 10 color mapping functions for each micro-camera when iteratively setting each camera as the source label, and then calculates the averaged map. Another 0.496 second is required when performing the correction for 10 1080p snapshots at any specific time stamp. The total

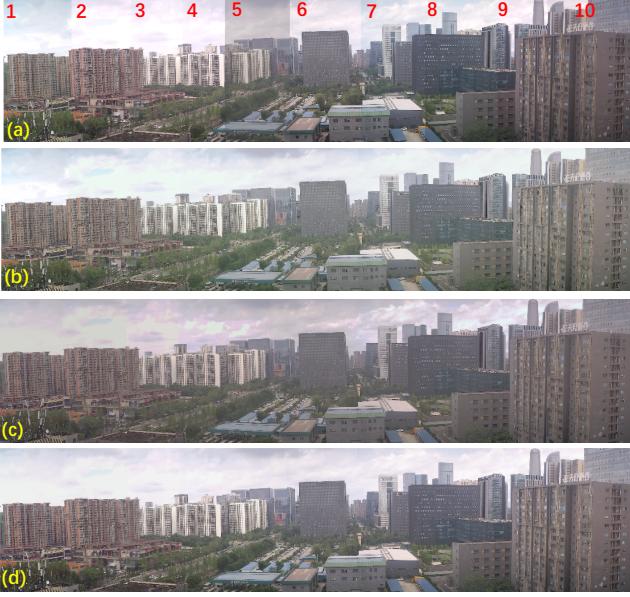


Fig. 8. **Impact of various source labels.** (a) original image snapshots captured by 10 microcameras (from #1 to #10) for panoramic viewing experience; (b) color correction assuming the source label using #1 camera; (c) color correction assuming the source label using #5 camera; (d) color correction using average mapping function.

TABLE IV

OBJECTIVE EVALUATION OF DIFFERENT COLOR CORRECTION METHODS ON DATASET D_1 PROVIDED IN [15] (THE **BEST** PERFORMANCE IS IN **RED**; AND THE **SECOND BEST** IS IN **BLUE**)

Method	PSNR↑	SSIM(%)↑	FSIM(%)↑	iCID(%)↓
R/GL [8]	22.09	83.60	91.90	34.02
AM/GL [12]	21.83	82.21	91.14	34.13
HM/GL [17]	22.46	83.30	91.57	33.78
G/GL [19]	21.89	82.78	90.90	34.09
3MS/GL [20]	21.92	82.32	90.57	34.66
FGPS/GL [15]	23.31	83.37	91.63	34.75
GPS/GL [15]	24.07	83.44	91.78	34.28
Prop. HHM_NS/GL	22.38	83.39	91.61	33.73
Prop. HHM/GL	22.41	83.52	91.74	33.49

running memory is about 140.72 MegaBytes. Note that this running memory includes the additional software overhead, such as the Matlab thread. We need to have $10 \times 10 \times 3 \times 256 = 75$ Kilobytes to cache all intermediate color mapping functions that are implemented using the lookup tables.

This part only gives a qualitative understanding on the time and space complexity of proposed HHM. Ideally, our HHM is a lightweight approach because only histogram matching and table lookup are involved. Our recent proprietary C++ implementation have reported that the HHM can be processed in real-time for multi-camera imaging.

G. Algorithm Limitations

Our HHM heavily relies on the color histogram of both source and target images for the derivation of color mapping function. In principle, it expects the sufficient amount of overlapped pixels to cover the full spectrum of color distribution for consistent color transfer. We have demonstrated

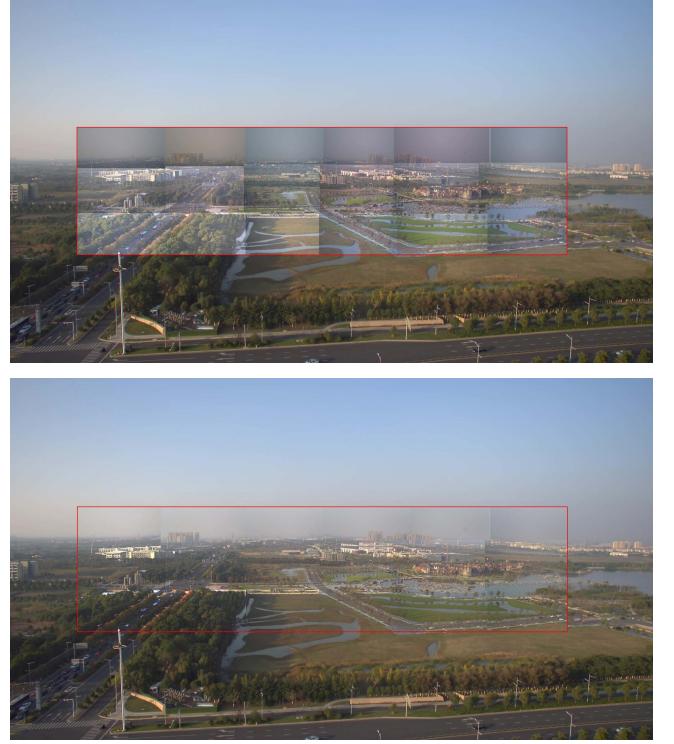


Fig. 9. **Multi-camera with reference.** Illustration of color correction using reference camera for Mantis 70 product. Color differences are effectively alleviated by applying the color correction using the reference camera as the source label (e.g., top versus bottom subplot).

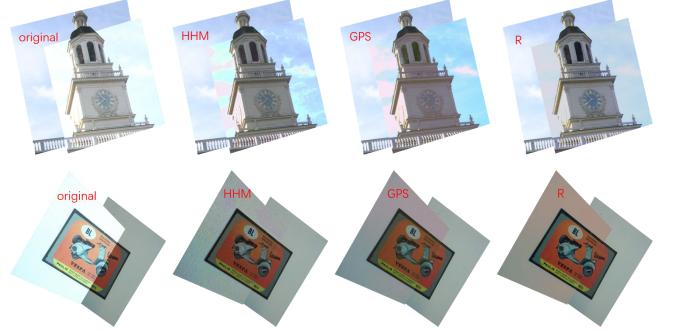


Fig. 10. **Visualized Samples in dataset D_1 .** Illustrative results of color-corrected images using proposed HHM, GPS [15] and R [8]. More saturated pixels are observed in HHM method, but colors are not well corrected for both GPS and R.

the superior efficiency of our algorithm for HD (1080p) and UHD (4K) cameras. It, however, may have performance loss for low-resolution cameras since the same overlapping percentile would not include sufficient pixel intensity variations. Fortunately, recent mainstream cameras are mostly with HD resolution, and the trend of deploying HD or UHD cameras still grows. This promises the potentials of our algorithm in practical system.

On the other hand, we have reused the D_1 dataset provided in [15] to quantitatively analyze the performance of color correction algorithms when having input image pairs at relatively small resolution, with large parallax, and poorly aligned. As shown in Table IV objectively, none of existing algorithms

has shown significant performance lead individually. On the contrary, our HHM offers the best performance for iCID metric, the R leads the performance with SSIM and FSIM measurements, and the GPS tops the PSNR evaluation. For SSIM index, the HHM closely follows the R at the second place. For FSIM measurement, the HHM is ranked at the third place with only 0.04 difference from the second best GPS methods. Both iCID and FSIM are highly recommended in [15] for evaluation because they are much closer to the human judgment. Thus, our HHM still offers relatively convincing performance on this D_1 dataset using the iCID and FSIM measurements. Note that our studies in Table IV do not apply the soft rank used in [15], making the fair comparison baseline for all algorithms.

Subjectively, we further dive into the dataset D_1 by visualizing two typical examples in Fig. 10. It can be seen that the HHM method can still correct the colors, but present saturated pixels in some areas. This is mainly because of the poor alignment of the overlapped area. Thus, it is incapable of producing effective cumulative histogram for accurate mapping function derivation. Though both GPS and R methods generally have less saturated pixels, they cannot adapt the color correctly. This partially explains why HHM does not give the best performance on the PSNR, SSIM, and FSIM metrics that are easily affected by pixel distortions, but it is the optimal approach when using the color sensitive indicator iCID.

Though our algorithm offers robust and reliable color correction, lens vignetting, focal distances, etc, may cause other problems as well (see slight gray edges even after color correction in Fig. 9). Thus, it is worth to further explore the combination of our HHM with other camera settings for more realistic application.

Similar to other algorithms, our method independently performs color correction per each color component. Thus, it could not resolve the problem if we encounter the color space rotation. One possible solution is exploring the cross-component correlation, by which the color channel rotation issue may be resolved to some extent. For example, we can combine the proposed HHM with Pitié's [24] color channel de-correlation approach instead of existing global mapping method as discussed in [15]. We would like to investigate this interesting topic for the next step.

Color consistency can be also partially alleviated by carefully selecting camera sensors, and deliberately applying thorough camera calibrations. In reality, it is difficult to have identical color response for any pair of camera, and on the other hand, sensor response efficiency would vary along with the time, and application conditions. Thus, our HHM still have its positive value as the software refinement, in addition to these hardware calibration.

VI. CONCLUSION

Multi-camera imaging plays a vital role in advanced photography. However, due to the sensor differences, illumination changes, etc, it is difficult to guarantee the uniform color response across various cameras. In this work, we have proposed the hybrid histogram matching-based color correction by integrating the global color mapping and local color straightening.

Experimental studies using real multi-camera system (e.g., self-built 10-camera prototype, and commercialized Mantis 70) have revealed superior efficiency of our algorithm, in comparison to the existing popular approaches, with leading gains both objectively and subjectively. Further ablation studies have examined different aspects of proposed algorithm to understand its capacity in practical applications. All of these have shown that our method can be generalized, promising the encouraging perspective in multi-camera-based imaging systems.

ACKNOWLEDGMENT

Our sincere gratitude is directed to the F. Bellavia and C. Colombo. Their tremendous contributions in [15], e.g., dataset, source codes, extensive simulations, etc, are greatly appreciated in this study.

REFERENCES

- [1] D. J. Brady, M. E. Gehm, R. A. Stack, D. L. Marks, D. S. Kittle, D. R. Golish, E. Vera, and S. D. Feller, "Multiscale gigapixel photography," *Nature*, vol. 486, no. 7403, p. 386, 2012.
- [2] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy, "Burst photography for high dynamic range and low-light imaging on mobile cameras," *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, vol. 35, no. 6, 2016.
- [3] M. Cheng, Z. Ma, M. S. Asif, Y. Xu, H. Liu, W. Bao, and J. Sun, "A dual camera system for high spatiotemporal resolution video acquisition," *IEEE Trans. Pattern Analysis and Machine Intelligence*, 2020.
- [4] H. Zheng, M. Ji, H. Wang, Y. Liu, and L. Fang, "Crossnet: An end-to-end reference-based super resolution network using cross-scale warping," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 88–104.
- [5] V. Boominathan, K. Mitra, and A. Veeraraghavan, "Improving resolution and depth-of-field of light field cameras using a hybrid imaging system," in *2014 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2014, pp. 1–10.
- [6] H. S. Faridul, T. Pouli, C. Chamaret, J. Stauder, E. Reinhard, D. Kuzovkin, and A. Tréneau, "Colour mapping: A review of recent methods, extensions and applications," in *Computer Graphics Forum*, vol. 35, no. 1. Wiley Online Library, 2016, pp. 59–88.
- [7] K.-J. Hsu, Y.-Y. Lin, Y.-Y. Chuang *et al.*, "Robust image alignment with multiple feature descriptors and matching-guided neighborhoods," in *CVPR*, 2015, pp. 1921–1930.
- [8] E. Reinhard, M. Adhikmin, B. Gooch, and P. Shirley, "Color transfer between images," *IEEE Computer graphics and applications*, vol. 21, no. 5, pp. 34–41, 2001.
- [9] X. Xiao and L. Ma, "Color transfer in correlated color space," in *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, 2006, pp. 305–309.
- [10] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International journal of computer vision*, vol. 74, no. 1, pp. 59–73, 2007.
- [11] A. Ilie and G. Welch, "Ensuring color consistency across multiple cameras," in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2. IEEE, 2005, pp. 1268–1275.
- [12] Y. Hwang, J.-Y. Lee, I. So Kweon, and S. Joo Kim, "Color transfer using probabilistic moving least squares," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 3342–3349.
- [13] M. Zhang and N. D. Georganas, "Fast color correction using principal regions mapping in different color spaces," *Real-Time Imaging*, vol. 10, no. 1, pp. 23–30, 2004.
- [14] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, no. 3, pp. 346–359, 2008.
- [15] F. Bellavia and C. Colombo, "Dissecting and reassembling color correction algorithms for image stitching," *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 735–748, 2017.

- [16] W. Xu and J. Mulligan, "Performance evaluation of color correction approaches for automatic multi-view image and video stitching," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 263–270.
- [17] U. Fecker, M. Barkowsky, and A. Kaup, "Histogram-based prefiltering for luminance and chrominance compensation of multiview video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 9, pp. 1258–1267, 2008.
- [18] Q.-C. Tian and L. D. Cohen, "Color correction in image stitching using histogram specification and global mapping," in *2016 Sixth International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2016, pp. 1–6.
- [19] M. Oliveira, A. D. Sappa, and V. Santos, "A probabilistic approach for color correction in image mosaicking applications," *IEEE Transactions on Image Processing*, vol. 24, no. 2, pp. 508–523, 2014.
- [20] H. Faridul, J. Stauder, J. Kervec, and A. Tréneau, "Approximate cross channel color mapping from sparse color correspondences," in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2013, pp. 860–867.
- [21] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [22] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "Fsim: A feature similarity index for image quality assessment," *IEEE transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, 2011.
- [23] J. Preiss, F. Fernandes, and P. Urban, "Color-image quality assessment: From prediction to optimization," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1366–1378, 2014.
- [24] F. Pitié, A. C. Kokaram, and R. Dahyot, "Automated colour grading using colour distribution transfer," *Computer Vision and Image Understanding*, vol. 107, no. 1-2, pp. 123–137, 2007.



Chunqiu Ding is a M.S. student in School of Electronic Science and Engineering, Nanjing University, Jiangsu, 210093, China. She received the B.S degree from Nanjing University in 2018. Her research interests include multi-camera system and image processing.



Zhan Ma (SM'19) is now on the faculty of Electronic Science and Engineering School, Nanjing University, Jiangsu, 210093, China. He received the B.S. and M.S. from Huazhong University of Science and Technology (HUST), Wuhan, China, in 2004 and 2006 respectively, and the Ph.D. degree from the New York University, New York, in 2011. From 2011 to 2014, he has been with Samsung Research America, Dallas TX, and Futurewei Technologies, Inc., Santa Clara, CA, respectively. His current research focuses on the learning-based video coding, and smart cameras. He is a co-recipient of 2018 ACM SIGCOMM Student Research Competition Finalist, 2018 PCM Best Paper Finalist, 2019 IEEE Broadcast Technology Society Best Paper Award, and 2020 IEEE MMSP Image Compression Grand Challenge Best Performing Solution.