

Guided Colorization Using Mono-Color Image Pairs

Zehua Sheng¹, Hui-Liang Shen¹, *Member, IEEE*, Bowen Yao, and Huaqi Zhang

Abstract—Compared to color images captured by conventional RGB cameras, monochrome (mono) images usually have higher signal-to-noise ratios (SNR) and richer textures due to the lack of color filter arrays in mono cameras. Therefore, using a mono-color stereo dual-camera system, we can integrate the lightness information of target monochrome images with the color information of guidance RGB images to accomplish image enhancement in a colorization manner. In this work, based on two assumptions, we introduce a novel probabilistic-concept guided colorization framework. First, adjacent contents with similar luminance are likely to have similar colors. By lightness matching, we can utilize colors of the matched pixels to estimate the target color value. Second, by matching multiple pixels from the guidance image, if more of these matched pixels have similar luminance values to the target one, we can estimate colors with more confidence. Based on the statistical distribution of multiple matching results, we retain the reliable color estimates as initial dense scribbles and then propagate them to the rest of the mono image. However, for a target pixel, the color information provided by its matching results is quite redundant. Hence, we introduce a patch sampling strategy to accelerate the colorization process. Based on the analysis of the posteriori probability distribution of the sampling results, we can use much fewer matches for color estimation and reliability assessment. To alleviate incorrect color propagation in the sparsely scribbled regions, we generate extra color seeds according to the existed scribbles to guide the propagation process. Experimental results show that, our algorithm can efficiently and effectively restore color images with higher SNR and richer details from the mono-color image pairs, and achieves good performance in solving the color bleeding problem.

Index Terms—Colorization, dual-camera system, patch sampling, statistical distribution analysis, image enhancement.

I. INTRODUCTION

MODERN digital cameras use a specially arranged color filter array (CFA) to capture color images on a square grid of sensors. However, due to the filtering structure, each sensor can only use part of the incoming light to record color

information, thus reducing the signal-to-noise ratio (SNR) of the imaging result especially in low-light conditions. Besides, CFA down-samples images to the half size to simultaneously record scene information in different spectral bands (usually including one red channel, one blue channel, and two green channels). As a result, some detailed structures may be lost in the imaging process. From the perspective of single-image denoising, despite the strong noise removal ability of current denoisers, it's still inevitable for them to over-smooth the detailed structures [1], [2], [3], [4], [5], [6], [7].

Recently, extensive studies on joint filtering [8], [9], [10], [11], [12], [13] and image fusion [14], [15], [16] indicate that, a degraded image can be better restored under the guidance of additional images captured in the same scene. Compared with conventional RGB cameras, monochrome cameras do not have CFA structures to record color information. Therefore, mono images basically have higher SNRs and contain richer textures. In this work, we aim to employ a dual-camera system to capture mono-color image pairs and generate enhanced images by integrating the luminance information of monochrome images with the color information of RGB images. In other words, we conduct image enhancement in a guided colorization manner.

Conventional joint filtering [9], [10], [11], [12], [13] and fusion [14], [15], [16] algorithms basically require input images to be well-aligned. In previous studies, several camera systems have been designed to acquire well-aligned images from different source cameras [17], [18]. However, it's not an easy job to directly deploy them into portable devices such as smartphones. In this work, therefore, a more practical way to acquire mono-color image pairs is employing a stereo camera system.

Since two cameras have different viewing angles, the captured mono and color images have disparity. Previous methods [19], [20] basically follow the scheme that they register the image pairs using stereo matching techniques before conducting the joint enhancement process. However, when the contents of the paired images are not consistent in terms of noise levels and details, it's hard to produce pixel-level accurate disparity estimates, leading to several artifacts such as color bleeding. In addition, conventional stereo matching approaches usually require steps such as cost volume aggregation and consistency check to guarantee the matching reliability, which sometimes can be relatively time-consuming.

In this work, without explicit stereo matching process, we propose a novel colorization framework for stereo mono-color image pairs. Overall, our algorithm is constructed based on two assumptions. First, adjacent pixels with similar luminance values are likely to also have similar colors. This

Manuscript received 11 August 2021; revised 23 October 2022 and 28 December 2022; accepted 30 December 2022. Date of publication 13 January 2023; date of current version 20 January 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 61371160 and in part by the ZJU-vivo Information Technology Joint Research Center. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Keigo Hirakawa. (*Corresponding author: Hui-Liang Shen.*)

Zehua Sheng and Bowen Yao are with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China (e-mail: shengzehua@zju.edu.cn; bw Yao36@zju.edu.cn).

Hui-Liang Shen is with the College of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China, and also with the Key Laboratory of Collaborative Sensing and Autonomous Unmanned Systems of Zhejiang Province, Hangzhou 310015, China (e-mail: shenhl@zju.edu.cn).

Huaqi Zhang is with vivo Mobile Communication Company Ltd., Hangzhou 310030, China (e-mail: zhanghuaqi@vivo.com).

Digital Object Identifier 10.1109/TIP.2023.3235536

1941-0042 © 2023 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

assumption is first presented in [21]. Given a few color hints, which are also called color scribbles, the indicated colors can be propagated to the rest of the image based on the luminance similarity among adjacent pixels. Considering the strong correlation between the mono image and the lightness channel of the color image, for a target monochrome pixel, we can search for pixels that have similar luminance values in the guidance color image and use their colors to estimate the target color value. To obtain more robust color estimates, we match multiple similar pixels for each target pixel, which is implemented through block matching using overlapped patches. With patches of size $S \times S$, each pixel can obtain at most S^2 matching results from the guidance image.

However, due to the occlusion issue in stereo images, the matched pixels do not always have similar luminance values to the target one. Therefore, it's unreliable to use all of them for color estimation. In this work, instead of exploring more complex similarity metrics to detect the outliers, we propose to analyze colorization reliability from a statistical perspective.

Our second assumption is that, supposing we match multiple pixels for a target pixel, if more of these matching results have similar luminance values to the target one, we can use them for color estimation with more confidence. In other words, we expect to determine a threshold T . Among the total S^2 matching results, if more than T ones have similar luminance to the target, we consider its color estimate to be reliable and define that it achieves a valid match. We only keep those reliable estimates as initial scribbles and then propagate them to the remaining pixels to generate the fully colorized image. For the sparsely scribbled contents, which are typically the occluded regions, we generate extra color seeds based on their neighboring information, so that the occluded contents can be locally consistent in chrominance after color propagation.

Although we aim to assess the colorization reliability based on multiple matching results, their color information is actually quite redundant. Hence, it's worth considering whether we can match fewer pixels to estimate colors more efficiently and still provide a proper assessment of the colorization reliability. In other words, just by looking at the statistical distribution of the N matching results sampled from the total S^2 ones ($N \ll S^2$), we still expect to achieve sufficiently reliable color estimation. In this work, we address this issue by analyzing the posteriori probability distribution of the sampled matching results. Based on this, we further introduce a sampling strategy to strike a balance between efficiency and reliability of color estimation. During block matching, sampling is conducted at patch level.

Overall, our patch sampling strategy is designed based on three considerations. First, the number of sampled pixels N should be as small as possible to guarantee high efficiency. Second, the initial scribbles should be as reliable as possible to generate accurate color estimates. Third, we expect a high rate of valid match, *i.e.*, dense scribbles. In other words, we hope to colorize the target mono image faithfully according to the guidance image as it records the color information of the scene. We'll give a detailed elaboration in Section III-B.

In summary, we design a novel probabilistic-concept guided colorization algorithm to generate color images of better visual quality in terms of higher SNR and richer textures from paired stereo mono-color images. The main contributions of this work are as follows:

- We present a novel insight to handle guided colorization for stereo mono-color image pairs based on the analysis of the statistical distribution of multiple matching results instead of conducting conventional stereo matching.
- We introduce a patch sampling strategy to accelerate the colorization process with much fewer matches while still producing reliable color estimates faithfully according to the guidance color image.
- Our proposed algorithm outperforms the state-of-the-art guided colorization and guided restoration algorithms on mono-color image pairs from various datasets in terms of both accuracy and visual quality.

II. RELATED WORK

In this section, we briefly review the current research on image colorization, guided image restoration, and single-image denoising.

A. Image Colorization

Generally, image colorization methods can be categorized into scribble-based and example-based ones.

Scribble-based methods [21], [22], [23], [24], [25] require users to manually add the color hints. By assuming that adjacent pixels with similar luminance values tend to have similar colors, the sparse color hints can be propagated to the entire image. This type of method was first popularized by [21], where colorization is modeled as an optimization problem constructed by a system of linear equations. To reduce color bleeding, [22] uses edge detection to prevent color hints from crossing the object boundaries, while [23] takes geodesic distance into consideration during color propagation. Based on the low-rank property of natural images, [24] performs colorization by solving a matrix completion problem. In [25], Zhang et al. fuse color hints with semantic information through a deep neural network, which improves the colorization quality and also enables its real-time use.

Example-based methods [26], [27], [28], [29], [30] use a set of user-provided reference images to colorize the grayscale image automatically. The most common way to make the color transfer is to establish feature correspondence between the reference image and the target grayscale image. The features can be pixels [19], [26], [27], [29], super-pixels [28], or even semantic contents [30]. The only customization required is to choose an appropriate reference image. To further reduce manual efforts, [29] introduces a deep neural network that can select the best reference image from a dataset and accomplish colorizing at the same time. Recently, extensive studies on the deep learning theory inspire [31], [32] that colorization can be fully automated even without reference images based on the various image priors learned by neural networks.

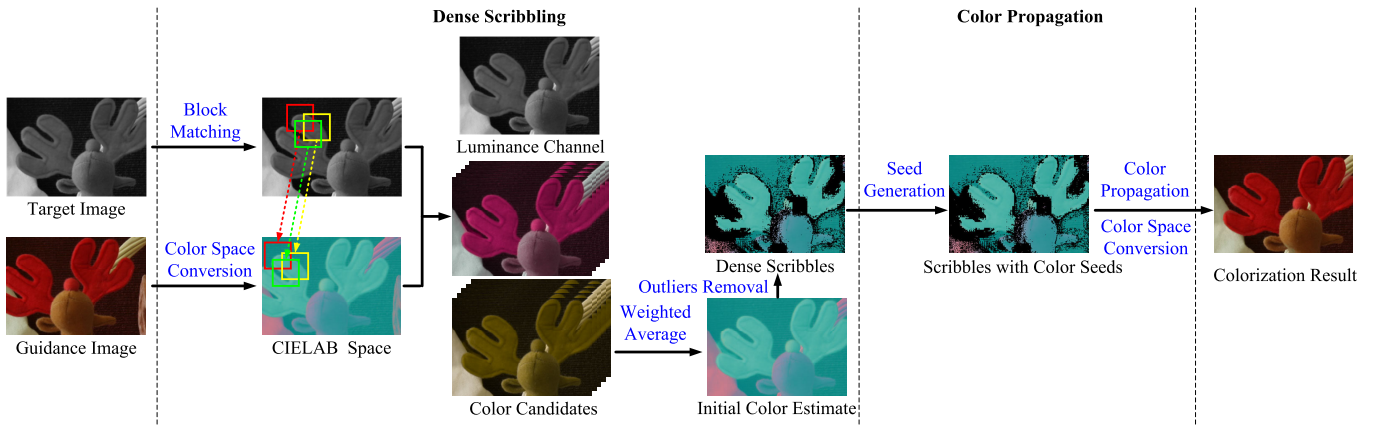


Fig. 1. Framework of our colorization algorithm. First, block matching is performed between the image pair to assign color candidates. Colors are computed by weighted averaging over the candidates. Then, outliers are removed to generate the initial dense scribbles. Finally, sparse color seeds are generated in the regions that have no scribbles at all to help color hints propagate to the entire image.

B. Guided Image Restoration

Guided restoration methods [8], [9], [10], [20], [33], [34], [35], [36], [37], [38], [39] aim to reconstruct images of higher quality from their degraded observations based on an additional guidance image. In [9], He et al. introduce the guided filter, which is the foundation of many current guided restoration studies. It leverages a local linear model to reconstruct the target image from the guidance, and can be treated as either an edge-preserving smoothing operator or a detail transfer operator. To alleviate halo artifacts and produce better edge reconstruction, [33] modifies the original cost function with an edge-aware weighting, while [34] performs the guided filtering in the gradient domain. In some specific application scenarios such as RGB image guided depth image super-resolution, textures are not supposed to be transferred. Hence, a new type of guided filtering, called as dynamic guided filtering, is introduced to solve the problem [35], [36], [37], [38]. As opposed to the aforementioned static filtering process, dynamic one conducts filtering iteratively. In each iteration, the guidance image is updated to meet its structural consistency with the target image. Currently, deep learning is also applied to the guided restoration tasks. Reference [39] trains an end-to-end neural network for fast guided filtering, while [10] constructs a spatially variant linear representation model with learnable coefficients. In [13], the authors propose to complete guided denoising in the frequency domain.

However, most of the guided restoration algorithms require images to be well-aligned. To handle misaligned image pairs, [8] generates a set of translated guidances, and then performs a weighted average of the filtering results using these different guidance images. Similar to our dual-camera system settings, [20] aligns the mono-color pairs before filtering the noisy RGB image. The misaligned problems can be resolved to a certain extent, but there is still a lack of more in-depth analysis.

C. Single-Image Denoising

Single-image denoising is one of the well-explored low-level vision tasks that aims to restore the clean image based

on one single noisy observation. Traditional denoising methods can be categorized into filtering-based and optimization-based ones. Filtering-based approaches such as Gaussian filtering and mean filtering usually exploit low-pass filters to alleviate the unwanted intensity changes caused by random noise. However, they often bring severe edge over-smoothing to the filtering results. For better edge preservation, bilateral filtering [40] combines adjacent pixels based on both their geometric closeness and photometric similarity. Optimization-based methods restore clean images by minimizing an objective function. They are usually constructed according to several image priors such as sparsity [2], [41], [42] and low-rankness [3], [43]. However, the optimization process usually takes quite a long time. It's not practical to deploy them into actual application scenarios. In [44], the authors introduce a non-local strategy, which effectively improves the denoising accuracy and has become one basic technique applied in quite a lot denoising frameworks [1], [2], [3], [41], [42], [43], [45].

Recently, the exploitation of deep learning theory further boosts denoising performance by a large margin. DnCNN [4], one of the earliest learning-based denoising models, demonstrates that a simple network architecture consisting of only convolutional blocks already outperforms traditional denoising algorithms. To deal with heteroscedastic noise, FFDNet [46] uses noise variance as the input parameters along with the image for training, while CBDNet [5] trains an additional sub-network for noise estimation. More recently, MPRNet [7] proposes a multi-stage progressive restoration architecture to generate contextually-enriched and spatially accurate denoising results. HINet [47] improves denoising accuracy with half instance normalization blocks. NAFNet [48] shows that the nonlinear activation functions are not necessary for image restoration and further derives a new baseline which achieves the state-of-the-art denoising accuracy.

Despite the tremendous advances in denoising performance, it's still difficult to preserve salient structures during noise removal, especially at high noise levels. In addition, it's also quite challenging to recover those lost textures caused by CFA down-sampling.

III. PROPOSED ALGORITHM

Overall, our guided colorization framework mainly contains two stages, *i.e.*, dense scribbling and color propagation. During dense scribbling, for each monochrome pixel, we aim to search for multiple pixels with similar luminance from the guidance image to estimate its color value. Based on the analysis of the statistical distribution of the matching results, we evaluate the colorization reliability and keep those reliable color estimates as the initial dense scribbles. Considering the redundant color information provided by multiple matched pixels, we further introduce a sampling strategy to accelerate the dense scribbling process while still giving a proper evaluation of the colorization reliability. Finally, in the color propagation stage, we use the initial scribbles to estimate the color value of the remaining pixels to generate the fully colorized image. The framework of our algorithm is displayed in Fig. 1.

A. Dense Scribbling

In the dense scribbling stage, we aim to estimate the colors of the target monochrome image according to a guidance color image captured in the same scene. Let $\mathbf{M} \in \mathbb{R}^{H \times W}$ and $\mathbf{C} \in \mathbb{R}^{H \times W \times 3}$ denote the paired monochrome and color images, respectively. In this work, $\mathbf{M}$ and $\mathbf{C}$ are acquired using a stereo dual-camera system, with a maximum disparity D .

In [21], the authors present a quite useful assumption that adjacent pixels with similar luminance should also have similar colors, which has become the foundation for several follow-up scribble-based colorization methods. Therefore, considering the strong correlation between the monochrome image and the lightness channel of the color image, we can adopt luminance similarity as a basic metric to estimate colors according to the guidance image.

Specifically, we convert $\mathbf{C}$ into the CIELAB color space to split the luminance components from the color components. The lightness channel is denoted as $\mathbf{C}_l$, describing the luminance information. Two chrominance channels are denoted as $\mathbf{C}_a$ and $\mathbf{C}_b$. As computations in $\mathbf{C}_a$ and $\mathbf{C}_b$ are identical, we refer to $\mathbf{C}_a$ as the chrominance channel below for brevity.

For a target pixel $\mathbf{M}(i, j)$ located at (i, j) , we search for N pixels $\{\mathbf{C}^{ij}(n)\}_{n=1, \dots, N}$ that have the smallest luminance difference within a search range in $\mathbf{C}$. The luminance values of all matching results are stacked into a 3D tensor $\mathbf{L} \in \mathbb{R}^{H \times W \times N}$, where $\mathbf{L}(i, j, n) = \mathbf{C}_l^{ij}(n)$. The corresponding color values are stacked into $\mathbf{U} \in \mathbb{R}^{H \times W \times N}$, where $\mathbf{U}(i, j, n) = \mathbf{C}_a^{ij}(n)$. Here, $\mathbf{C}_l^{ij}(n)$ and $\mathbf{C}_a^{ij}(n)$ are the luminance and the color values of pixel $\mathbf{C}^{ij}(n)$, respectively. Then, a weighted average is performed across these color candidates to estimate the target chrominance channel $\mathbf{M}_a \in \mathbb{R}^{H \times W}$. That is,

$$\mathbf{M}_a(i, j) = \sum_{n=1}^N \mathbf{W}(i, j, n) \cdot \mathbf{U}(i, j, n), \quad (1)$$

where $\mathbf{W} \in \mathbb{R}^{H \times W \times N}$ is the weight tensor.

Based on the aforementioned assumption that adjacent pixels have similar luminance are likely to have similar colors, we determine the value of each element in $\mathbf{W}$ by computing the

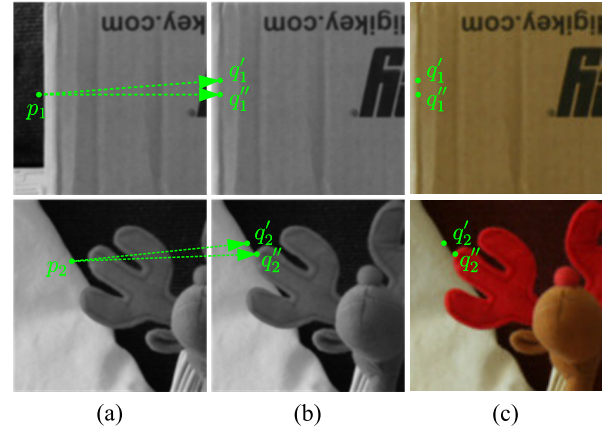


Fig. 2. Examples of two types of outliers. (a) Monochrome target image. (b) Lightness channel of the guidance image. (c) Guidance image. p_1 is the occluded pixel, whose luminance value is 0.11. But the luminance values of its matched pixels q_1' and q_1'' are 0.45 and 0.46, respectively. p_2 is the color-ambiguous pixel. The luminance values of p_2 and its matched pixels q_2' and q_2'' are all 0.44, but q_2' and q_2'' have different colors.

absolute luminance difference between the target monochrome pixel and the matched pixel, computed as

$$\mathbf{W}'(i, j, n) = \frac{1}{|\mathbf{L}(i, j, n) - \mathbf{M}(i, j)| + \varepsilon}, \quad (2)$$

and

$$\mathbf{W}(i, j, n) = \mathbf{W}'(i, j, n) \cdot \frac{1}{\sum_{n=1}^N \mathbf{W}'(i, j, n)}. \quad (3)$$

Here, ε has a small value and is used to prevent the denominator from being zero. In this work, we set $\varepsilon = 2^{-52}$, the default value of *eps* in MATLAB. Equation (3) ensures that the summation of the weight elements for each pixel is 1.

To obtain multiple color candidates for each target pixel, we conduct the matching process at the patch level. The major reason is that, the consistency between luminance and color does not always hold. As shown in Fig. 2, two pixels from different objects can have similar luminance but totally different colors. Therefore, the structural information provided by the patches is beneficial to reduce such questionable matches. In addition, compared with matching at the pixel level, block matching is less sensitive to image distortions and thus can achieve higher robustness. Using overlapped patches, each target pixel can obtain multiple matched pixels from the matching results of different related patches.

We divide $\mathbf{M}$ into overlapped patches of size $S \times S$. For a monochrome patch $\mathbf{p} \in \mathbb{R}^{S \times S}$, we search for its most similar patch $\mathbf{q} \in \mathbb{R}^{S \times S \times 3}$ in $\mathbf{C}$, within a window of size $W_h \times W_v$. In this work, we set $S = 16$, the horizontal search range $W_h = D$, and the vertical search range $W_v = 30$. The difference between $\mathbf{p}$ and $\mathbf{q}$ is measured by their Euclidean distance of luminance, computed as

$$D_l(\mathbf{p}, \mathbf{q}) = \|\mathbf{p} - \mathbf{q}_l\|_F^2, \quad (4)$$

where $\mathbf{q}_l$ is the lightness channel of $\mathbf{q}$. Commonly, the stereo image pairs require to be rectified so that they only have horizontal disparities. Considering the possible rectification errors and the occlusion issue, we also allow a vertical search

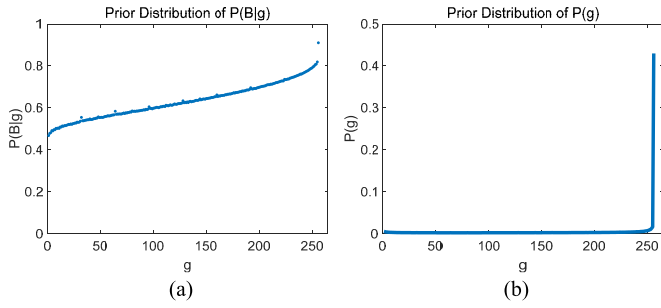


Fig. 3. Visualization of the prior probabilities $P(\mathcal{B}|g)$ and $P(g)$, which are statistically obtained using 800 mono-color image pairs from the Flickr1024 training set [49]. Here, g is the number of matched pixels that have similar luminance to the target, and $\mathcal{B}$ is the event that the color estimation is correct. For a larger number of g , the probability of correct color estimation is higher. These two probabilities are computed when the patch size is 256 (i.e., $S = 16$).

range. If we use all the patches for block matching, each target pixel can obtain $N = S^2$ matching results.

Different from conventional stereo matching methods that use strategies such as cost volume aggregation or consistency check to guarantee the matching confidence, we propose to analyze it from a statistical perspective. Based on the redundant color information provided by these multiple matching results, we expect to give a proper assessment of the colorization reliability. Due to the occlusion issue, as shown in Fig. 2, the matched pixels do not always have similar luminance values to the target one. Our second assumption is that, among all the matching results, if more of them have similar luminance to the target one, we can use their colors for color estimation more confidently.

To verify the above assumption, we obtain the statistical probability distribution of correct colorization using 800 stereo mono-color image pairs from the Flickr1024 training set [49], including about 1.48×10^8 pixels. We take the lightness channel of the left-view image as the monochrome target image and the right-view image as the guidance. To simulate the CFA-sampled situations, we down-sample the original right-view image to its half size and replicate the green channel to form the bayer image. Then, we employ demosaicing interpolation to obtain the full-size guidance image.

In this work, we use the just-noticeable difference (JND) threshold introduced in [50] to judge whether two pixels are similar or not considering that human visual system has different sensitivities in regions with different intensity and contrast. It is the smallest change that human eyes can detect, which is computed as

$$J(i, j) = J_L(i, j) + J_T(i, j) - 0.3 \min(J_L(i, j), J_T(i, j)), \quad (5)$$

where $J_L(i, j)$ is the intensity term and $J_T(i, j)$ is the contrast term. Suppose the range of luminance intensity is $0 \sim 255$. Human eyes can achieve the highest sensitivity at the luminance value of 127, and the lowest at the luminance values of 0 and 255. Therefore, the intensity term J_L has the lowest value at 127, and the highest values at 0 and 255. Human eyes also have different sensitivities to different contrast intensities. The intensity change can be perceived more easily in the flat

areas than in the texture-rich areas. Therefore, the texture term J_T is determined by the gradient values. A larger gradient leads to a higher JND threshold.

Denote g ($0 \leq g \leq S^2$) as the number of matched pixels that have similar luminance to the target one and denote $\mathcal{B}$ as the event of correct color estimation, both of which are judged by the JND threshold. Conducting the aforementioned colorization process, we expect to obtain the prior probability distribution of correct colorization under different values of g , denoted as $P(\mathcal{B}|g)$. A higher value of $P(\mathcal{B}|g)$ indicates higher colorization reliability. As displayed in Fig. 3, the value of $P(\mathcal{B}|g)$ does become larger as g increases. Therefore, we can determine a threshold T . For a target pixel, if more than T of the total matching results have similar luminance, we can assert that the color estimate is reliable and define that it achieves a valid match. Then, we retain these reliable color estimates as the initial dense scribbles.

In fact, random noise in the guidance image can affect the accuracy of dense scribbling, and can also cause noise residue in the colorization results. A simple and fast pre-denoising on the guidance image can help to alleviate the aforementioned problems directly. In this work, we apply the random redundant DCT (RRDCT) denoising algorithm introduced in [51] for noise suppression, which runs almost in real time. Although our algorithm already has some robustness to random noise due to the patch-wise matching scheme, the pre-denoising step can help to further improve the colorization quality, especially when the noise level is relatively high.

B. Patch Sampling

Based on the analysis of the statistical probability distribution of multiple matching results, we can yield reliable dense color estimates. However, the color information provided by these matched pixels is actually quite redundant. Therefore, we introduce a patch sampling strategy to estimate colors more efficiently with fewer matches.

From the perspective of a target pixel, sampling is equivalent to extracting N matching results from its total S^2 ones, under a sampling rate of $\rho = N/S^2$. Among them, r matched pixels have similar luminance to the target. Similarly, we expect to determine a threshold T . If $r \geq T$, we consider that it achieves a valid match.

Overall, our sampling strategy is constructed based on three considerations. First, to achieve high efficiency, we expect to use as few matches as possible for each target pixel to accomplish color estimation. That is, N should be set to a small value. Second, for each target pixel, its colorization confidence $\Phi_{\text{confidence}}(N, T)$ should be as high as possible to ensure the overall accuracy. Third, we expect to obtain a high rate of valid match, which is denoted as $\Phi_{\text{valid_match}}(N, T)$, to colorize as many pixels as possible in the dense scribbling stage. The main reason is that, the guidance color image records the exact color information of the scene. Hence, to ensure that the color of the target image is faithfully estimated according to the scene, it's essential to make full use of the guidance image to colorize

TABLE I
MAIN NOTATIONS USED IN PATCH SAMPLING

Notations	Meanings
g	Number of similar matched pixels without patch sampling
r	Number of similar matched pixels with patch sampling
$\mathcal{A}_r, \mathcal{A}_T$	Event of matching r, T similar pixels with patch sampling
$\mathcal{B}$	Event of obtaining correct color estimate
$P(g)$	Prior probability of matching g similar pixels without patch sampling
$P(\mathcal{A}_r)$	Probability of matching r similar pixels with patch sampling
$P(\mathcal{B} g)$	Prior probability of correct color estimation when matching g similar pixels without patch sampling
$P(\mathcal{B} \mathcal{A}_T)$	Posteriori probability of correct color estimation when matching T similar pixels with patch sampling

as many pixels as possible considering the strong correlation between the mono-color paired images.

For clarity, the main notations used in patch sampling are listed in Table I. Suppose that, among the total S^2 matching results, g of them have similar luminance to the target pixel. In the actual matching process, however, we only sample N of them and observe that r matched pixels have similar luminance values, which we denote as the event $\mathcal{A}_r$. Here, the probability distribution of $P(\mathcal{A}_r)$ can be computed by

$$\begin{aligned}
 P(\mathcal{A}_r) &= \sum_{g=1}^{S^2} P(\mathcal{A}_r, g) = \sum_{g=1}^{S^2} P(g) \cdot P(\mathcal{A}_r|g) \\
 &= \sum_{g=1}^{S^2} P(g) \cdot \frac{C_g^r \cdot C_{S^2-g}^{N-r}}{C_{S^2}^N}. \quad (6)
 \end{aligned}$$

Here, $P(g)$ is the prior distribution of g that is also statistically obtained using 800 stereo mono-color image pairs from the Flickr1024 training set [49]. Its probability distribution is displayed in Fig. 3. C_m^n denotes the combination function. $P(\mathcal{A}_r|g)$ is the probability that we can sample r out of a total of g matching results that have similar luminance to the target. $C_{S^2}^N$ is the number of combinations that we select N matched pixels from the total S^2 ones, and $C_g^r \cdot C_{S^2-g}^{N-r}$ denotes the number of combinations that r of them are sampled from the similar ones and the remaining $N - r$ ones are not.

Therefore, the rate of valid match can be estimated by the probability of obtaining at least T similar matched pixels, computed by

$$\Phi_{\text{valid_match}}(N, T) = \sum_{r=T}^N P(\mathcal{A}_r). \quad (7)$$

In Fig. 4, we visualize the criterion $\Phi_{\text{valid_match}}$ under different values of N and T . Since patch sampling requires a small N for high efficiency, we only display the cases where $N \leq 30$ and $T \leq 30$. It's natural to observe that, for a certain number of matched pixels N , a lower threshold T results in achieving valid match more easily. Similarly, for a fixed

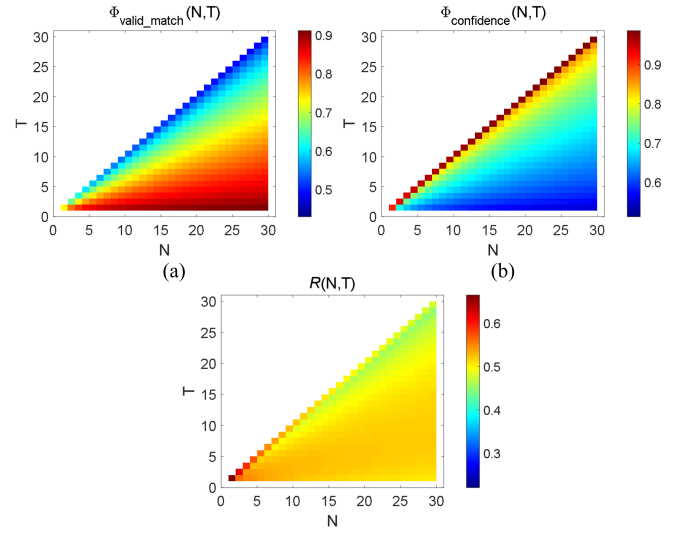


Fig. 4. Visualization of (a) probability $\Phi_{\text{valid_match}}(N, T)$, (b) probability $\Phi_{\text{confidence}}(N, T)$, and (c) criterion $\mathcal{R}(N, T)$ under different settings of N and T .

threshold T , if we conduct more matches, it's also easier to obtain the valid match.

The colorization confidence can be evaluated by the posteriori probability of correct color estimation based on the observation of the sampling result. That is,

$$\begin{aligned}
 \Phi'_{\text{confidence}}(N, T) &= P(\mathcal{B}|\mathcal{A}_T) = \frac{P(\mathcal{A}_T, \mathcal{B})}{P(\mathcal{A}_T)} \\
 &= \frac{\sum_{g=1}^{S^2} P(\mathcal{A}_T, \mathcal{B}, g)}{P(\mathcal{A}_T)} \\
 &= \frac{\sum_{g=1}^{S^2} P(\mathcal{A}_T, \mathcal{B}|g)P(g)}{P(\mathcal{A}_T)} \\
 &= \frac{\sum_{g=1}^{S^2} P(\mathcal{A}_T|g)P(\mathcal{B}|g)P(g)}{P(\mathcal{A}_T)}, \quad (8)
 \end{aligned}$$

where $P(\mathcal{A}_T|g)$ and $P(\mathcal{A}_T)$ can be computed according to (6). For a clearer indication, we normalize the confidence criterion using its maximum value, i.e., $\Phi_{\text{confidence}}(N, T) = \Phi'_{\text{confidence}}(N, T)/\Phi_{\text{max}}$.

As visualized in Fig. 4 (b), if we obtain N matched pixels, setting $T = N$ contributes to much higher confidence than the setting of $T < N$. In addition, if all the matching results have similar luminance to the target pixel, a larger number of matches can also indicate higher colorization reliability.

To strike a balance between these two criteria, we combine them together as a criterion Φ to determine the sampling parameters, formulated as

$$\mathcal{R}(N, T) = \Phi_{\text{valid_match}}(N, T) \cdot \Phi_{\text{confidence}}(N, T). \quad (9)$$

As demonstrated in Fig. 4 (c), by setting a small value of N and $T = N$, we can achieve the best balance between the rate of valid match and colorization confidence. In addition, it also guarantees higher matching efficiency.

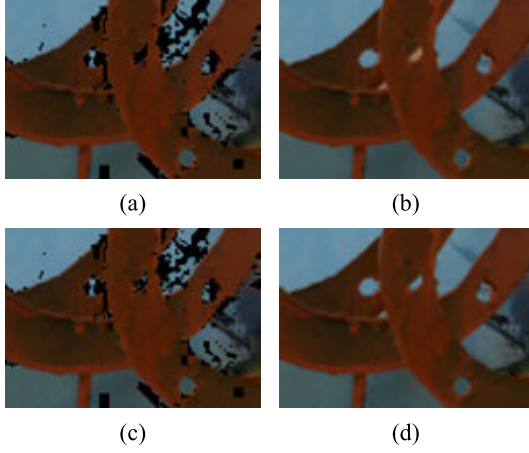


Fig. 5. Color propagation results. (a) Dense scribbles without color seeds. (b) Colorization result without color seeds. (c) Dense scribbles with color seeds. (d) Colorization result with color seeds.

C. Outliers Removal

As shown in Fig. 2, the major reasons that cause wrong color estimation are occlusion and color ambiguity. To yield a colorization result of higher visual quality, it's essential to detect these outliers, remove their initial color estimates, and re-colorize them using their adjacent information.

Occluded pixels can be easily detected in the block matching stage, without conventional left-right consistency check. Using the sampling strategy mentioned above, pixels that fail to achieve valid matches can be recognized as occluded pixels.

To check whether a pixel that completes valid match has ambiguous color candidates, we sort its color values in ascend order. If there exists an absolute difference between two adjacent values that is greater than the threshold τ , we can assert that the chrominance values of its color candidates are not consistent. In our work, leveraging a constant τ is enough to give plausible detection results of ambiguous-color pixels. Here, τ is set to 5/255.

D. Seed Generation and Color Propagation

After obtaining the scribbles, we propagate them to the entire image to generate the complete colorization result. Color propagation is one of the key problems in the study of scribble-based colorization algorithms, which basically follow the assumption that adjacent pixels with similar luminance values are likely to have similar colors. Hence, the existed color hints can gradually spread to their adjacent pixels with similar luminance, until all the pixels are colorized. Unlike the sparse, manually created color hints, our scribbles are dense and are less likely to cause color bleeding during propagation. The pioneering work [21] introduced by Levin et al. is already sufficient to propagate colors properly. The only concern lies in the large occluded regions where there are no color hints at all. To further alleviate color bleeding in these regions, we generate extra color seeds to guide the propagation process.

To begin with, the monochrome image is divided into non-overlapped blocks of size $B \times B$. In each block, we categorize pixels into several luminance levels. Specifically, we sort

Algorithm 1 Guided Image Colorization

Input: Mono image $\mathbf{M} \in \mathbb{R}^{H \times W}$, Color image: $\mathbf{C} \in \mathbb{R}^{H \times W \times 3}$
Input: Parameters: S, W_h, W_v, N, T
Output: Colorization result: $\mathbf{M}_C$
Initialize: $\mathbf{L}, \mathbf{U}_a, \mathbf{U}_b = \mathbf{0} \in \mathbb{R}^{H \times W \times N}$, $\mathbf{Z} = \mathbf{0} \in \mathbb{R}^{H \times W}$, $\varepsilon = 2^{-52}$

- 1: $\{\mathbf{C}_l, \mathbf{C}_a, \mathbf{C}_b\} = \text{RGB2LAB}(\text{PreDenoising}(\mathbf{C}))$
- 2: $\rho = N/S^2$, $\{\mathbf{p}_k, x_k, y_k\}_{k=1}^K = \text{PatchSampling}(\mathbf{M}, S, \rho)$
 $\triangleright \{x_k, y_k\}$ is the coordinate of patch $\mathbf{p}_k$
- 3: $\mathbf{J} = \text{JND}(\mathbf{M})$
- 4: **for all** k in K **do**
- 5: $\{\mathbf{q}_l, \mathbf{q}_a, \mathbf{q}_b\} = \text{Matching}(\mathbf{p}_k, \mathbf{C}_l, \mathbf{C}_a, \mathbf{C}_b, x_k, y_k, W_h, W_v)$
- 6: **for all** s_i from 1 to S **do**
- 7: **for all** s_j from 1 to S **do**
- 8: $i = x_k + s_i - 1, j = y_k + s_j - 1, n = \mathbf{Z}(i, j)$
- 9: **if** $|\mathbf{q}_l(s_i, s_j) - \mathbf{p}(s_i, s_j)| \leq \mathbf{J}(i, j)$ **and** $n < N$ **then**
- 10: $\mathbf{Z}(i, j) = \mathbf{Z}(i, j) + 1, n = \mathbf{Z}(i, j)$
- 11: $\mathbf{L}(i, j, n) = \mathbf{q}_l(s_i, s_j)$
- 12: $\mathbf{U}_a(i, j, n) = \mathbf{q}_a(s_i, s_j)$
- 13: $\mathbf{U}_b(i, j, n) = \mathbf{q}_b(s_i, s_j)$
- 14: $\mathbf{W}' = 1/(\|\mathbf{L} - \mathbf{M}\| + \varepsilon)$, $\mathbf{W} = \text{Normalization}(\mathbf{W}')$
- 15: $\mathbf{M}_a = \text{WeightedAveraging}(\mathbf{U}_a, \mathbf{W})$
- 16: $\mathbf{M}_b = \text{WeightedAveraging}(\mathbf{U}_b, \mathbf{W})$
- 17: $\text{Mask} = \text{IsValidMatch}(\mathbf{M}_a, \mathbf{M}_b, N, T)$
- 18: $\text{Mask} = \text{IfColorAmbiguous}(\mathbf{M}_a, \mathbf{M}_b, \text{Mask})$
- 19: $\{\mathbf{M}_a, \mathbf{M}_b, \text{Mask}\} = \text{SeedGeneration}(\mathbf{M}, \mathbf{M}_a, \mathbf{M}_b, \text{Mask})$
- 20: $\{\mathbf{M}_a, \mathbf{M}_b\} = \text{ColorPropagation}(\mathbf{M}_a, \mathbf{M}_b, \text{Mask})$
- 21: $\mathbf{M}_C = \text{LAB2RGB}(\mathbf{M}, \mathbf{M}_a, \mathbf{M}_b)$

their luminance values in ascend order. A level splitter is set where the absolute luminance difference between two adjacent values is greater than the threshold τ_l . We have to ensure that each luminance level of pixels has at least one color hint. Otherwise, we select one pixel p from that level as the seed pixel, and search for N_p of its most similar pixels in luminance, denoted as $q_n, 1 \leq n \leq N_p$, in the initial scribbles within a window of size $W_N \times W_N$. Here, we set $B = 20$, $\tau_l = 9/255$, $N_p = 3$ and $W_N = 50$. The color of p is computed as

$$p_a = \sum_{n=1}^{N_p} z_p \cdot \frac{1}{|p - q_{l,n}| + \varepsilon} \cdot q_{a,n}, \quad (10)$$

where $q_{l,n}$ and $q_{a,n}$ denote the luminance value and the chrominance value of pixel p_n respectively, and z_p denotes the normalization parameter. With the initial dense scribbles and the extra color seeds, we apply the color propagation algorithm [21] to colorize the remaining pixels. Fig. 5 (b) and (d) compare the colorization results with and without color seeds, showing that color seeds help to reduce incorrect color propagation.

In practice, extra color seeds only account for very small part of the total pixels ($<0.05\%$ in the Middlebury dataset), and have little impact on the overall accuracy. To estimate more reliable colors for them, we use a window centered at each seed pixel rather than using just a single pixel for matching. Sometimes the color seeds may be incorrect in the fully occluded regions, but fetching color from neighbors can make the occluded contents locally consistent in chrominance.

The pseudo codes of our guided colorization framework are presented in Alg. 1 to give a clearer demonstration of the entire pipeline.

TABLE II

THE AVERAGE PSNR (dB), SSIM, DELTA E AND CID VALUES ON THE IMAGE PAIRS FROM THE MIDDLEBURY DATASETS [52], [53] OBTAINED BY OUR COLORIZATION ALGORITHM UNDER DIFFERENT PATCH SAMPLING PARAMETERS

$N = T$	1	2	3	4	5	6
PSNR $\uparrow$	36.10	37.05	37.22	37.27	37.29	37.26
SSIM $\uparrow$	0.9857	0.9886	0.9891	0.9893	0.9893	0.9894
DeltaE $\downarrow$	2.3955	2.1936	2.1588	2.1458	2.1392	2.1396
CID $\downarrow$	0.0349	0.0278	0.0268	0.0265	0.0264	0.0265

IV. EXPERIMENTS

In this section, we first introduce the experimental setup. Then, we validate our algorithm by performing it step-by-step to show the contributions of each stage. Next, comparisons with different methods are conducted on both synthetic and real-world mono-color image pairs. Finally, we have a discussion on several implementation details.

A. Experimental Setup

Our algorithm is implemented in MATLAB R2020b. All of our experiments are conducted on a personal computer with 2.70GHz CPU (Intel Core i5-6400) and 24 GB RAM.

We evaluate our algorithm on the public Middlebury 2005, 2006 datasets [52], [53] and a SceneFlow test set [54]. The Middlebury datasets consist of 30 groups of multi-view color images for stereo matching evaluation. In our experiments, we convert the left view image into the CIELAB color space, and take the lightness channel as the target monochrome image. Its adjacent right view is taken as the guidance image. The original left-view color image is taken as ground truth. Further, to simulate realistic CFA-sampling scenarios, we down-sample the original guidance image into its half size and replicate the green channel to generate the pseudo raw image preserved in the bayer format. Then, we employ demosaicing interpolation to obtain the full-size guidance image. The SceneFlow test set contains 120 stereo images, where the synthetic image pairs are generated in the same way. Four quality metrics, *i.e.*, peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM) [55], DeltaE, and color-image-difference (CID) [56], are employed to assess the colorization performance. Here, DeltaE and CID are two common measures to detect color deviation. Higher PSNR and SSIM values as well as lower DeltaE and CID values indicate better performance.

Based on the criterion $\mathcal{R} = \Phi_{\text{valid_match}} \cdot \Phi_{\text{confidence}}$ presented in Section III-B, we determine the patch sampling parameters N and T . First, to obtain higher colorization reliability, we should set $N = T$ according to $\Phi_{\text{confidence}}$ visualized in Fig. 4 (b). Second, to ensure a high rate of valid match, according to $\Phi_{\text{valid_match}}$ visualized in Fig. 4 (a), T is expect to have a small value. Third, a small value of N also contributes to a high matching efficiency. Therefore, in this work, we set $N = T$ to small values, which can strike a good balance between among the aforementioned three considerations, as shown in Fig. 4 (c). In Table II, we list the quantitative results obtained by our algorithm where N and T

range from 1 to 6 on the clean mono-color image pairs from the Middlebury Dataset. Balancing the above criteria, we set $N = T = 5$ in this work.

We also compare our proposed algorithm to the state-of-the-art guided colorization, guided restoration, and single-image denoising approaches. The colorization algorithms include Jeon16 [19], He18 [29], He19 [30], Jampani17 [57], and Zhang19 [58]. Jeon16 [19] uses stereo matching to register the mono-color image pair before colorization. He18 [29] completes matching and colorization with two separated neural networks. Its following work, Zhang19 [58], is applied to video colorization tasks. He19 [30] is also a deep learning algorithm that transfers color from guidance to target by constructing semantic correspondence. Jampani17 [57] is a video colorization algorithm that aims to propagate colors frame by frame from the reference one. Its application scenario is similar to ours, for the mono-color pairs can be recognized as two adjacent video frames. The guided restoration algorithms include Pan19 [10], Shibata17 [8], and Jung17 [20]. Since their target images are the RGB ones, we only conduct visual comparisons. To show the superiority of multi-source image processing, we also compare our algorithm with three state-of-the-art single-image denoisers including MPRNet [7], HINet [47], and NAFNet [48].

To validate our algorithm in real scenes, we set up a dual-camera system and create our own mono-color image pairs. The monochrome camera captures the left-view image, while the color camera captures the right view. Since the image pairs do not have ground truth, we only conduct the visual assessment as well.

B. Performance Assessment

To validate our colorization algorithm, we perform it step by step and show the contributions of each stage to the colorization results. In the dense scribbling stage, block matching allocates each monochrome pixel an initial color value. As shown in Fig. 6 (c), the initial colors are already plausible in most regions, even in some occluded areas. With outliers detection, the wrong colorization areas are effectively located. We remove these incorrect color assignments and obtain the dense scribbling, displayed in Fig. 6 (d). Fig. 6 (e) shows the final colorization result after color propagation.

In the real application scenarios, the resolutions of the monochrome image and the color image can be different. Hence, we evaluate our algorithm in the situation where the color image is smaller than the monochrome image. Before colorization, we up-sample the guidance image using bicubic interpolation to match the size of the target image for block matching. As shown in Fig. 7 (b), the up-sampled guidance image is over-smooth, but we can still achieve good colorization results thanks to the robustness of block matching. Similarly, over-smoothing caused by image denoising on the guidance image also has little adverse effect on the colorization accuracy.

We also capture mono-color image pairs with a dual-camera system to evaluate our algorithm in real scenarios. Due to the filtering structure of RGB cameras, intensities of the

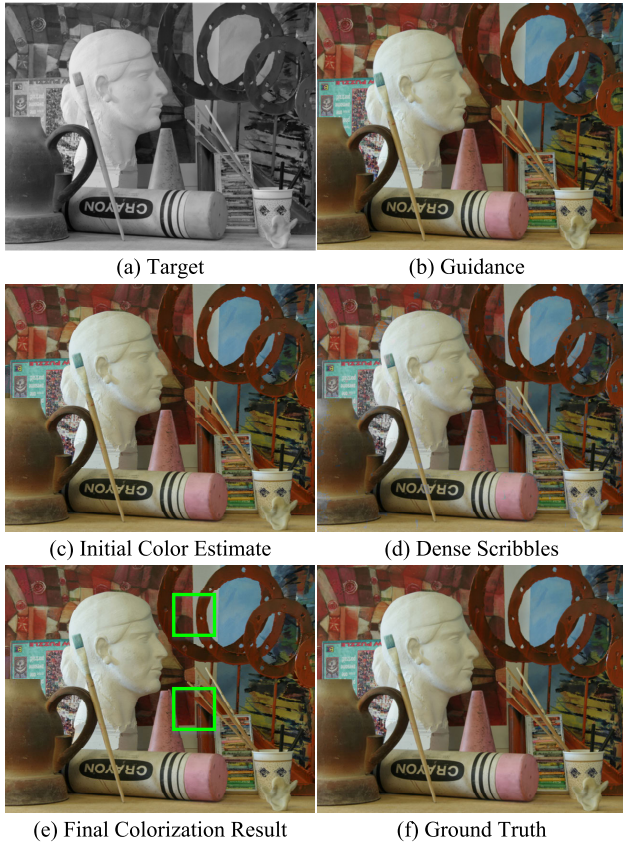


Fig. 6. Colorization results of each step in our algorithm. Compared with the initial estimate, wrong colors are effectively corrected in the final colorization result.

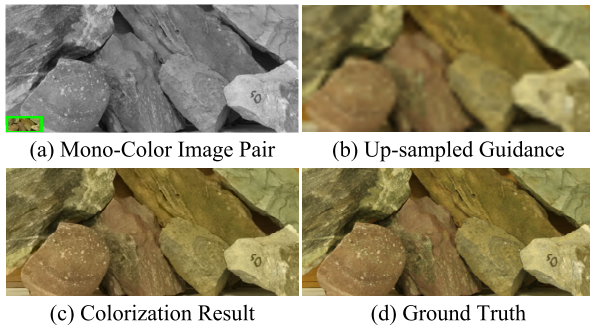


Fig. 7. Colorization result when the size of the target image is $64\times$ the size of the guidance image.

color images are lower than the monochrome images. The color image requires to match the intensity level of the monochrome image for accurate feature matching. Let $\bar{C}$ denote the grayscale version of the color image C obtained by averaging its R, G and B channels. Intensities of $\bar{C}$ and the monochrome image M have a linear relationship, *i.e.*, M and $\lambda\bar{C}$ are of the same intensity level. The linear parameter λ is computed as the average luminance ratio between the image pair. Hence, block matching can be performed between M and $\bar{C}' = \lambda\bar{C}$ for realistic mono-color image pairs. Here, λ can also be computed locally for each search region to achieve higher robustness. It's worth noting that the matching process is conducted on the raw data. To handle non-linear

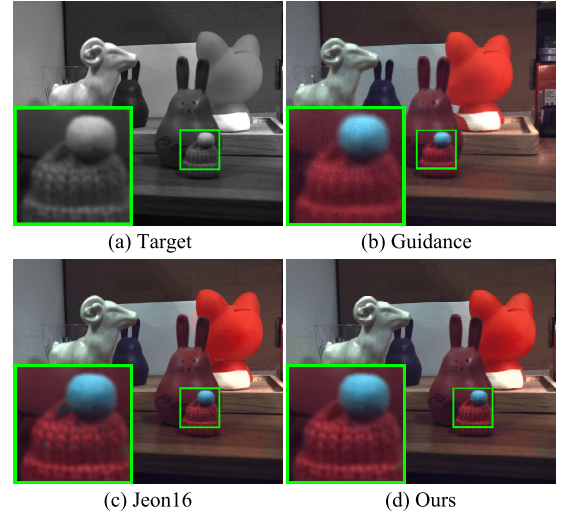


Fig. 8. Colorization result of a realistic mono-color image pair. The colorization results are obtained by Jeon16 [19] and our algorithm, respectively.

situations in the image signal processing pipeline such as Gamma transformation, the linear model still works well.

Fig. 8 (d) shows one of our colorization results. Compared to the guidance image displayed in Fig. 8 (b), whose intensity level has been matched to the target image, our restored color image has lower noise intensities and contains richer textures that conventional denoising algorithms cannot obtain. Fig. 8 (c) is obtained by Jeon16 [19] that registers the image pair via stereo matching and then perform colorization. Its colorization result suffers from the color bleeding problem. One of the reasons is that it's hard to estimate an accurate disparity map when the contents of the image pair are not consistent in terms of noise and textures. In comparison, our algorithm is more robust and can restore color images with better visual quality in terms of less color bleeding, demonstrating that image alignment is not an essential step for accurate colorization using mono-color image pairs.

C. Comparisons to Colorization Algorithms

We quantitatively evaluate our algorithm on synthetic mono-color image pairs generated from the Middlebury 2005, 2006 datasets [52], [53] and the SceneFlow test set [54], and compare it to the state-of-the-art example-based colorization algorithms including Jeon16 [19], He18 [29], He19 [30] and Zhang19 [58], as well as the video color propagation algorithm Jampani17 [57]. Here, we consider both clean and noisy situations. As mentioned in Section III-A, to generate the CFA-sampled guidance image, we down-sample the original right-view color image to its half size and replicate the green channel to mimic the bayer pattern of the raw image. Then, we use demosaicing interpolation to generate the full-size guidance image from the pseudo raw data.

To simulate noise in realistic scenarios, we add synthetic Poisson and Gaussian noise to the pseudo raw version of the input images, which are two main types of noise in modern camera systems [59]. The noise variance maps of the mono image and the color image, denoted as $\Gamma_m \in \mathbb{R}^{H \times W}$ and

TABLE III

THE AVERAGE PSNR (dB), SSIM, DELTA E AND CID VALUES OF DIFFERENT ALGORITHMS ON IMAGE PAIRS FROM THE MIDDLEBURY DATASETS

	$\alpha = 0, \sigma = 0$				$\alpha = 0.01, \sigma = 0.03$				$\alpha = 0.02, \sigma = 0.05$			
	PSNR $\uparrow$	SSIM $\uparrow$	DeltaE $\downarrow$	CID $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	DeltaE $\downarrow$	CID $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	DeltaE $\downarrow$	CID $\downarrow$
Jeon16 [19]	36.79	0.9876	2.2228	0.0308	32.42	0.9644	4.1277	0.0773	31.22	0.9537	4.7963	0.0979
He18 [29]	31.87	0.9656	4.8948	0.0951	30.48	0.9552	5.4398	0.1070	29.70	0.9487	5.8258	0.1210
He19 [30]	29.48	0.9378	5.6728	0.1421	27.82	0.9182	6.8970	0.1778	27.07	0.9083	7.5500	0.1948
Zhang19 [58]	29.45	0.9542	5.7863	0.1277	28.78	0.9459	6.1487	0.1355	28.49	0.9415	6.3193	0.1403
Jampani17 [57]	23.25	0.7273	16.7273	0.4539	22.59	0.7142	17.2086	0.4714	22.56	0.7113	17.2160	0.4695
Ours	37.29	0.9893	2.1392	0.0264	33.14	0.9691	3.8529	0.0672	31.92	0.9595	4.4148	0.0848

TABLE IV

THE AVERAGE PSNR (dB), SSIM, DELTA E AND CID VALUES OF DIFFERENT ALGORITHMS ON IMAGE PAIRS FROM THE SCENEFLOW TEST SET

	$\alpha = 0, \sigma = 0$				$\alpha = 0.01, \sigma = 0.03$				$\alpha = 0.02, \sigma = 0.05$			
	PSNR $\uparrow$	SSIM $\uparrow$	DeltaE $\downarrow$	CID $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	DeltaE $\downarrow$	CID $\downarrow$	PSNR $\uparrow$	SSIM $\uparrow$	DeltaE $\downarrow$	CID $\downarrow$
Jeon16 [19]	31.23	0.9469	3.7355	0.0709	26.85	0.8375	8.1495	0.1954	25.43	0.7759	10.1501	0.2672
He18 [29]	29.11	0.9185	5.8268	0.1256	26.08	0.8749	7.8414	0.1790	25.26	0.8352	9.0835	0.2262
He19 [30]	25.87	0.8337	8.8544	0.2292	24.82	0.7792	10.1562	0.2695	24.36	0.7547	10.9167	0.2909
Zhang19 [58]	27.45	0.9003	7.2393	0.1728	25.23	0.8499	8.6908	0.2119	24.74	0.8220	9.2041	0.2310
Jampani17 [57]	25.05	0.7828	10.6355	0.2709	24.56	0.7579	11.4748	0.2928	24.36	0.7497	11.7318	0.3004
Ours	32.27	0.9561	3.3463	0.0616	27.56	0.8777	6.7656	0.1562	27.06	0.8626	7.3411	0.1741

$\Gamma_c \in \mathbb{R}^{H \times W \times 3}$, are indicated by α_0 and σ_0^2 , respectively. That is,

$$\Gamma_m(i, j) = \alpha_0 \cdot \mathbf{M}(i, j) + \sigma_0^2, \quad (11)$$

and

$$\Gamma_c(i, j, k) = \alpha_0 \cdot \mathbf{C}(i, j, k) + \sigma_0^2. \quad (12)$$

Suppose the intensities of the monochrome image are 3 times those of the color image due to the CFA filtering structure. Therefore, we have to ensure that the mono-color image pairs have the same intensity levels and conduct block matching between $\mathbf{M}$ and the lightness channel of $\mathbf{C}' = 3\mathbf{C}$. In this case, the noise variance map of the color image is

$$\begin{aligned} \Gamma'_c(i, j, k) &= \alpha \cdot \mathbf{C}'(i, j, k) + \sigma^2 \\ &= 9\alpha_0 \cdot \mathbf{C}(i, j, k) + (3\sigma)^2. \end{aligned} \quad (13)$$

In this work, we evaluate our algorithm under two noise settings, *i.e.*, $\alpha = 0.01, \sigma = 0.03$ and $\alpha = 0.02, \sigma = 0.05$. Since guided colorization models usually don't focus on noise removal, for fair comparisons, we exploit one state-of-the-art single-image denoiser, NAFNet [48] to process the noisy monochrome images for both our and competing algorithms to generate the clean lightness channel of the colorization result.

Results of He18 [29], He19 [30], Zhang19 [58] and Jampani17 [57] are obtained by the source codes and network models from the authors' websites, while results of Jeon16 [19] are obtained by our own implementation that follows the exact steps and parameter settings according to the paper. As feature matching can be directly performed between the target image and the lightness channel of the guidance image, decolorization of the guidance image in Jeon16 [19]

are not included in the experiments. For fair comparison, deep learning methods are fine-tuned using 15000 image pairs from the SceneFlow train set. Besides, the lightness channel of each colorization result obtained by the compared algorithms is replaced by the target monochrome image as in our algorithm.

Table III and Table IV lists the average PSNR, SSIM, DeltaE and CID values obtained by different algorithms on the 30 mono-color image pairs from the Middlebury dataset. Visual results under noise parameters $\alpha = 0.01$ and $\sigma = 0.03$ are shown in Fig. 9. Jeon16 [19] achieves good PSNR and SSIM, and also shows plausible colorization results. It enhances the colorization method in [21] by introducing an additional weight term. The entire image is segmented into super-pixels and the weight term is computed by the median chrominance of each super-pixel. However, restricted by the accuracy of disparity estimation and super-pixel segmentation, color bleeding still exists. He18 [29], He19 [30] and Zhang19 [58] obtain results with better visual quality in terms of less color bleeding. However, they do not achieve better quantitative values. The main reason is that, they are designed to transfer colors between two dissimilar images. The use of perceptual loss causes the colorization results to be faithful to the overall style rather than the accurate colors of the guidance images. Jampani17 [57] propagates colors between two adjacent video frames where the motions are relatively small. Hence, it can be less effective when handling large disparity situations, as depicted in Fig. 10. In comparison, our algorithm achieves the highest PSNR and SSIM as well as the lowest DeltaE and CID values, and also the best visual quality. As shown in Fig. 11, our algorithm basically out-performs other state-of-the-art algorithms on every image pair from the Middlebury datasets.

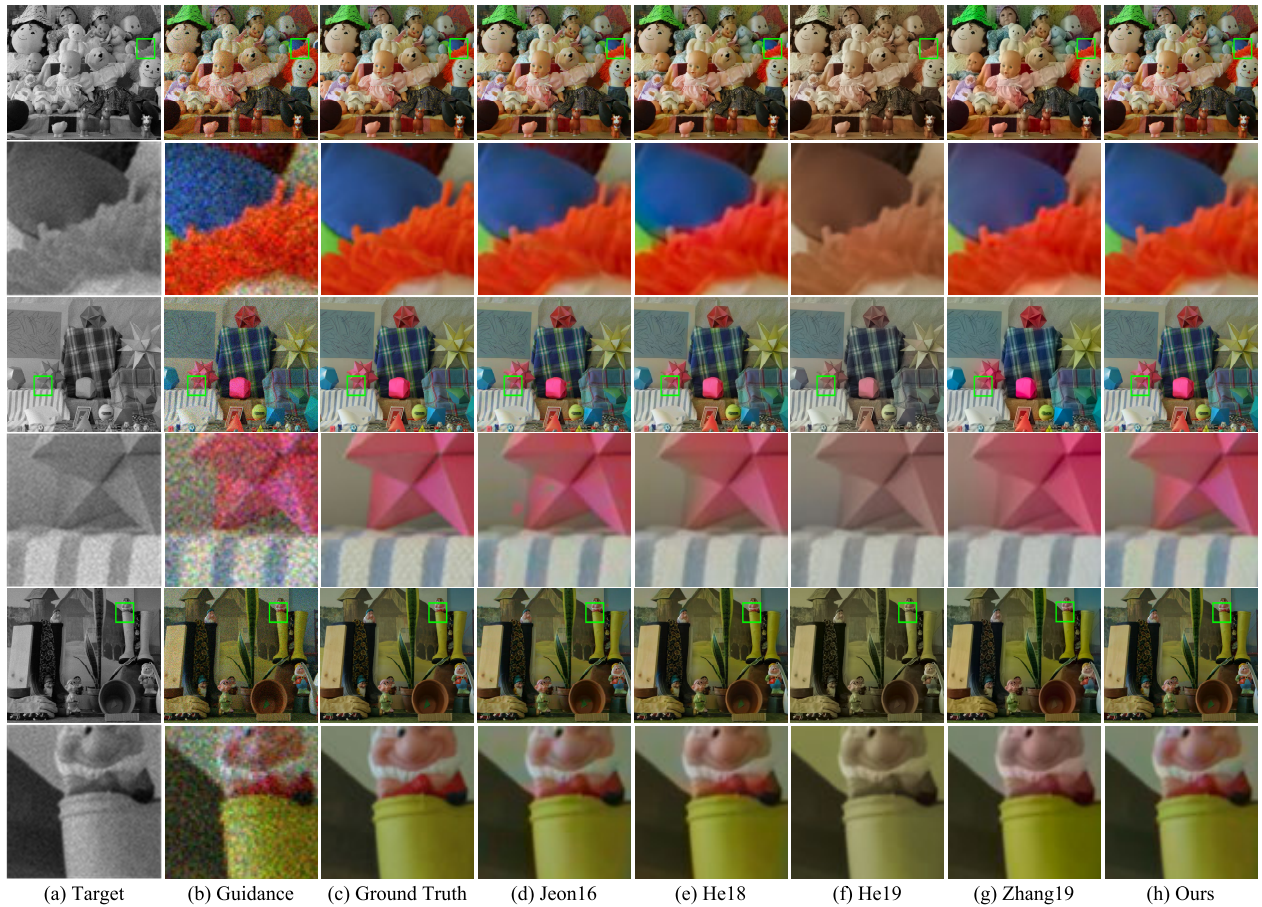


Fig. 9. Visual comparison of colorization results on the image pairs from the Middlebury datasets under Poisson-Gaussian noise ($\alpha = 0.01$, $\sigma = 0.03$), obtained by Jeon16 [19], He18 [29], He19 [30], Zhang19 [58] and our algorithm, respectively.

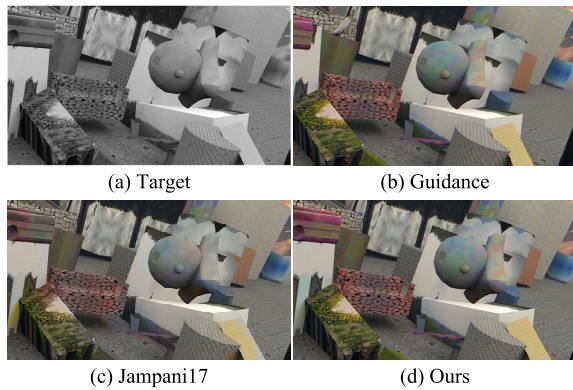


Fig. 10. Visual comparison of colorization results on the image pair from the SceneFlow test set, obtained by Jampani17 [57] and our algorithm, respectively.

The codes of Jeon16 [19] and our algorithm are both implemented in MATLAB R2020b. The average size of the test images from the Middlebury datasets is 555×660 . Jeon16 [19] requires 6.16 minutes to colorize one mono image averagely. In comparison, our algorithm only takes about 11.62 seconds per image (7.40 seconds for dense scribbling and 4.22 seconds for color propagation). Using C++ implementation and parallel computing can further reduce the runtime. If dense scribbling is processed without patch sampling, it will take

about 184.14 seconds per image, which is computationally expensive.

In addition, we set up a dual-camera system to evaluate algorithms on real-world image pairs. Fig. 12 displays the colorization results of the scenes captured using mono and RGB cameras with identical settings. Fig. 13 shows the colorization results, where the source images are captured in the low-light environment using a smartphone equipped with the mono-color cameras. In comparison, colorization performance of each algorithm on the real-world data is similar to that on the synthetic image ones. Our algorithm can achieve colorization results with no color bleeding, even around tiny details, and restore colors faithfully according to the guidance image.

D. Comparisons to Guided Restoration Algorithms

The other option to fuse a mono-color image pair is to denoise the color image under the guidance of the monochrome one. Hence, we also compare our algorithm to the state-of-the-art guided restoration methods. Pan19 [10] proposes a spatially variant linear representation model for guided restoration but requires the input images to be well-aligned. Jung17 [20] and Shibata17 [8] are designed for restoration with unaligned image pairs. The visual comparison is displayed in Fig. 14. All comparative guided restoration methods can achieve plausible results. However, in the case of stereo images, Pan19 [10] can lead to noticeable ghosting

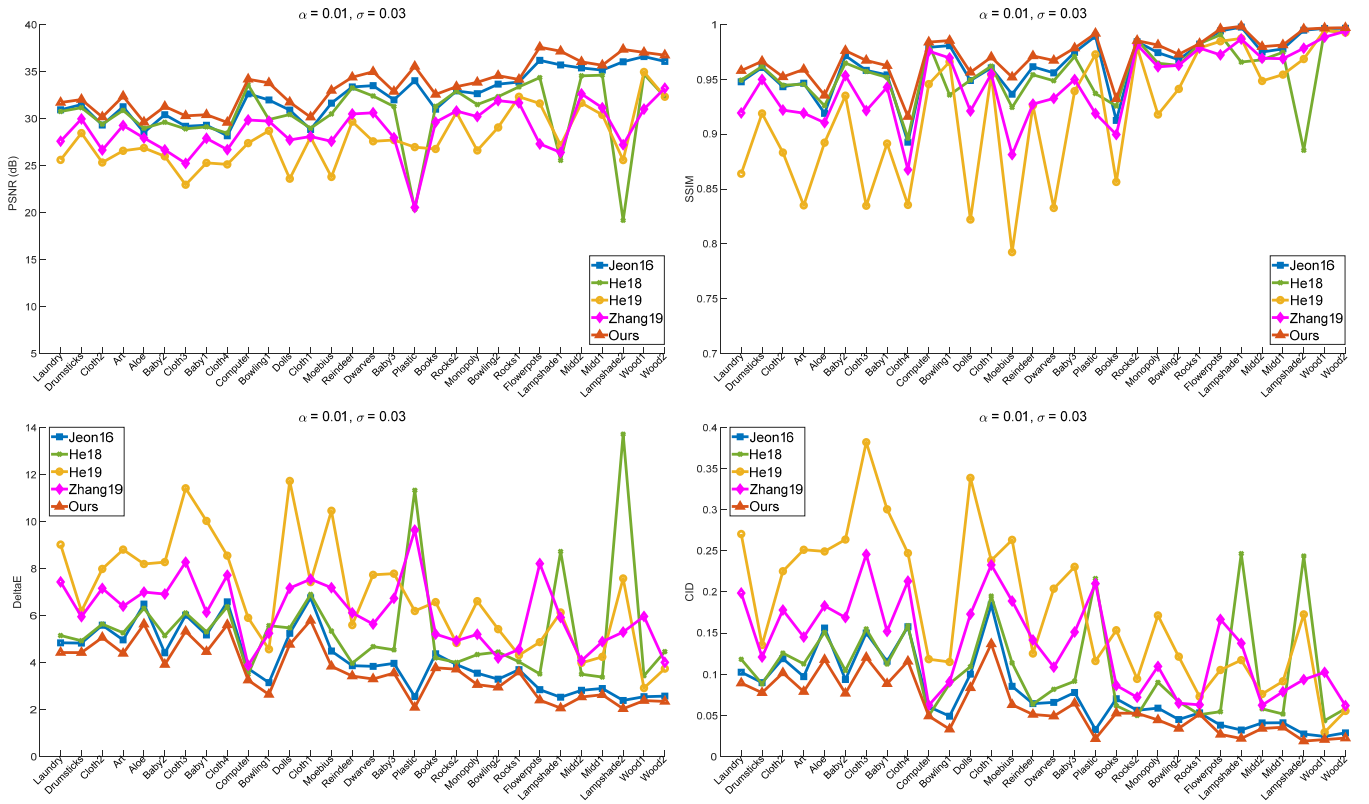


Fig. 11. The PSNR (dB), SSIM, DeltaE and CID values on each single image pair from the Middlebury datasets [52], [53] obtained by Jeon16 [19], He18 [29], He19 [30], Zhang19 [58] and our algorithm, respectively. The x-axis represents the image title, and the y-axis represents the corresponding metric values.



Fig. 12. Visual comparison of colorization results on the realistic image pairs, obtained by Jeon16 [19], He18 [29], He19 [30], and our algorithm, respectively.

artifacts. Jung17 [20] shows its effectiveness in preserving structures, but they are not as salient as in the noisy input. Shibata17 [8] handles misalignment by shifting the guidance image, and is constructed based on the classic guided filtering theory [9]. However, its assumption that the input paired images are linearly correlated in local patches does not always hold, which can result in over-smoothing the details. It also requires high computation if the displacement between two images is large. To process images of size 555×660 from the Middlebury Datasets, Jung17 [20] takes approximately 1.11 minutes per image, while Shibata17 [8] requires about

21.18 minutes. In comparison, ours is much more computationally efficient, which only takes about 11.62 seconds to process one image. Besides, the colorization option can produce images of higher visual quality in terms of better detail preservation and artifact control.

E. Comparisons to Single-Image Denoising Algorithms

To show the superiority of multi-source image processing, we also compare our algorithm with state-of-the-art single-image denoising models including MPRNet [7], HINet [47], and NAFNet [48]. Since the reference images for

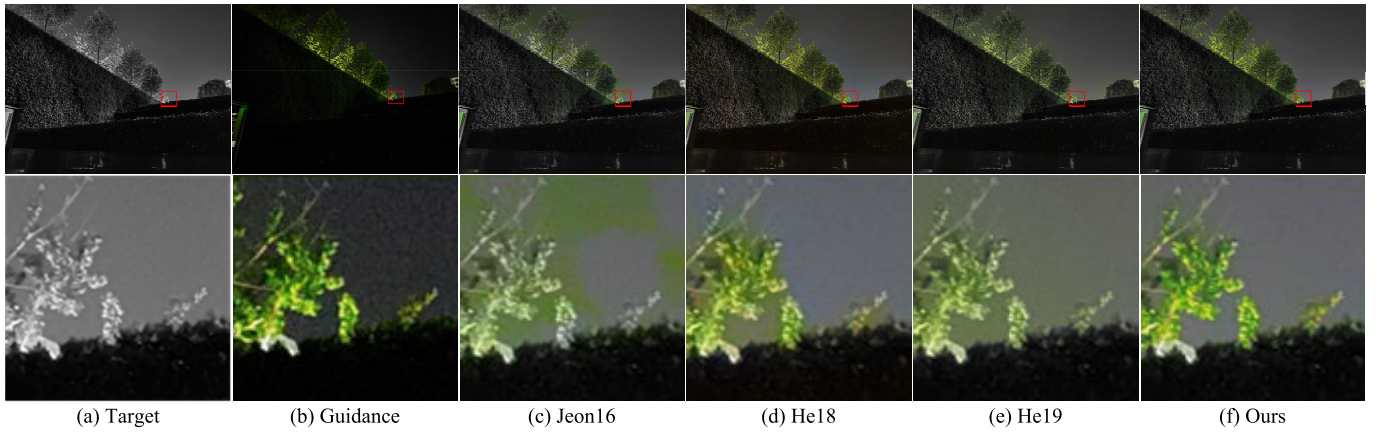


Fig. 13. Visual comparison of colorization results on the realistic image pairs, obtained by Jeon16 [19], He18 [29], He19 [30], and our algorithm, respectively.

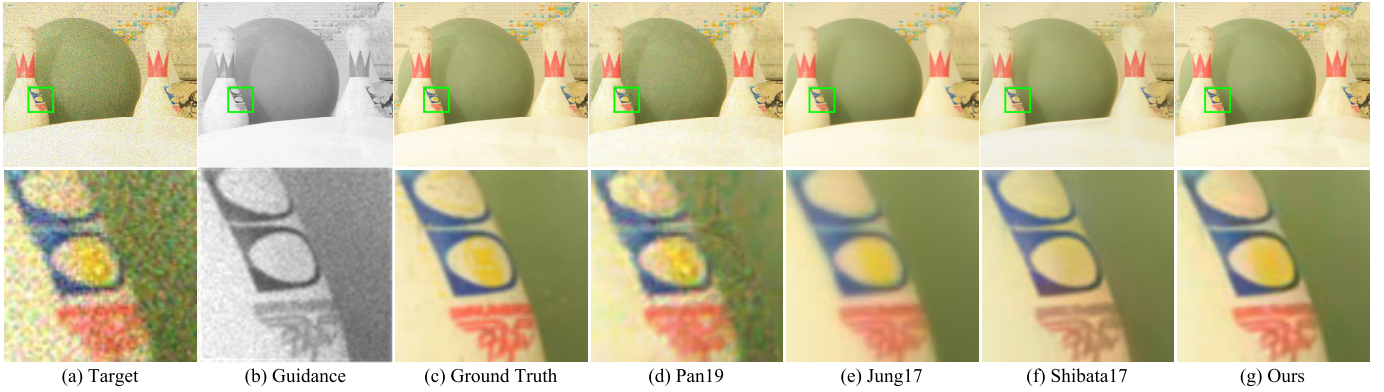


Fig. 14. Visual comparison of results obtained by our colorization algorithm and guided restoration algorithms including Pan19 [10], Jung17 [20], and Shibata17 [8] under mixed Poisson-Gaussian noise ($\alpha = 0.01$, $\sigma = 0.03$).

denoising and our guided colorization are different, we only give qualitative comparisons as well. The visual results are shown in Fig. 16. To generate clean lightness channels of the colorization results, we process the noisy monochrome images with MPRNet, HINet and NAFNet respectively for fair comparisons, displayed in Fig. 16 (f) Ours-M, (g) Ours-H and (h) Ours-N. It's observed that it's difficult for these denoisers to preserve salient structures during noise removal. In comparison, our guided colorization scheme can restore the lost structures caused by CFA-sampling according to the monochrome images. Although the monochrome images are noisy as well, they basically have much lower noise levels than the color images captured in the same scene. Therefore, it can be much easier to denoise these mono images and preserve salient structures.

F. Discussions

1) *How Does Patch Sampling Influence the Accuracy?:* In this work, the size of each patch is set to 16×16 . We also set the sampling parameters $N = T = 5$ so that 2% of the patches are selected for dense scribbling. To show that 2% of the patches are indeed sufficient for reliable color estimation, we perform guided colorization without patch sampling for comparison.

TABLE V
THE AVERAGE PSNR (dB), SSIM, DELTA E, AND CID VALUES ON THE IMAGE PAIRS FROM THE MIDDLEBURY DATASETS [52], [53] OBTAINED BY OUR COLORIZATION ALGORITHM WITH AND WITHOUT PATCH SAMPLING

	$\alpha = 0, \sigma = 0$			
	PSNR $\uparrow$	SSIM $\uparrow$	DeltaE $\downarrow$	CID $\downarrow$
w/o Sampling	37.37	0.9896	2.1400	0.0267
w/ Sampling	37.29	0.9893	2.1392	0.0264
	$\alpha = 0.01, \sigma = 0.03$			
	PSNR $\uparrow$	SSIM $\uparrow$	DeltaE $\downarrow$	CID $\downarrow$
w/o Sampling	33.58	0.9726	3.6143	0.0615
w/ Sampling	33.14	0.9691	3.8529	0.0672
	$\alpha = 0.02, \sigma = 0.05$			
	PSNR $\uparrow$	SSIM $\uparrow$	DeltaE $\downarrow$	CID $\downarrow$
w/o Sampling	32.45	0.9646	4.1031	0.0766
w/ Sampling	31.92	0.9595	4.4148	0.0848

Table V lists the average PSNR, SSIM, DeltaE and CID values on the image pairs from the Middlebury datasets obtained by our colorization algorithm with and without patch sampling respectively. Their results are similar, but patch sampling significantly reduces the amount of computation.

2) *How Does the Pre-Denoising Step Influence the Accuracy?:* We use the random redundant DCT (RRDCT)

TABLE VI

THE AVERAGE PSNR (dB) AND SSIM VALUES ON THE IMAGE PAIRS FROM THE MIDDLEBURY DATASETS [52], [53] OBTAINED BY OUR COLORIZATION ALGORITHM WITH AND WITHOUT DENOISING ON THE GUIDANCE IMAGE

	$\alpha = 0.01, \sigma = 0.03$			
	PSNR $\uparrow$	SSIM $\uparrow$	DeltaE $\downarrow$	CID $\downarrow$
w/o Denoising	30.17	0.9429	5.8371	0.1369
w/ Denoising	33.14	0.9691	3.8529	0.0672

	$\alpha = 0.02, \sigma = 0.05$			
	$\sigma^2 = 0$		$\sigma^2 = 0.03^2$	
	PSNR $\uparrow$	SSIM $\uparrow$	DeltaE $\downarrow$	CID $\downarrow$
w/o Denoising	27.66	0.9095	7.9277	0.2190
w/ Denoising	32.45	0.9595	4.4148	0.0848

TABLE VII

THE AVERAGE PSNR (dB), SSIM, DELTA E AND CID VALUES ON THE IMAGE PAIRS FROM THE MIDDLEBURY DATASETS [52], [53] OBTAINED BY OUR COLORIZATION ALGORITHM USING DIFFERENT WEIGHTS

Weighting Function	$\mathbf{W}'$	$\mathbf{W}'_{sd}$	$\mathbf{W}'_{nc}$
PSNR $\uparrow$	37.29	37.08	37.25
SSIM $\uparrow$	0.9893	0.9891	0.9892
DeltaE $\downarrow$	2.1392	2.1528	2.1470
CID $\downarrow$	0.0264	0.0267	0.0265

denoising algorithm introduced in [51] to remove the mixed Poisson-Gaussian noise in the guidance image. One of the major reasons we choose RRDCT is that it can runs in almost real time. Here, we give a rough introduction of the denoising process. The noisy image is first divided into multiple overlapped patches. For each patch, hard-thresholding is performed on its DCT coefficients for noise removal, with a threshold related to the noise variance. Noise intensity is approximated to be patch-wisely consistent, computed by the average noise variance of each patch. Finally, the denoised patches are aggregated to construct the complete denoised image.

Table VI shows the average PSNR, SSIM, DeltaE and CID values on the 30 image pairs from the Middlebury datasets [52], [53], obtained by our colorization algorithm with and without denoising on the guidance image, respectively. It's clear that the pre-denoising step helps further improve the accuracy, especially when the noise level is relatively high. The main reason is that noise will affect the accuracy of block matching and cannot be completely eliminated by the weighted average of the color candidates. Spatial denoising is one of the most direct and effective manners to remove the noise of the color candidates.

Fig. 15 (c) displays the denoised guidance images under the noise level $\alpha = 0.01$ and $\sigma = 0.03$, while Fig. 15 (d) shows the corresponding colorization results. Visually, the denoised images suffer from artifacts and edge smoothing, but our colorization results are not effected by these problems thanks to the robustness of block matching. Hence, RRDCT is already sufficient for robust colorization, and it's not necessary to apply a more complicated denoising algorithm that increases the amount of calculation.

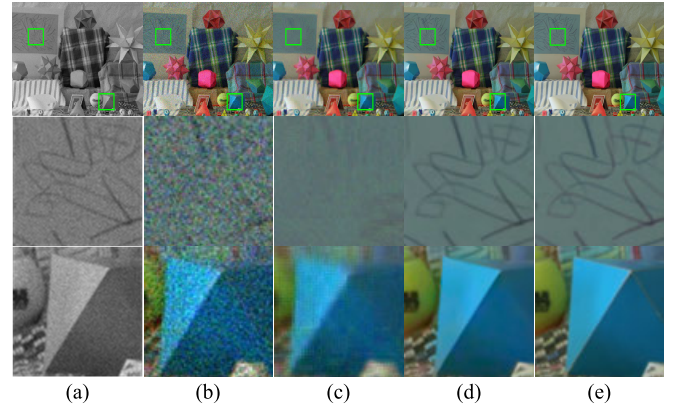


Fig. 15. Visual comparison between the denoised guidance images and the colorization results. (a) Monochrome target image. (b) Guidance image. (c) Denoised guidance image. (d) Colorization result. (e) Ground truth.

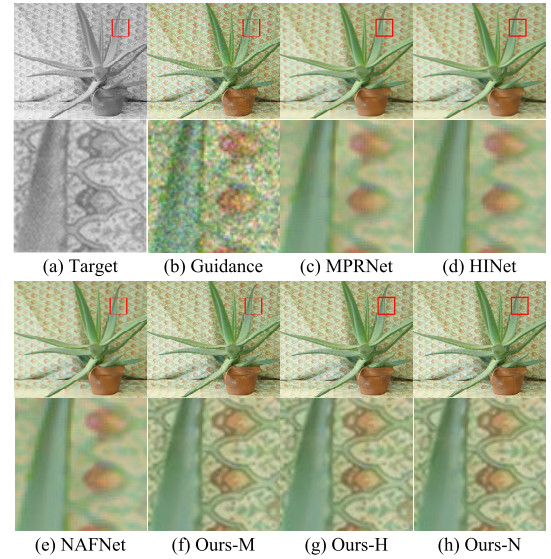


Fig. 16. Visual comparison between the colorization results and the denoising results achieved by three state-of-the-art single-image denoising models including MPRNet [7], HINet [47], and NAFNet [48] under mixed Poisson-Gaussian noise ($\alpha = 0.01, \sigma = 0.03$). For fair comparisons while also obtaining clean lightness channels of the denoising results, we process the noisy mono images using these denoisers respectively, displayed in (f) Ours-M, (g) Ours-H, and (h) Ours-N.

3) *How Does the Weighting Function Influence the Accuracy?*: As shown in Equation (2), our weight tensor is computed by the luminance difference between the target and the matched pixels. To demonstrate that our algorithm is not sensitive to the design of the weight tensor, we adopt the other two weighting functions presented in [21] for comparisons. The first one is based on the squared difference between the two intensities:

$$\mathbf{W}'_{sd}(i, j, n) = e^{-(\mathbf{L}(i, j, n) - \mathbf{M}(i, j))^2 / 2\sigma_l^2}, \quad (14)$$

while the second one is computed based on the normalized correlation:

$$\mathbf{W}'_{nc}(i, j, n) = 1 + \frac{1}{\sigma_l^2} (\mathbf{L}(i, j, n) - \mu_l) (\mathbf{M}(i, j) - \mu_l). \quad (15)$$

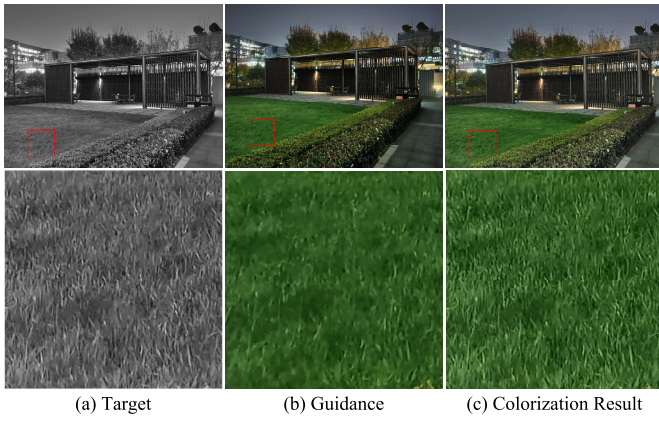


Fig. 17. Colorization result where the input mono-color paired images are not rectified.

Here, μ_l and σ_l^2 are the mean and variance of the luminance values in a window of size 16×16 around the target pixel. Quantitative results obtained by our algorithm on the Middlebury Dataset using different weighting functions are listed in Table VII, demonstrating that it achieves similar colorization accuracy in different cases.

4) *Is Image Rectification an Essential Step?*: In common settings of stereo image processing, the input images are usually rectified to only contain horizontal disparities, which can significantly reduce the computational burden for tasks such as stereo matching. In this work, we don't rely on conventional stereo matching techniques to construct the one-to-one pixel correspondences. During the matching process, we don't require complicated cost aggregation and consistency check. Therefore, allowing matching in a reasonable vertical range will not impose a heavy computational burden, especially when using patch sampling. To verify this, we capture a pair of realistic mono-color images using our dual-camera system, where the color image has a larger field of view than the mono one and the paired images are not rectified. Our result is shown in Fig. 17, demonstrating the proposed colorization framework can also handle this situation. Compared to the input guidance image, the colorization output contains richer details.

V. CONCLUSION

In this work, we propose a guided colorization to restore color images with higher SNR and richer details from the mono-color image pairs. Instead of conventional stereo matching process, we analyze guided colorization from a statistical perspective. For each target pixel, we obtain multiple matched pixels from the guidance image for color estimation. Based on the statistical distribution of these matching results, we can directly assess the colorization reliability. Further, considering that the color information provided by these matched pixels are quite redundant, we introduce a patch sampling strategy to significantly accelerate the matching process while still obtaining reliable color estimates. Experimental results show that, our algorithm is computationally efficient, and achieves good performance in solving the color bleeding problem.

REFERENCES

- [1] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.
- [2] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, Apr. 2013.
- [3] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2862–2869.
- [4] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
- [5] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1712–1722.
- [6] S. W. Zamir et al., "Learning enriched features for real image restoration and enhancement," in *Proc. Eur. Conf. Comput. Vis. Heidelberg, Germany: Springer*, 2020, pp. 492–511.
- [7] S. W. Zamir et al., "Multi-stage progressive image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14821–14831.
- [8] T. Shibata, M. Tanaka, and M. Okutomi, "Misalignment-robust joint filter for cross-modal image pairs," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3295–3304.
- [9] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.
- [10] J. Pan, J. Dong, J. S. Ren, L. Lin, J. Tang, and M.-H. Yang, "Spatially variant linear representation models for joint filtering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1702–1711.
- [11] X. Deng and P. L. Dragotti, "Deep convolutional neural network for multi-modal image restoration and fusion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3333–3348, Oct. 2021.
- [12] Z. Shi, Y. Chen, E. Gavves, P. Mettes, and C. G. M. Snoek, "Unsharp mask guided filtering," *IEEE Trans. Image Process.*, vol. 30, pp. 7472–7485, 2021.
- [13] Z. Sheng, X. Liu, S.-Y. Cao, H.-L. Shen, and H. Zhang, "Frequency-domain deep guided image denoising," *IEEE Trans. Multimedia*, early access, Oct. 13, 2022, doi: [10.1109/TMM.2022.3214375](https://doi.org/10.1109/TMM.2022.3214375).
- [14] S. K. Nayar and T. Mitsunaga, "High dynamic range imaging: Spatially varying pixel exposures," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2000, pp. 472–479.
- [15] S. Li, X. Kang, and J. Hu, "Image fusion with guided filtering," *IEEE Trans. Image Process.*, vol. 22, no. 7, pp. 2864–2875, Jul. 2013.
- [16] D. Connah, M. S. Drew, and G. D. Finlayson, "Spectral edge image fusion: Theory and applications," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 65–80.
- [17] X. Zhang, T. Sim, and X. Miao, "Enhancing photographs with near infra-red images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
- [18] M. D. Tocci, C. Kiser, N. Tocci, and P. Sen, "A versatile HDR video production system," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 1–10, Jul. 2011.
- [19] H.-G. Jeon, J.-Y. Lee, S. Im, H. Ha, and I. S. Kweon, "Stereo matching with color and monochrome cameras in low-light conditions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4086–4094.
- [20] Y. J. Jung, "Enhancement of low light level images using color-plus-mono dual camera," *Opt. Exp.*, vol. 25, no. 10, pp. 12029–12051, 2017.
- [21] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," in *Proc. ACM SIGGRAPH Papers*, Aug. 2004, pp. 689–694.
- [22] Y.-C. Huang, Y.-S. Tung, J.-C. Chen, S.-W. Wang, and J.-L. Wu, "An adaptive edge detection based colorization algorithm and its applications," in *Proc. 13th Annu. ACM Int. Conf. Multimedia (MULTIMEDIA)*, 2005, pp. 351–354.
- [23] L. Yatziv and G. Sapiro, "Fast image and video colorization using chrominance blending," *IEEE Trans. Image Process.*, vol. 15, no. 5, pp. 1120–1129, May 2006.
- [24] S. Wang and Z. Zhang, "Colorization by matrix completion," in *Proc. AAAI Conf. Artif. Intell.*, vol. 26, no. 1, 2012, pp. 689–694.
- [25] R. Zhang et al., "Real-time user-guided image colorization with learned deep priors," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–11, 2017.

- [26] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," in *Proc. 29th Annu. Conf. Comput. Graph. Interact. Techn. (SIGGRAPH)*, 2002, pp. 277–280.
- [27] R. Ironi, D. Cohen-Or, and D. Lischinski, "Colorization by example," in *Proc. Eurograph. Symp. Rendering*. Goslar, Germany: The Eurographics Association, 2005, pp. 201–210.
- [28] R. K. Gupta, A. Y.-S. Chia, D. Rajan, E. S. Ng, and H. Zhiyong, "Image colorization using similar images," in *Proc. 20th ACM Int. Conf. Multimedia (MM)*, 2012, pp. 369–378.
- [29] M. He, D. Chen, J. Liao, P. V. Sander, and L. Yuan, "Deep exemplar-based colorization," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–16, 2018.
- [30] M. He, J. Liao, D. Chen, L. Yuan, and P. V. Sander, "Progressive color transfer with dense semantic correspondences," *ACM Trans. Graph.*, vol. 38, no. 2, pp. 1–18, Apr. 2019.
- [31] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 415–423.
- [32] J.-W. Su, H.-K. Chu, and J.-B. Huang, "Instance-aware image colorization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7968–7977.
- [33] Z. Li, J. Zheng, Z. Zhu, W. Yao, and S. Wu, "Weighted guided image filtering," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 120–129, Jan. 2015.
- [34] F. Kou, W. Chen, C. Wen, and Z. Li, "Gradient domain guided image filtering," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4528–4539, Nov. 2015.
- [35] B. Ham, M. Cho, and J. Ponce, "Robust image filtering using joint static and dynamic guidance," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4823–4831.
- [36] B. Ham, M. Cho, and J. Ponce, "Robust guided image filtering using nonconvex potentials," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 192–207, Jan. 2018.
- [37] X. Shen, C. Zhou, L. Xu, and J. Jia, "Mutual-structure for joint filtering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 3406–3414.
- [38] X. Guo, Y. Li, J. Ma, and H. Ling, "Mutually guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, pp. 694–707, 2020.
- [39] H. Wu, S. Zheng, J. Zhang, and K. Huang, "Fast end-to-end trainable guided filter," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1838–1847.
- [40] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. 6th Int. Conf. Comput. Vis.*, 1998, pp. 839–846.
- [41] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2272–2279.
- [42] J. Xu, L. Zhang, and D. Zhang, "A trilateral weighted sparse coding scheme for real-world image denoising," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 20–36.
- [43] W. Dong, G. Shi, and X. Li, "Nonlocal image restoration with bilateral variance estimation: A low-rank approach," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 700–711, Feb. 2013.
- [44] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2, Jun. 2005, pp. 60–65.
- [45] Y. Hou et al., "NLH: A blind pixel-level non-local method for real-world image denoising," *IEEE Trans. Image Process.*, vol. 29, pp. 5121–5135, 2020.
- [46] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.
- [47] L. Chen, X. Lu, J. Zhang, X. Chu, and C. Chen, "HINet: Half instance normalization network for image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 182–192.
- [48] L. Chen, X. Chu, X. Zhang, and J. Sun, "Simple baselines for image restoration," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 1–21.
- [49] Y. Wang, L. Wang, J. Yang, W. An, and Y. Guo, "Flickr1024: A large-scale dataset for stereo image super-resolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1–6.
- [50] J. Wu, G. Shi, W. Lin, and C. C. J. Kuo, "Enhanced just noticeable difference model with visual regularity consideration," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 1581–1585.
- [51] S. Fujita, N. Fukushima, M. Kimura, and Y. Ishibashi, "Randomized redundant DCT: Efficient denoising by using random subsampling of DCT patches," in *Proc. SIGGRAPH Asia Tech. Briefs*, 2015, pp. 1–4.
- [52] D. Scharstein and C. Pal, "Learning conditional random fields for stereo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
- [53] H. Hirschmuller and D. Scharstein, "Evaluation of cost functions for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
- [54] N. Mayer et al., "A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4040–4048.
- [55] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [56] I. Lissner, J. Preiss, P. Urban, M. S. Lichtenauer, and P. Zolliker, "Image-difference prediction: From grayscale to color," *IEEE Trans. Image Process.*, vol. 22, no. 2, pp. 435–446, Feb. 2013.
- [57] V. Jampani, R. Gadde, and P. V. Gehler, "Video propagation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 451–461.
- [58] B. Zhang et al., "Deep exemplar-based video colorization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 8052–8061.
- [59] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1737–1754, Oct. 2008.



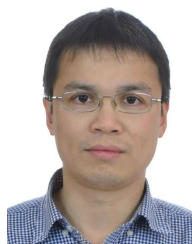
Zehua Sheng received the B.E. degree from Zhejiang University, China, in 2017, where he is currently pursuing the Ph.D. degree with the College of Information Science and Electronic Engineering. His research interests include image denoising and multimodal image processing.



Hui-Liang Shen (Member, IEEE) received the B.E. and Ph.D. degrees in electronic engineering from Zhejiang University, China, in 1996 and 2002, respectively. He was a Research Associate and a Research Fellow with The Hong Kong Polytechnic University, Hong Kong, from 2001 to 2005. He is currently a Full Professor with the College of Information Science and Electronic Engineering, Zhejiang University. His research interests include multispectral imaging, image processing, computer vision, and machine learning.



Bowen Yao received the B.E. degree in information engineering from Zhejiang University, Hangzhou, China, in 2020, where he is currently pursuing the master's degree with the College of Information Science and Electronic Engineering. His research interests include color constancy and image alignment.



Huaqi Zhang received the M.D. degree from Hangzhou Dianzi University in 2004. He is currently working at vivo Mobile Communication Company Ltd. His main research interests include computer vision and machine learning.