Unsupervised Color Retention Network and New Quantization Metric for Blind Motion Deblurring

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

20-10-2021 / 21-10-2021

CITATION

Zhao, Suiyi; Zhang, Zhao; Hong, Richang; Xu, Mingliang; Zhang, Haijun; Wang, Meng; et al. (2021):
Unsupervised Color Retention Network and New Quantization Metric for Blind Motion Deblurring. TechRxiv.
Preprint. <https://doi.org/10.36227/techrxiv.16835314.v1>

DOI

[10.36227/techrxiv.16835314.v1](https://doi.org/10.36227/techrxiv.16835314.v1)

Unsupervised Color Retention Network and New Quantization Metric for Blind Motion Deblurring

Suiyi Zhao, Zhao Zhang, *Senior Member, IEEE*, Richang Hong, Mingliang Xu, Haijun Zhang, *Senior Member, IEEE*, Meng Wang, *Fellow, IEEE*, and Shuicheng Yan, *Fellow, IEEE*

Abstract—Unsupervised blind motion deblurring is still a challenging topic due to the inherent ill-posed properties, and lacking of paired data and accurate quality assessment method. Besides, virtually all the current studies suffer from large chromatic aberration between the latent and original images, which will directly cause the loss of image details. However, how to model and quantify the chromatic aberration appropriately are difficult issues urgent to be solved. In this paper, we propose a general unsupervised color retention network termed CRNet for blind motion deblurring, which can be easily extended to other tasks suffering from chromatic aberration. New concepts of blur offset estimation and adaptive blur correction are introduced, so that more detailed information can be retained to improve the deblurring task. Specifically, CRNet firstly learns a mapping from the blurry image to motion offset, rather than directly from the blurry image to latent image as previous work. With obtained motion offset, an adaptive blur correction operation is then performed on the original blurry image to obtain the latent image, thereby retaining the color information of image to the greatest extent. A new pyramid global blur feature perception module is also designed to further retain the color information and extract more blur information. To assess the color retention ability for image deblurring, we present a new chromatic aberration quantization metric termed Color-Sensitive Error (CSE) in line with human perception, which can be applied to both the cases with/without paired data. Extensive experiments demonstrated the effectiveness of our CRNet for the color retention in unsupervised deblurring.

Index Terms—Unsupervised blind motion deblurring, image restoration, color retention, new chromatic aberration quantization metric.

1 INTRODUCTION

IMAGE deblurring is a process of removing the blurring artifacts and recovering the latent image from the blurred ones, which has been attracting increasing attention in emerging applications. Motion blur is the most common kind of blur in reality. For example, when taking pictures using handheld devices, it is common to obtain motion-blur pictures due to the hand's shaking or rapid movement of the targets in lens. For the motion blur, the relationship between the blurred and sharp images can be formulated as follows:

$$B = S \bullet K + N \quad (1)$$

where B , S and N denote the blurred image, sharp image, and noise, respectively. K is a blur kernel. $\bullet$ and $+$ denote the convolution and element-wise addition operations, respectively. When K is known, the related task is called non-blind deblurring that is also a classical deconvolution problem [1], [2], [3], [4], [5], [6], [7], [8]. However, K is often unknown in reality, and the related task is called blind deblurring. Traditional blind deblurring methods usually require a time-consuming iterative process [9], [10], [11]. With the fast development of deep neural networks (DNN), some

Fig. 1. Comparison of the chromatic aberration suffered in current unsupervised deblurring model (left), and the color retention ability of our CRNet (right). The chromatic aberration in top example is obvious, while the chromatic aberration in bottom example is not obvious. The values of PSNR, SSIM and our CSE metrics are also described.

end-to-end blur-kernel-free DNN-based deep deblurring models have been proposed for blind deblurring. Besides, deep deblur models can avoid the complicated and time-consuming learning process, and have better generalization ability to other images.

Because of using massive labeled data for training and strong fitting ability of DNN, supervised deep deblur models have obtained better quantitative and visual performance [12], [13], [14], [15], [16], [17]. However, obtaining paired data (i.e., blurry image and its sharp image) from specific domains is usually expensive, and the data-driven

S. Zhao, Z. Zhang, R. Hong, and M. Wang are with the School of Computer Science and Information Engineering; also with the Key Laboratory of Knowledge Engineering with Big Data (Ministry of Education); also with the Intelligent Interconnected Systems Laboratory of Anhui Province, Hefei University of Technology, Hefei 230601, China. E-mail: suiye_zhao@163.com, cszzhang@gmail.com, hongrc.hfut@gmail.com, eric.mengwang@gmail.com. M. Xu is with the School of Information Engineering, Zhengzhou University, Zhengzhou, China. E-mail: iexumingliang@zzu.edu.cn. H. Zhang is with the Department of Computer Science, Harbin Institute of Technology (Shenzhen), Shenzhen, China. E-mail: hjzhang@hit.edu.cn. S. Yan is with Sea AI Lab (SAIL), Singapore; also with the National University of Singapore, Singapore 117583. E-mail: shuicheng.yan@gmail.com.

Fig. 2. CRNet architecture. The entire architecture consists of two branches that jointly constrain the model, where the upper one is the deblur branch, and the lower one is the blur branch. A_S and B_S represents unpaired images in sharp domain (S-domain), and A_B and B_B represents unpaired images in blurry domain (B-domain). For the deblur branch, given a blurry image A_B , the blur offset estimation operation generates the blur offset and the adaptive blur correction phase corrects the original image to obtain the latent image A_S . Similarly, inverse blur offset estimation and adaptive blur correction operations are performed to obtain the reconstructed image A_B^* in B-domain, e.g., $A_B \rightarrow A_S \rightarrow A_B^* \approx A_B$. For the blur branch, the entire pipeline can be represented as $B_S \rightarrow B_B \rightarrow B_S^* \approx B_S$ in S-domain. In addition, two multi-scale cropping discriminators are used to distinguish the real and fake samples in both S-domain and B-domain.

deep models also cannot be generalized to the blurry images in different domains. As such, unsupervised deblur models offers advantages over fully-supervised ones, due to the fact that no paired data are required in unsupervised cases [18], [19]. However, current unsupervised deblur models still cannot handle the natural images well, due to the complex contents and details. Besides, current unsupervised deblur models usually obtain the sharp image in an end-to-end manner, by focusing on judging whether the image is sharp. As a result, it may lead to serious color information loss in training, and cause unfriendly visual experience. In fact, most supervised deblur models also suffer from the chromatic aberration issue of a different degree, although paired data are used for training. Some latent images generated by current unsupervised methods are visualized in Figure 1 as examples, where we see clearly that the chromatic aberration clearly exists. Specifically, when the chromatic aberration is obvious, human eyes can easily identify the difference, but cannot figure out how large the numerical color difference. When the chromatic aberration is not obvious, it will make the human eyes inept in observing the color difference. Since color retention ability is an important factor to evaluate the performance, however how to quantitatively evaluate the color retention ability of unsupervised models is still a challenging issue. It is urgent to explore effective metrics to asses and quantify the chromatic aberration produced in unsupervised tasks.

In this paper, we therefore propose effective strategies to solve the chromatic aberration issue and quantify the chromatic aberration in the case without paired data. The major contributions of this paper are summarized as follows:

- Technically, we propose a novel unsupervised color retention network called CRNet for deep blind motion deblurring, which presents novel strategies of blur offset estimation and adaptive blur correction for preserving the color information. CRNet has a strong generation ability, which can well handle both the natural and domain-specific images. To the best

of our knowledge, this is the first study on designing specific unsupervised color retention network for image deblurring. The proposed unsupervised color retention strategy is also general, which can be extended to other tasks involving chromatic aberration, e.g., image deraining/dehazing, image generation, and image transformation-based ones.

- To retain the color information of the original images to the greatest extent, we have a new solution, i.e., learning a mapping from the blurry image to motion offset firstly, and then performing adaptive blur correction on the motion offset to obtain the latent image, thereby preserving the color information and detailed information. A dedicated pyramid global blur feature perception (PGBFP) module is also designed to perceive the non-local information, which can further improve the deblurring performance.
- To quantify the chromatic aberration and assess the color retention ability of our CRNet, we introduce a new metric called Color-Sensitive Error (CSE) based on the color histograms. The objective evaluation of CSE is more consistent with the human's perception and can reflect the detailed information, which will be more accurate than the widely-used PSNR and SSIM metrics. To the best of our knowledge, this is one of very few studies on designing chromatic aberration assessment method. Note that CSE metric can be used for evaluating the color retention abilities of both unsupervised and supervised methods.
- Extensive experiments verified the effectiveness of our CRNet for deblurring in terms of overall restoration performance and color retention, which also has great advantages in restoring the image details.

This paper is outlined as follows. Firstly, a brief review of related deblurring models is presented in Section 2. In Section 3, we introduce the unsupervised CRNet and the quantization metric CSE. Extensive experiments on both natural and domain-specific image datasets are shown in

Section 4. Finally, Section 5 offers the conclusion.

2 RELATED WORK

We briefly introduce the related image deblurring methods to our proposed CRNet.

2.1 Traditional Deblur Methods

Blind motion deblurring aims at restoring a blurry image to a sharp image, with the blur kernel unknown. Traditional methods usually assume that the sharp image or blur kernel meets a certain prior to constrain the solution space of the latent image, e.g., heavy-tailed gradient prior [20], patch prior [21], dark channel prior [22], local maximum gradient prior [23] and multi-scale latent structure prior [24]. But these methods have obvious drawbacks. On one hand, processing the blurry images requires a complex and time-consuming iterative learning process. On the other hand, the restored image is either under-deblurred (the restored image is still blurry) or over-deblurred (obvious artifacts existed in the restored image), due to various prior inaccuracies.

2.2 GAN-Based Deep Deblur Models

With the rapid development and strong learning ability of deep networks, deep deblur models have obtained better performance than traditional ones [16], [25], such as generative adversarial networks (GAN) [26]-based methods.

2.2.1 GAN-Based Supervised Methods.

As a powerful neural network and due to its great success for image representation, most of existing deep deblur models are designed based on GAN. For example, multi-scale images are used as input [12], and then the up-sampled prediction result of small image and larger-scale image are concatenated as the next-scale input, which can well retain the detail of restored images. Strictly speaking, DeblurGAN is the first end-to-end GAN-based supervised deep deblur model that uses the WGAN-GP [27] for training [13]. DeblurGAN-v2 [14] further introduces the feature pyramid network (FPN) into the deblurring task and adds different backbones (e.g., Inception-ResNet and MobileNet), hence achieving a lightweight of the model and obtaining more promising deblurring performance.

2.2.2 GAN-Based Unsupervised Methods.

Training a supervised deep deblurring network has to involve massive labeled data, which suffers from the tricky and time-consuming labeling issue. As such, some unsupervised deep deblurring methods were proposed recently, such as [18] and DisentDeblur [19]. The method proposed in [18] is the first dedicated GAN based unsupervised deblur model, which uses a reblur loss and a multi-scale gradient loss to constrain the solution space and obtains better results on synthetic datasets. DisentDeblur [19] is a disentangled unsupervised deblur model that separates the content attribute and blur attribute in a blurry image, and adds the KL divergence loss to blur encoder to prevent the blur attribute from the process of encoding content attribute. DualGAN [28] and DiscoGAN [29] have similar ideas as the classic CycleGAN [30], but the structures of the generators

and discriminators are different. Although DiscoGAN and DualGAN are not specifically designed for the deblurring tasks, if the sharp domain (S-domain) and blurry domain (B-domain) are considered as two domains, deblurring can be performed accordingly.

3 CRNET: UNSUPERVISED COLOR RETENTION NETWORK FOR BLIND MOTION DEBLURRING

We introduce the idea and the framework of our CRNet in detail, including the chromatic aberration problem, and the corresponding solutions, network architecture and specific constraints for the color retention of our CRNet.

3.1 Problem Statement and Motivation

Chromatic aberration is rather common in the real-world applications. In recent decades, researchers have noticed this issue and started to study this topic [31]. For example, [32], [33] believe that the human eyes also have chromatic aberration, i.e., when we observe an object, the observed color may be different from the object's actual color. In fact, image processing results in vision tasks more or less have color difference in comparison to the original image, which is more obvious in unsupervised cases without paired data. As for deblurring task, even if the unsupervised deblur models can recover the structural information, they still cannot effectively retain the color information of the original image, due to lack of the constraint of paired data. This will directly lead to chromatic aberration with poor visual effects, which will directly limit the applicability in reality.

Generally, two solution schemes can be applied to handle the chromatic aberration. The first one is adding additional constraint on color information, such as the l_1 or l_2 norm on the color histogram vectors of both original blurry and latent images. But the constraint on the color information of entire image is still too weak, as it cannot ensure the color information of each pixel to be approximately accurate. In other words, it still may be a highly ill-posed constraint and cannot be directly used for training model.

The other one is by retaining the color information of original image as much as possible, which has two benefits: 1) there is no need to design complex and time-consuming constraints on color, which simplifies the model training and improves the training speed; 2) directly correcting the original image in a specific way to obtain the latent image, which can avoid the issue that color information of the image degrades in the training process. As such, in this paper we design an unsupervised color retention strategy along this line, and propose new strategies of blur offset estimation and adaptive blur correction operations. Next, we will elaborate the details.

3.2 CRNet Architecture

The architecture of our CRNet is shown in Figure 2. Similar to the current unsupervised models [28], [29], [30], CRNet also contains two branches, i.e., blur branch and deblur branch. The blur branch is used to improve the deblurring performance, although the deblur branch can complete the deblurring task independently. We use A and B to denote unpaired images, and use the subscripts to indicate the

Fig. 3. Structure of blur offset estimation (BOE) network. Given an original image, BOE network will generate a blur offset. The entire structure consists of an encoding process, a transformation process and a decoding process. PGBFP participates in the whole process.

domain in which the image is located. For example, A_S and B_S are unpaired images in sharp domain (S-domain), and A_B and B_B are unpaired images in blurry domain (B-domain). We use I to describe an image that does not need to distinguish which domain it belongs to.

In general, our CRNet includes three major parts: 1) two blur offset estimation (BOE) sub-networks G_{BS} and G_{SB} for B-domain and S-domain; 2) adaptive blur correction (ABC) scheme denoted by F ; 3) two multi-scale cropping discriminators D_B and D_S for two domains. Specifically, given a blurry image A_B and a sharp image B_S , both G_{BS} and G_{SB} respectively generate the corresponding blur offsets and inverse blur offsets. Then, the adaptive blur correction scheme F corrects the blurry image and sharp image into the images in opposite domains. Finally, D_B and D_S are used to distinguish the real and generated samples of the corresponding domains. Note that we use a multi-scale approach for the structural design of discriminators. Traditional “multi-scale” refers to resizing the input image to generate images of different sizes. Then, this set of images is used to compute the adversarial loss. In this case, the discriminator only needs to judge the authenticity of the entire image, while not focusing on the local information. As a result, it will be unreasonable for the deblurring task, since humans only need partial information to determine the authenticity of the entire image. Hence, inspired by the attention mechanism, we propose a variant of the original

multi-scale discriminator, called multi-scale cropping discriminator (MCD). To be specific, we aim at reducing the image size by cropping to make the discriminator pay more attention to the local details of the target image.

The blur offset estimation and adaptive blur correction operations are two key stages for color retention in our model. Different from the current unsupervised deblur models that obtain the restored latent image directly from the blurry image, our CRNet adopts a new “blurry image $\rightarrow$ motion offset $\rightarrow$ latent image” manner. That is, a mapping is firstly learnt from the blurry image to motion offset, and then according to obtained motion offset, the adaptive blur correction operation is performed to obtain the latent image. The adaptive blur correction operation starts from the original image and performs physical correction on the original resolution without learning, which can make the restored image have no chance to produce large chromatic aberration. At the same time, the blur offset estimation network introduces the self-attention mechanism and builds a new module PGBFP to learn global blur information, thereby further retaining the color information of image.

3.3 Blur Offset Estimation (BOE) Network

The BOE network designs a DNN to obtain the blur offset from original image. Since CRNet has two branches, this stage will produce different blur offsets according to different inputs. The structure of the BOE network is shown

Fig. 4. The gradient maps of the blur offsets generated during the training process, where the kind of the selected blur offset is residual.

in Figure 3. Clearly, the entire BOE network contains a dedicated PGBFP module and a blur offset estimation sub-network. The sub-network will play a core role in the BOE stage. The structure design of PGBFP needs to cooperate with that of the sub-network. PGBFP jointly performs the blur offset estimation by cascading with the sub-networks at different scales. For the convenience of description, we introduce the concept of cascade level. The cascade level $\aleph$ denotes the number of information exchanging times between PGBFP and sub-network. That is, a higher cascade level means a deeper sub-network structure.

3.3.1 Pyramid Global Blur Feature Perception (PGBFP)

Convolutional layer is one of the most important components of DNN. However, due to the limited receptive field, convolutional layer cannot perceive non-local information, which leads to poor non-local performance in some tasks. To perceive the non-local information, SAGAN [34] introduces the attention mechanism the DNN. In this paper, we also introduce PGBFP to learn global blur information. Unlike the self-attention manner of SAGAN, we further refine the self-attention block and embed it into our model. Specifically, after getting the self-attention feature map, we directly feed it into a special down-sampling block containing a 1-by-1 convolutional layer and an average pooling layer to change the dimension and size of the feature maps, so that the multiple self-attention blocks can be well connected. Figure 3 shows the specific details on PGBFP.

Through considering the structure of blur offset estimation sub-network and graphic memory limitation, we set the cascade level $\aleph$ to 3 in this study. For the i -th cascade, let r_i be the feature map from the sub-network into PGBFP, let m_i denote the feature map from the PGBFP into the sub-network, and let s_i be the internal feature maps to be fused, then the cascaded operation requires the size consistency between the feature maps generated by the PGBFP and sub-network, i.e., $size(r_i) = size(m_i)$. In addition, $size(r_i) = size(s_{i-1})$ is needed to ensure that r_i and s_{i-1} can be fused appropriately within PGBFP. Technically, the operations within the PGBFP can be described as follows:

$$\begin{cases} s_0 = O \\ n_i = r_i + s_{i-1} \\ m_i = \text{Softmax} \left((Q(n_i))^T \cdot K(n_i) \right) \cdot V(n_i) \\ s_i = A(S(m_i)) \end{cases} \quad (2)$$

where $i = 1, 2, \dots, \aleph$, $Q(\cdot)$, $K(\cdot)$, $V(\cdot)$ and $S(\cdot)$ denote 1-by-1 convolutional operations, $A(\cdot)$ is average pooling, O is the feature map with all zero values, $\bullet$ denotes matrix multiplication, and $+$ denotes element-wise addition.

For the PGBFP with a certain cascade level, the function is to obtain the output feature maps $m_1, m_2, \dots, m_\aleph$ according to the input feature maps $r_1, r_2, \dots, r_\aleph$. The process can be simply expressed by the following formula:

$$m_i = PGBFP(r_i), i = 1, 2, \dots, \aleph \quad (3)$$

3.3.2 Blur Offset Estimation Sub-network

For the macrostructure design of the sub-network, we follow the classical “encoding, transforming and decoding” structure. Specifically, the sub-network has a pre-processing layer, three down-sampling layers, nine residual blocks [35], three up-sampling layers and a post-processing layer. The pre-processing and post-processing layers change the channel dimension by a 7-by-7 convolution kernel. Meanwhile, the pre-processing layer uses a larger receptive field to perturb the information on original size, which can enhance the learning ability of network. The post-processing layer sets the affine parameters of instance normalization to a learnable state and uses the Tanh function in the latter layer to map the values to $[-1, 1]$. The up-sampling and down-sampling layers use a 3-by-3 convolution kernel to change the channel dimension and size of the feature maps simultaneously. The number of up-sampling and down-sampling layers correspond to the cascade level. For the residual blocks, we use the same settings, except that the instance normalization is used to replace the batch normalization.

Next, we will introduce how the blur offset estimation sub-network is cascaded with the PGBFP. For the i -th cascade, let d_i be the feature map generated after the down-sampling layer, let u_i be the feature map generated after the up-sampling layer, then d_i and r_i will be equal, i.e., $d_i = r_i$. Besides, the sizes of u_i and m_i should be equal as well, i.e., $size(u_i) = size(m_i)$. Thus, the process of each cascade in the sub-network can be described as follows:

$$\begin{cases} u_\aleph = Res(d_\aleph) \\ m_i = PGBFP(d_i) \\ u_{i-1} = Upsampling_i(u_i + m_i) \end{cases} \quad (4)$$

where $i = 1, 2, \dots, \aleph$, $Res(\cdot)$ denotes the module composed of 9 residual blocks, and $Upsampling_i(\cdot)$ denotes the up-sampling layer in the i -th cascade.

Fig. 5. The adaptive blur correction (ABC) stage with three kinds of schemes. The figure shows the process of correcting the blurry image to the S-domain. The reverse operation is the same, because the BOE stage has generated different blur offsets.

Finally, for the BOE network, given unpaired blurry image A_B and sharp image B_S , it will produce the normal blur offset f and inverse blur offset f^* respectively as follows:

$$\begin{aligned} f &= G_{BS}(A_B) \\ f^* &= G_{SB}(B_S) \end{aligned} \quad (5)$$

To understand how the blur offset works in subsequent adaptive blur correction, we visualize the blur offset of some images with different iteration times for observation in Figure 4. We see that BOE network firstly obtains rough image contour in early stage. As the number of iterations increases, the details of the blur offsets between blurry and sharp images can be gradually learned and becomes clearer.

3.4 Adaptive Blur Correction (ABC) Stage

Adaptive blur correction has the greatest contribution to the color retention. At this stage, the model selects a correction scheme for the blur offset and does not need to learn parameters except for the adaptive factor. However, it is precisely because the model basically does not need learn parameters at this stage and only directly corrects the original image at the physical level, so that the network can be prevented from losing useful color information in training.

3.4.1 ABC Calculation

This stage aims at designing a scheme F to correct the original image based on the blur offset generated in the BOE stage. By unifying the normal blur offset f and inverse blur offset f^* into f , the entire blur correction process can be performed as follows:

$$F(I, f) = I \odot (f * \varphi) \quad (6)$$

where $\odot$ denotes the correction scheme, $*$ denotes the multiplication operation, and φ denotes an adaptive factor.

The adaptive factor φ makes the blur correction an adaptive operation, which can further enhance the model's color retention ability. The adaptive factor is initialized to 0, and

finally a stable value will be learned after training. As can be seen from Eq. (6), φ is equivalent to the weight of blur offset. The larger the weight, the higher the correction degree of the original image, and the more color information will be lost. Next, we consider two cases: 1) no adaptive factor is introduced (this is equivalent to directly setting the adaptive factor to 1), i.e., the weight of the blur offset is equal to 1. In fact, the weight of the blur offset is high, and a lot of color information will be lost during the blur correction process; 2) introducing an adaptive factor. The involvement of φ provides a gradual process, i.e., φ is initially set to 0, and the restored image will be the original image, i.e., no color information loss. In the training process, φ will gradually increase, which leads to an increase in the weight of the blur offset. From the experimental observation, φ can eventually increase to 0.55, which is 45% lower than without introducing φ , i.e., the adaptive factor is helpful.

3.4.2 Blur Correction Scheme Selection

The scheme of blur correction is in fact not unique, since it corresponds to different kinds of blur offsets learned in ABC stage. In this paper, we tested three kinds of blur offsets. (1) **Residual**. This blur correction scheme is element-wise addition, i.e., the BOE stage will obtain the residual of the same size as the original image, and then an element-wise addition is performed between the original image and the residual to correct the blur; (2) **Element-wise weights**. This kind of blur correction scheme is element-wise multiplication, i.e., the BOE stage obtains the weights of each pixel in the original image, and then perform an element-wise multiplication to correct the blur; (3) **Non-local attention**. This kind of blur correction scheme is matrix multiplication, inspired by the attention mechanism. Specifically, a rectangular attention map on each channel is obtained, and then a matrix multiplication is performed to correct the blur.

Figure 5 shows the three kinds of blur offsets with corresponding correction schemes. The element-wise weight and residual blur offset are designed according to the original image scale. This has two advantages: 1) the obtained blur offset is easy to understand, and the specific content information can still be seen from the blur offset; 2) the design of the blur correction scheme is simple, since there is no complicated calculation. Unlike the blur offsets of original scale, non-local attention is more difficult to understand, and the calculation is more complicated and needs additional constraints. By quantitative comparison in our study, the original scale blur offsets obtain better performance.

3.5 Loss Function

We first describe the objective function of our CRNet as the weighted sum of several loss functions:

$$\mathcal{L} = \lambda_{adv}\mathcal{L}_{adv} + \lambda_{cc}\mathcal{L}_{cc} + \lambda_p\mathcal{L}_p \quad (7)$$

where $\mathcal{L}_{adv}$ is adversarial loss using to convert the image between two different domains, $\mathcal{L}_{cc}$ is cycle-consistency loss using to maintain the content information of the original image, $\mathcal{L}_p$ is perceptual loss used to alleviate the artifact generated in the deblurring process. $\mathcal{L}_{adv}$, $\mathcal{L}_{cc}$, $\mathcal{L}_p$ are the weighting factors of the corresponding loss. These losses

TABLE 1
Symbol table, which stores the symbols and the corresponding descriptions used in Section 4.

Sym	Description
C	HSV color space, $C = \{h, s, v\}$.
c	A certain component in HSV, $c \in C$.
V	The entire color histogram in HSV, containing three color histogram component vectors, $V = \{V^c c \in C\} = \{V^h, V^s, V^v\}$. The subscript can be added to indicate the image.
$\Delta^k V^c$	The K -th order difference of histogram vector V^c . Particularly, $\Delta^1 V^c$ is denoted as ΔV^c .
$\sigma(V^c)$	The discrete integral of vector $ V^c $.
p	The primary peak points: $((p)_x, (p)_y)$.
q	The auxiliary peak points: $((q)_x, (q)_y)$.
Q	The set of auxiliary peak points: $((q_1)_x, (q_1)_y), \dots, ((q_{N-1})_x, (q_{N-1})_y)$, where N denotes the sum of the number of the primary peak and auxiliary peak(s).
t	The uniform representation of peak points: $((t)_x, (t)_y)$, $t \in p \cup Q$. Superscripts and subscripts indicate a certain component and image, respectively.
ω	The weight of different components in the entire color histogram, $\omega = \{\omega_c c \in C\} = \{\omega_h, \omega_s, \omega_v\}$.
γ	The weight of different peaks in all peaks. Superscripts and subscripts indicate a certain component and the type of peak, respectively. $\gamma = \{\gamma_t^c c \in C, t \in \{p, q\}\} = \{\gamma_p^h, \gamma_p^s, \gamma_p^v, \gamma_q^h, \gamma_q^s, \gamma_q^v\}$.

are commonly used in deblurring tasks. Note that for perceptual loss, VGG19 [36] pre-trained on ImageNet [37] is applied in the experiment.

4 COLOR-SENSITIVE ERROR (CSE): NEW CHROMATIC ABERRATION QUANTIZATION METRIC

The proposed unsupervised color retention network cannot be directly used as a training loss, but from another perspective, this idea is suitable for designing new metric to assess and quantify the chromatic aberration. Specifically, we propose a new image quality assessment method, termed Color-Sensitive Error (CSE), to quantify the chromatic aberration between restored image and target image. Note that the target image in our method can be the ground truth for paired data in supervised case, and the original blurring image for unpaired data in unsupervised case. That is, CSE metric can be employed in both supervised and unsupervised cases. Technically, we design the CSE metric based on the color histograms of the two images. For an image, the gray values that account for more will converge into the peaks in the histogram, and it will be easier for human eyes to perceive the faint change of the peaks. As such, CSE is clearly designed around the peaks to conform to the human eye's intuition. Specifically, we describe the entire calculation process of CSE as the weighted average of the errors on the three components h , s and v . On each component, the error includes three parts: 1) **peak-area error (PAE)** that is used to measure the overall difference between each pair of peaks in the color histograms of the two images. This error is designed from the perspective of peak height and area; 2) **peak-offset error (POE)** that is used to measure the difference between the primary peaks of the two compared color histograms, and also the difference

between auxiliary peaks. This error is designed from the perspective of peak position; 3) **component weight** is used to further optimize the error.

Next, we will elaborate the details on the above three parts. Since the calculation and derivation process of the CSE metric involves some symbols, we firstly describe the used symbols in Table 1 for clear presentation.

4.1 Peak-Area Error (PAE)

We first describe the PAE in detail. As we know, HSV color space $C = \{h, s, v\}$ is closer to the human's perception than RGB space, so we quantify the chromatic aberration in HSV space. Given an RGB color image $I \in \mathbb{R}^{H \times W \times C}$, we firstly convert it into $\tilde{I} \in \mathbb{R}^{H \times W \times C}$ in HSV space. For any HSV image, the value ranges of the three components h , s , v are different. For each component $c \in C$, the value width R_c can be defined as the following formula:

$$R_c = \sup_{z \sim Z^c} [z] - \inf_{z \sim Z^c} [z], \quad c \in C \quad (8)$$

where $\sup[\cdot]$ and $\inf[\cdot]$ denote the supremum and infimum values respectively, and Z^c denotes all possible distributions of the color histogram of c component. For an integer HSV image, the value width of h component is 180, while the value width of s and v components is 256. Prior to computing the color histograms of images, a factor can be set, i.e., *histsize*, which determines the length of a single partition in the histogram vector.

In Figure 6, we show examples of the color histograms of an image under different *histsize*s. Clearly, a larger *histsize* corresponds to a smoother histogram, but it may not be helpful and conducive to the quantitative metric. Specifically, when *histsize* is large enough, the difference between the two histograms will become small. More importantly, large *histsize* may merge some important peaks in the histograms, directly making the difference between two histograms insignificant. On the contrary, a small *histsize* will lead to a sharper histogram that will be more numerically accurate and detailed. However, *histsize* should not be too small, since in such case it will be difficult to determine the peaks to be calculated. Therefore, we make a trade-off in this paper. According to our experiments, we set *histsize* to the greatest common divisor of 180 and 256, i.e., 4, to quantify the metric and visualize the color histograms.

Then, we can obtain a set of color histogram vectors $V = \{V^c | c \in C\}$ over each component of image $\tilde{I}$. Let L_c be the number of partitions on each component, i.e., the length of vector V^c , it can be calculated by the following formula:

$$L_c = \frac{R_c}{\text{histsize}}, \quad c \in C \quad (9)$$

In the color histogram, the peaks play a decisive role in human intuition and perceiving the difference. To highlight the importance of peaks and prevent the non-peak elements from participating in the calculation as much as possible, we consider the filterability of the difference to those non-peak elements and make a first-order difference to V^c to obtain the difference vector ΔV^c by the following formula:

$$\begin{cases} \Delta^0 V^c = V^c, & c \in C \\ \Delta^k V_i^c = \Delta^{k-1} V_i^c - \Delta^{k-1} V_{i+1}^c, & i = 1, \dots, L_c - k \end{cases} \quad (10)$$

Fig. 6. An image and its color histograms with different *histsize*s in HSV color space. In this paper, we consistently set *histsize* = 4 to quantify the chromatic aberration and visualize the color histogram in all experiments.

Fig. 7. Examples of computing the discrete integral according to two different color histogram vectors of the same component. Where the discrete integral represents the area enclosed by the histogram vector and the 0 axis, as shown in the green area.

where Δ^k denotes k -th order difference, V_i^c denotes the i -th element of V^c . After filtering out those non-peak elements and obtaining the difference vector ΔV^c , the discrete integral can be obtained to indirectly reflect the height and area of all the peaks in the histogram vector. Specifically, we compute the discrete integral $\sigma(\Delta V^c)$ of ΔV^c according to the difference vector ΔV^c as follows:

$$\begin{aligned} \sigma(V^c) &= \sum_{i=1}^{L_c-1} \left(\frac{1}{2} \times (|V_i^c| + |V_{i+1}^c|) \times 1 \right) \\ &= \frac{1}{2} |V_1^c| + |V_2^c| + \dots + |V_{L_c-1}^c| + \frac{1}{2} |V_{L_c}^c| \quad (11) \\ &= \sum_{i=1}^{L_c} |V_i^c| - \frac{1}{2} (|V_1^c| + |V_{L_c}^c|), \quad c \in C \end{aligned}$$

where V_i^c denotes the i -th element of V^c , $|\cdot|$ denotes absolute value operation.

For ease of understanding, Figure 7 shows two comparison examples of the calculation from V^c to $\sigma(\Delta V^c)$.

Similarly, given two HSV images $\tilde{I}_a \in \mathbb{R}^{H \times W \times C}$ and $\tilde{I}_b \in \mathbb{R}^{H \times W \times C}$, we can obtain the color histogram vectors V_a^c and V_b^c , the difference vectors ΔV_a^c and ΔV_b^c , and the discrete integral $\sigma(\Delta V_a^c)$ and $\sigma(\Delta V_b^c)$ on each component.

Finally, the peak-area error (PAE) is calculated as

$$\begin{aligned} PAE_c &= |\sigma(\Delta V_a^c) - \sigma(\Delta V_b^c)| \\ PAE &= \sum_c PAE_c, c \in C \end{aligned} \quad (12)$$

4.2 Peak-Offset Error (POE)

The calculation process of POE involves the numerical calculation of the peak, so we first give a unified expression of the peaks. Given a color histogram vector V^c , we define $t^c = ((t^c)_x, (t^c)_y)$ as the peak point, where x and y indicate the peak position and peak value. Specifically, $(t^c)_y$ is a certain value in V^c , while $(t^c)_x$ is the corresponding index in V^c . In general, for a peak point t^c in V^c , the following two conditions need to be satisfied: ①: $V_{(t^c)_x}^c > V_{(t^c)_x-1}^c$, ②: $V_{(t^c)_x}^c > V_{(t^c)_x+1}^c$. In particular, the peak point can either be the starting point satisfying condition ②, or the ending point satisfying condition ①. In fact, we do not need to solve all peak points in the distribution, since the values of some peak points are too small, so that they can be considered as non-peaks. Therefore, in the design of POE, we only select the peak point with the highest value (i.e., primary peak) and some peak points with the next highest values (i.e., auxiliary peak(s)). To facilitate the statement, we use p to denote primary peak, use q to denote auxiliary peak, and use Q to denote the set of auxiliary peaks, i.e., $Q = \{q_1, q_2, \dots, q_{N-1}\}$. Note that the primary peak is necessary and the auxiliary peaks are not necessary. In other words, Q can be an empty set in some cases. To define the POE metric, the following two steps will be performed:

Peak positioning. Given a histogram vector V^c , how to find and position the peaks p and q is core process. In this paper, we propose a peak positioning algorithm to obtain the needed peaks on histogram vector V^c , as summarized in Algorithm 1. Specifically, we perform the following three operations in order: 1) setting an empty peak set T to store the peaks; 2) sorting the vector V^c in descending order to obtain the index vector idx , i.e., $V^c[idx[i]] > V^c[idx[i+1]], i = 1, \dots, L$; 3) adding zero elements before the start point and after the end point of the vector V^c , i.e., $V^c \leftarrow [0, V_1^c, V_2^c, \dots, V_L^c, 0]$. Then, the vector idx is traversed to determine whether the current value satisfies the condition of becoming a peak. If yes, adding $(idx[i], V^c[idx[i]])$ to T . Finally, if the number of peaks in T reaches the user-specified maximal value N or the vector

Fig. 8. Aligned peaks between V_a and V_s , and the non-aligned peaks between V_a and V_b , V_b and V_s . Peak alignment eliminates misalignment by swapping the primary peak and auxiliary peak in the vectors.

idx traversal is completed, the peak set T will be returned. Note that the primary peak is necessary, i.e., N should be more than 1. In this paper, we set N to 2 to obtain the primary peak p and one auxiliary peak q .

Peak aligning. After obtain the primary peak and auxiliary peaks, it is still possible to encounter a tricky problem, i.e., peak misalignment. While the peak misalignment issue will directly affect the accuracy of the metric. For example, given two vectors V_a^c and V_b^c , the peaks p_a^c, q_a^c for V_a^c and the peaks p_b^c, q_b^c for V_b^c can be obtained by peak positioning. For the primary peaks p_a^c and p_b^c , if $|(p_a^c)_x - (p_b^c)_x|$ is very large, $|(p_a^c)_y - (p_b^c)_y|$ is very small, and meanwhile similar arguments for the auxiliary peaks, then the misalignment of peaks will exist, as shown in Figure 8. We can see that the misalignment of peaks exists between p_a and p_b , and between p_b and p_s , which will directly result in inaccurate calculation results. Thus, we design a peak aligning algorithm to overcome this issue, as described in Algorithm 2, where the parameter ϑ is set to 0.3 in this paper.

Note that the peak offset is defined as a distance between the x coordinates of the two peaks. Since we consider different kinds of peaks (i.e., primary and auxiliary peaks), both the primary and auxiliary offsets will be calculated. However, the primary peaks are usually more important than the auxiliary peaks, so we propose to weight the two kinds of offsets according to their y coordinates. That is, given two images $\tilde{I}_a, \tilde{I}_b \in \mathbb{R}^{H \times W \times C}$, we can obtain the aligned peaks t_a^c and t_b^c , $t \in (p \cup Q)$ based on the color histogram vectors $\tilde{V}_a^c$ and $\tilde{V}_b^c$ on each component of the HSV color space. Technically, the weights γ_t^c for the primary and auxiliary peaks are defined as follows:

$$\gamma_t^c = \frac{(t_a^c)_y + (t_b^c)_y}{\sum_t ((t_a^c)_y + (t_b^c)_y)}, c \in C, t \in (p \cup Q) \quad (13)$$

where the superscript and the subscript of γ_t^c indicate the component and type of peaks, i.e., $\gamma_t^c = \{\gamma_p^h, \gamma_p^s, \gamma_p^v, \gamma_{q_1}^h, \gamma_{q_1}^s, \gamma_{q_1}^v, \dots, \gamma_{q_{N-1}}^h, \gamma_{q_{N-1}}^s, \gamma_{q_{N-1}}^v\}$. After that, the peak offsets can be obtained by weighted sum of the primary and auxiliary peaks as follows:

$$d_c = \gamma_p^c |(p_a^c)_x - (p_b^c)_x| + \sum_q \gamma_q^c |(q_a^c)_x - (q_b^c)_x|, c \in C \quad (14)$$

Algorithm 1 Peak positioning.

Input:

V^c - An arbitrary vector.

N - The number of peak points to be found on this vector.

Output:

T - A set of peak points.

```

1:  $T \leftarrow \emptyset$ 
2:  $L \leftarrow \text{length}(V^c)$ 
3:  $idx \leftarrow \text{Descending-argsort}(V^c)$ 
4:  $V^c \leftarrow 0 \cup V^c \cup 0$ 
5: for  $i \leftarrow 1$  to  $L$  do
6:   if  $V^c[idx[i]] \geq V^c[idx[i] + 1]$  and  $V^c[idx[i]] \geq V^c[idx[i] - 1]$  then
7:      $T \leftarrow T \cup (idx[i], V^c[idx[i]])$ 
8:      $N \leftarrow N - 1$ 
9:   if  $N = 0$  then
10:    return  $T$ 
11:   end if
12: end if
13: end for
14: return  $T$ 

```

Finally, the POE metric is formally defined as follows:

$$POE_c = e^{\sqrt{d_c}} \quad (15)$$

$$POE = \sum_c POE_c, \quad c \in C$$

4.3 Component Weight and CSE Metric

Considering that the contributions of different components to the human's perception is different, we also weight the obtained errors PAE and POE in component dimension. Similar to the weights for peaks, we utilize the value of the peaks to design the weights on components as follows:

$$\omega_c = \frac{\max(V_a^c) + \max(V_b^c)}{\sum_c (\max(V_a^c) + \max(V_b^c))} \quad (16)$$

where $\max(\cdot)$ denotes the maximum value operation.

The purpose of weighting the components is to highlight the contribution of different components, but when the distribution of a component of the two histograms is flat enough and even without obvious peaks (such as the two distributions of component vectors in Figure 9), it will be difficult to appropriately weight the components. Specifically, in this case, the expected error value of the two histogram vectors should not be very large, since these two vectors looks similar. However, in fact, although the peak of the histogram vector is very inconspicuous, the peak-positioning algorithm still can locate the required peaks. However, such peaks are usually inaccurate and will lead to erroneous results, as shown in Figure 9. As a result, when the histogram vector of the target image is flat enough, we compute the root mean square error between restored image and target image, instead of using the peaks to accumulate the error due to the invalidity of the peaks in those cases. Note that this will bring one extra benefit. That is, only when the error of the remaining components is

Algorithm 2 Peak aligning.**Input:**

$T_a = \{t_{a_1}, t_{a_2}, t_{a_3}, \dots, t_{a_{N-1}}\}$ - A set of peaks for vector a , containing a primary peak and one or more auxiliary peaks, where t_{a_1} is primary peak.

$T_b = \{t_{b_1}, t_{b_2}, t_{b_3}, \dots, t_{b_{N-1}}\}$ - A set of peaks for vector b , containing a primary peak and one or more auxiliary peaks, where t_{b_1} is primary peak.

ϑ - The threshold to determine whether an alignment operation should be performed.

Output:

$P_a = \{t_{a_1}, t_{a_2}, t_{a_3}, \dots, t_{a_{N-1}}\}$ - A set of aligned peaks for vector a .

$P_b = \{t_{b_1}, t_{b_2}, t_{b_3}, \dots, t_{b_{N-1}}\}$ - A set of aligned peaks for vector b .

```

1:  $N_a \leftarrow |T_a|$ 
2:  $N_b \leftarrow |T_b|$ 
3: if  $N_a \neq N_b$  then
4:   ERROR
5:   return ;
6: end if
7: for  $i \leftarrow 1$  to  $N_a - 1$  do
8:   for  $j \leftarrow i + 1$  to  $N_a$  do
9:     if  $|(t_{a_i})_x - (t_{b_i})_x| < |(t_{a_i})_x - (t_{b_j})_x|$  and
        $|(t_{a_j})_x - (t_{b_j})_x| < |(t_{a_j})_x - (t_{b_i})_x|$  then
10:      continue
11:    end if
12:     $D \leftarrow 0.5 * |(t_{a_i})_y - (t_{a_j})_y| + 0.5 * |(t_{b_i})_y - (t_{b_j})_y|$ 
13:     $R \leftarrow 0.25 * ((t_{a_i})_y + (t_{a_j})_y + (t_{b_i})_y + (t_{b_j})_y)$ 
14:    if  $D/R \leq \vartheta$  then
15:       $t_{a_i}, t_{a_j} \leftarrow t_{a_j}, t_{a_i}$ 
16:    end if
17:  end for
18: end for
19: return  $T_a, T_b$ 

```

small enough, the error of the flat component will play a leading role, otherwise the flat component will only occupy a small proportion, which does not conflict with our original intention of using peaks to assess the color difference. In this paper, when the maximum value of a certain component of the histogram is less than 1/4 of the overall maximum value, we will consider this component as flat one.

Then, we utilize R_c to perform the normalization, and the CSE on certain component is calculated as follows:

$$CSE_c = \begin{cases} \frac{\omega_c}{R_c} \cdot e^{\sqrt{d_c}} \cdot |\sigma(\Delta V_a^c) - \sigma(\Delta V_b^c)| & \text{if not flat,} \\ \frac{\omega_c}{R_c} \sqrt{\frac{1}{L_c} \sum_{j=1}^{L_c} ((V_a^c)_j - (V_b^c)_j)^2} & \text{if flat.} \end{cases} \quad (17)$$

Finally, the entire CSE error can be calculated by accumulating the errors over all components as follows:

$$CSE = \sum_c CSE_c, \quad c \in C \quad (18)$$

Remarks. PSNR and SSIM are two widely-used quantitative evaluation metrics to assess the image deblurring quality, however they clearly have two shortcomings: 1) the objective evaluation result is not consistent with the human perception, which may be because the detailed

Fig. 9. Comparison of the color histogram vector of the same component of two images. Intuitively, the error between them seems small, but in fact the error is large. This error is mainly caused by the distributions of the histogram vectors, i.e., it is too flat to locate the peaks correctly. Without properly considering the weights, these two vectors will produce a large peak-offset error, with the order of magnitude being e^{56} .

Fig. 10. Comparison of visual and color histograms of the original blurry image (left) and sharp image (right). It is clear that the blurry image contains most of the color information in image, which can also be concluded from the similar distributions of color histograms.

and color information are not appropriately considered; 2) paired data are needed for calculating the values. In contrast, our CSE metric is based on the statistical principle on the HSV color histogram vectors, so it will be more close the human perception and sensitive to color changes. It is also noteworthy that although the CSE metric is originally proposed to assess the color retention power, the values of it can also be used for evaluating the ability of restoring the details of images to some extent. In addition, CSE metric is a general metric that can be used to evaluate different kinds of deblurring models with/without paired data, including unsupervised, fully-supervised and semi-supervised ones. Specifically, for paired data with sharp image, we calculate the CSE based on the restored and sharp images. While for the unpaired data without sharp image, we can directly calculate the CSE based on the restored image and original blurry image. Because we have visually and experimentally observed that the original blurry image can still contain most of the color information in image, as shown in Figure

Fig. 11. Visualization results of deblurring on GOPRO. We show four groups of visualizations. In each group, the top row contains the blurry image, sharp image (i.e., ground truth), and the latent images recovered by different methods. The second row represents the corresponding color histograms. The bottom row contains two parts, where the left part is the local details of the color histogram, and the right part is the chromatic aberration metrics (CSE-1 and CSE-2). The lower CSE-1 and CSE-2, the better of the deblurring result.

TABLE 2

Comparison of quantitative deblurring results between our CRNet and other state-of-the-art methods on GOPRO.

Method	PSNR↑	SSIM↑	CSE-1↓	CSE-2↓
CycleGAN [30]	23.9796	0.8211	119.2788	115.8199
DualGAN [28]	25.2129	0.8793	184.1219	179.3184
DisentDeblur [19]	26.7884	0.8830	58.6419	56.3577
Ours	28.3114	0.9054	48.5290	46.9309

10. As such, directly using the blurring image for evaluating the unsupervised methods will be reasonable, which has also been verified by experiments.

5 EXPERIMENTS

We evaluate CRNet qualitatively and quantitatively, and illustrate the comparison results with several closely related models. A natural image dataset GOPRO [12] and a domain-specific synthetic dataset CelebA [38] are employed for evaluations. We use an Nvidia RTX2080Ti graphics card with 10G memory for our experiments.

5.1 Evaluation and Implementation

Evaluation methods. Note that the related models that can be included for comparison are limited, since only a few unsupervised deblur models have been proposed in the field.

Fig. 12. Visualization results of image deblurring on CelebA. In each group, the first row contains the deblurred image comparison, histogram detail comparison and CSE comparison. The second row compares the color histograms of the deblurred images of each method. The lower CSE-1 and CSE-2, the better of the deblurring result.

TABLE 3

Comparison of quantitative deblurring results between our CRNet and other state-of-the-art methods on CelebA.

Method	PSNR $\uparrow$	SSIM $\uparrow$	CSE-1 $\downarrow$	CSE-2 $\downarrow$
CycleGAN [30]	19.2961	0.6211	473.2691	543.4212
DualGAN [28]	-	-	2433.7568	2979.2196
DisentDeblur [19]	20.3984	0.6484	372.0668	434.6888
Ours	20.3901	0.6516	265.9458	336.1518

i.e., CycleGAN [30], DualGAN [28] and DisentDeblur [19].

Evaluation metrics. For the numerical evaluations, the commonly-used PSNR and SSIM metrics, and our CSE metric are employed. Although we mainly discussed the unsupervised deblurring task without ground truth, we can still compute the CSE using the original blurry image or sharp image. Specifically, we use CSE-1 to quantify the chromatic aberration between the restored and blurry images, and use CSE-2 to quantify the chromatic aberration between the restored image and the sharp image.

In this study, three unsupervised methods are compared,

Fig. 13. Zooming in of partial details of the deblurred images on CelebA. For each pair of results, the upper one denotes the blurry image, the latent images restored by different methods and sharp image (ground truth). The lower part zooms in the local detail in the red boxes.

Implementation details of CRNet. The proposed deblur model is implemented on PyTorch [39]. In the training phase, the weighting factors λ_{adv} , λ_{cc} and λ_p are respectively set as 1, 20, and 0.1 for CRNet in all experiments. We use the Adam optimizer [40] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$, the initial learning rate is set to 0.0002, the learning rate remains unchanged in the first 40 epochs and decays linearly to 0 in the last 40 epochs. In the feedforward of the discriminator, we follow [30] to set the sample buffer pool to allow the model to learn more general inter-domain relationships. In the testing phase, the blur offset estimation network G_{SB} and the discriminators D_S and D_B will be removed. Given a blurry image I_B , the testing process is performed as follows:

$$I_S = F(I_B, G_{BS}(I_B)) \quad (19)$$

5.2 Experiments on GOPRO

We first evaluate each model for image deblurring on the most commonly used natural image dataset GOPRO. It

contains 2,103 pairs of training data and 1,111 pairs of test data. Each pair of data includes a blurry image and a sharp image, and the size of each image is 1280*720 pixels. Since GOPRO is a natural image dataset, it is less visually sensitive than the domain-specific dataset and we did not add any additional details for comparison.

Due to graphics card memory limitation, it is difficult to evaluate the performance at the original resolution, so we resize the original image to intermediate size and then crop the image size into 128*128 pixels for training. In the test phase, we resize the original image to 224*128 resolution to maintain the same length-width ratio as the original image.

The numerical evaluation results are described in Table 2, and some visual results are shown in Figure 11, where we use the same settings as GOPRO dataset. We see that: 1) our CRNet surpasses the other state-of-the-art methods in terms of PSNR, SSIM and our CSE metrics in Table 2. However, it should be noted that the performance improvement of the methods in terms of both PSNR and SSIM is usually limited, that is, PSNR and SSIM cannot distinguish the performance difference significantly due to their inability to fully consider the detail in images; 2) compared with our CRNet in Figure 11, other unsupervised methods cannot effectively preserve the color information of images, which can also be demonstrated by the results of CSE-1 and CSE-2. That is, our CRNet can obtain better deblurring performance, while effectively retaining the color information. Note that the orders of model performance for each method are always consistent between CSE-1 and CSE-2. That is, calculating the assessment results from the original blurry images or the ground-truth sharp images are indeed feasible, which will be important for the cases without paired data.

5.3 Experiments on CelebA

We then evaluate each model for deblurring on the domain-specific synthetic dataset CelebA that is a face attribute dataset. CelebA contains 202,599 sharp face images, and the size of each image is 178*218. In our experiment, we use an aligned and cropped version of the CelebA dataset [38]. Since CelebA only contains sharp images, we need to generate motion blur for them manually. In this study, we follow the motion blur generation algorithm of [13]. For the parameter setting, we set the impulsive shake probability as 0.005 and the max movement length as 20.

In the training phase, we similarly resize the original image to intermediate size and then crop the image size into 128*128 pixels. In the testing phase, we perform the deblurring on the original resolution.

In this study, the method proposed in [29] is not included for comparison, since it performs poorly on the domain-specific dataset. Due to different settings, we cannot directly use the pre-trained model provided by the authors. As such, we used the same settings to retrain the compared models on CelebA for the fair comparison, according to the codes provided by authors. The quantitative deblurring results on CelebA are described in Table 3, and some visual results are shown in Figure 12. We find from Figure 12 that other compared unsupervised methods produce obvious chromatic aberration in the latent images, as can also be seen from the quantitative results of CSE-1 and CSE-2. In contrast,

TABLE 4

Results of ablation studies. The upper part of shows the model performance after gradually adding components into the baseline, the middle part shows the model performance after deleting each component, and the lower part shows the comparison results via different blur offset types.

Method	Dual blur branch	MCD	Perceptual loss	PGBFP	PSNR↑	SSIM↑
Only deblur branch (baseline)					27.0340	0.8868
Add dual blur branch	✓				27.9178	0.8988
Add MCD	✓	✓			28.1765	0.8985
Add perceptual loss	✓	✓	✓		28.1588	0.9020
Add PGBFP	✓	✓	✓	✓	28.3114	0.9054
Only delete dual blur branch		✓	✓	✓	27.3346	0.8971
Only delete MCD	✓		✓	✓	28.0165	0.9007
Only delete perceptual loss	✓	✓		✓	27.9213	0.8991
Only delete PGBFP	✓	✓	✓		28.1588	0.9020
All components included	✓	✓	✓	✓	28.3114	0.9054
Non-local attention blur offset	✓	✓	✓	✓	21.1787	0.7075
Element-wise weight blur offset	✓	✓	✓	✓	28.1795	0.9029
Residual blur offset	✓	✓	✓	✓	28.3114	0.9054

our CRNet can effectively retain the color information of the original image after deblurring, i.e., smaller CSE-1 and CSE-2. As can be found from the numerical results in Table 3, our CRNet is highly competitive with the state-of-the-art DisentDeblur [19], although it is originally proposed for processing the domain-specific datasets. It is noteworthy that DisentDeblur obtains a higher PSNR value than ours, however a higher PSNR value may not indicate that the restored image is sharper, i.e., the PSNR is inaccurate in reality. To illustrate this issue, we visualize some partial details of the restored images in Figure 13 for observation. As can be seen, the detail of the recovered image by DisentDeblur are in fact worse than our CRNet, although it has a slightly higher PSNR value. In other words, we should assess the deblurring result of each model qualitatively and quantitatively in future. It is also noted that the performance difference evaluated by our CSE metric is more significant than the PSNR and SSIM metrics, because our CSE metric fully considers the color information that can also be considered as the detailed information contained in the images.

5.4 Ablation Study

We conduct ablation experiments to prove the effectiveness of different components and modules in our CRNet, and verify their effects on the performance. Specifically, we verify the effectiveness of four components on dataset GOPRO: 1) **dual blur branch**; 2) **multi-scale cropping discriminator**; 3) **perceptual loss**; 4) **PGBFP**.

Therefore, a complete ablation study in our study should have 16 items, since each component has two possibilities of adding or removing. In fact, we have carried out all these experiments, but considering the space limitation, we only report the ablation study by gradually adding them, or removing certain component for comparison. The results of ablation study are described in Table 4. We see that adding both dual blur branch and PGBFP can significantly improve the performance of deblurring. We also notice that deleting the dual blur branch greatly degrades the performance.

Note that we have also tried three different kinds of blur offset, i.e., non-local attention, element-wise weight and residual, and the results are described in the lower part of Table 4. We see that both the element-wise weight and residual blur offset are highly competitive, and both of them are more effective than the non-local attention.

6 CONCLUSION

We have discussed the color retention and quality assessment problems for the unsupervised blind motion deblurring. Technically, a general unsupervised color retention network CRNet and a new chromatic aberration quantization metric CSE are proposed. The deblurring architecture of CRNet solves the chromatic aberration problem by proposing a new strategy consisted of a blur offset estimation network and an adaptive blur correction stage. The advantages of the new metric CSE are threefold. First, it can be easily used for the supervised/unsupervised settings with/without paired data; Second, the quantization result is more consistent with the human perception than PSNR and SSIM metrics; Third, the performance difference evaluated by the CSE metric is more significant than PSNR and SSIM metrics. Extensive experiments on both natural image and domain-specific datasets show that our CRNet has obtained better performance for deblurring and color retention. In future, we will try to figure out the underlying cause of blurring at the pixel level, and also the difference of distributions between sharp and blurry images, so that the task can be performed better in unsupervised manner.

ACKNOWLEDGMENTS

This work is partially supported by National Natural Science Foundation of China (62072151, 62020106007), Anhui Provincial Natural Science Fund for Distinguished Young Scholars (2008085J30), and the Fundamental Research Funds for the Central Universities of China (JZ2019HGPA0102). Zhao Zhang is the corresponding author of this paper.

REFERENCES

- [1] N. Wiener *et al.*, *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications*. MIT press Cambridge, MA, 1964, vol. 8.
- [2] W. H. Richardson, "Bayesian-based iterative method of image restoration," *JoSA*, vol. 62, no. 1, pp. 55–59, 1972.
- [3] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," *Advances in neural information processing systems*, vol. 22, pp. 1033–1041, 2009.
- [4] L. Sun, S. Cho, J. Wang, and J. Hays, "Good image priors for non-blind deconvolution," in *European conference on computer vision*. Springer, 2014, pp. 231–246.

- [5] R. Das, A. Bajpai, and S. M. Venkatesan, "Fast non-blind image deblurring with sparse priors," in *Proceedings of International Conference on Computer Vision and Image Processing*. Springer, 2017, pp. 629–641.
- [6] S. A. Bigdeli, M. Jin, P. Favaro, and M. Zwicker, "Deep mean-shift priors for image restoration," *arXiv preprint arXiv:1709.03749*, 2017.
- [7] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3929–3938.
- [8] D. Ren, W. Zuo, D. Zhang, J. Xu, and L. Zhang, "Partial deconvolution with inaccurate blur kernel," *IEEE transactions on image processing*, vol. 27, no. 1, pp. 511–524, 2017.
- [9] L. Chen, F. Fang, J. Zhang, J. Liu, and G. Zhang, "Oid: Outlier identifying and discarding in blind image deblurring," in *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*. Springer, 2020, pp. 598–613.
- [10] M. Delbracio, I. Garcia-Dorado, S. Choi, D. Kelly, and P. Milanfar, "Polyblur: Removing mild blur by polynomial reblurring," *IEEE Transactions on Computational Imaging*, vol. 7, pp. 837–848, 2021.
- [11] Z. Xu, H. Chen, and Z. Li, "Fast blind deconvolution using a deeper sparse patch-wise maximum gradient prior," *Signal Processing: Image Communication*, vol. 90, p. 116050, 2021.
- [12] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3883–3891.
- [13] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8183–8192.
- [14] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 8878–8887.
- [15] M. Ye, D. Lyu, and G. Chen, "Scale-iterative upscaling network for image deblurring," *IEEE Access*, vol. 8, pp. 18316–18325, 2020.
- [16] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14821–14831.
- [17] D. Rozumnyi, M. R. Oswald, V. Ferrari, J. Matas, and M. Pollefeys, "Defmo: Deblurring and shape recovery of fast moving objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3456–3465.
- [18] T. M. Nimisha, K. Sunil, and A. Rajagopalan, "Unsupervised class-specific deblurring," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 353–369.
- [19] B. Lu, J.-C. Chen, and R. Chellappa, "Unsupervised domain-specific deblurring via disentangled representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10225–10234.
- [20] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *Acm transactions on graphics (tog)*, vol. 27, no. 3, pp. 1–10, 2008.
- [21] L. Sun, S. Cho, J. Wang, and J. Hays, "Edge-based blur kernel estimation using patch priors," in *IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2013, pp. 1–8.
- [22] J. Pan, D. Sun, H. Pfister, and M.-H. Yang, "Blind image deblurring using dark channel prior," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1628–1636.
- [23] L. Chen, F. Fang, T. Wang, and G. Zhang, "Blind image deblurring with local maximum gradient prior," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1742–1750.
- [24] Y. Bai, H. Jia, M. Jiang, X. Liu, X. Xie, and W. Gao, "Single-image blind deblurring using multi-scale latent structure prior," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 2033–2045, 2019.
- [25] F.-J. Tsai, Y.-T. Peng, Y.-Y. Lin, C.-C. Tsai, and C.-W. Lin, "Banet: Blur-aware attention networks for dynamic scene deblurring," *arXiv preprint arXiv:2101.07518*, 2021.
- [26] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.
- [27] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville, "Improved training of wasserstein gans," *arXiv preprint arXiv:1704.00028*, 2017.
- [28] Z. Yi, H. Zhang, P. Tan, and M. Gong, "Dualgan: Unsupervised dual learning for image-to-image translation," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2849–2857.
- [29] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 1857–1865.
- [30] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.
- [31] M. K. Johnson and H. Farid, "Exposing digital forgeries through chromatic aberration," in *Proceedings of the 8th workshop on Multimedia and security*, 2006, pp. 48–55.
- [32] R. Bedford and G. Wyszecki, "Axial chromatic aberration of the human eye," *JOSA*, vol. 47, no. 6, pp. 564_1–565, 1957.
- [33] L. Thibos, A. Bradley, D. Still, X. Zhang, and P. Howarth, "Theory and measurement of ocular chromatic aberration," *Vision research*, vol. 30, no. 1, pp. 33–49, 1990.
- [34] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *International conference on machine learning*. PMLR, 2019, pp. 7354–7363.
- [35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [37] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [38] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.
- [39] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.
- [40] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.