IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 June 2022, accepted 18 July 2022, date of publication 26 July 2022, date of current version 15 August 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3194032*

RESEARCH ARTICLE

# RAID-Net: Region-Aware Image Deblurring Network Under Guidance of the Image Blur Formulation

**LIANJUN LIAO[1,2,3], (Member, IEEE), ZIHAO ZHANG[2,3], (Member, IEEE), AND SHIHONG XIA[2,3], (Member, IEEE)**

[1]Department of Computer Science and Technology, School of Information, North China University of Technology, Beijing 100144, China
[2]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China
[3]Department of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China

Corresponding author: Shihong Xia (xsh@ict.ac.cn)

This work was supported in part by the China National Key Research and Development Program of Science and Technology for Winter Olympics under Grant 2020YFF0304701.

**ABSTRACT** Image deblurring is a challenging field in computational photography and computer vision. In the deep learning era, deblurring methods boosted with neural networks achieve significant results. However, the existing methods mainly focus on solving specific image deblurring problem, and overlook the origin of the motion blur. In this paper, we revisit how blur occurs, and divide them into three categories, i.e. caused by relative motion between camera and scene, caused by the movement of the object itself and the edges of a blurring image, which may meet discontinuity because of the pixels trajectory sampled outside the image. To address the issues of different blurs in an image, we propose a two-stage neural network for image deblurring named RAID-Net. In order to remove the global blurry region caused by camera movements, we first use a U-shape network to get the coarse deblurred image. Then we leverage an adaptive reasoning module to model the relationship between different blurry regions within one image jointly and remove the other two categories of motion blur. Experiments on two public benchmark datasets demonstrate that our method achieves comparable or better results over the state-of-the-art methods.

**INDEX TERMS** Attention mechanism, graph neural network, graph reasoning network, image deblur, image processing.

## I. INTRODUCTION

Image deblurring aims to recover an image with sharp edges and fine details from a blurry image, and is one of the fundamental topics in computer vision. Image deblurring has many applications. For example, an image captured with hand-held devices such as mobile phones or video cameras often contains severe blurs, and it is highly desired to remove undesired blurs.

The performance of image deblurring has been significantly improved in recent years by a great number of image deblurring methods [1]–[5]. However, these methods were developed mainly for one or two specific types of blur and therefore paid less attention to the relationship between the causes of the different image blur types and their corresponding solutions. Moreover, the blur in the image may have multiple causes. The approach and the order for removing

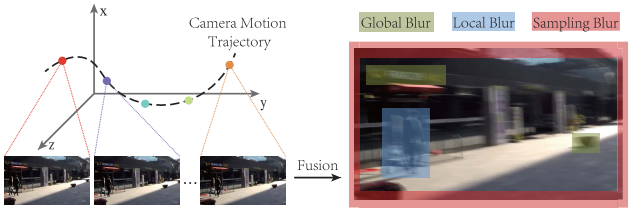



**FIGURE 1.** Illustration of how blur occurs. On the left of the figure, we show the motion trajectory in 3D space and the corresponding sharp image. On the right of the figure, we show the blurry image and the image regions with three different categories of blur.

The associate editor coordinating the review of this manuscript and approving it for publication was Charith Abhayaratne.

different types of blur may also influence the final deblurring results.

To address the above issues, we first formulated three different causes of the image blur and propose a two-stage deblurring network to specifically remove each type of image blur. As shown in Fig. 1, we show three different types of image blur. The first type of image blur is caused by camera movement, which leads to the overall blur in the image due to the pixel change during the shutter closes. The second type of image blur is due to the object movement, which causes the partial image blur because of the change of a specific image region during the shutter closes. The third type of image blur is caused by discontinuity sampling pixels among the edges of a blurring image. In this case, the pixel on the edge of the blurry image will lose the information of the trajectory outside the image. Based on our formulation of image blur, we propose a two-stage coarse-to-fine single image deblurring framework named RAID-Net, which can adaptively remove the dominant blur and intricate blur in the same image. In the first stage, a simple deblurring network is used aiming to remove the dominant blur such as the camera-movement-caused blur. In the second stage, the coarse deblurred image is deblurred in a divide-and-conquer strategy. Since convolution operation in previous stage lacks the ability to model the intricate and irregular blurry region located in different parts of the image, we jointly model these regions using the network with the ability to model irregular data structure, i.e. the graph convolutional network. The main contribution of this work can be summarized as follows:

1. We formulate three different causes of the image blur and propose the corresponding deblurring methods for each type of image blur;
2. Unlike traditional convolutional neural network, we use the graph convolutional neural network GCN) to model the irregular data structure provided by the attention mechanism and depict the relationship between image regions which may have a different position but the same blurring pattern;
3. Experiments on benchmark datasets are carried out to prove that our method can achieve state-of-the-art accuracy and real-time deblurring efficiency with only a lightweight model.

## II. RELATED WORK

In this section, we review related works including image deblurring methods, attention mechanism, and graph convolutional neural network.

### A. IMAGE DEBLURRING

Recently, the deep neural network brings excellent improvements to the image deblurring task. Prior arts that use a convolutional neural network to solve the deblurring task include [1], [4], [6]–[15]. Tao *et al.* [1] uses shared weight network through different scales. Similarly, the work [16] also use multiscale methods to solve the blind image deblurring problem. Zhang *et al.* [8] propose a deep hierarchical multi-patch network to solve the deblurring task through a fine-to-coarse hierarchical representation. Gao *et al.* [4] propose the parameter selective sharing scheme to solve the dynamic scene deblurring problem. Pan *et al.* [11] obtain a high-quality blur kernel directly from the frequency domain. Zhou *et al.* [12] try to use temporal information and propose a filter adaptive convolutional layer to align the deblurred features in the temporal domain. Similarly, Pan *et al.* [14] also use the temporal information to solve the video deblurring problem. Liu *et al.* [15] propose a differentiable re-blur model so that they can train the model in a self-supervised manner.

However, CNN mainly focus on extracting the local feature from the image which may lack the generalization ability for the dynamic scene deblurring problem. Recently, there are also works that focus on solving the limitation of the convolutional network. In the work [17], the authors propose a two-stage coarse-to-fine deblurring method, which is similar to ours. However, they use a uniform way to treat different blur categories. Suin *et al.* [13] propose an adaptive motion deblurring method by considering both the global spatial dependencies and neighboring pixel information. Purohit [18] propose an efficient motion deblurring by estimating the spatial attention-maps to model the local features and their global inter-dependencies. In the work [19], the authors propose a multi-stage architecture. In each stage, the local features are extracted and reweighted through a per-pixel adaptive module. Chi and his colleagues [20] propose a self-supervised meta-auxiliary learning method to enhance the performance of the deblurring task. In the work [21], the authors propose a method that can handle unseen blurry image by learning a blur kernel space. In the work [22], the authors use a hierarchical network architecture search strategy to find the optimal network for deblurring. In the work [23], the authors use the disparity probability volume module to leverage the disparity information for deblurring task.

Other than image deblurring problem, there are also many works related to us. In the work [24], the authors propose a residual-guided multiscale fusion network for the Bit-Depth Enhancement problem and can get excellent results. The multiscale strategy is similar to that of the work [1]. The researchers then propose a target attention network by using the nonlocal block to capture the global features. In the work [25], the authors try to solve the ''temporal interpolation problem'' for fast moving object(FMO) based on a image with FMO and its corresponding clear background. In the work [26], the authors mainly focus on the image deblurring problem for the event camera.

As discussed above, the causes of image blur vary from image to image. Directly using CNN or even the combination of attention mechanism and CNN may be difficult to process such image blur. Therefore, different to the previous methods, we first investigate the causes of the image blur and then propose the image deblurring framework under the guidance of our image-blur formulation. Moreover, we refine

the deblur results by leveraging the attention mechanism and graph convolutional neural network to model the relationship of different blur regions on an image.

### B. ATTENTION MECHANISM

The attention mechanism in deep learning is trying to imitate such human behavior. The differences between various attention mechanisms mainly lie in the domain they focus on. Some methods mainly focus on the spatial domain [27], [28]. The key idea of these methods is to transform the spatial information in the original picture into another space and retain the key information. Other methods focus on the channel domain [29]–[31]. These methods mainly focus on building the relationship on the channel domain. Recently, with the development of transformer network, increasing works start us transformer-based network for image restoration tasks.

In our work, we mainly focus on the attention mechanisms on the spatial domain. Our key idea is to use the attention mechanisms to find the image regions that have residual blur after the coarse deblurring network.

### C. GRAPH CONVOLUTIONAL NETWORK

There are a lot of data in our real life that do not have a regular spatial structure which we call it as Non-Euclidean data. Graph Convolutional network (GCN) [32], [33] is a neural network especially for this kind of data that is represented as graphs. Recently, GNN becomes a widely-used method in computer vision because of its superior ability to model the relationship between the irregular data. Yan *et al.* [34] propose a spatial-temporal GCN for action recognition. Their method can learn both the temporal and the spatial patterns from the data by using the graph structure. Similarly, Ferrari [35] also model the spatial and temporal relationship between humans and objects to learn the dynamic relationship from videos. Moreover, Velickovic [36] integrate the attention mechanism with the graph structures.

In our work, we mainly use the GCN to model the intrinsic relationship between different blurry region. These region may be separated from each other but has the same blur pattern. Compared with the other methods using attention mechanism and Graph Convolutional network, our work is the first one to combine them and introduce them in the image deblurring problem. The attention mechanism we use can help us adaptively find the blurry image region and the Graph Convolutional network can help us jointly model the intricate blur in different blurry regions better.

## III. BLUR FORMULATION

The mainstream method for image deblurring is to increase the receptive field of the convolutional neural network and the depth of the deblurring network. Recently, another popular method for image deblurring is to model the spatial relationship in the image [13], [18]. However, these methods still ignore the programmatic deblurring strategy. We claim that removing the image blur based on its causes will improve the motion deblurring performance. The experiments of our work also support that the cause-specific image deblurring solution is a considerable better strategy. Now, we will introduce the formulation of the proposed three types of image blur.

**TABLE 1. The summary of the formulated image blur type.**

| Blur Type | Unknown Parameters | Equations |
|---|---|---|
| Global Blur | m | n |
| Local Blur | $>$ m | n |
| Sampling Blur | m | $<$ n |

The original blurry images contain many redundant movement information of the background which may reduce the overall deblurring results. For example, the pixel of an object may contain the movement information of the camera and the object itself. To resolve the blurs of these pixels, we use the coarse deblurring network to remove one kind of blur in this and then the rest in the next stage. We call the blur caused by camera movement as *global blur* since the camera movement influence all pixels on the image. The formulation of the global blur is relatively simple. Given a period $T$ when the shutter close, the image containing camera movement $b(i, j)$ can be considered as the average image, which can be formulated as follows:

$$b(i,j) = \frac{1}{T}\sum_{t=1}^{T} s(i + tr_x(t), j + tr_y(t)) \tag{1}$$

where $s(i, j)$ is the sharp image at pixel $(i, j)$ on the frame $t$. $tr_x(t)$ and $tr_y(t)$ represent the $x$-component and $y$-component of the camera movement on the frame $t$.

The case introduced in Equ. 1 is a rather strong constraint that the blur is generated by only the camera motion. However, the image blur in our daily life has many sources. One of the most common blur causes is object movement, which we call *local blur*. Considering the object movement, the Equ. 1 can then be updated as:

$$b(i,j) = \frac{1}{T}\sum_{t=1}^{T} s(i + tr_x(t) + m_x(i + tr_x(t)), \\ j + tr_y(t) + m_y(j + tr_y(t)) \tag{2}$$

where $m_x(t)$ and the $m_y(t)$ represent the $x$-component and $y$-component of the blur caused object movement at frame $t$. Here, we assume that the global blur is overlaid by the local blur. The difficulties of removing this kind of blur are mainly two-fold. Firstly, since the object movement has great diversity in its motion patterns such as type, velocity, and object shape, the blur of each individual region in an image varies from each other. Moreover, the blur of the different images also varies from each other.

The third type of the blur cause is the one that is ignored in most of the image deblurring network which we call sampling blur. Different from the previous two types of blur causes, the sampling blur is not caused by the movement of any object. As shown in Eq. 1 and Eq. 2, the blurry pixel is sampled from a set of corresponding sharp image. The pixel near the edge

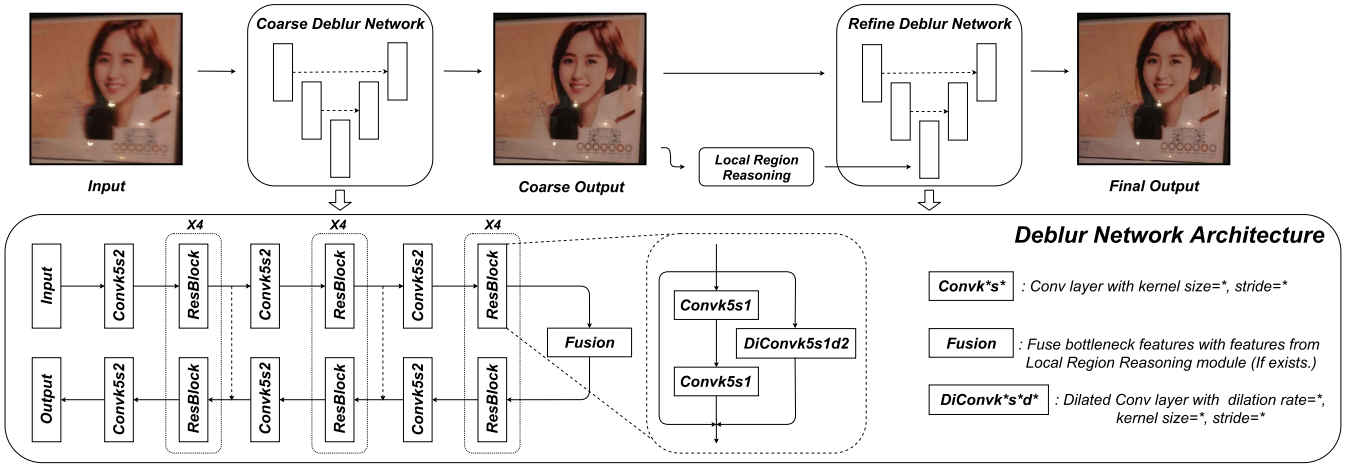



**FIGURE 2.** The overall pipeline of RAID-Net, The network consists of two stages, the coarse deblur network and the refine deblur network.

of the image may not be able to sample from the whole set of the sharp image. Therefore, this type of blur is mainly caused by the break of the trajectory on the image boundary which can be formulated as follows:

$$b(i,j) = \frac{1}{T}\sum_{i=1}^{T} s(i + tr'_x(t), j + tr'_y(t)) \tag{3}$$

where $tr'_x(\cdot)$ and $tr'_y(\cdot)$ represents the $x$-component and $y$-component of the hybrid blur. What should be noted is the value range that $min(i + tr(i)) \leq 0 \leq i + tr(i) \leq c \leq max(i + tr(i))$ and $c$ is the boundary of the image. The value range indicates that some of the sampling pixel may exceed the boundary of the image and thus be truncated to $[0, c]$. The difficulty of removing this kind of blur is that the problem itself is an ill-posed problem. The missing information on the edge of the image is not available.

As shown in Table 1, we also further discuss the mathematical solutions of each blur type. Given the blurry image $b \in B$, our deblurring network aims to learn the mapping $\mathcal{F} : b \mapsto s(t), s \in S, t \in [1, T]$ between the blurry image space $B$ and the sharp image space $S$. Treating the above equations as system of linear equations, we then analysis the rank of different blur types. The information we want to solve can be regarded as the unknown parameters and the information we can get can be regarded as the number of equations. We assume that the unknown parameters for global blur is $m$ and the number of equations is $n$. The unknown parameters for local blur should be larger than $m$ since we have to solve the object movement mask $m_x(t)$ and $m_y(t)$. The number of equations of sampling blur should be less than $n$ since the sampling trajectory is truncated to $[0, c]$.

The above observation inspires us that using convolution alone may be a favorable solution for global blur while not that satisfactory for the rest two types of image blur causes. To solve this problem, extra information should be used so that the balance between the unknown parameters and the number of equations could be rebuilt. Therefore, we need to treat the global blur and the rest two types of blur differently and model the image blur region from a different region of the image to get more hidden information.

## IV. METHOD

Fig. 2 shows the pipeline of our framework. Given a blurry image containing different types of blur, our goal is to recover the sharp image considering the causes of the blur using a two-stage network. Our key idea is that different types of blur can be mathematically formulated and we can remove the blur based on how this blur could emerge. This section will be organized as follows. First, we will introduce the architecture of the corresponding stage 1 for remove. Then, we will introduce the corresponding solution for the rest two types of blur. Finally, we will give the implementation details of our method.

### A. COARSE DEBLURRING NETWORK

The coarse deblurring network aims to remove the global blurry region caused by camera movement in the image. The task is relatively simple, and has been explored by many previous literatures. Therefore, we use a U-shape network similar to the work [1] to fulfill it. The input of the U-shape network is the blurry image and the output of this module is the coarse deblurred image.

As stated in the InceptionNet [37], we try to extract both the sparse and dense features from the same layer, so that we can improve the accuracy while avoiding tons of parameters. The details of the network architecture can be found in Fig. 2. Different with the ResBlock used in previous deblurring methods, we make two main modifications to help us explore the global blur feature better. Firstly, instead of using two consecutive convolution layers with a residual connection, we divide the block into three parallel parts, which is shown in the dotted box in Fig. 2. In the first part, two convolution layers are used to extract the dense feature. In the second part, one dilation convolution layer is used to extract more sparse

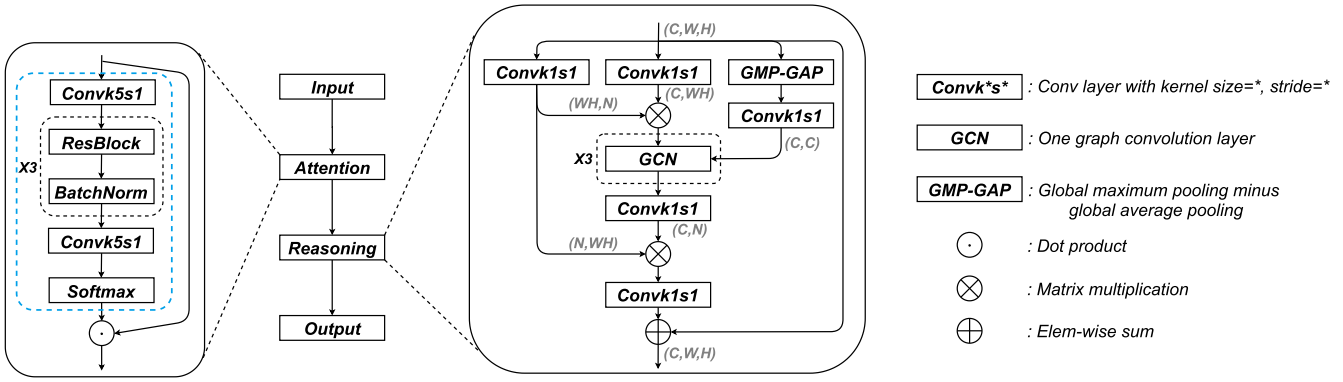



**FIGURE 3.** The network architecture of our local region reasoning module.The network architecture of multi-head attention module is shown in the box of "Attention", and the graph reasoning network is shown in the box of "Reasoning".

feature. In the third part, the residual connection is used to avoid gradient vanishing. After this, we aggregate these three features with the sum function to get the output feature of the ResBlock. Secondly, we can sacrifice some accuracy for higher efficiency. Thus, we supply more flexible choices on the channel size for different uses to have a trade-off between accuracy and efficiency.

### B. REFINE DEBLURRING NETWORK

As illustrated in Fig. 2, the refine deblurring network shares the same backbone network as the coarse deblurring network. What is different is that we propose a *local region reasoning (LRR)* module to model the intricate blurry region in the coarse output of the previous stage. Since the CNN are spatially invariant, using CNN alone is not suitable for the deblurring task that the deblurring degree varies from region to region. To address this issue, we jointly model all the blurry regions and learn the hidden blur feature using the graph convolutional network. The blurry regions are selected adaptively using a multi-head attention mechanism. Shown in the fuse block of Fig.2, the hidden blur feature is then added with the feature from the bottleneck of the U-shape network. This step can ensure that the blurry information from the blurry regions is aggregate by the network and can help to improve the performance of the deblurring task. The details of the LRR module are shown in Fig. 3. We will introduce how the attention mechanism and the graph reasoning works in the LRR module for the rest two types of blur in the following part.

#### 1) MULTI-HEAD ATTENTION MECHANISM

The multi-head attention mechanism (MHAM) aims to adaptively find out the image regions caused by the rest two types of blur from the coarse sharp image so that we can learn the hidden blur feature from these regions. Shown as Attention module in Fig 3, this method can ensure that the blurry region after removing the camera movement blur can be found regardless of their position on the image. These blurry regions represent the image blur caused by the local blur and sampling blur since we assume that the global blur $tr(t)$ has been removed in the previous stage. The advantages of our attention mechanism are two-fold. First, it can in some way solve the limitation of convolutional spatial invariance by ignoring the neighborhood relationship. Moreover, this method also follows the cause of the blur stated in Sec III that only some regions on the image have these kinds of blur. The above features make our attention module suitable for removing the rest two kinds of blur since we start to focus on the non-contiguous regions on the image.

As shown in Fig 3, the first component of each head is a 2D convolution, then the features are fed to a ResBlock with batch normalization. The output feature is then fed to another 2D convolution block with a softmax to get a weighted mask $\mathcal{M}_i$, where $i$ is the index of attention head among all the $C$ heads. The element-wise multiplication is conducted between the residual map and the predicted mask $\mathcal{M}_i$. Denote the input feature of MHAM as $f_{in}$, the output weighted feature as $f^i_{MHAM}$ and the operation shown in the blue dotted box in the leaf part of Fig 3 as $K(\cdot)$. Then the MHAM can be formulated as follows:

$$f^i_{MHAM} = K_i(f_{in}) \odot \mathcal{M}_i \tag{4}$$

Since the multi-head attention module does not share weights, the parameters in $K_i$ is different from each other, and the predicted mask $\mathcal{M}_i$ is also different and it can be regarded as the numerical solution of the $m_k(t)$ in Eq. 2. The multi-head attention mechanism can enable the network to adaptively select different key image regions to deblur, and avoid being stuck in fixed patterns or regions.

#### 2) GRAPH REASONING NETWORK

Given the intricate blurry regions in the coarse sharp image, the graph reasoning network aims to learn the hidden blur feature and model the correlation from these regions, and then remove the rest two kinds of blur. In our problem, we treat the given blurry region from MHAM module as the nodes of the graph. We assume that these nodes are all connected with each other so that we can learn the relationship between these blurry regions better. Basically, we predict the adjacency

matrix $\mathbf{W}$ and calculate the feature of the graph node, and then use GCN to calculate the output feature vector $y \in \boldsymbol{R}^{C\times N}$. The output feature then passes through the up-sampling block to generate the final deblurred image. This allows us to aggregate all the blur information from all blurry regions and fulfill the down-streaming tasks. Then we will give a detailed introduction on the adjacency matrix prediction and the graph feature generation.

#### a: ADJACENCY MATRIX PREDICTION

As shown in Fig. 3, given the output regions $\{f^i_{MHAM}\}, i = 1, \ldots, C$ of the MHAM block, we first feed them into the global average pooling (GAP) and global maximum pooling (GMP) layer. Since the average function and maximum function are all symmetric functions, the features after GAP and GMP can be symmetric and invariant to the order of the blurry regions. Only when the features being invariant to the order of the blurry regions, can we get the most representative blurry features. The symmetric feature is then subtracted with each other and fed to the 1D convolution to get the adjacency matrix $\mathbf{W} \in \boldsymbol{R}^{N\times N}$, where $N$ is the pre-defined feature dimension.

#### b: GRAPH GENERATION

The output feature from the MHAM $\{f^i_{MHAM} \in \boldsymbol{R}^{C\times W\times H}\}, i = 1, \ldots, C$ of the MHAM block, as shown in the middle box part of Fig. 3, is fed to two 2D convolution layers. The output from the first path is $f_{p1-out} \in \boldsymbol{R}^{C\times WH}$, the output from the second path is $f_{p2-out} \in \boldsymbol{R}^{N\times WH}$. The node feature matrix $\mathcal{V}$ is given by:

$$\mathcal{V} = f_{p1-out} \times f^T_{p2-out} \tag{5}$$

With the node feature matrix $\mathcal{V}$ and the adjacency matrix $\mathbf{W}$, we can easily feed these variables into the GCN. The output feature $f_{GCN} \in \boldsymbol{R}^{C\times N}$ is then multiplied with the transposed feature $f_{p2-out}$ so that the output dimension becomes $C \times WH$. We then add it with the output feature from the MHAM and use a 1D convolution to get the hidden blur feature.

### C. IMPLEMENTATION DETAILS

In this part, we mainly introduce the training strategy of our method. Our training strategy can be divided into two steps. We first pre-train the network using the classical mean square error loss $\mathcal{L}_{mse}$ to warm up the the network. The mean square error loss is given by:

$$\mathcal{L}_{mse}(I, K) = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \tag{6}$$

where $I$ and $K$ are the original image and the predicted image, $(m, n)$ is the size of the image.

In second step of our training strategy, we focus on recovering those complicated details. These details are lost mainly because of the local blur and the sampling blur. The mean square error constraint works well on the overall deblurring tasks. For the details deblurring task, we additionally use the structure similarity loss $\mathcal{L}_{ssim}$ as an auxiliary loss to capture the local details better. The structure similarity loss between the image $I$ and $K$ is formulated as follows

$$\mathcal{L}_{ssim}(I, K) = \frac{2(\mu_I\mu_K + c_1)(2\sigma_{IK} + c_2)}{(\mu_I^2 + \mu_K^2 + c_1)(\sigma_I^2 + \sigma_K^2 + c_2)} \tag{7}$$

where $\mu$ is the mean pixel value, $\sigma$ is the standard deviation of the pixels, $\sigma_{IK}$ is the covariance matrix, $C$ is a constant value to avoid division by zero.

**TABLE 2. The results of our ablation study on GoPro dataset, different method means different comparing items.**

| Method | PSNR/SSIM |
|---|---|
| RAID-Net(-) | 28.47/0.928 |
| RAID-Net | 30.90/0.956 |
| RAID-Net(+) | 31.59/0.960 |
| Baseline | 30.90/0.956 |
| Single Stage 1 | 26.69/0.898 |
| Double Stage 1 | 30.08/0.947 |

Our experiments are conducted on a PC with Intel (R) Xeon (R) Silver 4110 CPU@2.10GHz and an NVIDIA GTX 3090 GPU. The network framework is implemented on PyTorch. Unless noted specifically, all the experiments mentioned in the following part are conducted on the above equipment. All the experiments are trained on the same dataset for a fair comparison. The hyper-parameters of our method are as follows. The batch size we use during training is 16. We use the "CosineAnnealingWarmRestarts" as the learning rate scheduler which $T_0$ and $T_{mul}$ is set to 1 and 2 respectively. We use the Adam optimizer and we train the model on the training set for 2000 epochs.

## V. EXPERIMENTS

### A. DATASETS AND METRICS

In our experiment, we use the GoPro dataset [38], RealBlur dataset [39] and HIDE dataset [40] to evaluate our method. The GoPro dataset is the most classical dataset for image deblurring problem. It has 3,214 image pairs for blurry and sharp images. Following the training strategy of GoPro [38], we use 2,103 pairs as training samples and the other 1,111 as testing samples. As for the RealBlur dataset, it has 3,758 image pairs from 182 different scenes for training and 980 image pairs of 50 different scenes in the testing set. The HIDE dataset is the deblurring dataset for human motion-related deblurring tasks. It has 6,397 training samples and 2,025 testing samples with both wide-range and close-range scenes.

To evaluate the performance of our image deblurring method, we adopt both the qualitative metrics and quantitative metrics. For the quantitative evaluation, we use the Peak Signal to Noise Ratio (PSNR) and structure similarity (SSIM). For the qualitative evaluation, we present the qualitative results in Fig. 4 and Fig. 5.

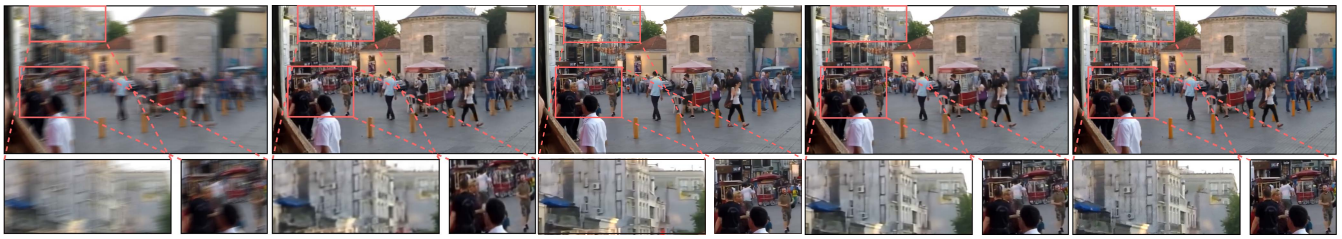


**FIGURE 4.** Quantitative results on GoPro dataset. From left to right, we show the images of 1) input blurry image; 2) Result of [5]; 3) Result of [19]; 4) Our Result; 5) Ground truth.

### B. ABLATION STUDY

In this part, we will evaluate the effect of the key terms in our methods. The dataset we use for the ablation study is the GoPro dataset. When changing the value of one term, we keep the remaining terms fixed for a fair comparison. The results are shown in Table 2.

#### 1) EFFECT OF BASIC CHANNEL SIZE

In this set of experiments, we mainly evaluate two variants of RAID-Net, which is RAIN-Net(−) and RAIN-Net(+). RAID-Net is the original network described in the previous section. RAID-Net(−) and RAID-Net(+) have the same architecture as RAID-Net. Differently, we set the channel size in the ResBlock introduced in Sec IV-A as 8 and 32, respectively.

In the upper part of Table 2, we show the performance of different variants of our method. We can observe that the smaller the channel size is, the lower performance will be get. However, the performance is relatively close to each other. Conceptually, the network capacity is smaller as the channel size decreases.

#### 2) EFFECT OF KEY MODULES

In this set of experiments, we mainly investigate the effect of key modules in our model. We use RAID-Net as the baseline model for the comparison, and the results are shown in Table 2.

##### *a: EFFECT OF TWO-STAGE ARCHITECTURE*

To investigate whether the two-stage method works for improving the deblurring results, we compare the performance of our method with the model using only the first stage (denoted as Single Stage 1). As shown in Table 2, we observe that the models using two stages achieve better results than the model using the first stage. Therefore, the two-stage network architecture can improve the deblurring performance.

##### *b: EFFECT OF LOCAL REGION REASONING MODULE*

To investigate the effect of local region reasoning (LRR) module, we compare the performance with the baseline model and the model without LRR module (denoted as Double Stage 1). As shown in the fifth row and seventh row of Table 2, the model using the LRR module performs better than the model without the LRR module. Therefore, the LRR module in the refine deblur network can help improving the deblurring performance. Conceptually, the LRR module helps find the hidden deblurring patterns hidden in the severe blurry regions of the image.

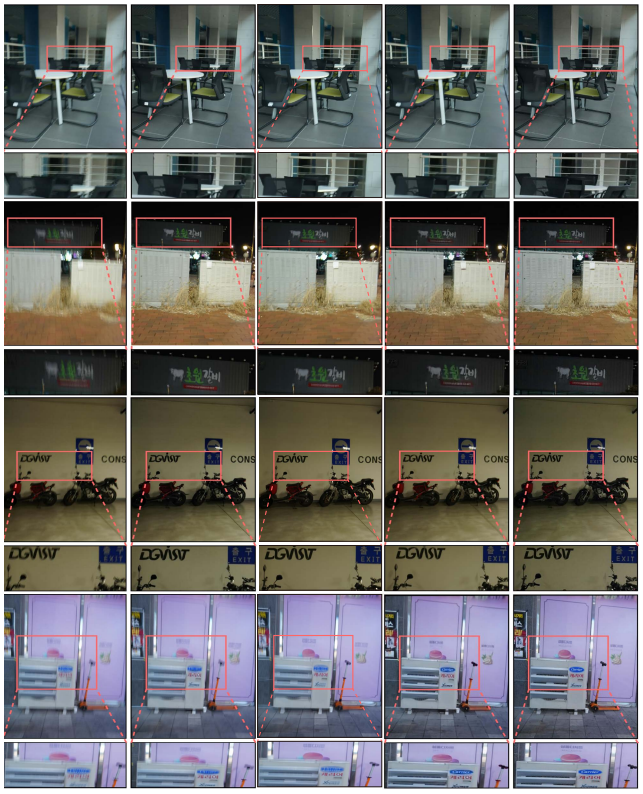


**FIGURE 5.** Quantitative results on RealBlur dataset. From left to right, we show the images of 1) input blurry image; 2) Result of [1]; 3) Result of [19]; 4) Our Result; 5) Ground truth.

### C. COMPARISONS WITH THE STATE-OF-THE-ART METHODS

We compare the performance of our method on GoPro dataset, RealBlur dataset and HIDE dataset with other state-of-the-art methods. The results of each method on different dataset are shown in Table 3 and Table 4. Fig. 4 and Fig. 5 also show the qualitative results of our methods.

**TABLE 3.** Results on GoPro and HIDE datasets.

| Method | GoPro PSNR/SSIM | HIDE PSNR/SSIM |
|---|---|---|
| DeblurGAN [2] | 28.70/0.858 | 24.51/0.871 |
| SRNet [1] | 30.26/0.934 | 28.52/0.925 |
| DeblurGANv2 [3] | 29.55/0.934 | 26.62/0.876 |
| DBGAN [5] | 31.10/0.942 | 28.92/0.913 |
| MPRNet [19] | **32.66**/0.959 | **30.96**/0.939 |
| Test-time [20] | 32.50/0.958 | 30.55/0.935 |
| RAID-Net(+) (Ours) | 31.59/**0.960** | 29.70/**0.952** |

**TABLE 4.** Results on RealBlur datasets. "*" indicates using additional data.

| Method | RealBlur-J PSNR/SSIM | RealBlur-R PSNR/SSIM |
|---|---|---|
| SRNet [1] | 31.02/0.8987 | 36.47/0.9515 |
| SRNet* [1] | 31.38/0.9091 | 38.65/0.9652 |
| DeblurGANv2* [3] | 29.69/0.8703 | 36.44/0.9347 |
| MPRNet [19] | **31.76/0.922** | **39.31/0.972** |
| RAID-Net (Ours) | 31.09/0.9120 | 38.87/0.9685 |

On GoPro dataset, we compare with the state-of-the-art methods, including SRNet [1], DeblurGAN [2], DeblurGANv2 [3], DBGAN [5],MPRNet [19] and a self-supervised meta-auxiliary learning method [20]. As shown in Table 3, the PSNR value of our method on GoPro is 31.59dB, though 0.91dB lower than the MPRNet, still achieves comparable performance with the state-of-the-art methods. Moreover, the SSIM value of our method outperforms the other methods. We also provide a set of qualitative results tested on GoPro dataset. From Fig. 4, we can see that our method can recover both the moving foreground object such as the pedestrian and the background object such as the exterior walls.

On the HIDE dataset, as is shown in Table 3, our method also achieves high performance. The PSNR value of our method on HIDE is 29.70dB and the SSIM value is 0.952, which shows the similar performance with the state-of-the-art works. For a fair comparison, we train our model on the GoPro dataset and test it on the HIDE dataset.

On the RealBlur dataset, the performance of our method is also significant. From Table 4, we observe that the result is 0.67 dB lower in the PSNR on the RealBlur-J dataset compared with MPRNet and SRNet that uses additional GoPro data for training. However, our method can outperform other methods on the rest metrics, especially on SSIM. Our model mainly tends to generate a visually sharp image and the high PSNR values may not indicate that the deblurring results is satisfactory. From Fig. 5, we can see that our method can recover the image details better. For example, the line of the brickwork is more clear and the text, especially on the fourth image, is much more clear than the results of the other methods.

We also compare the efficiency of our method and other state-of-the-art methods. As shown in Table 5, the RAID-Net(-) only use 0.04 second to process an image, which is almost 20 times faster than the other methods. For a fair comparison, all the comparison experiments are conducted on the same computer.

**TABLE 5.** Runtime analysis.

| Method | Processing time (s) |
|---|---|
| SRNet [1] | 1.2 |
| Nested [4] | 1.0 |
| DeblurGANv2 [3] | 0.48 |
| Pyramid-architecture [22] | 0.026 |
| Ours (RAID-Net(-)) | **0.04** |
| Ours (RAID-Net) | 0.25 |
| Ours (RAID-Net(+)) | 0.63 |

## VI. CONCLUSION

In this paper, we formulate three different types of image blur and propose a two-stage pipeline to correspondingly remove the blur. Our formulation of the blur categories provide a self-contained theoretical analysis of the blur causes and could contribute to the future works. For example, the third type of blur caused by discontinuity in sampling, could be reduced by introducing the temporal information which can be easily solved in the video deblurring problem.

The core idea of our method is to use the attention mechanism and GCN to model the latent relationship of different blurry regions regardless of their spatial distances. Extensive experiments demonstrate that our method is a simple and effective image deblurring framework.

## REFERENCES


[1] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 8174–8182.

[2] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8183–8192.

[3] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang, "DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8878–8887.

[4] H. Gao, X. Tao, X. Shen, and J. Jia, "Dynamic scene deblurring with parameter selective sharing and nested skip connections," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 3843–3851.

[5] K. Zhang, W. Luo, Y. Zhong, L. Ma, B. Stenger, W. Liu, and H. Li, "Deblurring by realistic blurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2737–2746.

[6] D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. Van Den Hengel, and Q. Shi, "From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 3806–3815. [Online]. Available: https://ieeexplore.ieee.org/document/8099888/

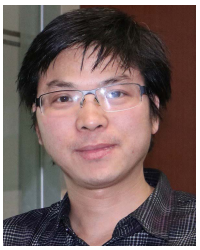
[7] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 237–246. [Online]. Available: http://ieeexplore.ieee.org/document/8099516/

[8] H. Zhang, Y. Dai, H. Li, and P. Koniusz, "Deep stacked hierarchical multi-patch network for image deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5978–5986.

[9] A. Kaufman and R. Fattal, "Deblurring using analysis-synthesis networks pair," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 5810–5819. [Online]. Available: https://ieeexplore.ieee.org/document/9157378/

[10] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 769–777.

[11] L. Pan, R. Hartley, M. Liu, and Y. Dai, "Phase-only image based kernel estimation for single image blind deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 6027–6036. [Online]. Available: https://ieeexplore.ieee.org/document/8954150/
[12] S. Zhou, J. Zhang, J. Pan, W. Zuo, H. Xie, and J. Ren, "Spatio-temporal filter adaptive network for video deblurring," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2482–2491.
[13] M. Suin, K. Purohit, and A. N. Rajagopalan, "Spatially-attentive patch-hierarchical network for adaptive motion deblurring," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3606–3615.
[14] J. Pan, H. Bai, and J. Tang, "Cascaded deep video deblurring using temporal sharpness prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 3040–3048. [Online]. Available: https://ieeexplore.ieee.org/document/9157525/
[15] P. Liu, J. Janai, M. Pollefeys, T. Sattler, and A. Geiger, "Self-supervised linear motion deblurring," *IEEE Robot. Autom. Lett.*, vol. 5, no. 2, pp. 2475–2482, Apr. 2020.
[16] Y. Bai, H. Jia, M. Jiang, X. Liu, X. Xie, and W. Gao, "Single-image blind deblurring using multi-scale latent structure prior," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 2033–2045, Jul. 2020.
[17] K.-H. Liu, C.-H. Yeh, J.-W. Chung, and C.-Y. Chang, "A motion deblur method based on multi-scale high frequency residual image learning," *IEEE Access*, vol. 8, pp. 66025–66036, 2020.
[18] K. Purohit and A. Rajagopalan, "Region-adaptive dense network for efficient motion deblurring," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 11882–11889.
[19] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14821–14831.
[20] Z. Chi, Y. Wang, Y. Yu, and J. Tang, "Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 9137–9146.
[21] P. Tran, A. T. Tran, Q. Phung, and M. Hoai, "Explore image deblurring via encoded blur kernel space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 11956–11965.
[22] X. Hu, W. Ren, K. Yu, K. Zhang, X. Cao, W. Liu, and B. Menze, "Pyramid architecture search for real-time image deblurring," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4298–4307.
[23] S. H. Jung and Y. S. Heo, "Disparity probability volume guided defocus deblurring using dual pixel data," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, 2021, pp. 305–308.
[24] J. Liu, X. Wen, W. Nie, Y. Su, P. Jing, and X. Yang, "Residual-guided multiscale fusion network for bit-depth enhancement," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 5, pp. 2773–2786, May 2022.
[25] D. Rozumnyi, M. R. Oswald, V. Ferrari, J. Matas, and M. Pollefeys, "DeFMO: Deblurring and shape recovery of fast moving objects," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 3456–3465.
[26] F. Xu, L. Yu, B. Wang, W. Yang, G.-S. Xia, X. Jia, Z. Qiao, and J. Liu, "Motion deblurring with real events," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2583–2592.
[27] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 2017–2025.
[28] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.
[29] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 3–19.
[30] Y. Du, C. Yuan, B. Li, L. Zhao, Y. Li, and W. Hu, "Interaction-aware spatio-temporal pyramid attention networks for action classification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 373–389.
[31] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[32] M. Niepert, M. Ahmed, and K. Kutzkov, "Learning convolutional neural networks for graphs," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2014–2023.
[33] S. I. Ktena, S. Parisot, E. Ferrante, M. Rajchl, M. Lee, B. Glocker, and D. Rueckert, "Distance metric learning using graph convolutional networks: Application to functional brain networks," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Springer, 2017, pp. 469–477.
[34] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 7444–7452.
[35] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 399–417.
[36] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[38] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3883–3891.
[39] J. Rim, H. Lee, J. Won, and S. Cho, "Real-world blur dataset for learning and benchmarking deblurring algorithms," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 184–201.
[40] Z. Shen, W. Wang, X. Lu, J. Shen, H. Ling, T. Xu, and L. Shao, "Human-aware motion deblurring," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5572–5581.



**LIANJUN LIAO** (Member, IEEE) received the M.S. degree from the North China University of Technology, Beijing, China, in 2009. He is currently pursuing the Ph.D. degree in computer application technology with the University of Chinese Academy of Sciences and the Institute of Computing Technology, Chinese Academy of Sciences, Beijing. His current research interests include image processing, human motion, and computer graphics.

**ZIHAO ZHANG** (Member, IEEE) received the B.S. degree in mathematics from Sichuan University, Chengdu, China. He is currently pursuing the Ph.D. degree with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China. His research interests include point cloud processing and human motion modeling.

**SHIHONG XIA** (Member, IEEE) received the B.S. degree in mathematics from Sichuan Normal University, Chengdu, China, in 1996, and the Ph.D. degree in computer software and theory from the University of Chinese Academy of Sciences, Beijing, China, in 2002. He is currently a Professor with the Institute of Computing Technology, Chinese Academy of Sciences. His research interests include computer graphics, virtual reality, and artificial intelligence.