Low-Light Raw Video Denoising With a High-Quality Realistic Motion Dataset

Ying Fu, *Senior Member, IEEE*, Zichun Wang, Tao Zhang, and Jun Zhang

***Abstract*—Recently, supervised deep-learning methods have shown their effectiveness on raw video denoising in low-light. However, existing training datasets have specific drawbacks, e.g., inaccurate noise modeling in synthetic datasets, simple motion created by hand or fixed motion, and limited-quality ground truth caused by the beam splitter in real captured datasets. These defects significantly decline the performance of network when tackling real low-light video sequences, where noise distribution and motion patterns are extremely complex. In this paper, we collect a raw video denoising dataset in low-light with complex motion and high-quality ground truth, overcoming the drawbacks of previous datasets. Specifically, we capture 210 paired videos, each containing short/long exposure pairs of real video frames with dynamic objects and diverse scenes displayed on a high-end monitor. Besides, since spatial self-similarity has been extensively utilized in image tasks, harnessing this property for network design is more crucial for video denoising as temporal redundancy. To effectively exploit the intrinsic temporal-spatial self-similarity of complex motion in real videos, we propose a new Transformer-based network, which can effectively combine the locality of convolution with the long-range modeling ability of 3D temporal-spatial self-attention. Extensive experiments verify the value of our dataset and the effectiveness of our method on various metrics.**

***Index Terms*—Raw video denoising, transformer, convolutional neural network, temporal-spatial self-attention.**

I. Introduction

SINCE the rapid development of smartphone cameras, with the increasing need to shoot videos in night scenes, low-light videography has been of great importance. However, due to the low photon count, noise is almost inescapable in the low-light environment. The noise highly degrades the quality of videos. To tackle this, several hardware-based solutions aim to gather more photons [1], [2]. For example, one may use a larger aperture size, open a flashlight, or take a long-exposure image. However, the aperture size is always limited by the camera size, especially for smartphone cameras. The flashlight can only illuminate nearby objects, and the long exposure time is only suitable for static scenes. These limitations weaken its effectiveness.

In contrast, computation-based denoising has its own merit, because of its better compatibility with various devices [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21]. Their performances heavily depend on the amount of information in the original noisy image.

In pursuit of more information in input data, researchers favor performing denoising directly in raw domain [22], [23], [24]. The readings in the raw domain are not destroyed by the non-linear image signal processor (ISP), so it can strictly reflect scene irradiance and record original values. Consequently, we put our emphasis on Raw2Raw video denoising in this work.

Owing to this, several raw image denoising datasets [22], [23], [25] have been proposed. These are typically composed of noisy/clean pairs, based on either shooting long/short exposure pairs or averaging multiple noisy images for ground truth. Unfortunately, it is hard to directly adapt this way from images to videos, since objects may move along the temporal dimension. Neither of these two techniques can capture high-quality ground truth in dynamic scenes.

It is hard to collect real raw video denoising datasets on dynamic scenes in low-light circumstances. For this reason, some use synthetic data for training [26], while the inaccuracy can harm the final performance. Additionally, researchers try to capture paired clean and low-light videos in simplified settings, which can be categorized into: i) containing static scenes only, e.g., SMID [27], ii) using manually created motion, e.g., CRVD [26] or fixed motion, e.g., SDSD [28], iii) low luminous flux caused by the beam splitter in a co-axis optical system, where the beam splitter can create two spatially aligned clean scenes. Then, one of the clean scenes is made noisy by adding a neutral density (ND) filter, e.g., SMOID [29], etc. For the first and second settings, manually created motion or no motion is much simpler than in real-world cases. For the third setting, it takes precise control to align two frames at the pixel level in the co-axis optical system, making this system hard to be assembled. Also, photons are halved by the beam splitter, making ground truth frames drowned in noise, which limits the dataset quality. In general, these datasets are all collected in degraded conditions, which may significantly decline the performance of the network trained on them when tackling real scenes. There exists

Manuscript received 9 October 2022; revised 5 December 2022; accepted 16 December 2022. Date of publication 30 December 2022; date of current version 8 December 2023. This work was supported by the National Natural Science Foundation of China under Grants 62171038, 61827901, and 62088101. The Associate Editor coordinating the review of this manuscript and approving it for publication was Dr. Vladan Velisavljevic. *(Corresponding author: Jun Zhang.)*

Ying Fu is with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China, and also with the Yangtze Delta Region Academy of Beijing Institute of Technology, Jiaxing 314000, China (e-mail: fuying@bit.edu.cn).

Zichun Wang and Tao Zhang are with the Beijing Laboratory of Intelligent Information Technology, School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China (e-mail: wangzichun@bit.edu.cn; tzhang@bit.edu.cn).

Jun Zhang is with the MIIT Key Laboratory of Complex-field Intelligent Exploration, Beijing Institute of Technology, Beijing 100081, China, and also with the Yangtze Delta Region Academy of Beijing Institute of Technology, Jiaxing 314000, China (e-mail: zhjun@bit.edu.cn).

Digital Object Identifier 10.1109/TMM.2022.3233247

TABLE I
COMPARISON OF THE EXISTING VIDEO DENOISING DATASET AND OUR CAPTURED DATASET

	Realistic scene motion	Number of paired videos	Number of noise levels	High-quality ground truth	No extra equipment
SMID	static	202	5	✓	✓
CRVD	static background with simple motion created by hand	55	5	✓	✓
SDSD	scene moving in one direction	150	1	✓	electric slide rail and controller
SMOID	✓	179	5	limited quality since photons are halved by the beam splitter	co-axis optical system
Ours	✓	210	6	✓	✓

✓ is used if the dataset meets the corresponding requirement. Otherwise, we add a brief explanation. Our dataset features realistic motion, more numbers of noise levels, high-quality ground truth, and no extra equipment.

no high-quality raw video denoising dataset in the low-light with realistic motion.

In addition to the dataset, increased interest lies in video denoising methods. For image denoising, extensive researches have shown the validity of U-net [30], owing to its encoder-decoder architecture with skip connection. Despite the astonishing results of image denoising, one major issue in video denoising is how to utilize temporal information. To temporally align multi-frames, some directly use convolution temporally [31], while others use optical-flow [32], [33] or deformable convolution [26] to aggregate temporal features. For video denoising, spatial-temporal information is of vital importance. However, existing methods often use auxiliary modules for alignment, where sub-optimal alignment can harm their performance. Also, the fusion of features in multi-frames of existing methods may not fully exploit joint self-similarity in temporal-spatial dimension.

In this paper, towards a high-quality realistic motion raw video denoising dataset, and also the better utilization of temporal-spatial self-similarity, we present a low-light raw video denoising dataset containing realistic motion, with a new Transformer method containing 3D window-based self-attention mechanism for video denoising.

First, for the dataset, we directly obtain complex motion in real-world cases, instead of manually setting simple or fixed motion. Under carefully considered shooting conditions to avoid moire patterns, we capture complex motion in real videos displayed on the monitor. In this way, motion is made controllable by pausing or playing the video. Therefore, we are able to capture high-quality pairs in a frame-by-frame manner. We collect and choose the videos being shot, making sure all the scenes vary enough. Meanwhile, although the data is captured on a monitor, we validate that its noise distribution and depth perception is closed enough to the real-world data.

Besides a new dataset, a novel method for raw video denoising is also proposed. Spatial self-similarity has been proven to be indispensable for image denoising, and the redundant temporal information has further increased their importance for video tasks. However, it may not be fully utilized by earlier networks. Our network considers self-similarity in temporal-spatial dimension based on the attention mechanism, while following the design of the classic image denoising method, *i.e.*, U-net. By adding 3D temporal-spatial self-attention in the encoder and decoder, also with the temporal fusion block for multi-frame aggregation, we can use the abundant information in natural videos more efficiently.

In summary, our main contributions are that:

1) We present a dataset for raw video denoising in low-light on dynamic scenes, which contains high-quality frames with realistic motion and multiple noise levels, bridging the gap in raw video denoising datasets.
2) We design a novel Transformer-based method for low-light raw video denoising, which exploits temporal-spatial self-similarity and combines locality with the long-range interaction of Transformer blocks.
3) Experimental results prove the value of our dataset and the effectiveness of the proposed method on various metrics.

II. RELATED WORKS

In this section, we briefly review various related video denoising methods and video denoising datasets.

A. Video Denoising Methods

Numbers of methods have been proposed for video denoising. For traditional state-of-the-art methods, VBM4D [34] groups similar patches in spatial and temporal dimensions.

Recently, deep-learning based methods have achieved better performance. Some studies use recurrent neural networks [35], [36], or multi-branch networks [37], while other methods adapt the classic U-net for video, by changing the convolution type, e.g., 3D convolution in SMOID [29] and 3D deformable convolution in $3D^2Unet$ [38]. Besides methods following a sliding window manner, ViDeNN [31] has two CNN subnetworks for spatial and temporal denoising. Since temporal self-similarity can be better extracted between aligned consecutive frames, several works focus on the alignment between reference and neighbor frames. DVDnet [32] and TOFlow [33] warp input frames for the temporal denoising module by explicitly estimating optical flow. FastDVDnet [39] advances it through replacing explicit estimation with U-net blocks. Furthermore, TDAN [40] features deformable convolution for temporal alignment. Moreover, RViDeNet [26] makes use of non-local attention and

Fig. 1. The comparison of motion in multiple video denoising datasets.[1] For each dataset, we show three examples. The left is the original frame and the right is the displacement between this frame and the next frame (not shown). Different colors represent the motion direction, and the saturation represents the intensity of the movement. The background in CRVD is static and the direction of the object is fixed. SMID only contains videos with static scenes. The direction of motion in SDSD is fixed by an electric slide rail. Our displacement figures are the most colorful and complex, which represents our dataset containing complex motion patterns matching real-world cases.

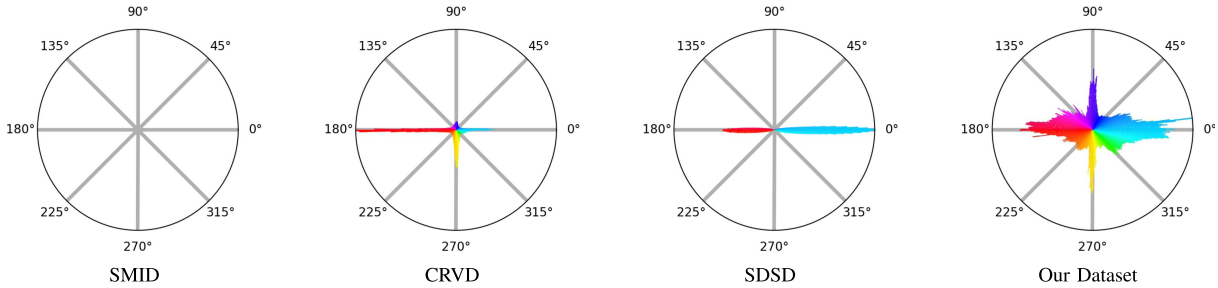


Fig. 2. Statistics for motion directions in four datasets.[1] We plot a circular histogram, where the color of each bin represents the direction of motion, and the height of the bar represents the proportion of specific direction to all the directions. SMID contains no motion, CRVD and SDSD contain motion mostly in the horizontal direction. In contrast, motion in our dataset can cover all directions, representing the complex and realistic motion in our dataset.

pyramid, cascading deformable convolution to enhance the performance, achieving state-of-the-art for raw video denoising.

Since self-similarity greatly exists in natural videos, corrupted details can be recovered by both spatially nearby pixels and temporally consecutive frames. However, current video denoising methods may either suffer from sub-optimal alignment, or not fully utilize the underlying spatial-temporal information. In this work, we aim to extensively exploit the intrinsic spatial-temporal self-similarity based on 3D attention.

B. Video Denoising Dataset

Deep-learning based methods have achieved promising results for video denoising. However, its data-driven property attaches extreme importance to the dataset. Unfortunately, due to the movement in videos, it is hard to directly use long/short exposure pairs or average multiple shots for ground truth. Consequently, most raw video denoising datasets are synthesized or captured in degraded conditions.

For synthetic datasets, early studies add gaussian noise to clean videos. SRVD [26] synthesizes the raw video dataset by unprocessing sRGB videos to raw videos, but the fixed ISP parameters make the color inaccurate. For real captured datasets, SMID [27] contains only static scenes for training. Even if an extra consistency loss is added, the network is still hard to generalize on scenes with motion. CRVD [26] uses controllable objects to create simple motion manually, which may not be accurate compared to real motion. Recently, SDSD [28] creates motion by moving the camera on a rail system. However, videos with nearly-identical camera motions can not match the complex motion in our lives. Besides, only one noisy level is not enough for real lighting conditions. SMOID [29] uses a co-axis optical system to split the light into two beams, with the ND filter on one of the beams for creating aligned noisy scenes. It can capture clean/noisy videos simultaneously, while the low photon count caused by the beam splitter may lead to noisy ground truth frames, thus harming the data quality. Also, it takes effort to precisely align two scenes after splitting the light. A detailed comparison is listed in Table I.

Current low-light video denoising datasets are either synthetic or captured in simplified conditions. In contrast, we directly obtain realistic motion in our raw low-light video denoising dataset, featuring high-quality data, multiple noise levels, and no need for extra equipment.

III. Low-Light Realistic Motion Video Dataset

Several low-light raw image denoising datasets have been proposed [22], [23], [25], where ground truth images are captured by either using long exposure or averaging multiple shots. However, these techniques may induce blur in dynamic scenes, so cannot be directly applied to videos. Hindered by this, existing low-light raw video datasets degrade the quality in exchange for

[1]SMOID is not included since it has not been released.

the accessibility of capturing such datasets. These degradations include manually created motion or fixed motion [26], [27], [28], low luminous flux caused by beam splitter in the co-axis optical system for spatially aligned pairs [29].

Both of these settings may significantly reduce both the quality and quantity of collected data. For the former, its motion is much simpler than the real-world cases. For example, SMID [27] is completely static, while CRVD and SDSD only contain motion along a single direction [26], [28]. Also, dynamic scenes in the presence of moving objects, e.g., pedestrians or vehicles cannot be contained. This may lead to poor generalization of networks trained on them to realistic videos. Besides, CRVD [26] needs to move the objects by hand, and SDSD [28] needs an electric rail for camera movement and the same static scene when capturing clean/noisy videos, all of which reduce the capture efficiency. For the latter, SMOID [29] uses a beam splitter to capture aligned videos with the cost of limited dataset quality. Photons are halved by the beam splitter, which reduces the quality of ground truth frames. Also, the co-axis optical system needs to be assembled with effort.

We need a low-light raw video denoising dataset with realistic motion and high-quality frames. It would be better if no extra equipment is required, which is great for reproduction. However, one main challenge comes from collecting ground truth videos with realistic motion due to motion blur. Instead of setting simple or fixed motion like in previous works, we directly acquire complex motion in real-world videos. In this work, we manage to control complex motion by playing or pausing real-world videos on the monitor, since scenes are static when paused. This enables us to capture clean/noisy video pairs in a frame-by-frame manner. By taking this simple yet effective approach, we obtain realistic motion with high-quality frames from real videos, as shown in Fig. 1. The corresponding analysis of motion directions is shown in Fig. 2, where our dataset features complex and realistic motion patterns.

As shown in Algorithm 1, we first collect 70 high-quality 4 k videos from the internet, then play them on the DELL U2720QM monitor. We use a Sony Alpha 7R IV full-frame mirrorless camera. The size of the Bayer image is 9504 $\times$ 6336. The scenes of the video clips contain indoor and outdoor, ranging from natural landscapes to extreme sports. This relatively large range of scenes also has an advantage compared to previous datasets. Examples of our data are in Fig. 3.

Another problem generated while shooting a monitor is the appearance of the moire pattern. To solve this, we carefully posit the camera until the moire pattern disappears. Also, the distance between the screen and the camera is set far enough to ensure every monitor pixel is smaller than a camera sensor pixel. All shooting processes are performed in the darkroom, so the lighting condition is strictly controlled.

We totally capture 210 video pairs including 70 scenes. For each frame, it contains one long-exposure ground truth and three noisy counterparts randomly chosen from six low-light ratios that range from 100 to 320. As shown in Table I, our dataset contains the most noise levels. We have about 3000 noisy/clean pairs with dynamic scenes in total. All paired video frames are in an unsigned integer format organized in the RGBG-based Bayer pattern, since we put our emphasis on the Raw2Raw video denoising task. Still, the raw frames can further be post-processed for sRGB data, so as to support Raw2RGB and RGB2RGB denoising tasks.

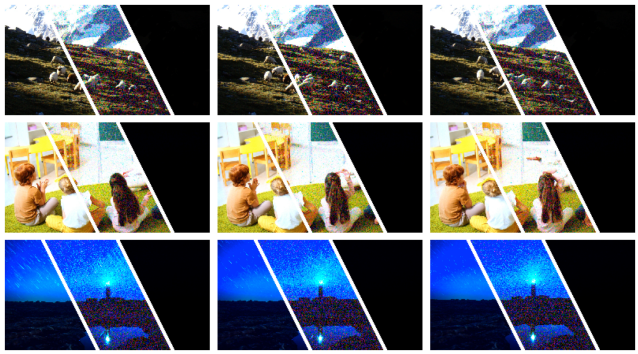


Fig. 3. Examples of the videos we collect. Each row includes three example frames. The left is the ground truth frame. The right is the low-light frame in short exposure, which appears entirely black due to our low-light capture condition. The center is the linearly scaled low-light frame matching the brightness of ground truth. The white lines are added to separate each component only for visualization. (Best viewed on screen with zoom).

Algorithm 1: Dataset Capture Protocol.

Require: $t_b = 1\ s, g_b = ISO\ 100$;
Posit camera until moire pattern disappears. Distance between the camera and monitor should ensure one monitor pixel is smaller than a camera sensor pixel;
for Each Scene **do**
  Meter the scene to find the aperture size f that well exposes the video;
  Choose 3 out of 6 random low-light ratios r, $r \in [100, 320]$;
  for Each frame in the video **do**
    Take the reference frame at exposure setting (f, t_b, g_b);
    for Each low light ratio r **do**
      Take the noisy frame at $(f, t_b/r, g_b)$;
    end
    Play the next frame of the video;
  end
end

A. Realistic Scene Motion Matters

For data-driven methods, the quality of the training dataset always decides the final performance. Specifically, for video denoising, since motion in real-world videos is mostly complex and random, the existence of such complex motion in the training dataset may be better than containing simple or fixed motion only. However, this is barely discussed in previous works, and the effect of a wide variety of motions has hardly been figured out. Now, we discuss whether the complexity of motion is instrumental to deep-learning based methods.

To answer this question, the key is to know whether performance on complex motion videos declines remarkably, if there

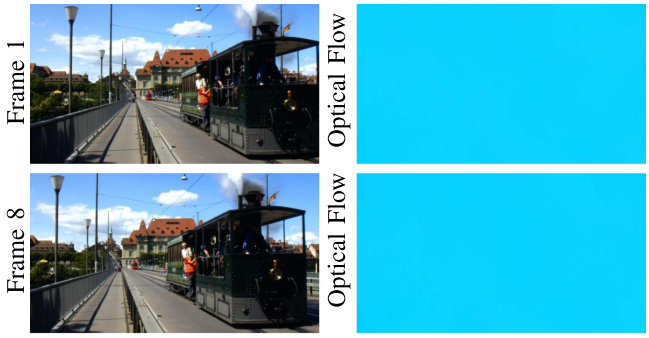


Fig. 4. Examples of synthesized videos with simple fixed motion. We present the 1st and 8th frames of the video on the first and the second row. In each row, the original frame is on the left and the corresponding motion (visualized by optical flow) is on the right. The optical flow appears entirely blue because synthesized video with simple fixed motion contains only one kind of motion.

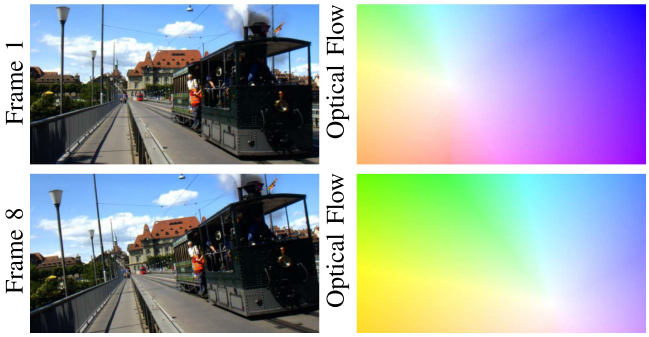


Fig. 5. Examples of synthesized videos with complex and random motion. We present the 1st and 8th frames of the video on the first and the second row. In each row, the original frame is on the left and the corresponding motion (visualized by optical flow) is on the right. Note that in our synthesized videos, complex and random motion varies *between frames* and *within each frame*, which can be found in the diverse colors of optical flow.

is a lack of corresponding complex motion videos for training. In other words, we need to train the same network on simple and complex motion videos. Then, the value of motion types can be determined by the gap between the performance of these two networks.

We need paired videos with simple and complex motion respectively. Also, the scenes in each video should be nearly the same for fair comparison. However, under this setting, real paired videos are almost impossible to be captured. Therefore, we synthesize videos by rotating or shifting the same image, in order to simulate videos with both simple and complex motion. By this means, we ensure the scenes in both videos are almost identical, thus excluding the gap brought by scene differences. Also, the complexity of motion can be easily controlled.

In detail, we synthesize videos with two types of motion. *i) Simple and fixed motion*, as shown in Fig. 4. It is implemented by horizontally shifting the image to the right, where the shifting distance is fixed. *ii) Complex and random motion*, as shown in Fig. 5. This complex motion includes rotation, horizontal and vertical movement with random shifting distance. For each motion type, we synthesize 64 videos in sRGB color space, each for 10 frames in total. Gaussian noise ($\sigma \in [1, 55]$) is then added for the noisy counterpart. After obtaining videos with two types of motion, we train a convolutional neural network [29] with temporal fusion. Training settings are kept the same for the two networks. For the testing set, we use synthesized complex motion videos with Gaussian noise ($\sigma = 15, 25, 50$). Results are listed in Table II.

TABLE II
COMPARISON OF DENOISING PERFORMANCE ON COMPLEX MOTION VIDEOS

	Trained on videos with simple and fixed motion		Trained on videos with complex and random motion	
Sigma	PSNR	SSIM	PSNR	SSIM
15	26.32	0.777	**31.53**	**0.880**
25	25.58	0.735	**29.73**	**0.833**
50	23.82	0.627	**26.81**	**0.721**
Average	25.24	0.713	**29.36**	**0.811**

We train the same network with two types of synthesized videos, Including: i) Simple and fixed motion, ii) Complex and random motion. Note that the scenes are kept the same for fair comparison. Network trained on complex motion videos performs better than network trained on simple motion videos. It indicates the value of a wide variety of motion in training dataset for video denoising.

The network trained on complex motion performs 4.12 dB higher in PSNR than the network trained on simple motion. Furthermore, since the scenes are kept nearly the same, the performance gap is almost all brought by the bias between motion types. This great gap validates the significance of complex motion in the video denoising dataset. Accordingly, a wide variety of motion types matters for video denoising, which confirms the significance of our proposed dataset. It is noteworthy that the pattern of motion in real-life video is still more complicated than our synthesized complex motion, so the performance gap between different types of motion may even be amplified.

B. Accurate Noise Distribution for Monitor

Another determinant factor for deep-learning based video denoising methods is the noise distribution. Earlier studies may use inaccurate synthetic noise, while the bias between synthetic and real noise hinders its application in real scenes.

Realistic noise is fundamental. Therefore, we justify that our dataset, though captured with a monitor, also contains realistic noise distribution just like capturing real objects. We first show how the noise is generated in the camera, then conduct corresponding experiments to support our argument.

For the CMOS sensor, which is the dominating imaging sensor nowadays [41], the incident light conversion process involves three stages [42], including i) from photons to electrons, ii) from electrons to voltage, and iii) from voltage to digital numbers. Corresponding noise is generated in each stage, which can be categorized into light-dependent noise and light-independent noise. For the latter, it generates even if there is no light, thus unaffected by the light source [42]. For the former, the number of electrons collected in a fixed exposure time is inevitably uncertain, owing to the quantum nature of light. This follows the Poisson distribution [43]. Therefore, given the same luminous intensity of one signal, the source of a signal has no effect on the

Fig. 6. Example ground truth frame of our captured demo dataset. We first capture noisy/clean real-world videos (a clean frame is shown on the left), then play the captured clean videos on the monitor. Then, the same scene is captured again on the monitor (a clean frame is shown on the right). Black gaps of frames captured on the monitor are then cut off to match the real-world video. We then subtract the noisy and clean frames for real-world scenes and monitor scenes respectively, in order to compare the noise distribution.

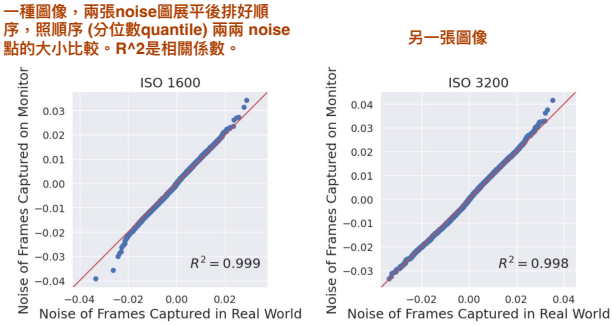


Fig. 7. The Q-Q (quantile-quantile) plot of noise on real scenes and monitor scenes. *To exclude the effect of diverse scenes, we first capture videos on real-world stationary scenes, then play the ground truth videos on the monitor and capture them again.* On both noise levels, it can be observed that the points in the Q-Q plot (blue points) are closely fitted to the $y = x$ line (red line). This means the noise distributions on real and monitor scenes are close enough.

noise. Whether the light is from the light-emitting diode (LED) monitor or reflected by real objects, the noise distribution should not have much difference.

To further validate our argument, we capture a demo dataset to compare the distribution of monitor-captured and real-world videos, as shown in Fig. 6. First, we capture noisy and clean videos in real-world situations in a frame-by-frame manner. Then, all the collected ground truth videos are played on the monitor, and captured again for each frame. Finally, we can get the monitor-captured counterparts for each real-world video. Overall, we collect 6 videos, each with 5 frames and 2 noise levels of real objects. To compare the noise distribution, we subtract the ground truth from the noisy counterpart for videos captured in the real-world and on the monitor, respectively. The subtraction results are then normalized with the ground truth overall intensity. We plot the quantile-quantile (Q-Q) plot of two sets of videos, as shown in Fig. 7.

It can be seen that the points in the Q-Q plot are closely fitted to the $y = x$ line, which means the two distributions are almost the same. The coefficient of determination ($i.e.$, R^2) is 0.999 and 0.998 for two noise levels, proving the similarity between the two noise distributions. This justifies that there is no bias between the noise distribution for monitor and real objects, and our dataset contains realistic noise distribution.

C. Depth Perception for Monitor

Compared with previous works, our dataset contains realistic motion with high-quality ground truth videos. However, since the videos are captured on the monitor, one may concern if the limited depth will result in a discrepancy between our dataset and real-world videos. Although the precise recording of depth is crucial for specific tasks, e.g., depth estimation, we justify that our dataset, used for video denoising instead, will not suffer from insufficient depth information.

For common CCD and CMOS sensors, captured raw frame F can be expressed as the summation of i) the potential clean frame and ii) the noise components [21], while neither of them will be affected when capturing monitor in terms of depth.

First, for the potential clean frame, *i.e.,* the real-world scenarios, the impact of depth is mainly reflected in the out-of-focus effect, which occurs on objects out of the camera's depth-of-field [44]. For example, a person close to the camera may stay clear, while the background far from the camera may be blurred [45]. Though capturing a monitor contains the same depth, videos played on the monitor are already blurred when being recorded [46], which represents the depth of each object. This makes the potential clean videos in our dataset the same as real-world conditions. A similar approach can be found in the City100 dataset [47], where researchers capture city scenes printed on postcards as ground truth images.

Second, for the noise components, common sensors record the number of photoelectrons, which is proportional to the scene irradiation [21], [48]. Therefore, noise distribution will remain the same for the signal with the same luminosity but different depths, making the depth information irrelevant to the noise components when capturing the monitor.

IV. Low-Light Video Denoising Transformer

In this section, we show the overall architecture of our proposed method, and describe the basic 3D spatial-temporal self-attention block with convolution.

A. Overall Pipeline

Encoder-decoder is a classic architecture for low-level image tasks, exemplified by U-net [30]. The main issue for adopting the design of U-net to video denoising is how to efficiently use the redundant temporal information. To align temporal features, existing methods often use an auxiliary module for alignment, including convolution only [31], [39], deformable convolution [26], optical flow [33]. However, sub-optimal alignment operation may harm its performance.

Besides, most existing methods use convolution for multi-frame features fusion, where the lacking of long-range modeling ability may decline their recovery result. Some methods utilize spatial self-similarity, e.g. [26], while the abundant temporal-spatial self-similarity in the extra temporal dimension is not fully exploited.

Self-attention is suited for aggregating self-similarity since it can dynamically allocate weight for each pixel. To this end, we combine 3D temporal-spatial attention with the hierarchical design of U-net. Nonetheless, Transformer may suffer from the deficiency of local feature extraction, which is indispensable for recovering image details. Thus, we combine the locality of convolution with the long-range interaction of self-attention in each Transformer block.

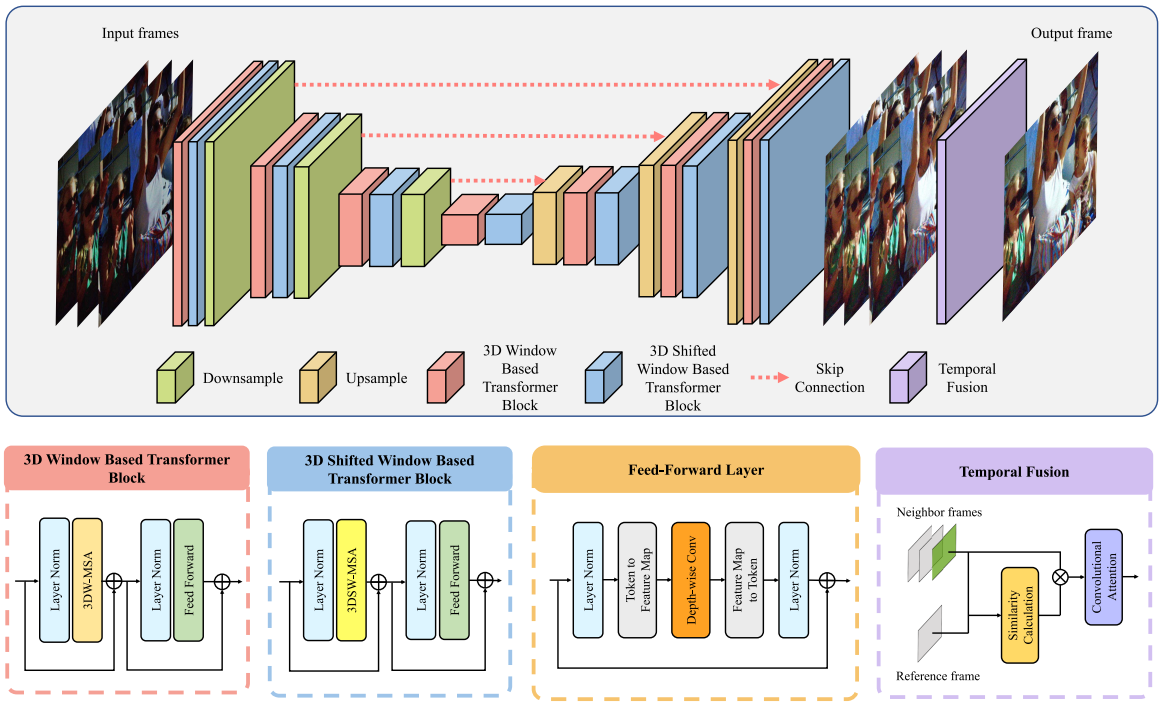


Fig. 8. Overview of network architecture. Swin Transformer denotes Shifted Window-based Transformer. 3D (S)W-MSA denotes 3D (Shifted)Window-based Multi-head Self-attention. LN denotes Layer Normalization. Convolutional Attention denotes our final fusion block.

The overview architecture of our network is shown in Fig. 8. We focus on the Raw2Raw video denoising task, where the input and output are all in the raw domain. The input is of size $T \times H \times W \times 4$. T represents the number of input frames, with each frame containing $H \times W \times 4$ pixels in the Bayer pattern. The output frame is of size $H \times W \times 4$. To embed the pixels in images as tokens, we first apply a 3×3 convolution. After embedding, all the tokens pass through K encoders and patch merging layers. Each encoder contains M Shifted Window Transformer blocks. For downsampling, we use the 4×4 convolution and double the dimensions. Symmetrically, the decoder includes K Transformer blocks and patch expanding layers. The output of decoder layers is then projected back to image patches. Finally, the extracted multi-frame features are temporally fused to handle the misalignment.

B. 3D Swin Transformer Block

Since vanilla self-attention [49] is computationally consuming, directly adopting it to video denoising is not affordable due to the extra temporal dimension. Besides, Transformer [49] has strong long-range modeling ability but neglects local features, which is vital for recovering details. To extract locality with less computational effort, we apply 3D shifted window-based multi-head self-attention (3DSW-MSA) and 3D window-based multi-head self-attention (3DW-MSA) [50], together with depth-wise convolution in the feed-forward layer. In this way, we can effectively extract the local features by convolution, at the same time fully taking advantage of intrinsic temporal-spatial self-similarity by the long-range modeling ability of the Transformer.

Two consecutive 3D shifted window-based Transformer blocks are computed as:

$$\begin{aligned}
\hat{\mathbf{z}}^{l} &= \text{3DW} - \text{MSA}\left(\text{LN}\left(\mathbf{z}^{l-1}\right)\right) + \mathbf{z}^{l-1}, \\
\mathbf{z}^{l} &= \text{FFN}\left(\text{LN}\left(\hat{\mathbf{z}}^{l}\right)\right) + \hat{\mathbf{z}}^{l}, \\
\hat{\mathbf{z}}^{l+1} &= \text{3DSW} - \text{MSA}\left(\text{LN}\left(\mathbf{z}^{l}\right)\right) + \mathbf{z}^{l}, \\
\mathbf{z}^{l+1} &= \text{FFN}\left(\text{LN}\left(\hat{\mathbf{z}}^{l+1}\right)\right) + \hat{\mathbf{z}}^{l+1},
\end{aligned} \tag{1}$$

where $\hat{\mathbf{z}}^{l}$ and $\mathbf{z}^{l}$ represent the output features of the 3DW-MSA and 3DSW-MSA for lth block. A LayerNorm (LN) is added before MSA and after the FeedForward layer (FFN). Following the previous studies, we add the relative position encoding $B \in \mathbb{R}^{T^2 \times M^2 \times M^2}$ to the 3D attention block. The self-attention is computed as:

$$\text{Attention}\left(Q, K, V\right) = \text{SoftMax}\left(QK^{T}/\sqrt{d} + B\right)V, \tag{2}$$

where $Q, K, V \in \mathbb{R}^{TM^2 \times d}$ are the query, key and value matrices. d is the dimension of the query and key features. TM^2 is the number of tokens per window. And, the values of B are taken from the 3D bias matrix $\hat{B} \in \mathbb{R}^{(2T-1)\times(2M-1)\times(2M-1)}$, corresponding to the temporal range of $[-T+1, T-1]$ and the spatial range of $[-M+1, M-1]$.

C. Temporal Fusion

After the exploitation of spatial-temporal self-similarity, features in neighbor frames are fused for the recovery of the reference frame. However, it is not appropriate to simply combine these frames, since the complex motion in real videos makes

each neighbor frame contribute variously to the central reference frame. Intuitively, the closer between the features in the neighbor frame and reference frame, the more information a neighbor frame can provide for recovery. Therefore, we first extract the features by embedding, then compute the similarity between the features of each neighbor and the reference features in an embedded space:

$$S\left(F_{t+i}, F_t\right) = \mathrm{Sim}\left(\theta\left(F_{t+i}\right)^T, \phi\left(F_t\right)\right), \tag{3}$$

where θ and ϕ are embedding functions. Sim denotes the similarity calculation function. Here we also adopt the dot product following previous work [51] for similarity calculation. F_t refers to the reference frame and F_{i+t} refers to the neighbor frames where $i \in [-T+1, T-1]$. After getting the similarity matrix, we adaptively re-weight the features in the temporal dimension,

$$\tilde{F}_{t+i} = F_{t+i} \odot S\left(F_{t+i}, F_t\right), \tag{4}$$

$$F_{\text{fusion}} = \mathrm{Conv}\left(\left[\tilde{F}_{t-T}, \ldots, \tilde{F}_t, \ldots, \tilde{F}_{t+T}\right]\right), \tag{5}$$

where $\odot$ and $[\cdot,\cdot,\cdot]$ denote the element-wise multiplication and concatenation respectively. We then concatenate all the features and gather them together for the reconstructed frame by convolution layer. Finally, a convolutional attention module [52] is used to spatially enhance the feature representation.

V. Experiments

In this section, we first introduce the datasets and setup. Then, we compare our method with state-of-the-art methods. Finally, we compare the restored performance on real captured videos to verify the value of our dataset.

A. Datasets and Setup Details

Two raw video denoising datasets are used to evaluate the performance, including CRVD [26] and our dataset. We compare raw video denoising on the dataset with dynamic scenes. SDSD [28] is on the sRGB domain, SMOID [29] is not released, and SMID [27] only contains static scenes, thus we do not compare the performance of them.

For CRVD, we follow the settings in their paper. For our dataset, the patch size for training is set to 256×256. We randomly select 6 videos from all scenes for testing, overall including 360 pairs of noisy-clean frames of size 1400×2600. We sample three consecutive frames from the full length as input. A batch size of 1 is used in the experiment. We adopt $\mathcal{L}_1$ loss between ground truth and output for training. We use Adam optimizer with momentum terms of $(0.9, 0.999)$. We first train the network with a learning rate of 10^{-4}. After 20 epochs, the learning rate is set to 10^{-5}. The network is trained with 50 epochs until it fully converges. We implement the method in PyTorch, and train it on the Nvidia 1080Ti GPU.

Two metrics are utilized to evaluate the performance of methods, including peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) [53]. The larger value of PSNR and SSIM implies better fidelity.

TABLE III
Ablation Studies on Our Proposed Method

Spatial window size	8	4	8	16
Attention type	2D	3D	3D	3D
PSNR	43.64	44.01	44.18	**44.27**
SSIM	0.986	0.987	0.988	**0.988**

The best result is highlighted in bold.

TABLE IV
Comparison of State-of-The-Art Video Denoising Methods on Two Datasets

	Our Dataset		CRVD	
Methods	PSNR	SSIM	PSNR	SSIM
ViDeNN [31]	30.05	0.688	32.00	0.732
TOFlow [33]	28.09	0.881	36.83	0.943
FastDVDnet [39]	36.39	0.945	41.17	0.974
SMOID [29]	37.07	0.958	42.08	0.982
EMVD [36]	35.01	0.911	42.63	0.985
EDVR [51]	37.50	0.962	43.53	0.985
RViDeNet [26]	37.39	0.962	43.97	0.987
Ours	**37.74**	**0.965**	**44.18**	**0.988**

The best result is highlighted in bold.

B. Ablation Study

In this section, we do ablation studies on CRVD [26] dataset to analyze the effectiveness of each component of our method. It includes 3D temporal-spatial self-attention and local window size. Results can be found in Table III.

3D temporal-spatial attention vs. 2D spatial attention Our method considers temporal-spatial self-similarity in a more effective way. Compared with 2D self-attention, our 3D temporal-spatial self-attention can exploit similar features in the temporal axis. To prove its validity, we replace 3D attention with 2D spatial attention. PSNR drops from 44.18 dB to 43.64 dB as we cut down the temporal information.

Different local window size Self-similarity is more likely to be found in the larger local window, owing to the bigger search space within the local window. To find the influence of local window size, we set different local window sizes to show their performance respectively. PSNR drops 0.17 dB as we halve the spatial size from 8 to 4. Note that though we adopt the window size as 8 in this work, doubling the size from 8 to 16 further yields about 0.1 dB gain, showing the scalability of our method.

C. Comparison With State-of-The-Art Methods

We compare our method with six state-of-the-art methods of video denoising, including standard convolution-based methods (ViDeNN [31], FastDVDnet [39], SMOID [29] and EMVD [36]), optical flow based method (ToFlow [33]), and deformable based methods (EDVR [51] and RViDeNet [26]). All methods are compared on our captured dataset and CRVD dataset [26]. Table IV provides the averaged results over multiple scenes.

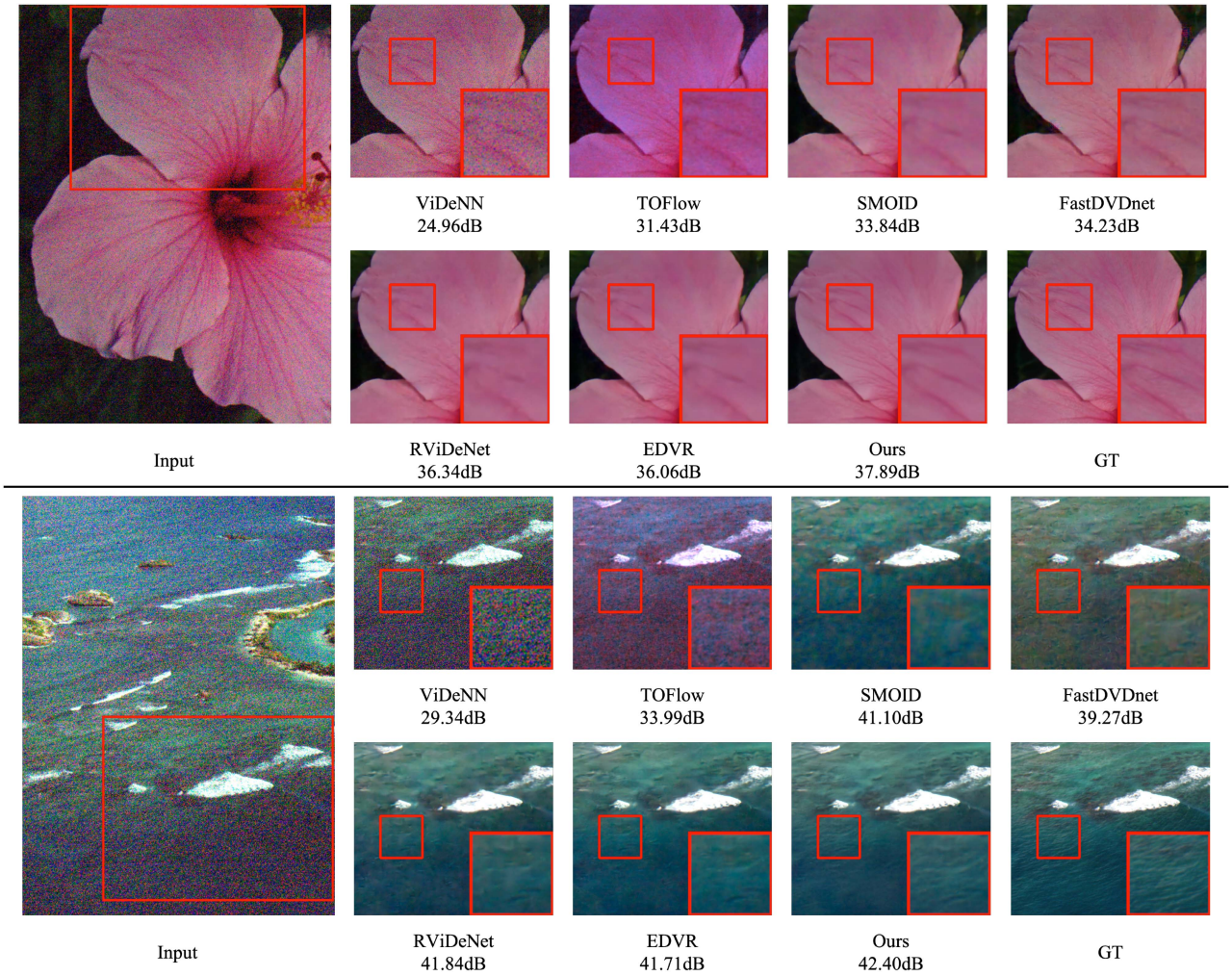


Fig. 9. Visualization of the denoised results on our dataset (zoom in for details). Our method shows the best visual quality.

We adjust all methods to raw video denoising. The best results are highlighted in bold. It can be observed that our method outperforms all compared methods on both datasets. Standard convolution-based methods, e.g., ViDeNN, FastDVDnet, EMVD and SMOID, are often constrained by the inefficiency of extracting temporal information. Also, due to the abundant self-similarity in natural videos, neighboring pixels and temporally consecutive frames are essential for recovery. However, the limited receptive field of convolution can make them not fully utilize the underlying spatial-temporal information. For optical-flow based alignment methods, e.g., TOFlow may suffer from the misalignment of sub-optimal optical flow computation. This is even more pronounced under high noise levels. For deformable convolution-based methods, e.g., EDVR and RViDeNet, they may lack the ability to model long-range interaction, thus being unable to fully exploit the abundant self-similarity in temporal and spatial dimensions. Besides, the fixed amplitudes from all the different spatial locations may also be not proper [54].

To visualize the results, representative denoised frames on our captured dataset and CRVD dataset [26] are shown in Figs. 9 and 10 respectively. Our method can better recover the natural textures, owing to the effective utilization of abundant self-similarity in videos.

TABLE V
COMPARISON OF THE EFFICIENCY OF STATE-OF-THE-ART VIDEO DENOISING METHODS

	GMACs (G)	Param (M)
ViDeNN [31]	46.67	1.43
TOFlow [33]	11.56	1.51
FastDVDnet [39]	10.55	2.48
EMVD [36]	1.88	0.82
SMOID [29]	27.62	21.88
EDVR [51]	70.88	3.45
RViDeNet [26]	66.16	8.58
Ours	54.86	6.36

The comparison of efficiency is also reported in Table V. For methods that achieve 37 dB or above on our dataset, our method outperforms EDVR [51] by 0.65 dB on CRVD [26] and RViDeNet [26] by 0.35 dB on our dataset, while requiring 77% and 82% of computation cost, respectively. Also, our method requires 28% of parameters compared to SMOID [28]. This indicates the effectiveness of our proposed method.

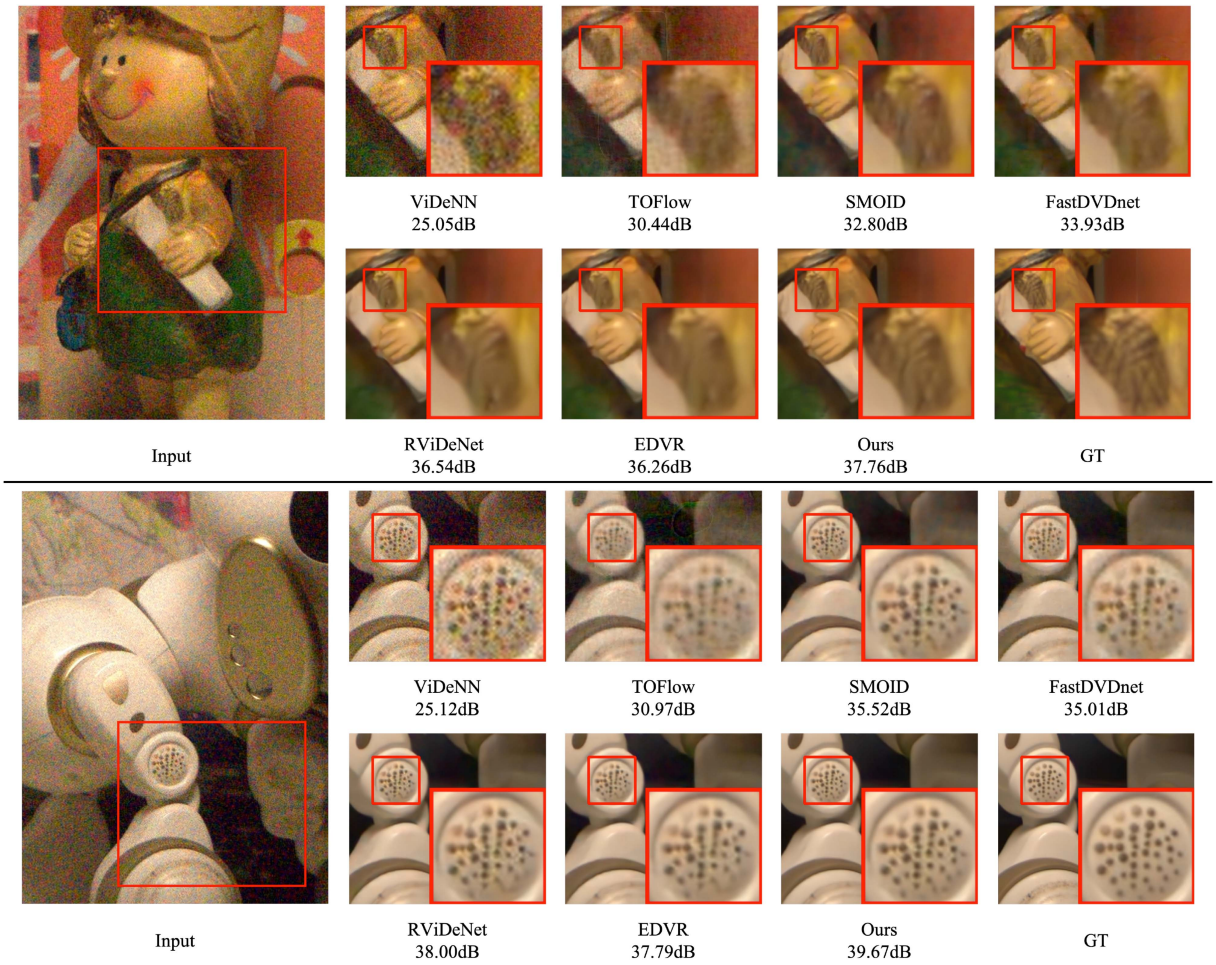


Fig. 10. Visualization of the denoised results on CRVD dataset (zoom in for details). Our method shows the best visual quality.

D. Evaluation on Real-World Videos

To prove that our captured low-light video denoising dataset can be generalized to real-world videos, we train the proposed method with several video enhancement datasets, including SDSD [28], SMID [27], CRVD [26] and our dataset.[2] For qualitative comparison, we shoot raw videos by iPhone 13 Pro Max. We apply denoising on raw videos, then provide sRGB frames for better visualization.

As Fig. 11 shows, the method trained on our dataset can achieve better visual quality, in terms of smoothness in the flat area and the preservation of fine textures. The superiority owes to the realistic motion in our dataset, while SMID only includes static scenes. Also, motion in CRVD is manually created, thus being much simpler. These all decline their performances when tackling real-world videos.

E. Performance on Different Stages of Image Processing Pipeline

Low-light video denoising can be performed at various stages of the image processing pipeline, in which the role these methods play varies. For most cases, the final goal of low-light video denoising is to achieve higher quality in color videos, $i.e.$, sRGB format videos, which can be perceived by human eyes and better enhance the downstream tasks. However, this relation between the final performance on color videos and applied stages has not been fully discussed.

For existing video denoising studies, different applied stages can be categorized into: i) Raw2Raw [26], where the denoising for raw frames can be seen as pre-processing before the image processing pipeline. ii) Raw2RGB [26], [27], [28], where the network itself will learn both the denoising process and the non-linear color transformation from raw to RGB, including white balance, sensor response and color correction matrix. iii) RGB2RGB [31], [33], [39], where denoising is performed in color space after the whole image signal processing procedure.

We adopt the Raw2Raw pipeline in this work. To compare the performance of different applied stages, we re-train the Raw2RGB and RGB2RGB versions of our proposed method, using our low-light realistic motion video dataset and a pre-defined ISP. Here, following previous work [21], we assume that color frames are rendered by a simple ISP, including white balance, color correction and a camera response function.[3] All the used parameters for ISP are directly read from the captured raw files.

[2]SMOID is not included because it has not been released at the time we submit the paper.

[3]We do not perform demosaicking. Instead, we average two green channels in each Bayer pattern block to form the three-channel raw RGB frames.

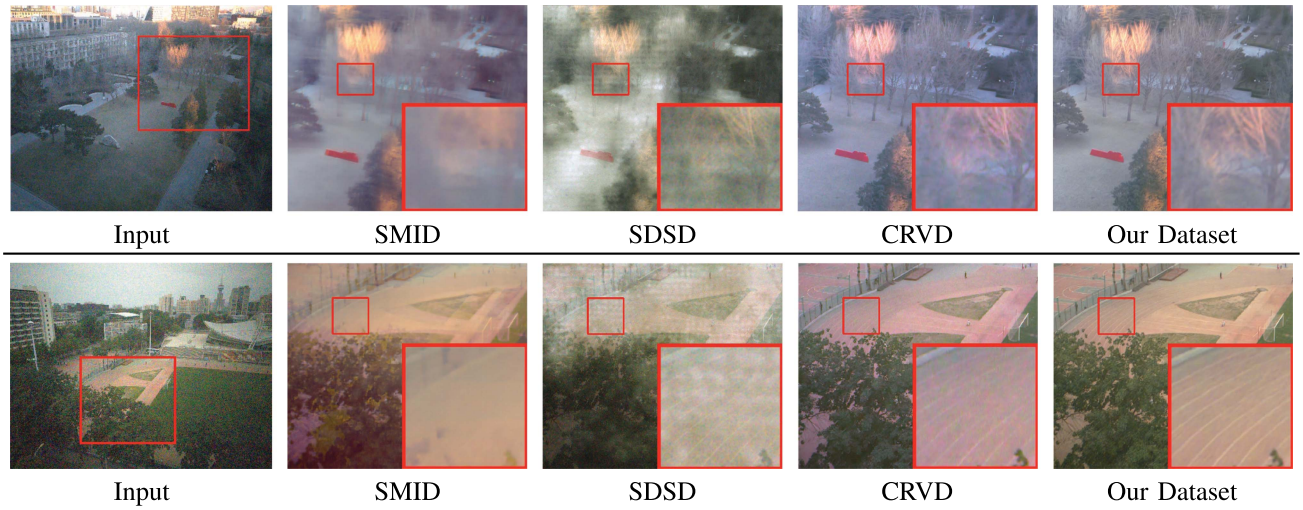


Fig. 11. Recovery results by our method trained on various datasets. Input noisy videos are captured by iPhone 13 Pro Max. The method trained on our dataset shows the best visual quality. We randomly choose one frame for visualization.

TABLE VI
COMPARISON OF DENOISING PERFORMANCE OF OUR METHOD APPLIED ON DIFFERENT IMAGE PROCESSING STAGES

Denoising Stages of the Image Processing Pipeline	PSNR	SSIM
Raw2Raw + ISP	**32.38**	**0.884**
Raw2RGB	32.04	0.883
RGB2RGB	31.06	0.881

We test the videos in our proposed dataset, which go through a pre-defined ISP to generate the RGB format counterparts. The outputs of Raw2Raw method are post-processed by image signal processing for RGB results. All the results are evaluated on RGB color space.

The results are listed in Table VI. Our Raw2Raw pipeline with a pre-defined ISP achieves the best recovery performance on RGB color space videos, and it is slightly better than the Raw2RGB pipeline. Also, both of these two pipelines can outperform the RGB2RGB pipeline. The reason may be that: i) for the Raw2RGB pipeline, the network has to learn both the denoising process and the non-linear transformation, which increases the difficulty of the task. In this case, networks with more complexity may be needed to achieve comparable performance. ii) For the RGB2RGB pipeline, all the input frames go through an 8-bit quantization process. Consequently, there may exist information loss during the conversion from original raw files, especially for low-light circumstances where lower pixel intensities may be more susceptible to quantization.

VI. CONCLUSION

In this paper, we present a high-quality raw video denoising dataset in low-light containing realistic motion. Our dataset is composed of paired noisy/clean frames in raw format, covering various noise levels. Compared with previous video denoising datasets, which either contain fixed and simple motion, or be inferior in ground truth quality, our dataset contains both complex motion and high-quality ground truth. Also, our dataset collecting pipeline requires no extra equipment used in previous datasets, e.g., the co-axis optical system or electric slide rail with the controller. Since it takes much effort to assemble the equipment and temporally align the video sequences, our capture pipeline requires no extra effort, and can be easily reproduced. In this way, video denoising datasets with realistic motion can be easily collected for other cameras and scenes. Besides, we propose a new method for the low-light raw video denoising. To effectively exploit the abundantly existing self-similarities both spatially and temporally, the network utilizes 3D temporal-spatial attention in encoders and decoders, while following the design of the U-net. Extensive results demonstrate the superiority of our network and the value of our dataset.

We will continue to do more research in the future. Our dataset only contains limited types of cameras. We will extend the scale of the dataset by capturing more videos with various cameras under more diverse scenes, since real-world videos can be captured by any kind of camera and in any scene. The dataset will be released to facilitate further research. Besides, like in previous works, our method needs extra information for brightness alignment. It is worth focusing on automatic brightness alignment instead of explicitly providing the low light ratio to the network, and also the network that is capable of handling various kinds of camera inputs. While being more challenging, this is more in line with practical applications. We foresee future works to deal with these challenges, in order to push the low-light video denoising a step forward.

REFERENCES

[1] D. Dussault and P. Hoess, "Noise performance comparison of ICCD with CCD and EMCCD cameras," *Proc. SPIE*, vol. 5563, pp. 195–204, 2004.
[2] A. Kirmani et al., "First-photon imaging," *Science*, vol. 343, no. 6166, pp. 58–61, 2014.
[3] X. Hu, Y. Cai, Z. Liu, H. Wang, and Y. Zhang, "Multi-scale selective feedback network with dual loss for real image denoising," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 729–735.

[4] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2392–2399.

[5] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1712–1722.

[6] G. Varghese and Z. Wang, "Video denoising based on a spatiotemporal gaussian scale mixture model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 7, pp. 1032–1040, Jul. 2010.

[7] L. Guo, O. C. Au, M. Ma, and Z. Liang, "Temporal video denoising based on multihypothesis motion compensation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 10, pp. 1423–1429, Oct. 2007.

[8] J. Dai, O. C. Au, C. Pang, and F. Zou, "Color video denoising based on combined interframe and intercolor prediction," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 23, no. 1, pp. 128–141, Jan. 2013.

[9] H. Liuet al., "Image denoising via low rank regularization exploiting intra and inter patch correlation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 12, pp. 3321–3332, Dec. 2018.

[10] L. Zhang, P. Bao, and X. Wu, "Multiscale LMMSE-based image denoising with optimal wavelet selection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 4, pp. 469–481, Apr. 2005.

[11] K. Zhang, W. Zuo, and L. Zhang, "FFDNet: Toward a fast and flexible solution for CNN-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, Sep. 2018.

[12] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[13] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-D nonlocal spatiotemporal transforms," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 3952–3966, Sep. 2012.

[14] Y. Du, G. Han, Y. Tan, C. Xiao, and S. He, "Blind image denoising via dynamic dual learning," *IEEE Trans. Multimedia*, vol. 23, pp. 2139–2152, 2021.

[15] S. R. S. P. Malladi, S. Ram, and J. J. Rodríguez, "Image denoising using superpixel-based PCA," *IEEE Trans. Multimedia*, vol. 23, pp. 2297–2309, 2021.

[16] R. Ma, S. Li, B. Zhang, and Z. Li, "Towards fast and robust real image denoising with attentive neural network and PID controller," *IEEE Trans. Multimedia*, vol. 24, pp. 2366–2377, 2022.

[17] J. Ma, C. Peng, X. Tian, and J. Jiang, "DBDNet: A deep boosting strategy for image denoising," *IEEE Trans. Multimedia*, vol. 24, pp. 3157–3168, 2022.

[18] H. Chen, Y. Jin, K. Xu, Y. Chen, and C. Zhu, "Multiframe-to-multiframe network for video denoising," *IEEE Trans. Multimedia*, vol. 24, pp. 2164–2178, 2022.

[19] C. Chen, Z. Xiong, X. Tian, Z.-J. Zha, and F. Wu, "Real-world image denoising with deep boosting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 12, pp. 3071–3087, Dec. 2020.

[20] K. Zhanget al., "Plug-and-play image restoration with deep denoiser prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6360–6376, Oct. 2022.

[21] K. Wei, Y. Fu, Y. Zheng, and J. Yang, "Physics-based noise modeling for extreme low-light photography," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 11, pp. 8520–8537, Nov. 2022.

[22] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3291–3300.

[23] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1692–1700.

[24] T. Brookset al., "Unprocessing images for learned raw denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11036–11045.

[25] T. Plotz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1586–1595.

[26] H. Yue, C. Cao, L. Liao, R. Chu, and J. Yang, "Supervised raw video denoising with a benchmark dataset on dynamic scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2301–2310.

[27] C. Chen, Q. Chen, M. N. Do, and V. Koltun, "Seeing motion in the dark," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 3185–3194.

[28] R. Wang et al., "Seeing dynamic scene in the dark: A high-quality video dataset with mechatronic alignment," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9700–9709.

[29] H. Jiang and Y. Zheng, "Learning to see moving objects in the dark," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 7324–7333.

[30] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.- Assist. Interv.*, 2015, pp. 234–241.

[31] M. Claus and J. van Gemert, "ViDeNN: Deep blind video denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 1843–1852.

[32] M. Tassano, J. Delon, and T. Veit, "DVDNet: A fast network for deep video denoising," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 1805–1809.

[33] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, 2019.

[34] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising using separable 4d nonlocal spatiotemporal transforms," *Proc. SPIE*, vol. 7870, pp. 9–19, 2011.

[35] X. Chen, L. Song, and X. Yang, "Deep RNNs for video denoising," *Proc. SPIE*, vol. 9971, pp. 573–582, 2016.

[36] M. Maggioniet al., "Efficient multi-stage video denoising with recurrent spatio-temporal fusion," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3466–3475.

[37] F. Lv, F. Lu, J. Wu, and C. Lim, "MBLLEN: Low-light image/video enhancement using CNNs," in *Proc. Brit. Mach. Vis. Conf.*, 2018, Art. no. 4.

[38] Y. Zeng, Y. Zou, and Y. Fu, "$3D^2Unet$: 3D deformable Unet for low-light video enhancement," in *Proc. Chin. Conf. Pattern Recognit. Comput. Vis.*, 2021, pp. 66–77.

[39] M. Tassano, J. Delon, and T. Veit, "FastDVDnet: Towards real-time deep video denoising without flow estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1354–1363.

[40] Y. Tian, Y. Zhang, Y. Fu, and C. Xu, "TDAN: Temporally-deformable alignment network for video super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3360–3369.

[41] G. V. Research, "Image sensors market analysis," 2016. [Online]. Available: https://www.grandviewresearch.com/industry-analysis/image-sensors-market

[42] K. Wei, Y. Fu, J. Yang, and H. Huang, "A physics-based noise formation model for extreme low-light raw denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2758–2767.

[43] A. Foi, M. Trimeche, V. Katkovnik, and K. Egiazarian, "Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data," *IEEE Trans. Image Process.*, vol. 17, no. 10, pp. 1737–1754, Oct. 2008.

[44] N. Wadhwa et al., "Synthetic depth-of-field with a single-camera mobile phone," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–13, 2018.

[45] X. Jiet al., "Real-world super-resolution via kernel estimation and noise injection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 466–467.

[46] H. Nagahara, S. Kuthirummal, C. Zhou, and S. K. Nayar, "Flexible depth of field photography," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 60–73.

[47] C. Chen, Z. Xiong, X. Tian, Z.-J. Zha, and F. Wu, "Camera lens super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1652–1660.

[48] Y. Zhang, H. Qin, X. Wang, and H. Li, "Rethinking noise synthesis and modeling in raw denoising," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4593–4601.

[49] A. Dosovitskiy et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Representations*, 2021.[Online]. Available: https://openreview.net/forum?id=YicbFdNTTy

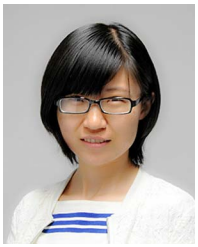
[50] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.

[51] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy, "EDVR: Video restoration with enhanced deformable convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 1954–1963.

[52] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 3–19.

[53] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[54] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable convNets V2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9308–9316.



Ying Fu (Senior Member, IEEE) received the bachelor's degree in electronic engineering from Xidian University, Xi'an, China, in 2009, the master's degree in automation from Tsinghua University, Beijing, China, in 2012, and the Doctoral degree in information science and technology from The University of Tokyo, Tokyo, Japan, in 2015. She is currently a Professor with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China. Her research interests include physics-based vision, image processing, and computational photography.

Zichun Wang received the bachelor's degree in 2021 from the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China, where he is currently working toward the master's degree with the School of Computer Science and Technology. His research interests include computational photography, image processing, and deep learning.

Tao Zhang received the bachelor's degree in 2017 from the School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China, where he is currently working toward the Doctoral degree with the School of Computer Science and Technology. His research interests include deep learning, image processing, and computational photography.

Jun Zhang received the bachelor's, master's, and Doctoral degrees in communications and electronic systems from Beihang University, Beijing, China, in 1987, 1991, and 2001, respectively. He is currently a Professor with the Beijing Institute of Technology, Beijing, where he is also the the Secretary of the Party Committee. His research interests include networked and collaborative air traffic management systems, covering signal processing, integrated and heterogeneous networks, and wireless communications. He is a Member of the Chinese Academy of Engineering, Beijing. He was the recipient of the awards for science and technology in China many times.