Efficient Burst Raw Denoising with Variance Stabilization and Multi-frequency Denoising Network

Dasong Li · Yi Zhang · Ka Lung Law · Xiaogang Wang · Hongwei Qin · Hongsheng Li

Received: 21 September 2021 / Accepted: 7 May 2022 / Published online: 20 June 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
With the growing popularity of smartphones, capturing high-quality images is of vital importance to smartphones. The cameras of smartphones have small apertures and small sensor cells, which lead to the noisy images in low light environment. Denoising based on a burst of multiple frames generally outperforms single frame denoising but with the larger computational cost. In this paper, we propose an efficient yet effective burst denoising system. We adopt a three-stage design: noise prior integration, multi-frame alignment and multi-frame denoising. First, we integrate noise prior by pre-processing raw signals into a variance-stabilization space, which allows using a small-scale network to achieve competitive performance. Second, we observe that it is essential to adopt an explicit alignment for burst denoising, but it is not necessary to integrate an learning-based method to perform multi-frame alignment. Instead, we resort to a conventional and efficient alignment method and combine it with our multi-frame denoising network. At last, we propose a denoising strategy that processes multiple frames sequentially. Sequential denoising avoids filtering a large number of frames by decomposing multiple frames denoising into several efficient sub-network denoising. As for each sub-network, we propose an efficient multi-frequency denoising network to remove noise of different frequencies. Our three-stage design is efficient and shows strong performance on burst denoising. Experiments on synthetic and real raw datasets demonstrate that our method outperforms state-of-the-art methods, with less computational cost. Furthermore, the low complexity and high-quality performance make deployment on smartphones possible.

Keywords Burst denoising · Poisson-Gaussian distribution · Variance stabilization · Sequential denoising · Multi-frequency denoising

1 Introduction
Mobile photography becomes increasingly popular because of the surging number of smartphones worldwide. However, the raw images captured by low-cost sensors of mobile devices usually show heavy noise, especially in low-light environments. The noise augmented in the imaging processing pipeline would dramatically deteriorate the visual quality. To improve the imaging quality, capturing and denoising a burst of multiple raw frames becomes a common choice to remove unpleasant noise. Therefore, burst raw denoising becomes an essential task in mobile photography.

Burst raw denoising aims at reproducing the underlying scene from a burst of noisy raw image frames. Classical
efficient burst denoising methods (Liba et al. 2019; Hasinoff et al. 2016) generally consist of some or all of the following three processing steps, including noise prior integration, multi-frame alignment, and multi-frame denoising. Raw images can be captured with different shooting parameters, such as exposure time and sensor gain, which lead to a wide range of noise levels with unstable variances. Integrating the noise prior helps the methods to be aware of the noise level of the input image so that they can handle different noise levels with different denoising strengths. The multi-frame alignment stage tries to align multiple neighboring frames to the reference frame. The representative methods include block matching Hasinoff et al. (2016) and optical flow-based Liu et al. (2014) methods. The final multi-frame denoising stage aggregates spatial and temporal information to reproduce the clean frame from the multiple frames with consideration of possible alignment failures. Classical fusion operations include collaborative filtering Dabov et al. (2007) and frequency domain fusion Hasinoff et al. (2016).

Recently deep learning-based methods outperform traditional methods by improving the different processing stages. Kernel Prediction Network (KPN) and its variants Mildenhall et al. (2018); Marinc et al. (2019); Xia et al. (2020) have been exploited to conduct multi-frame alignment and denoising jointly. RViDeNet Yue et al. (2020) and BDNet (Liang et al., 2020) improve the performance of multi-frame denoising by adopting Deformable Convolution Dai et al. (2017); Zhu et al. (2019) to achieve explicit alignment. However, less effects are paid to noise prior integration, which hampers the performance of burst raw denoising. Moreover, most learning-based methods Mildenhall et al. (2018); Liang et al. (2020) generally require large memory consumption and high computational cost.

In this paper, we improve the efficiency and effectiveness of burst denoising by improving all three steps:

1.1 Noise Prior Integration

In traditional methods Hasinoff et al. (2016); Liba et al. (2019), the variations of different noise levels of the input image make no difference to the computational complexity as the estimated noise variance is used as the coefficients of Wiener filter Liba et al. (2019); Hasinoff et al. (2016) to modulate the denoising strength. For deep learning-based methods Mildenhall et al. (2018); Zhang et al. (2017), they need to tackle a wide range of noise levels with a single network. The denoising strengths need to be adaptively adjusted by explicitly or implicitly estimating noise variances Mildenhall et al. (2018). The network capacity needs to be large enough to handle the varying noise levels.

We first analyze the key factors causing unstable variances of raw images by modeling CMOS signals. Then we introduce a variance-stabilizing transformation to stabilize the varying variances caused by these factors. The follow-up denoising can be then performed in the variance-stabilization space. The proposed strategy allows using a single lightweight network to handle different noise levels.

1.2 Multi-Frame Alignment

The mainstreaming burst denoising methods utilize kernel prediction to achieve multi-frame alignment and multi-frame denoising jointly. Learning-based alignment Wang et al. (2019); Liang et al. (2020) with Deformable Convolution Dai et al. (2017); Zhu et al. (2019) also show improved performance in burst denoising Liang et al. (2020) or video denoising Yue et al. (2020). In our experiments, it is shown that explicit alignment is essential to handle the global misalignment among multiple frames. Furthermore, we find that learning-based alignment do not show competitive performance against conventional alignment but brings much more computational cost. Therefore, we adopt a conventional alignment and combine it with learning-based multi-frame denoising.

Specifically, we adopt a coarse-to-fine alignment algorithm and process coarse and fine scales hierarchically. At the coarse scales, we use homography flow to achieve global alignment. At the fine scales, we adopt block-matching to refine the alignment results. To make our denoising network adaptive to alignment results, our follow-up multi-frame denoising network is trained on multiple frames aligned by our alignment. The proposedalignment strategy achieves competitive performance compared with learning-based alignment but has less computational cost.

Multi-frame Denoising. Processing a burst of frames simultaneously requires a large network, which increases the computational cost and memory significantly. Therefore, it is difficult to deploy deep learning-based multi-frame denoising Mildenhall et al. (2018); Wang et al. (2019) algorithms on mobile devices. We propose a sequential denoising strategy to process multiple frames in sequential manners. The proposed denoising network first performs spatial denoising on the reference frame to produce an intermediate denoising result. Then, each neighboring frame is sequentially input into the network to gradually refine the intermediate result to integrate the multiple neighboring frames’ temporal information. The denoising system adopts a novel and efficient multi-frequency network architecture to take advantages of the neural network’s capability on eliminating high-frequency noise and to perform denoise at different frequencies.

We evaluate our proposed method on the KPN dataset Mildenhall et al. (2018) and CRVD raw dataset Yue et al. (2020). Extensive experiments demonstrate the effectiveness of our proposed burst denoising approach. In summary, our proposed method has the following contributions:
We propose an efficient burst denoising system by improving the three stages of the burst denoising framework, noise prior integration, multi-frame alignment, and multi-frame denoising.

We analyze the variation of CMOS raw images and propose a comprehensive variance stabilization technique for learning-based denoising, which demonstrates its superior performance on raw burst denoising.

We propose a multi-frame framework to integrate multiple frames’ temporal information sequentially and a multi-frequency denoising network to handle noise of different frequencies effectively.

2 Related Work

Burst raw denoising involves a complex processing pipeline that aggregates spatial and temporal information from multiple frames and should be capable of handling a wide range of noise levels. The related work about noise prior integration, multi-frame alignment, and multi-frame denoising are discussed as follows.

2.1 Noise Prior

Given an observed intensity $x$ and its underlying clean intensity $x^*$, we have the following relation:

$$x = x^* + n,$$

where $n$ is the noise. Additive white Gaussian noise is widely used in previous works Buades et al. (2005); Dabov et al. (2007). However, Gaussian distribution cannot represent the signal-dependent photon noise in CMOS sensors. To approximate real noise of CMOS sensors, multiple types of noise are explored for noise modeling, such as Poisson-Gaussian distribution Foi et al. (2008); Liu et al. (2014), heteroscedastic Gaussian distribution Healey and Kondepudy (1994) and more complicated modelings Wei et al. (2020); Zhang et al. (2021).

2.1.1 Prior-Based Traditional Methods

Representative traditional denoising methods usually are based on different priors. This category of methods include anisotropic diffusion, total variation denoising Rudin et al. (1992), wavelet domain denoising Portilla et al. (2003), sparse coding Mairal et al. (2009), image self-similarity Buades et al. (2005); Dabov et al. (2007) and etc. Total variation denoising Rudin et al. (1992) uses the statistical characteristics of images to remove noise. Sparsity coding Mairal et al. (2009) enforces sparsity in dictionary learning methods to learn over-complete dictionaries from clean images. As an important prior, image self-similarity shows the excellent performance against other methods. NLM Buades et al. (2005) and BM3d Dabov et al. (2007) explore the presence of similar features or patterns in a non-local manner. Although above models are limited due to the assumptions on the prior of spatially invariant noisy or clean images, they can be applied to real raw data when a generalized anscombe transformation Starck et al. (1998) is applied.

2.1.2 Learning-Based Noise Prior Integration

With the development of convolution neural networks, end-to-end CNNs have achieve great success in handling real-world image denoising. DnCNN proposes residual learning for Gaussian denoising. A series of methods Zhang et al. (2018); Mildenhall et al. (2018); Wang et al. (2020) proposed to integrate the noise prior of raw image into deep neural networks. Based on Poisson-Gaussian distribution or Gaussian Mixture distribution, Zhang et al. (2018); Mildenhall et al. (2018) approximate pixel-wise noise variance to deal with the different noise levels. PMRID Wang et al. (2020) proposes a linear transform to handle different noise variances in different sensor gains. In this work, we propose a novel variance stabilization technique and perform learning-based denoising in the space transformed with stable variance.

2.2 Multi-Frame Alignment

The most long-standing method for multi-frame alignment is based on optical flow Horn and Schunck (1981); Bruce (1981); Baker et al. (March 2011); Menze and Geiger (2015); Weinzaepfel et al. (2013). Hierarchical structures Hasinoff et al. (2016); Liu et al. (2014) are explored to improve the efficiency of conventional alignment. Many learning-based alignment methods have been proposed in video-related tasks as the substitute for the conventional alignment. Learning-based optical flow Xue et al. (2019) and deformable convolution Wang et al. (2019) have been exploited for video interpolation, video super-resolution and video denoising. RViDeNet Yue et al. (2020) proposed pre-denoising modules to denoise each frame before the deformable convolution alignment, which, however, increases the computational burden significantly.

2.3 Multi-Frame Denoising

The multi-frame denoising aims at merging multiple frames with alignment errors to reproduce clean image. Collaborative filtering Maggioni et al. (2012, 2011) and frequency domain fusion (Hasinoff et al. 2016; Liba et al. 2019) are representative approaches. The mainstreaming learning-based methods implement multi-frame denoising without explicit alignment. KPN Mildenhall et al. (2018) proposes kernel...
prediction network to jointly conduct multi-frame alignment and denoising. MPKN (Marinc et al., 2019) extends single kernel prediction to multiple kernels prediction. BPN (Xia et al., 2020) proposes basis prediction networks for larger kernels. FastDVDNet Tassano et al. (2020) proposes two-step cascaded methods for efficient video denoising without alignment modules.

3 Methodology

Our raw burst denoising system adopts a three-stage design, which includes noise prior integration, multi-frame alignment, and multi-frame denoising. To produce a clean frame, N raw frames in a burst are first transformed into the noise variance stabilized space via noise prior integration and then aligned by multi-frame alignment. The transformed and aligned multiple frames are input into the multi-frame denoising network to produce the clean frame corresponding to the key frame.

3.1 Noise Prior Integration

Burst denoising in real-world scenarios needs to handle a wide range of noise levels. Most previous methods Mildenhall et al. (2018); Zhang et al. (2018) integrate noise prior by using estimated per-pixel variance as the extra input. However, these networks still need to remove the noise with varying variance. To improve both performance and efficiency, we would like to reduce the learning complexity of the network by eliminating the unstable variances. We first discuss the key factors causing unstable noise variances of raw intensity values via formulating the noise of raw CMOS signals. To eliminate the unstable variances of different noise levels, we use the variance-stabilizing techniques to eliminate the unstable noise variance caused by these factors. Further analysis is provided to reveal that stabilizing noise variances allows using lightweight networks to achieve effective denoising.

3.1.1 Noise Modeling of CMOS Signals

The raw data of CMOS signals contains two primary sources of noise: shot noise and read noise. Shot noise is produced as a Poisson process with a variance equal to signal level. The read noise, an approximately Gaussian process, is caused by the sensor readout effects. The raw data is usually modeled as Poisson-Gaussian distribution Foi et al. (2008); Liu et al. (2014):

$$ x \sim \sigma_s \mathcal{P} \left( \frac{x^*}{\sigma_s} \right) + \mathcal{N}(0, \sigma_r^2), \quad (2) $$

where $x$ is noisy measurement of the true intensity $x^*$. Two noise parameters $\sigma_s$ and $\sigma_r$ change across different images as the sensor gain (ISO) changes.

The variance of the noisy measurement $x$ is formulated as

$$ \text{Var}[x] = \sigma_s x^* + \sigma_r^2. \quad (3) $$

For a fixed sensor, the sensor gain (ISO) is the only factor affecting $\sigma_s$ and $\sigma_r$. The connection between the sensor gain and noise parameters $\sigma_s$, $\sigma_r$ are shown in Appendix A. Therefore the variance is affected by the sensor gain and underlying intensity $x^*$. When the sensor gain increases, the variance of each pixel at one image increases. When the sensor gain is fixed, different brightness shows different variances in the image.

3.1.2 Variance Stabilization

We propose to transform the pixel values into a new space to eliminate the varying variances.

Firstly, we eliminate the unstable variance caused by the sensor gain. The observed intensity $x$ and the underlying true intensity $\hat{x}$ are scaled by $\frac{1}{\sigma_s}$,

$$ \hat{x} = \frac{x}{\sigma_s}, \quad \hat{x}^* = \frac{x^*}{\sigma_s}. \quad (4) $$

With the above transformation, $\hat{x}$ becomes a Poisson variable corrupted by additive Gaussian noise of variance $\hat{\sigma}^2 = \frac{\sigma_r^2}{\sigma_s^2}$.

$$ \hat{x} = \mathcal{P}(\hat{x}^*) + \mathcal{N}(0, \hat{\sigma}^2). \quad (5) $$

The variance of $\hat{x}$ thus becomes

$$ \text{Var}[\hat{x}] = \hat{x}^* + \hat{\sigma}^2. \quad (6) $$

Then we need to eliminate the unstable variance caused by the signal-dependent property of the Poisson distribution, which indicates that different intensities within the same image have varying variances.

We generalize the Freeman-Tukey transformation Freeman and Tukey (1950) to transform the Poisson-Gaussian distribution (Eq. (5)) to

$$ y = \sqrt{\hat{x}} + \hat{\sigma}^2 + \sqrt{\hat{x} + 1} + \hat{\sigma}^2, \quad (7) $$

which is a Gaussian distribution with a constant variance at different intensities, i.e., the transformed intensity values is contaminated by Gaussian noise with unit variance. For
details of the interpretation in the transformed space, please refer to Mäkitalo and Foi (2012); Freeman and Tukey (1950). Then the following denoising network perform denoising on this space.

3.1.3 Inverse Transformation

We perform the algebraic inverse of Eq. (4) and Eq. (7) to map the denoising output back into the raw linear space,

$$\text{Inv}(y) = \left(\frac{y^4 - 2y^2 + 1}{4y^2} - \sigma_s^2\right)\sigma_s. \quad (8)$$

This pixel transformation facilitates the training and generality of the follow-up denoising network as the signal-dependent components of the noise are eliminated in the transformed intensity space. The follow-up denoising network can be more effectively trained based on the transformed intensities with signal-independent Gaussian noise.

3.1.4 Analysis

We here discuss related methods on handling the problem of varying noise levels in denoising for comparison. First, most learning-based denoising methods Zhang et al. (2018); Mildenhall et al. (2018) estimate per-pixel variance map as

$$\text{Var}[x] = \sigma_s \text{max}(x, 0) + \sigma_r^2. \quad (9)$$

However, the ideal noise estimation should be

$$\text{Var}[x] = \sigma_s \alpha x^* + \sigma_r^2. \quad (10)$$

Since the true intensity $x^*$ cannot be observed, using observed intensity $x$ to replace $x^*$ introduces the errors in noise estimation. Furthermore, these denoising networks still need to handle varying noise levels. In contrast, applying variance stabilization would avoid the errors of noise estimation and allows the denoising network to handle stable variance.

To stabilize the variance from sensor gain, PMRID Wang et al. (2020) proposed a $k$-sigma transform

$$f_k(x) = \frac{x}{\sigma_s} + \frac{\sigma_r^2}{\sigma_s^2} \quad (11)$$

to transform the images into an ISO-invariant space. This transformation only eliminates the unstable variance caused by the sensor gain but neglects unstable variance of the Poisson-Gaussian distribution. PMRID Wang et al. (2020) can be considered as only the first step of our proposed stabilization.

For stabilization of the Poisson-Gaussian distribution, Generalized Anscombe Transformation (GAT) Starck et al. (1998) extended Anscombe transformation Anscombe (1948) to stabilize the variance of Poisson-Gaussian distribution. In contrast, our proposed method extends Tukey-Freeman Transformation Freeman and Tukey (1950) for stabilization of the Poisson-Gaussian distribution with simple first-order approximation Starck et al. (1998) provided in Appendix A. It is observed in our experiments that our variance-stabilization technique for learning-based denoising shows better performance than GAT Starck et al. (1998).

3.2 Multi-Frame Alignment

Given the multiple frames for denoising reference frame, it is natural to utilize frame alignment methods Liu et al. (2014); Hasinoff et al. (2016); Liba et al. (2019) to align the frames before multi-frame denoising to optimally utilize neighboring frames’ contextual information. In video reconstruction tasks, learning-based optical flow Xue et al. (2019) and Deformable Convolution Wang et al. (2019) have been explored to perform multi-frame alignment as a substitute for the conventional alignment methods. However, it is not practical to deploy learning-based alignment on mobile processors because of its large amount of computational cost and running time. We decide to buck the trend and resort to conventional alignment methods, to achieve multi-frame alignment in an efficient manner. It is observed in our experiments (Sect. 4.5) that conventional alignment and learning-based alignment actually show comparable denoising performance with learning-based denoising networks.

3.2.1 Coarse-to-Fine Alignment

Our alignment is based on the combination of block-matching Hasinoff et al. (2016) and homography flow Liu et al. (2014). We build a four-scale alignment pyramid to estimate the motions between pairs of frames in a coarse-to-fine manner. We use homography flow at the top two levels and use block-matching for the bottom pyramid levels. The homography flow Liu et al. (2014) is used to model accurate camera motions. The two-layer homography flow acts as cascaded global alignment, while the block matching acts as refining local alignment. The cascade design makes our alignment both robust and fast under extreme low light environments.

The detailed steps of pyramid alignment is conducted as follows:

- At the top pyramid level, we estimate global homography between the reference frame and other frames. All pixels share the same flow vector. The homography is calculated based on corner keypoints detected by FAST Rosten and
Fig. 1 The overview of the proposed sequential denoising network. We propose a sequential denoising strategy to process the multiple frames. \( N \) frames of a burst are considered as a reference frame and \( N - 1 \) alternate frames. We adopt multi-frequency denoising network \( S_i \) as the backbone for efficient denoising. At the first stage, we perform spatial denoising \( S_0 \) on the reference frame to obtain intermediate results 0. Then we refine the intermediate result via integrating the information from \( N - 1 \) alternate frames. At each refinement stage, the multi-frequency denoising network \( S_i \) takes intermediate result and an alternate frame as inputs and refines the intermediate result.

Drummond (October 2005, M) and feature matching with BRIEF descriptors Calonder et al. (2010).

- At the second level, the whole plane is split into 4 blocks. Then we estimate the homography flow for each block separately. The BRIEF descriptors are extracted again from each block. When a block has insufficient matched features, the estimated homography would be substituted by the homography estimated from the previous level.
- At the two bottom levels, we use tile-based search to align frames. The pre-aligned images are split into 16 × 16 tiles. We perform \( L_1 \) -distance search for each tile within ±2 pixels neighborhood. In extreme low light case, tile size would be set to 32 × 32 to reduce the impact of noises. We accelerate the \( L_1 \) search on ARM Neon Intrinsics [1].

After the above alignment, there still remains some misalignments caused by the following reasons. (1) The large displacements would cause inevitable misalignments. (2) Pixels in one tile are required to share the same motion vector, which causes the unsmoothed translations between neighboring tiles. (3) We remove the operation of sub-pixel translation in Hasinoff et al. (2016) to avoid interpolation. To handle these misalignments, we train our follow-up multi-frame denoising network on images aligned by our proposed alignment method.

### 3.3 Multi-Frame Denoising

After the multiple frames in a burst are aligned with the above stage, the multi-frame denoising stage needs to aggregate temporal information from the multiple frames to produce one clean frame. However, processing a large number of frames Mildenhall et al. (2018); Tassano et al. (2020); Yue et al. (2020) simultaneously needs to adopt heavy networks as the relations between the too many frames might be challenging to model. To mitigate the need of heavy networks, we process multiple frames sequentially by a series of efficient networks. All networks shares the multi-frequency denoising architecture with different parameters. The overview of our proposed multi-frame denoising network is shown in Fig. 1. In Sect. 3.3.1, we introduce our sequential denoising strategy. In Sect. 3.3.4, we present the proposed multi-frequency denoising network (Fig. 2).

#### 3.3.1 Sequential denoising

For the input \( N \) frames, one of them is selected as the reference frame and the others \( N - 1 \) are the alternate frames. The denoising network \( S \) consists of \( N \) sub-networks \( S_0, S_1, \ldots, S_{N-1} \) that process these frames sequentially.

#### 3.3.2 Stage-1: Single-Frame Denoising of the Reference Frame

The multi-frame denoising network is trained to produce one clean image with the same content of the reference frame. Intuitively, the reference frame generally makes more contributions than alternate frames in the input clip for multi-frame denoising. We process the reference frame separately as the first stage of denoising. Our first sub-network \( S_0 \) performs single-frame denoising on the reference frame to produce an intermediate result \( I_0 \), which contains the same content with the noisy reference frame but has sharper edges and less noise. Performing the single-frame pre-denoising on the reference frame as the first step offers the following benefits: (1) Processing the reference frame separately emphasizes the priority of reference frame over other alternate frames and generates an intermediate result. It is observed in our experiments that refining the intermediate result \( I_0 \) shows better performance than directly refining noisy reference
Fig. 2  The overview of multi-frequency denoising network. The noises of different frequencies are separately handled by the multi-frequency denoising network. The network adopts three scales to handle different frequencies. The input of three scales are $m_2, m_1, m_0$, where $m_2$ and $m_1$ are downsampling versions of $m_0$. The output $o_2, o_1, o_0$ of three scales contain noises of different structures. At last, Multi-frequency Aggregation takes and aggregates $o_0, o_1, o_2$ to remove noise of all frequencies and preserve as many details as possible. The input $m_0, m_1, m_2$ and output $o_1, o_2, o_3$ are in raw Bayer pattern. For visualization, we utilize the post-process pipeline in RViDeNet Yue et al. (2020) to process them into the sRGB space.

3.3.3 Stage-2: Multi-Frame Denoising of the Reference Frame

Even with the alignment, the alternate frames might still show various degrees of misalignment to the reference frame caused by occlusions, blurring and sub-pixel translations. Therefore, different alternate frames should have different contributions for denoising the reference frame. To avoid too much extra computational cost, our system processes $N - 1$ alternate frames sequentially. The intermediate result $I_0$, produced by the single-frame denoising of the reference frame, is refined sequentially by the $N - 1$ separate sub-networks, $S_1, S_2, \ldots, S_{N-1}$. Specifically, $S_i$ takes the intermediate result $I_{i-1}$ from the previous sub-network and the $i$-th group of alternate frames as inputs, and generate the refined intermediate result $I_i$. The intermediate result $I_{N-1}$ of the last sub-network $S_{N-1}$ generates the final denoising output.

This proposed sequential denoising strategy is quite efficient as it avoids handling all frames jointly and only processes one alternate frame at a time. A series of lightweight sub-networks can be adopted. But the proposed method can still incorporate all information of the neighboring frames to denoise the reference frame gradually.

3.3.4 Multi-Frequency Denoising Network

Each of the denoising sub-network $S_1, \ldots, S_{N-1}$ takes the intermediate result and one alternate frame as input and produces the clean frame corresponding to the reference frame. As observed by Vogels et al. (2018), denoising neural networks are typically good at removing high-frequency noise but have more difficulties on handling low-frequency noise. Therefore, we decompose denoising of whole images into denoising on multiple frequencies. Our sub-network consists of a multi-frequency denoising and a multi-frequency aggregation module.

3.3.5 Multi-Frequency Denoising

We first perform bilinear downampling on the input frames by a factor of 2 for two times to obtain three-scale image pyramids $\{m^0, m^1, m^2\}$. Noise of the same pattern would correspond to different frequencies when downsampled to different scales. For instance, the low-frequency noise would be of high-frequency if the image is downsampled. Then we perform denoising at the three scales to remove noise of different frequencies.

At scale $i$, the denoising sub-network $F_i$, containing a multi-frame encoder $E_i$ and a denoising decoder $D_i$, predicts the residual noise for the image $m^i$. The multi-frame
encoder $E^i$ is implemented by three-layer Convolution neural network. The denoising decoder $D^i$ utilize four residual blocks He et al. (2016). The intermediate denoised images $o^i$ is obtained as

$$o^i = D^i(E^i(m^i)) + m^i. \quad (12)$$

Inspired by previous image restoration networks (Gu et al., 2019; Nah et al., 2017), we further propagate the intermediate result $o_i$ of lower frequency to the decoder $D^{i-1}$ for handling higher-frequency noise. The intermediate denoised images $o_i$ are calculated as

$$o^2 = D^2(E^2(m^2)) + m^2, \quad (13)$$
$$o^1 = D^1(E^1(m^1), o^2) + m^1, \quad (14)$$
$$o^0 = D^0(E^0(m^0), o^1) + m^0. \quad (15)$$

In our network, $D^i$ takes the encoder feature $E^i(m^i)$ and the intermediate denoising result $o^{i+1}$ from the lower frequency as inputs and generates the intermediate result at scale $i$. Intuitively, $D^2$ works on the smallest-scale image and the low-frequency noise in the original image is mostly removed. $D^1$ takes the intermediate result from $D^2$ and further conducts denoising in a higher range of frequency, and $D^3$ repeats this procedure to work on noise of even higher frequency. In this way, noise of different frequencies are efficiently handled by different sub-networks.

### 3.3.6 Multi-Frequency Aggregation

After performing multi-frequency denoising, we design a lightweight multi-frequency aggregation module to combine the denoising results $o^0$, $o^1$, $o^2$ of the three scales to generate the final output. To capture low frequency noise of $o^0$ and $o^1$,
the noise residuals are calculated
\[ n^1 = \downarrow (o^0) - o^1, \quad n^2 = \downarrow (o^1) - o^2. \] (16)

Intuitively, \( o^0 \) has its high-frequency noise at scale 0 removed. To further capture its low-frequency noise, it is downsampled to scale 0 as \( \downarrow (o^0) \), so that its remaining low-frequency noise can be converted to high-frequency noise at scale 1. Since \( o^1 \) has the high-frequency noise at scale 1 removed, the residual \( n^1 = \downarrow (o^0) - o^1 \) highlights the lower-frequency noise in \( o^0 \). Similarly, \( n^2 \) captures even lower-frequency noise of \( o^0 \) than \( n^1 \). In Fig. 2, we illustrate the gradually removed noise from high frequency to low frequency. The captured lower-frequency noise \( n^1 \) and \( n^2 \) can then be removed at scale 0 to obtain the refined output image \( I \),
\[ I = o^0 - \uparrow (n^1) - \uparrow (\uparrow (n^2)). \] (17)

Since \( n^1 \) and \( n^2 \) have smaller size than \( o^0 \), they are upsampled by a factor of 2 once and twice respectively to match the size of \( o^0 \). The proposed denoising network conducts denoising at different frequencies and achieves optimal performance in a multi-scale manner.

### 3.3.7 Loss Function

We optimize our denoising network \( S \) in the space of variance stabilization. Based on Eq. (7), the network input \( y \) is in the space of variance stabilization and the denoising output is denoted as \( \hat{y} = S(y) \). The ground truth \( y^* \) in this space is obtained as
\[ y^* = \sqrt{\hat{x}^* + \sigma^2} + \frac{\hat{x}^* + 1 + \sigma^2}{\sqrt{\sigma^2}}, \] (18)

where \( \hat{x}^* \) is obtained via Eq. (4). We use the average \( \mathcal{L}_1 \) distance and gradient loss Mildenhall et al. (2018) as the main loss term. Our loss function can be formulated as
\[ \mathcal{L}_r = \mathcal{L}_1(y^*, \hat{y}) + w_1 \mathcal{L}_1(\nabla y^*, \nabla \hat{y}), \] (19)

where \( \nabla \) is the finite difference operator that convolves its input with [-1, 1] and [-1, 1]T, and \( w_1 \) is set to 0.5 in our experiments.

When we have an ISP to process the raw output to sRGB domain for assessment, we add the \( \mathcal{L}_1 \) distance in the sRGB domain as Yue et al. (2020) does. The loss function then becomes
\[ \mathcal{L}_r = \mathcal{L}_1(y^*, \hat{y}) + w_1 \mathcal{L}_1(\nabla y^*, \nabla \hat{y}) + w_2 \mathcal{L}_1(\text{ISP}(\text{Inv}(y^*)), \text{ISP}(\text{Inv}(\hat{y}))), \] (20)

where \( \text{Inv}(y^*) \) and \( \text{Inv}(\hat{y}) \) follows Eq. (8), and \( w_2 \) is set to 0.5.

### 4 Experiments

We first evaluate the overall performance of our proposed method on the KPN dataset Mildenhall et al. (2018) and raw video benchmarks CRVD dataset Yue et al. (2020). We compare our method against state-of-the-art burst and video denoising methods, including VBM4D Maggioni et al. (2011) FastDVDNet Tassano et al. (2020), KPN Mildenhall et al. (2018) and RViDeNet Yue et al. (2020).

To evaluate specific designs of the three modules, we conduct ablation study to investigate the influence of each module. Finally, we present our methods’s actual deployment and inference speed on smartphones on a Snapdragon 888 processor.

#### 4.1 Evaluation Datasets

Raw videos are captured with different sensors and different sensor gains, which lead to different noise strengths and different types of motions. The datasets we chose contain videos with such rich variations.

##### 4.1.1 KPN Synthetic Dataset

Mildenhall et al. (2018) consists of 73 raw sequences with four different noise levels. Each noise level corresponds to a specific sensor gain. The relation between gains and the corresponding noise parameters \((\sigma_s, \sigma_r)\) is listed as follows: \(1 \rightarrow (2.7^{-3}, 6.8^{-3})\), \(2 \rightarrow (6.2^{-3}, 1.5^{-2})\), \(4 \rightarrow (1.4^{-2}, 3.6^{-2})\), \(8 \rightarrow (3.3^{-2}, 8.3^{-2})\). Each sequence contains 8 grayscale raw frames. The misalignment is synthesized in the range \([2, 16]\) pixels between 8 burst frames.

##### 4.1.2 CRVD Dataset

Yue et al. (2020) consists of real raw videos of 12 scenes captured by a SONY IMX385 sensor. Videos in the first 6 scenes are taken as the training set and videos in the remaining 6 scenes are used as the evaluation set. For each scene, five videos of 5 different ISOs ranging from 1,600 to 25,600 (corresponding gain from 1 to 16) are captured. The frames of each video only contain object motions without any camera motions. The ISOs correspond to the noise parameters \((\sigma_s, \sigma_r)\): \(1600 \rightarrow (8.6^{-4}, 8.4^{-4})\), \(3200 \rightarrow (1.7^{-3}, 1.5^{-3})\), \(6400 \rightarrow (3.3^{-3}, 2.8^{-3})\), \(12800 \rightarrow (6.5^{-3}, 5.4^{-3})\), \(25600 \rightarrow (1.3^{-2}, 1.0^{-2})\). We take clips of \( N = 3 \) and \( N = 5 \) frames as inputs to our method.
4.1.3 HDR + Dataset

Hasinoff et al. (2016) consists of 3,640 bursts stored in DNG format [2]. The bursts are captured by a variety of Android mobile cameras (Nexus 5/6/5X/6P, Pixel, Pixel XL). The maximum number of burst frames is 10 and the maximum exposure time is 100ms. The noise parameters can also be found in DNG format. Since the dataset cannot provide ground truth for quantitative evaluation, we perform qualitative evaluation on the bursts containing 8 frames captured in extreme low light scenes.

4.2 Training

We train our method on the CRVD dataset and KPN synthetic dataset.

4.2.1 Training for KPN Dataset

At first, we perform unprocessing Brooks et al. (2019) on sRGB images from the Open Images dataset Krasin et al. (2017) to obtain synthetic raw images. The three channels in each sRGB image are averaged to produce a single-channel image. The single-channel image are transformed into the raw linear space. Then we synthesize motion Mildenhall et al. (2018) on the single-channel images to simulate a clip of 8 frames. The mis-alignments between the reference frame and alternate frames are uniformly sampled in 2 to 16 pixels. When synthesizing raw noise, the sensor gain for each sequence is randomly sampled from $[1, 16]$, which corresponds to ISO from 1,600 to 25,600. We can add Poisson-Gaussian noise to the frames according to the corresponding $\sigma_s, \sigma_r$.

For finetuning, CRVD dataset (scene 1-6) with raw videos is used, where we can obtain pairs of noisy input and clean ground truth. Clips of three consecutive noisy frames are randomly selected as input and ground truth are the clean frames corresponding to reference noisy frames.

4.2.2 Training for CRVD dataset

Following the training setup of RViDeNet Yue et al. (2020), we train denoising models on SRVD dataset and finetune them on CRVD dataset (scene 1-6). Training set creation Since SRVD dataset is an sRGB video dataset, we perform the following operations to create the training pairs for raw video denoising. To synthesize raw clean videos, three consecutive sRGB frames are randomly selected from SRVD videos and unprocessed into RGBG raw frames with the camera parameters (white balance, color correction matrix and Gamma factor provided in CRVD dataset).

To construct training pairs, we add Poisson-Gaussian noise to synthesize three noisy frames from clean frames. The sensor gain is randomly sampled from $[1, 16]$, which corresponds to ISO from 1,600 to 25,600. We can add Poisson-Gaussian noise to the frames according to the corresponding $\sigma_s, \sigma_r$.

For finetuning, CRVD dataset (scene 1-6) with raw videos is used, where we can obtain pairs of noisy input and clean ground truth. Clips of three consecutive noisy frames are randomly selected as input and ground truth are the clean frames corresponding to reference noisy frames. Training settings For our proposed method, we stabilize and align the three noisy frames before feeding them into the proposed denoising network. Then the multi-frame denoising network produces the clean outputs in the variance-stabilization space, which are then transformed back into the raw linear space. The loss function for training adopts Eq. (19). The loss function of finetuning adopts Eq. (20). All the networks are trained with learning rate $10^{-4}$ for 85,000 iterations and finetuned with learning rate $10^{-5}$ for 30,000 iterations. The proposed network is implemented in PyTorch Paszke et al. (2019) and trained with NVIDIA 1080TI GPUs.

4.3 Evaluation

We evaluate our method and compare it with state-of-the-art multi-frame denoising methods, including V-BM4d Maggioni et al. (2011), FastDVDNet Tassano et al. (2020), RViDeNet Yue et al. (2020), KPN Mildenhall et al. (2018) and BPN Xia et al. (2020), for evaluation on the two datasets.

| Method          | Gain $\propto 1$ | Gain $\propto 2$ | Gain $\propto 4$ | Gain $\propto 8$ | Average |
|-----------------|------------------|------------------|------------------|------------------|---------|
| Noisy ref.      | 28.70            | 24.19            | 19.80            | 15.76            | 22.11   |
| V-BM4D          | 34.60            | 31.89            | 29.20            | 26.52            | 30.55   |
| KPN Mildenhall  | 36.47            | 33.93            | 31.19            | 27.97            | 32.39   |
| MKPN            | 36.88            | 34.22            | 31.45            | 28.52            | 32.77   |
| BPN Xia et al.  | 38.18            | 35.42            | 32.54            | 29.45            | 33.90   |
| Ours            | **39.39**        | **36.52**        | **33.47**        | **30.20**        | **34.90** |

Table 1 The PSNR results of different burst denoising methods on KPN dataset.
Table 2 lists the average PSNR and SSIM of raw domain and sRGB domain for video scenes 6-11. When we use only $N = 3$ frames for denoising, it can be observed that our method outperforms the compared denoising methods. Compared with the state-of-the-art RViDeNet Yue et al. (2020), our improvement is 0.35dB PSNR in raw domain and 0.46dB PSNR in sRGB domain. When the network takes $N = 5$ frames as inputs, our methods still achieve the best performance. We visualize the denoising results of outdoor scenes in Fig. 4. The proposed model generates better pleasing details in both static and dynamic regions.

4.3.3 Qualitative Evaluation on HDR+ Dataset

For evaluation on HDR+ dataset Hasinoff et al. (2016), we train the color version of our denoising network. The training settings are similar to training for KPN synthetic dataset as described in Sect. 4.2.1. We compare our method with HDR+ Hasinoff et al. (2016) and VLL Liba et al. (2019) as shown Fig. 5. These two images (from Nexus 6p) are captured with ISO 5760 and 100ms exposure time. Since the post-processing pipeline of HDR+ Hasinoff et al. (2016) is not released to the public, we utilize the post-processing pipeline.
in RViDeNet (Yue et al., 2020) to transform raw outputs into the sRGB space. Therefore, there exists inevitable color shift between our method and HDR+. Our method generally produces less noisy than compared methods. More details on edges and texture regions are recovered by our methods.

4.4 Computational Expense

We first report the number of floating point operations (FLOPs) required in our denoising network and BPN Xia et al. (2020) in Table 3. It is observed that our denoising network requires fewer FLOPS than BPN. Then we
Fig. 5  Burst denoising results on HDR+ dataset Hasinoff et al. (2016). Our method produces better images on extreme low light scenes with more details on edges and texture regions.
evaluate our method and BPN Xia et al. (2020) on the smartphone with the Snapdragon 888 processor. Snapdragon 888 processor contains one Qualcomm Kryo CPU and one Qualcomm Adreno 660 GPU. We implement the variance stabilization and alignment on CPU processor. These two algorithms are accelerated by ARM Neon Intrinsics (2014). As for 8 images with resolution 1024 $\times$ 768, running time of variance stabilization is 20 ms and running time of alignment is 100 ms. Then we utilize the PPLNN platform [4] to measure the running time of denoising networks on Snapdragon 888 processor. Because of the limited memory on smartphones, 1024 $\times$ 768 images are divided into 48 128 $\times$ 128 non-overlapping patches, which are processed separately. To accelerate execution on smartphones, we run our denoising network and BPN on both CPU and GPU processors. The per-patch and overall running times on CPU and GPU are listed in Table 3. We assign different numbers of patches on CPU and GPU processors according to their actual running times. Taking our denoising network as an example, we assign 30 patches to GPU processor and 18 patches to CPU processor, which leads to the most efficient running time of denoising network (30 $\times$ 44.5 ms = 1335 ms).

### 4.5 Ablation Study

We conduct ablation study to demonstrate the effectiveness of noise prior integration, multi-frame alignment and multi-frame denoising. These improvements are evaluated to illustrate that our methods run with limited computational cost but with competitive performance.

### 4.5.1 Noise Prior Integration

Since CRVD dataset Yue et al. (2020) is the only multi-frame raw dataset with ground truth, we perform ablation study of noise prior integration on CRVD dataset (burst number $N = 3$). In Table 4, we evaluate the different methods for implementing noise prior integration on CRVD dataset. We first remove the noise prior integration stage (denoted as “Ours w/o prior integration”), to let the network to handle unstable variances directly. Removing noise prior integration leads to about 0.43dB drop. Then we compare our method with the mainstreaming noise adaption method: noise map Mildenhall et al. (2018); Tassano et al. (2020); Maggioni et al. (2021). In training and finetuning, we replace the noisy input of network in our system by concatenation of noise map and noisy images (denoted as “Ours w/ noise map”). This method brings a drop of 0.17dB PSNR. To further analyze the importance of variance stabilization, we evaluate the effect of stabilizing sensor gains and brightness in Poisson distribution. We experiment using Eq. (5) (denoted as “Ours w/ VS-gain”) and $k$-$\sigma$ transform Eq. (11) proposed in PMRID Wang et al. (2020) (denoted as “Ours w/ $k$-$\sigma$”). Surprisingly, they achieves almost the same performances and shows a drop of about 0.14dB PSNR compared with our method. It reveals that $k$-$\sigma$ transform Wang et al. (2020) can also be used to stabilize sensor gains. Finally, we test stabilizing the variances of different brightness in Poisson distribution. We compare our generalized Tukey-Freeman transformation with well-known generalized anscombe transformation (GAT) Starck et al. (1998) ((denoted as “Ours w/ GAT”)). Using GAT brings a drop of 0.09dB PSNR. For the reason that our generalized Tukey-Freeman transformation surpasses generalized...
Fig. 6 Visual comparisons of different noise prior integration on CRVD dataset Yue et al. (2020) (burst number $N = 3$)

anscombe transformation in raw denoising, please refer to Appendix B.

Figure 6 shows the visualization of denoising results with different types of noise prior integration. Our method and GAT Starck et al. (1998) show more details and textures than noise map Mildenhall et al. (2018); Zhang et al. (2018) and $k$-sigma transformation Wang et al. (2020). Our method also achieves slight improvements on recovering edges against GAT Starck et al. (1998).

4.5.2 Multi-Frame Alignment

We demonstrate that performing explicit alignment is necessary in our denoising system and state-of-the-art burst denoising methods. As the mainstream burst denoising methods, kernel prediction methods Mildenhall et al. (2018); Xia et al. (2020) do not require an explicit alignment module. In contrast, RViDeNet Yue et al. (2020) and BDNet Liang et al. (2020) utilize Deformable Convolution Dai et al. (2017); Zhu et al. (2019) as an explicit alignment module before multi-frame denoising. To demonstrate the effectiveness of explicit alignment, we integrate our alignment module into two kernel prediction methods: KPN Mildenhall et al. (2018) and BPN Xia et al. (2020).

As there is no KPN model Mildenhall et al. (2018) released to the public, we implement KPN and report its result (denoted as “KPN*” in Table 5). Our implementation shows similar performance compared with the original results in Mildenhall et al. (2018). Adding an alignment module into KPN* (denoted as “KPN* + alignment”) leads to about 0.7dB at gain $\propto 4$. The results in Table 5 also show that adding our alignment module on top of BPN Xia et al. (2020) achieves an increase of about 0.4 dB PSNR at gain $\propto 2, 4, 8$. When we remove the alignment module (denoted as “Ours w/o alignment”) in our method, our denoising network is trained on the synthetic frames with misalignment in [2, 16] pixels. It achieves the approximate performance with “KPN* + alignment”. Then we perform our alignment (denoted as “Ours + alignment”) on the above model (trained in “Ours w/o alignment”). It is shown that directly applying our alignment brings improvement of 0.84dB average PSNR. Finally, we train our denoising network on images aligned by our method (denoted as “Ours”). The result shows that training on aligned images brings about an increase of 0.72 dB average PSNR.

In Table 6, we compare our alignment with learning-based alignment in CRVD dataset Yue et al. (2020) (denoted as “±16 CRVD”). The misalignments between the reference frame and alternate frames are uniformly sampled in [2, 16] pixels.

We combine Deformable Convolution alignment (DCN) in Liang et al. (2020); Yue et al. (2020) with our denoising network (denoted as “Ours w/ DCN”). It is shown in Table 6, deformable convolution alignment only leads to marginal improvements over our method. We also evaluate the running times of the alignment modules on mobile processors. With the help of ARM Neon Intrinsic, our alignment is much faster than the DCN alignment.

4.5.3 Multi-Frame Denoising Network

We first evaluate the effectiveness of sequential denoising on CRVD dataset (burst number $N = 5$). When we remove the sequential denoising, the single denoising network would take all frames as inputs simultaneously (denoted as “Ours-SQ1”). It is shown in Table 7 that the performance suffers
Table 5 Ablation study of influence of alignment on KPN dataset ((burst number \(N = 8\))

| Algorithm                      | Gain \(\propto 1\) | Gain \(\propto 2\) | Gain \(\propto 4\) | Gain \(\propto 8\) | Average |
|-------------------------------|---------------------|---------------------|---------------------|---------------------|---------|
| KPN Mildenhall et al. (2018)  | 36.47               | 33.93               | 31.19               | 27.97               | 32.39   |
| KPN* Mildenhall et al. (2018) | 36.47               | 33.94               | 31.21               | 28.24               | 32.47   |
| KPN* + alignment              | 37.14               | 34.62               | 31.87               | 28.73               | 33.09   |
| BPN Xia et al. (2020)         | 38.18               | 35.42               | 32.54               | 29.45               | 33.90   |
| BPN + alignment               | 38.28               | 35.79               | 32.96               | 29.86               | 34.22   |
| Ours w/o alignment            | 37.63               | 34.92               | 32.13               | 28.89               | 33.39   |
| Ours + alignment              | 38.48               | 35.79               | 32.87               | 29.62               | 34.19   |
| Ours                          | 39.39               | 36.52               | 33.47               | 30.20               | 34.90   |

KPN* denotes our implementation of KPN

Table 6 Ablation study of influence of different alignment methods on CRVD dataset Yue et al. (2020) (burst number \(N = 3\)). \(\pm 16\) CRVD denotes performing \(\pm 16\) pixels misalignment on CRVD dataset. Running times of the alignment modules is calculated on the Snapdragon 888 mobile processor

| Algorithm                        | CRVD          | \(\pm 16\) CRVD | Running time (ms) |
|----------------------------------|---------------|-----------------|-------------------|
| Ours w/o alignment               | 44.36 / 0.989 | 43.68 / 0.978   | –                 |
| Ours w/ DCN Yue et al. (2020); Wang et al. (2019) | 44.46 / 0.989 | 44.11 / 0.981   | 2957              |
| Ours                             | 44.43 / 0.989 | 44.05 / 0.980   | 38                |

Table 7 Comparison of different sequential denoising strategies on CRVD dataset (burst number \(N = 5\))

| Algorithm   | Raw          | sRGB         |
|-------------|--------------|--------------|
| Ours-SQ1    | 44.37 / 0.989 | 40.52 / 0.982 |
| Ours-SQ2    | 44.46 / 0.989 | 40.56 / 0.982 |
| Ours-SQ3    | 44.61 / 0.990 | 40.76 / 0.983 |
| Ours        | 44.70 / 0.990 | 40.88 / 0.983 |

from a drop of 0.33 dB PSNR when removing sequential denoising. When we use a network for spatial denoising on reference frame and another network for temporal denoising of the 4 alternate frames (denoted as “Ours-SQ2”), the performance also drops by 0.24 dB PSNR. Furthermore, we test using a network for spatial denoising and two networks for sequential denoising and each network would handle temporal information of two neighboring alternate frames at a time (denoted as “Ours-SQ3”). This design also causes a drop of \(~0.09\) dB PSNR. In the above variants, we adjust the model size by changing the channel numbers so that different setups have similar FLOPs for fair comparison.

To demonstrate the advantages of the proposed multi-frequency aggregation, we remove the multi-frequency aggregation and directly use the output \(o_0\) as the final result (denoted as “Ours w/o aggregation”). The network would be a little similar to SGN Gu et al. (2019). It is shown in Table 8 that removing multi-frequency aggregation causes a drop of about 0.26 dB PSNR. This result demonstrates the advantages of the multi-frequency denoising. Then we test replacing the proposed multi-frequency aggregation with pointwise convolution (denoted as “Ours w/ pointwise conv”) and attention-based fusion Vogels et al. (2018) (denoted as “Ours w/ attention”). When we adopt pointwise convolution for multi-frequency aggregation, it aggregates outputs of the three scales after \(o_2, o_1\) are upsampled to the same size as \(o_0\). It suffers a significant degradation of 0.72 dB PSNR. As for attention-based fusion Vogels et al. (2018), we utilize an 8-layer convolutional network followed by a sigmoid layer to predict per-pixel aggregation weights between two neighboring frequencies. It increases computational cost but still shows degraded performance of 0.15 dB PSNR.

Table 8 Ablation study of multi-frequency aggregation on CRVD dataset (burst number \(N = 3\))

| Algorithm                      | Raw          | sRGB         |
|-------------------------------|--------------|--------------|
| Ours w/o aggregation          | 44.17 / 0.988 | 40.10 / 0.980 |
| Ours w/ pointwise conv        | 43.71 / 0.987 | 39.69 / 0.979 |
| Ours w/ attention Vogels et al. (2018) | 44.28 / 0.988 | 40.34 / 0.981 |
| Ours                          | 44.43 / 0.989 | 40.51 / 0.982 |

5 Conclusions

In this work, we proposed an efficient video denoising method via the improvements of three stages of the denoising framework noise prior integration, multi-frame alignment and multi-frame fusion.

Transforming raw images into a variance stabilization space can significantly reduce the model complexity without
impacting its performance. From the perspective of on-chip running and efficiency, we combine classical image alignment and learning-based denoising to achieve comparable denoising performance with faster speed. As for multi-frame denoising, we introduce sequential denoising strategy and multi-frequency denoising to achieve efficient multiple-frame denoising.

We have deployed these three stages on a commercial SoC. It demonstrates that our method can be employed for burst image denoising on mobile devices.

Acknowledgements This work is supported in part by Centre for Perceptual and Interactive Intelligence Limited, in part by the General Research Fund through the Research Grants Council of Hong Kong under Grants (Nos. 14204021, 14207319, 14203118, 14208619), in part by Research Impact Fund Grant No. R5001-18, in part by CUHK Strategic Fund.

Availability of Data and Materials All datasets mentioned in this manuscript are the open datasets.

Code Availability The code of variance stabilization and network is available from the authors upon reasonable request.

Declarations

Conflict of interest This research is sponsored by The Chinese University of Hong Kong and may lead to the development of products which may be licensed to Sensetime, in which I have a business interest. I have disclosed those interests fully to Taylor & Francis, and have in place an approved plan for managing any potential conflicts arising from this arrangement.

Ethics Approval Not applicable.

Consent to Participate Not applicable.

Consent for Publication All authors consent to participate in this work. No extra dataset is proposed in this study.

Appendix A Noise Modeling of CMOS Signals

We provide the detailed noise modeling of CMOS signals to obtain the relation between sensor gain and $\sigma_r$, $\sigma_s$. We define the observed intensity as $x$ and underlying true intensity as $x^*$. Following Wang et al. (2020), the raw signal is modeled as

$$x \sim q_e a \mathcal{P}\left(\frac{x^*}{q_e a}\right) + \mathcal{N}(0, \alpha^2 \sigma_0^2 + \sigma_{adc}^2), \quad (A.1)$$

where $q_e$ is quantum efficiency factor, $\alpha$ is the sensor gain, $\sigma_0$ is the variance of read noise caused by sensor readout effects and $\sigma_{adc}$ is the variance of amplifier noise. Then we have:

$$\sigma_s = q_e a$$

$$\sigma_r^2 = \alpha^2 \sigma_0^2 + \sigma_{adc}^2. \quad (A.2)$$

For one fixed sensor, $q_e$, $\sigma_0$, $\sigma_{adc}$ is unchanged. Then sensor gain $\alpha$ is the only factor to affect $\sigma_s$, $\sigma_r$.

Appendix B Generalized Version of Freeman-Tukey Transformation

For variable $x$ in Poisson distribution of the mean value $x^*$, the general form of variance stabilization transformation in root-type is

$$y = 2\sqrt{x + c}. \quad (B.1)$$

The core problem of variance stabilization is to stabilize Poisson distribution to have unit variance. But no exact stabilization is possible Curtiss (1943). In practice, approximate transformations are generally used. The mainstreaming transformations include $2\sqrt{x}$, $2\sqrt{x + 1}$, $2\sqrt{x + \frac{1}{2}}$ Bartlett (1936), $2\sqrt{x + \frac{1}{8}}$ Anscombe (1948) and $\sqrt{x + \sqrt{x + 1}}$ Freeman and Tukey (1950). $\sqrt{x + \sqrt{x + 1}}$ can be taken as the linear combination of two general forms with $c = 0$ and $c = 1$. We visualize the variance of transformed $y$ in Fig. 7. When the value $x$ is enough large, the variance of $2\sqrt{x + \frac{1}{2}}$ Bartlett (1936), $2\sqrt{x + \frac{1}{8}}$ Anscombe (1948) and $\sqrt{x + \sqrt{x + 1}}$ Freeman and Tukey (1950) approach the unity. However, $\sqrt{x + \sqrt{x + 1}}$ Freeman and Tukey (1950) shows better approximation than other transformations when the mean value $x^*$ is close to zero. The SNR (signal-to-noise ratio) in dark areas is usually lower than that of other areas.

![Fig. 7 Visualization of different variance stabilization transformations.](image-url)
Therefore, we seek the generalized version of Freeman-Tukey Transformation Freeman and Tukey (1950) to handle Poisson-Gaussian distribution for raw denoising.

Firstly, we start from the transform of Poisson distribution. We define variable $x$ to be a Poisson variable of mean $m$. Its variance is $\text{Var}(x) = m$. We define $y$ to be the transformed $x$. Then we have $\text{Var}(y) \approx (\frac{d}{dx})^2\text{Var}(x)$ based on Doob (1935) and Bartlett (1947). The core problem of variance stabilization is stabilize Poisson distribution into unity variance. Hence we let $\text{Var}(y) = 1$ and obtain:

$$\frac{dy}{dx} = \sqrt{\frac{\text{Var}(y)}{\text{Var}(x)}} = \frac{1}{\sqrt{m}}. \quad (B.2)$$

For the general transform $y = 2\sqrt{x} + c$, we have

$$\frac{dy}{dx} = \frac{1}{\sqrt{x + c}}. \quad (B.3)$$

From Eqs. (B.2) and (B.3), we obtain the approximation:

$$m = x + c. \quad (B.4)$$

Secondly, we consider the transform of Poisson-Gaussian distribution. Similar to Eq. (5), we define variable $z$ as $z = x + y$, where $x$ is a Poisson variable of mean $m$ and $y$ is a Gaussian variable of mean $g$ and standard deviation $\sigma$. The variance of transformed $z$ is given by $\text{Var}(y) \approx (\frac{d}{dx})^2\text{Var}(z)$ based on Doob (1935) and Bartlett (1947). Similarly, we let $\text{Var}(y) = 1$ and obtain:

$$\frac{dy}{dz} = \sqrt{\frac{\text{Var}(y)}{\text{Var}(z)}} = \frac{1}{\sqrt{m + \sigma^2}}. \quad (B.5)$$

We take the first-order approximation in Starck et al. (1998) to approximate the Gaussian distribution $y \approx g$. From Eq. (B.4), we have $m = x + c - g$. Thus we have:

$$\frac{dy}{dx} = \frac{1}{\sqrt{z + c + \sigma^2 - g}}. \quad (B.6)$$

By integral of Eq. (B.6), we have the transformation $y(z)$ for Poisson-Gaussian distribution:

$$y(x) = 2\sqrt{z + c + \sigma^2 - g}. \quad (B.7)$$

Finally, we move to the generalized version of Freeman-Tukey Transformation Freeman and Tukey (1950): $y = \sqrt{x + \sqrt{x + T}}$. From the Eq. (B.7), we generalize $2\sqrt{x}$ and $2\sqrt{x + T}$ respectively. By using linear combination of two generalized transformations ($c = 0$ and $c = 1$), we obtain the generalized version of Freeman-Tukey Transformation:

$$y(x) = \sqrt{x + 1 + \sigma^2 - g} + \sqrt{x + \sigma^2 - g}. \quad (B.8)$$

### Appendix C Algebraic Inverse of transform

It is known that algebraic inverse is usually avoided due to bias in previous methods Starck et al. (1998). However the bias is already handled when we calculate the loss in the space of variance stabilization. Moreover, algebraic inverse can be used for both Anscombe transformation Anscombe (1948); Starck et al. (1998) and Freeman-Tukey transformation Freeman and Tukey (1950) in our framework.

Let $x$ and $x^*$ denote noisy signal and clean signal, respectively. The transform (Anscombe transform or Freeman-Tukey transform) is denoted as $f$ and the algebraic transform is denoted as $f^{-1}$. The bias is produced by the nonlinearity of the transformation $f$. We calculate the loss in the variance stabilization space. The denoising network would learn the mapping from $f(x)$ to $f(x^*)$ directly. Therefore, the bias is already handled when the denoising output approximates $f(x^*)$.

We further conduct experiments on CRVD dataset (burst number $N = 5$) to compare algebraic inverse and exact unbiased inverse under different training settings. The results are shown in Table 9. We first training with Generalization Anscombe transformation (GAT) Starck et al. (1998) and calculate the loss function before the inverse. Then we test the model with algebraic inverse (denoted as “GAT-4”) and exact unbiased inverse (denoted as “GAT-3”). It is shown that algebraic inverse outperforms the exact unbiased inverse.

| Algorithm | Variance stabilization | Training loss | Inverse | PSNR |
|-----------|------------------------|---------------|---------|------|
| GAT-1     | GAT (Anscombe)         | After inverse | Unbiased | 44.60 |
| GAT-2     | GAT (Anscombe)         | After inverse | Algebraic | 44.60 |
| GAT-3     | GAT (Anscombe)         | Before inverse| Unbiased | 44.50 |
| GAT-4     | GAT (Anscombe)         | Before inverse| Algebraic | 44.63 |
| Ours-1    | Freeman-Tukey          | After inverse | Algebraic | 44.68 |
| Ours      | Freeman-Tukey          | Before inverse| Algebraic | 44.70 |
Makitalo and Foi (2013) by 0.13 dB PSNR, which demonstrates that the bias is handled in calculating loss before inverse. Then we train with GAT with algebraic inverse (denoted as “GAT-2”) and optimal inverse (denoted as “GAT-1”) and calculate the loss function after the inverse. In Table 9, it can be observed that both two inverses show the same performance (44.60 dB PSNR) but are 0.03 dB PSNR lower than calculating the loss before inverse. It might be because the bias produced in the space of variance stabilization becomes more complicated after the non-linear inverse transformation. Handling the bias before inverse is more direct. The same phenomenon can also be observed in the Freeman-Tukey transformation (“Ours-1” VS “Ours”).

Appendix D More Ablation of Denoising Network

Input order of alternate frames We conduct experiments on CRVD dataset Yue et al. (2020) (burst number \(N = 5\)) to compare three input orders: a) preserving the temporal order of an input burst (denoted as “Keep”), b) shuffling the burst order randomly (denoted as “Shuffle”), and c) reversing the burst order (denoted as “Reverse”). In training and testing, 4 alternate frames are re-arranged following the same ordering strategies. It is shown in Table 10 that training with preserving the temporal order achieve the best performance of 44.70 dB PSNR, which slightly outperforms random shuffling by 0.03 dB PSNR. Furthermore, reversing the temporal order achieve the worst performance of 44.63 dB PSNR, which suffers a drop of 0.07 dB PSNR. It can be observed that preserving the temporal order is helpful in sequential denoising.

Specializing the network weights In our denoising network \(\mathcal{S}\), we have a series of sub-networks for sequential denoising. For burst denoising on CRVD dataset Yue et al. (2020) (burst number \(N = 5\)), \(\mathcal{S}_0\) is for spatially denoising of reference frame and \(\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\) are for sequential denoising of the 4 alternate frames. We conduct experiments on CRVD dataset (burst number \(N = 5\)) to compare \(\mathcal{S}_i\) with different weights (denoted as “specializing”) and \(\mathcal{S}_i^\prime\) with shared weights (denoted as “sharing”). It is shown in Table 11 that using shared weights of \(\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4\) just achieves 44.44 dB PSNR, which has a drop of 0.26 dB PSNR compared with specializing each \(\mathcal{S}_i\) (44.70 dB PSNR).

Different Scales in denoising backbone We conduct experiments on CRVD dataset (burst number \(N = 5\)) to explore using different scales (frequencies). We define the number of scales (frequencies) as \(s\). When \(s = 4\), we use four frequencies \((m_0, m_1, m_2, m_3)\) to achieve multi-frequency denoising. It can be observed in Table 12 that using two frequencies achieves 44.49 dB PSNR, which is a drop of 0.21 dB compared with using three frequencies (44.70 dB PSNR). But when we use four scales (frequencies), the denoising performance is 44.71 dB PSNR and just outperform using three frequencies by only 0.01 dB PSNR but its model size increases from 1.57M to 2.10M.

### References

ARM Neon Intrinsics. (2014). [https://developer.arm.com/architectures/instruction-sets/intrinsics/](https://developer.arm.com/architectures/instruction-sets/intrinsics/).

Anscombe, F. J. (1948). The transformation of poisson, binomial and negative-binomial data. *Biometrika, 35*(3–4), 1948.

Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., & Szeliski, R. (2011). A database and evaluation methodology for optical flow. *International Journal of Computer Vision, 92*(1), 1–31.

Bartlett, M. S. (1936). The square root transformation in analysis of variance. *Supplement to the Journal of the Royal Statistical Society, 3*(1), 68–78.

Bartlett, M. S. (1947). The use of transformations. *Biometrics, 3*(1), 39–52.

Brooks, T., Mildenhall, B., Xue, T., Chen, J., Sharlet, D., & Barron, J. T. (2019). Unprocessing images for learned raw denoising. CVPR.

Buades, A., Coll, B., & Morel, J.-M. (2005). A non-local algorithm for image denoising. In: Conference on Computer Vision and Pattern Recognition CVPR (2), IEEE Computer Society, pp. 60–65.

Calonder, Michael, Lepetit, Vincent, Strecha, Christoph, & Fua, Pascal. (2010). Brief: Binary robust independent elementary features. In: Costas Daniilidis, Petros Maragos, & Nikos Paragios (Eds.), *Computer Vision - ECCV 2010*. Berlin Heidelberg: Springer.

Curtiss, J. H. (1943). On Transformations Used in the Analysis of Variance. *Annals of Mathematical Statistics, 14*(2), 107–122.

Dabov, K., Foi, A., & Egiazarian, K. (2007). Video denoising by sparse 3D transform-domain collaborative filtering. In: 2007 15th European Signal Processing Conference (pp. 145-149). IEEE.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., & Wei, Y. (2017). Deformable convolutional networks. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp. 764–773.

Doob, J. L. (1935). The Limiting Distributions of Certain Statistics. *Annals of Mathematical Statistics, 6*(3), 160–169.

### Tables

| Network weights | Sharing | Specializing |
|-----------------|---------|--------------|
| PSNR (dB)       | 44.44   | 44.70        |

| # Scales | \(s = 2\) | \(s = 3\) | \(s = 4\) |
|----------|-----------|-----------|-----------|
| PSNR (dB)| 44.49     | 44.70     | 44.71     |
| Params. (M)| 1.06     | 1.57      | 2.10      |
Digital negative (dng). https://helpx.adobe.com/camera-raw/digital-negative.html#dng

Foi, Alessandro, Trimeche, Mejdi, Katkovnik, Vladimir, & Egiazarain, Karen. (2008). Practical poissonian-gaussian noise modeling and fitting for single-image raw-data. IEEE Transactions on Image Processing, 17(10), 1737–1754.

Freeman, Murray F., & Tukey, John W. (1950). Transformations Related to the Angular and the Square Root. Annals of Mathematical Statistics, 21(4), 607–611.

Gu, S., Li, Y., Gool, LV., & Timofte, R. (2019) Self-guided network for fast image denoising. In: The IEEE International Conference on Computer Vision (ICCV).

Hasinoff, S. W., Sharlet, D., Geiss, R., Adams, A., Barron, J., Kainz, F., et al. (2016). Burst photography for high dynamic range and low-light imaging on mobile cameras. ACM Transactions on Graphics, 35(6), 1–12.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

Healey, G. E., & Kondepudy, R. (1994). Radiometric ccd camera calibration and noise estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 16(3), 267–276.

Horn, B. K., & Schunck, B. G. (1981). Determining optical flow. Artificial Intelligence, 17(1–3), 185–203.

Krasin, I., Duering, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kaznetsova, A., Rom, H., Uijlings, J., Popov, S., Veit, A., Belongie, S., Victor G., Akinvole, Ch., Sen, S., Gal, C., David C., Zheyun F., Dhiyanesh N., and Kevin M. (2017) OpenImages: A public dataset for large-scale multi-label and multi-class image classification. Dataset available from https://github.com/openimages.

Liang, Z., Guo, S., Gu, H., Zhang, H., & Zhang, L. (2020) A decoupled learning scheme for real-world burst denoising from raw images. In: European Conference on Computer Vision (ECCV).

Liu, Xinrui, Tanaka, Masayuki, & Okutomi, Masatoshi. (2014). Practical signal-dependent noise parameter estimation from a single noisy image. IEEE Transactions on Image Processing, 23(10), 4361–4371.

Liu, Z., Yuan, L., Tang, X., Uyttendaele, M., & Sun, J. (2014). Fast burst images denoising. ACM Transactions on Graphics, 33(6), 1–232.

Bruce, D. (1981). Lucas and Takeo Kanade. IJCAI: An iterative image registration technique with an application to stereo vision.

Maggioni, M., Huang, Y., Li, C., Xiao, S., Fu, Z. & Song, F. (2021). Efficient multi-stage video denoising with recurrent spatio-temporal fusion.

Maggioni, M., Boracchi, G., Foi, A., & Egiazarain, K. (2011). Video denoising using separable 4d nonlocal spatiotemporal transforms. In: Image Processing: Algorithms and Systems, vol. 7870 of SPIE Proceedings, pp. 787003. SPIE

Maggioni, Matteo, Boracchi, Giacomo, Foi, Alessandro, & Egiazarian, Karen O. (2012). Video denoising, deblocking, and enhancement through separable 4d nonlocal spatiotemporal transforms. IEEE Transactions on Image Processing, 21(9), 3952–3966.

Mairal, J., Bach, F., Ponce, J., Sapiro, G., & Zisserman, A. (2009). Non-local sparse models for image restoration. In: IEEE 12th International Conference on Computer Vision ICCV, pp. 2272–2279. IEEE Computer Society

Makitalo, Markku, & Foi, Alessandro. (2013). Optimal inversion of the generalized anscombe transformation for poisson-gaussian noise. IEEE Transactions on Image Processing, 22(1), 91–103.

Marin, T., Srinivasan, V., Güll, S., Hellge, C., & Samek, W. (2019). Multi-kernel prediction networks for denoising of burst images. CoRR, abs/1902.05392

Menze, M., & Geiger, A. (2015). Object scene flow for autonomous vehicles. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3061–3070

Mildenhall, B., Barron, JT., Chen, J., Sharlet, D., Ng, R., & Carroll, R. (2018). Burst denoising with kernel prediction networks. In CVPR

Mäkitalo, M., & Foi, A. (2012). Poisson-gaussian denoising using the exact unbiased inverse of the generalized ancombe transformation. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

Nah, S., Kim, TH, & Lee, KM. (2017). Deep multi-scale convolutional neural network for dynamic scene deblurring. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché-Buc, E., Fox, E., and R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc.

OLiba, O., Murthy, K., Tsai, Y.T., Brooks, T., Xue, T., Karmad, N., He, Q., Barron, J.T., Sharlet, D., Geiss, R., Hasinoff SW., Yael P., and Marc L. Handheld mobility phone in very low light. SIGGRAPH Asia, 2019.

Pplnn is a primitive library for neural network. https://github.com/openppl-public/ppl.nn, 2021.

Portilla, Javier, Strela, Vasily, Wainwright, Martin J., & Simoncelli, Eero P. (2003). Image denoising using scale mixtures of gaussians in the wavelet domain. IEEE Transactions on Image Processing, 12(11), 1338–1351.

Rosten, E., & Drummond, T. (2005). Fusing points and lines for high performance tracking. In IEEE International Conference on Computer Vision, 2, 1508–1511.

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. European Conference on Computer Vision, 1 430–443.

Rudin, L. I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. Physica D: Nonlinear Phenomena, 60(1–4), 259–268.

Snapdragon 888 5g mobile platform. https://www.qualcomm.com/products/snapdragon-888-5g-mobile-platform, 2020.

Starck, J. L., Murtagh, F., & Bijaoui, A. (1998). Image processing and data analysis. Cambridge: In Cambridge University Press.

Tassano, M., Delon, J., Veit, T. (2020). Fastvdnet: Towards real-time deep video denoising without flow estimation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

Vogels, T., Rousselle, F., Mcwilliams, B., Röthlin, G., Harvill, A., Adler, D., et al. (2018). Denoising with kernel prediction and asymmetric loss functions. ACM Transactions on Graphics, 37(4), 1–5.

Wang, X., Chan, KC., Yu, K., Dong, C., Change LC. (2019). Edvr: Video restoration with enhanced deformable convolutional networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

Wang, Y., Huang, H., Xu, Q., Liu, J., Liu, Y., & Wang, J. (2020). Practical deep raw image denoising on mobile devices. In: European Conference on Computer Vision (ECCV), pp.1–16.

Wei, K., Fu, Y., Yang, J. & Huang, H. (2020) A physics-based noise formation model for extreme low-light raw denoising. In: IEEE Conference on Computer Vision and Pattern Recognition

Weinzaepfel, P., Revaud, J., Harchaoui, Z., & Schmid, C. (2013). Deepflow: Large displacement optical flow with deep matching. In: 2013 IEEE International Conference on Computer Vision, pp. 1385–1392.

Xia, Z., Perazzi, F., Gharbi, M., Sunkavalli, K., & Chakrabarti, A. (2010). Basis prediction networks for effective burst denoising with large kernels. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).
Xue, T., Chen, B., Wu, J., Wei, D., & Freeman, W. T. (2019). Video enhancement with task-oriented flow. *International Journal of Computer Vision (IJCV)*, 127(8), 1106–1125.

Yue, H., Cao, C., Liao, L., Chu, R., & Yang, J. (2020). Supervised raw video denoising with a benchmark dataset on dynamic scenes. In: IEEE Conference on Computer Vision and Pattern Recognition, 2020.

Zhang, K., Zuo, W., Wangmeng, C., Yunjin, Meng, Deyu, & Zhang, Lei. (2017). Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7), 3142–3155.

Zhang, K., Zuo, W., & Zhang, L. (2018). Ffdnet: Toward a fast and flexible solution for CNN based image denoising. *IEEE Transactions on Image Processing*, 27(9), 4608–22.

Zhang, Y., Qin, H., Wang, X., & Li, H. (2021). Rethinking noise synthesis and modeling in raw denoising. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 4593–4601.

Zhu, X., Hu, H., Lin, S., & Dai, J. (2019). Deformable convnets v2: More deformable, better results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).

**Publisher’s Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.