This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2943214, IEEE Transactions on Image Processing

Deep Non-local Kalman Network for Video Compression Artifact Reduction

Guo Lu, Xiaoyun Zhang, Wanli Ouyang, Dong Xu, *Fellow, IEEE,* Li Chen, and Zhiyong Gao

***Abstract*—Video compression algorithms are widely used to reduce the huge size of video data, but they also introduce unpleasant visual artifacts due to the lossy compression. In order to improve the quality of the compressed videos, we proposed a deep non-local Kalman network for compression artifact reduction. Specifically, the video restoration is modeled as a Kalman filtering procedure and the decoded frames can be restored from the proposed deep Kalman model. Instead of using the noisy previous *decoded* frames as temporal information, the less noisy previous *restored* frame is employed in a recursive way, which provides the potential to generate high quality restored frames. In the proposed framework, several deep neural networks are utilized to estimate the corresponding states in the Kalman filter and integrated together in the deep Kalman filtering network. More importantly, we also exploit the non-local prior information by incorporating the spatial and temporal non-local networks for better restoration. Our approach takes the advantages of both the model-based methods and learning-based methods, by combining the recursive nature of the Kalman model and powerful representation ability of neural networks. Extensive experimental results on the Vimeo-90k and HEVC benchmark datasets demonstrate the effectiveness of our proposed method.**

***Index Terms*—Video Compression Artifact Reduction, Deep Neural Network, Kalman Model, Recursive Filtering, Video Restoration**

I. Introduction

Considering the increasing amount of video data over the Internet, compression algorithms (e.g., H.264 and HEVC) [1]–[3] have been applied to reduce the storage size and bandwidth. However, these algorithms also introduce compression artifacts, such as blocking, blurring and ringing artifacts. In order to obtain high quality images/videos at the decoder side, a lot of compression artifact reduction algorithms have been proposed to generate artifact-free images in the past decades.

Previously, manually designed filters [4], [5] and sparse coding based methods [6]–[9] are proposed to solve this problem. Recently, learning based approaches have been successfully applied to a lot of computer vision tasks [10]–[20], such as super-resolution [15], [16], denoising [17] and artifact reduction [18]–[20]. In [18], the convolutional neural networks (CNN) are firstly utilized for image compression artifact reduction, which demonstrates the effectiveness of CNN model.

Guo Lu, Xiaoyun Zhang, Li Chen and Zhiyong Gao are with Institude of Image Communication and Network Engineering, Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai, China. (email: {luguo2014,xiaoyun.zhang,hilichen, zhiyong.gao}@sjtu.edu.cn). Corresponding authors: Xiaoyun Zhang and Zhiyong Gao.

Wanli Ouyang is with The University of Sydney, SenseTime Computer Vision Research Group, Australia. (email: wanli.ouyang@sydney.edu.au)

Dong Xu is with The University of Sydney, Australia. (email: dong.xu@sydney.edu.au)

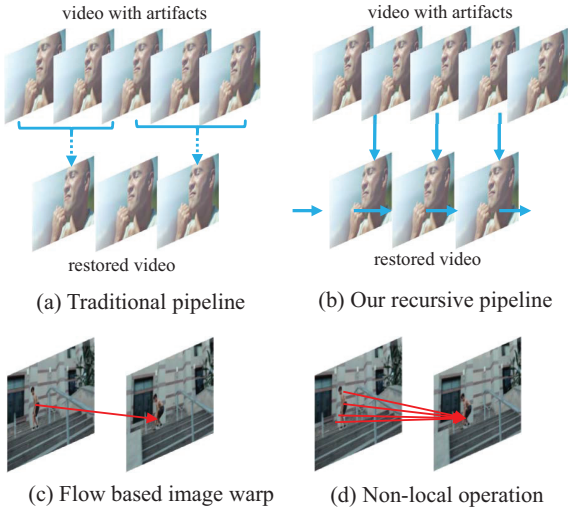


Fig. 1. Different methodologies for video artifact reduction (a) the traditional pipeline without considering previous restored frames. (b) our Kalman filtering pipeline. Exploiting the temporal information by (c) flow based image warp and (d) our non-local operation. w_i and w_j represent the non-local weight.

In this paper, we propose a deep non-local Kalman filtering network for video compression artifact reduction and our motivations are two-fold. First, the restoration process for the current frame can benefit from the previous restored frames. It is expected that the previous restored frame can provide more accurate temporal information compared with the original decoded frame. Therefore, we can employ more precise temporal information from previous restored frames and build a robust video artifact removal system with high performance. It is obvious that the dependence of previous restored frames will lead to a dynamic recursive solution for video artifact removal. More importantly, this scheme provides the opportunity to utilize long term temporal information through the recursive pipeline. As we know, most learning based approaches [16]–[19], [21] for artifact reduction focus on image artifact reduction. Although the temporal information is utilized in video artifact reduction [22] or video super-resolution [23]–[25], each frame is restored separately (see Fig. 1(a)) without considering the previous restored frames. In summary, we aims to build a dynamic filtering scheme (see Fig. 1(b)) to exploit accurate temporal information in previous frame for high quality restoration.

Second, spatial/temporal non-local prior information is beneficial for image/video restoration task. In the past decades, non-local prior has been successfully applied to image restoration tasks (e.g., image denoising [26] and image super-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2943214, IEEE Transactions on Image Processing

resolution [8]). However, it is still unclear how to utilize this powerful information for the learning based approach, especially for video restoration. More importantly, motion clue is crucial for the video restoration task. Most learning based video restoration approaches [22], [24] tried to use the optical flow to align the temporal neighbour frames for reconstruction (see Fig. 1(c)). Therefore, the quality of restored frame heavily rely on the accuracy of estimated optical flow and may degrade for the complex regions. At the same time, the non-local prior can capture the similarity between two neighbouring frames and can be used as an implicit approach to utilize the motion clue (see Fig. 1(d)), which is more robust and lightweight. Therefore, it is feasible to enhance the restoration by employing the non-local prior information.

In this paper, a deep non-local Kalman filtering network is proposed for video compression artifact reduction. Our proposed framework is designed as a post-processing module and can be easily extended to different compression algorithms. Specifically, the video artifact reduction is formulated as a Kalman filtering procedure, which means the decoded frame can be refined recursively by utilizing the information propagated from previous restored frames. In our deep non-local Kalman model, two CNN based neural networks (i.e., prediction network and measurement network) are proposed to perform Kalman filtering procedure. The prediction network tries to calculate *prior estimation* based on the previous restored frame, while the measurement network aims at obtaining the *measurement* through the current decoded frame. Both prediction network and measurement network incorporate the spatial/temporal non-local prior information by utilizing the well-designed non-local network. Then the *prior estimation* and the *measurement* are combined together in the Kalman framework to reduce the artifacts and restore the current frame. Our framework integrates the recursive nature of the Kalman filtering and highly non-linear transform ability of neural network, which bridges the gap between model-based method and learning-based method. To the best of our knowledge, this is the first deep neural network under Kalman model for video artifact reduction.

In summary, the main contributions of our paper are two-fold. First, the video artifact reduction is formulated as a Kalman filtering process, which leads to a recursive restoration procedure for the decoded frames. Several CNN models are employed to predict and update the state in the Kalman filtering procedure. Second, we employ the non-local network to exploit the powerful prior information in spatial/temporal domain for robust estimation. Extensive experimental results validate the effectiveness of our proposed framework for video compression artifact reduction.

The proposed framework builds upon our previous method in [27], we make the following notable improvements. First, we utilize the non-local network to exploit the spatial and temporal prior information and improve the quality of restored frames. Second, our proposed framework does not rely on task-specific prior information as [27] and is successfully extented to other video restoration task (e.g., video denoising) in this paper. Third, we provide the in-depth analysis and more experimental results to demonstrate the effectiveness of our framework.

II. Related Work

A. Single Image Compression Artifact Reduction

In the past decades, a lot of methods have been proposed to reduce the artifacts introduced by compression. In [28], [29], the hand-crafted filters are proposed for reducing blocking and ringing artifacts. Although these methods utilized sophisticated designed filters, the decoded frames are still far from satisfactory. In order to improve the performance for artifact reduction, the sparse coding technique is also widely utilized [8], [9], [30]. For example, Chang *et al.* [8] employed sparse coding and tried to learn powerful representations to reduce the compression artifacts. In [30], Wang *et al.* built a deep dual-domain based fast restoration model by leveraging the large capacity of deep networks and problem-specific knowledge.

Recently, a lot of convolutional neural network based methods have been proposed for the low-level vision tasks, such as super-resolution, denoising and artifact reduction. For example, a CNN based neural network (ARCNN) for JPEG artifact reduction was proposed in [18]. Inspired by ARCNN, various techniques (such as residual learning [31], skip connection [31], [32], batch normalization [17], perceptual loss [33], residual block [20] and generative adversarial network [20]) have been employed to generate high quality reconstructed frames. In [17], a 20-layer neural network by using batch normalization and residual learning was proposed for both denoising and other low-level vision tasks. In [16], Tai *et al.* built a memory persistent network based on a recursive unit and a gate unit for image restoration task. It should be mentioned that a lot of works [34]–[36] also tried to learn the image prior by using CNN and embedded the prior information into the traditional pipeline for image restoration. Venkatakrishnan *et al.* [37] built on the ADMM [38] algorithm and proposed to replace the proximal operator of the regularizer with a denoiser such as BM3D [39], which motivates subsequent works to learn the proximal operator using CNNs [34], [35].

B. Deep Learning for Video Restoration

In addition to the methods for image restoration, a lot of CNN based methods [22], [23], [40]–[42] have been proposed for video restoration. In [40], the optical flow was utilized to generate an ensemble of super-resolution draft, then the high resolution frame was restored from all drafts by a CNN model. In [41], Kappeler *et al.* also estimated the optical flow and then the corresponding patches were selected to generate the high resolution frame. Jaderberg *et al.* proposed the spatial transformation network (STN) to actively spatially transform feature maps. Inspired by this idea, several works [22]–[25] used the optical flow(transform parameters) to perform motion compensation and employed the aligned reference frames to increase the temporal coherence for high quality video super-resolution. For example, Liu *et al.* [9] aligned the reference frames and fused different temporal neighbouring frames through a multiple stream network. In [23], Tao *et al.* proposed a sub-pixel motion compensation scheme to obtain finer motion representation for video restoration. In [22], Xue

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2943214, IEEE Transactions on Image Processing

et al. observed that the optical flow itself is not tailored for video restoration task, and used a joint training strategy to optimize the optical flow and the following video restoration network, which leads to state-of-the-art results. Recently, Wang *et al.* [43] used a pyramid, cascading and deformable (PCD) convolution module to align the neighboring frames to current frames. The PCD will learn the offset to align the feature through deformable convolution, while our method use non-local module to achieve the temporal alignment.

For video artifact reduction, several approaches have been proposed [21], [44]–[46]. Dai *et al.* proposed the VRCNN for intra coding artifact reduction in [44]. Yang *et al.* built a decoder side CNN for video artifact reduction in [21] and tried to use temporal information in [46]. In [47], Guan *et al.* used bidirectional long short-term memory module to obtain the peak quality frames and employed these two high quality frames to restore current frame. Both [47] and our proposed method try to use more accurate temporal information to reduce the compression artifact. The difference is that our method utilized previous restored frame in a recursive way while Guan *et al.* selected two best neighbouring frames in compressed videos.

Instead of employing the single image for compression artifact reduction, the video restoration methods try to exploit temporal information. However, these methods only use noisy/low-resolution videos separately without building a recursive pipeline by employing the previous restored frames. In other words, these methods cannot use the more accurate temporal information to improve the performance. Although the work [48], [49] try to combine deep neural network and Kalman filter, they are not designed for the image/video enhancement tasks. In our previous work [27], we use deep Kalman network for video restoration. However, the prediction residual is required in [27], which means the framework in [27] is not easy to extended to other related video restoration task (such as video denoising). More importantly, the motion estimation/compensation is not exploited in [27]. In this paper, we do not rely on the task-specific prior information and employ the temporal non-local network to implicitly obtain the motion information.

C. Non-local Prior based Image Restoration

Non-local prior has been widely used in image restoration [26], [39]. In [26], non-local means method calculates the weights of all pixels in an image and removes the noise by weight averaging the pixels. Recently, Wang *et al.* proposed a non-local neural network for video action recognition [50]. Liu *et al.* proposed to utilize non-local network for image restoration [51]. However, their method only considers the prior information in spatial domain, while the temporal information is also very important for video restoration task. In [52], a non-local patch search module is integrated into the deep neural network to exploit the non-local prior for image/video denoising. [53] follows the similar idea and extended this scheme for CT image. Specically, the scheme in [52], [53] first performs non-local search on the image level (based on raw pixel value) and select n most similar neighbors to construct a new feature vector for restoration. In contrast, we use the extracted feature to compute the similarity between neighboring frame and current frame. And instead of selecting n similar neighbors, our approach try to use all features in the search window and combine them based on the estimated similarity.

III. Methodology

In this section, we first provide the basic knowledge of Kalman filter and then formulate the video compression artifact reduction task as a Kalman procedure and introduce the corresponding network design.

Introduction of Denotations Let $\mathcal{V} = \{X|X_1, X_2, ..., X_{t-1}, X_t, \ldots\}$ denote an uncompressed video sequence, where $X_t \in \mathcal{R}^{mn\times 1}$ is a video frame at time step t and mn represents the spatial resolution. In order to simplify the description, we only analyze video frame with a single channel, although we restore the video with 3 channels (RGB or YUV) in our implementation. After compression, X_t^c represents the decoded frame of X_t. $\hat{X}_t^-$ is the prior estimation and $\hat{X}_t$ denotes the posterior estimation for restoring X_t from the decoded frame X_t^c.

A. Brief Introduction of Kalman Filter

In the past decades, Kalman filter is widely used to estimate the states of a dynamic system [54]. The Kalman filter builds a recursive pipeline to estimate the state (e.g., speed or pixel intensity value) according to the observed measurements. In the proposed framework, we assume that the original frame in uncompressed video is the to-be-estimated internal state, while the compressed frames are the measurements.

1) Preliminary Formulation: One basic assumption in Kalman model is that state X_t at time t can be obtained based on the state X_{t-1} at time $t-1$ in the following way,

$$X_t = A_t X_{t-1} + w_{t-1}, \tag{1}$$

where we define A_t as the transition matrix at time t and w_{t-1} represents the process noise for Kalman filter. In the basic Kalman model, we can also get the measurement Z_t for the true state X_t as follows,

$$Z_t = HX_t + v_t, \tag{2}$$

where we use H to represent the measurement matrix and v_t is the corresponding measurement noise. In addition, the system may be non-linear in practical application, which means it is necessary to formulate the non-linear relationship between state X_{t-1} and X_t in the transition process. Namely,

$$X_t = f(X_{t-1}, w_{t-1}), \tag{3}$$

where $f(\cdot)$ represents the non-linear transition model.

In summary, we use Eq. (1) and Eq. (2) to describe the linear Kalman procedure while the non-linear Kalman procedure is formulated by Eq. (3) and Eq. (2).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2943214, IEEE Transactions on Image Processing

2) Kalman Filtering: Generally, there are two steps in the Kalman filtering model: the prediction step and the update step.

In the *prediction step*, based on the posterior estimation in previous time step, we can estimate the prior estimation for current time step. Take the non-linear Kalman model as an example, the prediction step can be formulated in the following way,

$$\text{Prior state estimation: } \hat{X}_t^- = f(\hat{X}_{t-1}, 0), \tag{4}$$

$$\text{Covariance estimation: } P_t^- = A_t P_{t-1} A_t^{\mathrm{T}} + Q_{t-1}, \tag{5}$$

where Q_{t-1} represents the covariance of the process noise w_{t-1} at time $t-1$, and P_t^- is the corresponding covariance matrix to update the Kalman gain and posterior estimation. In Kalman model, A_t is the transition model that describes the relationship between X_t and X_{t-1}. For the non-linear model, we usually uses the Jacobian matrix of $f(\cdot)$ to estimate A_t [55]. Namely, $A_t = \frac{\partial f(\hat{X}_{t-1},0)}{\partial X_t}$. For the linear Kalman model, the prediction step is directly formulated as $f(\hat{X}_{t-1}, 0) = A_t \hat{X}_{t-1}$.

In the *update step*, the Kalman model will generate the posterior estimate by combining both the prior estimate from the prediction step and the measurement, which provides the potential to use more complementary information. More details will be discussed in Section III-G.

Fig. 2(a) provides the overall architecture of the Kalman model. Specifically, in the prediction step, a prior state estimation $\hat{X}_t^-$ at time t is calculated based on the estimated state $\hat{X}_{t-1}$ at time $t-1$. Then the Kalman model will fuses the prior estimation $\hat{X}_t^-$ and the measurement Z_t based on the Kalman gain in the update step. In order to restore the whole video sequences, we perform these two procedures at each time steps.

B. Overview of our Deep Non-local Kalman Filtering Network

Fig. 2(b) illustrates the architecture of the proposed Non-local Kalman filtering network. In our proposed framework, we use the prediction network to estimate the prior estimation $\hat{X}_t^-$ based on the previous restored frame $\hat{X}_{t-1}$ and the decoded frame X_t^c. The measurement network is also employed to generate the measurement Z_t for current frame. After that, we follow the Kalman procedure and get the final the posterior estimation $\hat{X}_t$ by combining the prior estimation and measurement.

Compared with the basic Kalman model in previous section, there are three main differences, which are summarized as follows,

First, we use the temporal non-local residual network as the non-linear function $f(\cdot)$ in Eq. (3) to obtain the prior state estimation. It should be mentioned that both the previous restored frame $\hat{X}_{t-1}$ and the decoded frame X_t^c are used as the input for the proposed temporal non-local residual network. The final prior estimate will depend on $\hat{X}_{t-1}$ and X_t^c. More details are provided in Section III-D.

Second, we approximate the transition matrix A_t in Eq. (5) by using a linearization network. In order to obtain the transition matrix A_t for the non-linear Kalman model, we usually have to calculate the Jacobian matrix of the non-linear function $f(\cdot)$. However, it will increase the computational complexity significantly. In our approach, instead of calculating the Jacobian matrix for each pixel location, we use an alternative method by employing the linearization to estimate the transition matrix A_t through neural network. Details are given in Section III-E.

Third, in the update procedure, we use a spatial non-local residual network to generate the measurement. In comparison, the conventional Kalman filter might directly use the decoded frame with compression artifacts as the measurement. Details are given in Section III-F.

C. Non-local Block

We first introduce the non-local block. The non-local operation can be generalized as follows,

$$R_i^z = \frac{1}{\mathcal{K}(R^x, R^y)} \sum_j w(R_i^x, R_j^y) g(R_j^y) \tag{6}$$

where R^x is the input feature, R^y is the reference feature and R^z is the output feature. i, j represent the indexes. R_i^x and R_j^y represent the feature vectors at location i and j. The pairwise function $w(\cdot)$ represents the relationship between R_i^x and R_j^y. $\mathcal{K}(\cdot)$ represents the sum of pairwise function $w(\cdot)$, i.e., $\mathcal{K}(R^x, R^y) = \sum_j w(R_i^x, R_j^y)$. The function $g(\cdot)$ computes the representation of R_j^y.

In the traditional non-local means algorithm, $g(\cdot)$ is the identity mapping. In our proposed framework, we use non-local operation to refine the input feature R_i^x based on the the relationship between input feature R_i^x and reference feature R_j^y. When R^x is equal to R^y, it means that the non-local operation is conducted in the spatial domain. We name it as spatial non-local operation, which is the standard non-local operation used in [26]. When R^x is not equal to R^y, e.g., R^x, R^y represent the feature from different time step, we name it as the temporal non-local operation.

The architecture of the non-local network is shown in Fig. 3. Inspired by [51], we do not use the whole feature map in R^y to calculate the similarity between R_i^x and R_j^y. In fact, we only use the corresponding neighbouring feature map P in R^y to obtain the similarity. In our implementation, we use the embedded Gaussian function to measure the similarity between R_i^x and R_j^y as follows,

$$w(R_i^x, R_j^y) = \exp((\mathrm{W}_\theta \mathrm{R}_\mathrm{i}^\mathrm{x})^{\mathrm{T}} \mathrm{W}_\phi \mathrm{R}_\mathrm{j}^\mathrm{y}), \mathrm{j} \in \mathrm{P} \tag{7}$$

where W_θ and W_ϕ are the weight metrixs and implemented by standard convolution operation. $g(R_j^y)$ is the representation of R_j^y, which is also implemented by convolution as follows,

$$g(R_j^y) = W_g R_j^y, j \in P \tag{8}$$

where the W_g is the weight matrix. To facilitate the training procedure, we use the residual connection in Figure 3. More importantly, the residual connection also allows to insert the non-local module into existing network. Then the output feature at location i is computed by,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2943214, IEEE Transactions on Image Processing

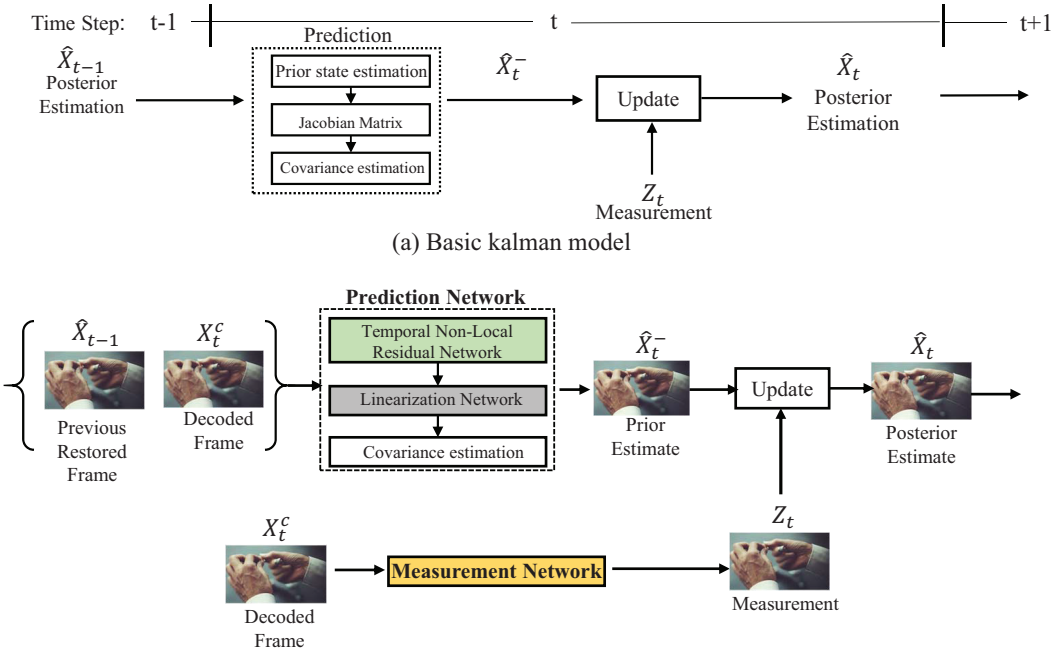


Fig. 2. (a) Basic Kalman model. (b) Overview of the proposed deep Kalman filtering network for video artifact reduction. X_t^c is the decoded frame at time t. $\hat{X}_{t-1}$ represents the restored frame from $t-1$. The prediction network generates prior estimate $\hat{X}_t^-$ for original frame based on X_t^c and $\hat{X}_{t-1}$. The measurement network takes the input of the decoded frame X_t^c to obtain an initial measurement Z_t. After that, we can build the posterior estimation $\hat{X}_t$ by fusing the prior estimate $\hat{X}_t^-$ and the measurement Z_t.

$$R_i^z = W_z softmax(\exp((\mathrm{W}_\theta \mathrm{R}_\mathrm{i}^\mathrm{x})^\mathrm{T} \mathrm{W}_\phi \mathrm{R}_\mathrm{j}^\mathrm{y}))\mathrm{g}(\mathrm{R}_\mathrm{j}^\mathrm{y}) + \mathrm{R}_\mathrm{i}^\mathrm{x} \quad (9)$$

where W_z is utilized to align the dimension.

D. Temporal Non-local Residual Network

1) Mathematical Formulation: In our proposed framework, in order to obtain the prior estimation, we use a CNN model to implement the non-linear function $f(\cdot)$ in the following way,

$$\hat{X}_t^- = \mathcal{F}(\hat{X}_{t-1}, X_t^c; \theta_f), \quad (10)$$

where θ_f represent the trainable parameters. The prior estimation of the current frame X_t depends on the estimated temporal neighbouring frame $\hat{X}_{t-1}$ and its decoded frame X_t^c at the current time step. The reasons are summarized in the following way. First, considering the strong correlation characteristic in temporal video sequences, it is natural to use the previous $\hat{X}_{t-1}$ to predict the X_t and get the corresponding prior estimation. Second, it is observed that the complex motion scenarios with occlusion will make it difficult to predict X_t by using $\hat{X}_{t-1}$ only. Therefore it is critical to employ the decoded frame X_t^c for a robust estimation for X_t. Based on these two assumptions, we model the transition function $f(\cdot)$ in Eq. (4) by adding decoded frame X_t^c as the extra input.

2) Network Implementation: The temporal non-local residual network architecture is illustrated in Fig. 4(a). Specifically, several residual blocks (pre-activation structure [56]) are used to build the non-local residual network. The kernel size is set to 3×3 and the output channel number of convolution layer is set as 64 except for the last layer, which is set to 1 for gray image. We employ the non-local network in our implementation to exploit the temporal non-local prior information. Specifically, the feature from previous restored frame $\hat{X}_{t-1}$ will be utilized to refine the feature extracted from current decoded frames. In contrast to other video restoration approaches [22], [24], we do not directly use the optical flow to warp the reference frame. Instead, we rely on the temporal non-local module to obtain the effective feature from reference frame, which is more robust and lightweight. More training details are discussed in Section III-H.

E. Linearization Network

In our proposed framework, we use the linearization network to learn the corresponding transition matrix A_t in Eq. (5). Traditional methods try to compute the Jacobian matrix of transition function $\mathcal{F}(\cdot)$ by using Taylor expansion, which increases the computational complexity, especially for learning based method. In our approach, the linearization network generate a liner matrix to approximate the non-linear procedure. Specifically, based on the the prior estimation $\hat{X}_t^-$, previous restored frame $\hat{X}_{t-1}$ and decoded frame X_t^c, we formulate the problem in the following way,

$$\hat{X}_t^l = \tilde{A}_t \hat{X}_{t-1} \text{ where } \tilde{A}_t = \mathcal{G}(\hat{X}_{t-1}, X_t^c; \theta_m) \quad (11)$$

where $\hat{X}_t^l$ represents the linearized prior estimation. In our proposed framework, we optimize the network $\mathcal{G}(\cdot)$ and expect $\hat{X}_t^l$ to be close to prior estimation $\hat{X}_t^-$. The network architecture is provided in Fig. 4(b). In our implementation, we use several convolution layers and residual blocks to build the linearization network.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2943214, IEEE Transactions on Image Processing

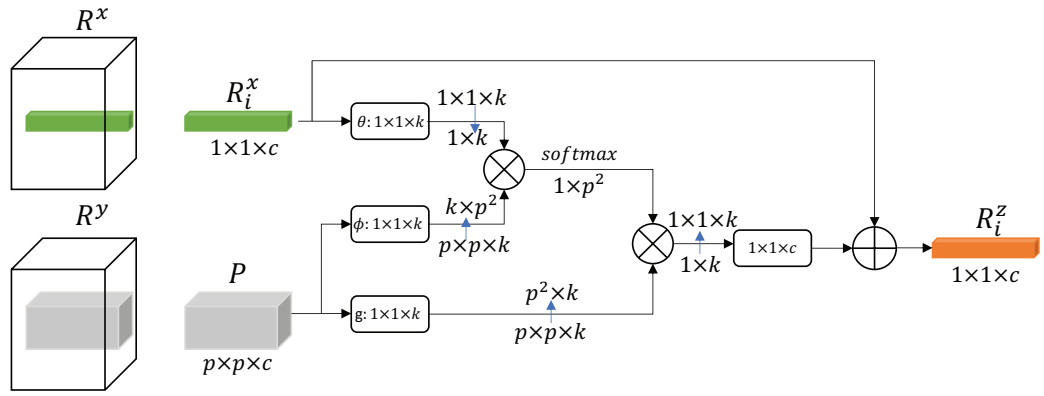
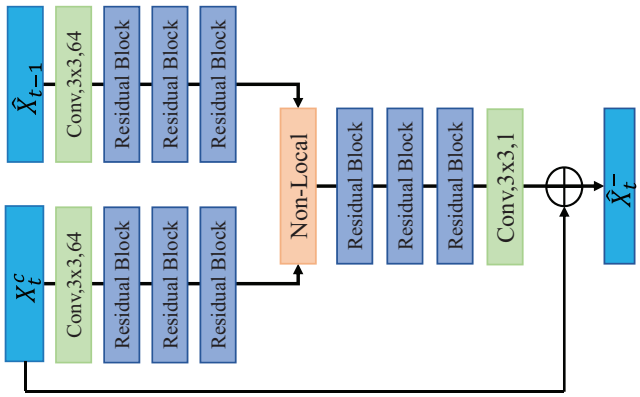
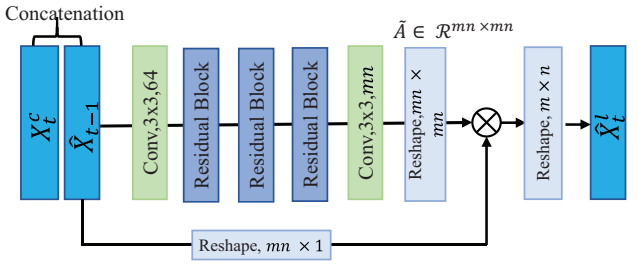


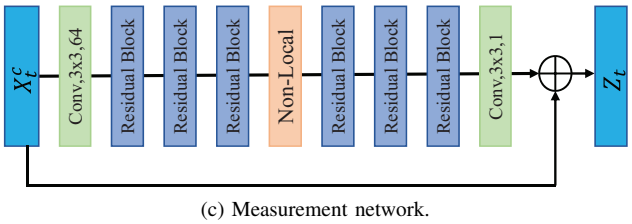
Fig. 3. The architecture of generalized non-local network. For simplified illustration, we only provide the non-local operation for one specific location. P represents the neighbouring features centered at location i in reference feature y. The blue lines represent the reshape operation. $\otimes$ denotes the matrix multiplication. $\oplus$ denotes the element-wise sum. θ, ϕ and g represent the 1×1 convolution. 'c' represents the channel number of feature.



(a) Temporal Non-lcoal Residual Network.



(b) Linearization Network.



(c) Measurement network.

Fig. 4. Network architecture of the proposed (a) Temporal Non-local Residual Network (b) Linearization network. (c) Measurement Network. Here 'Conv,3x3,64' represents the convolution operation with the 3x3 kernel and 64 feature maps. 'Reshape, $m\times n$' is the operation that reshapes one matrix to $m \times n$. $\oplus$ and $\otimes$ represent element-wise addition and matrix multiplication.

F. Measurement Network

In [27], the task-specific prior information (i.e., quantized prediction error) is utilized to obtain robust measurement estimation. However, it is not easy to utilize this information in other video restoration task, such as video denoising. In this paper, we utilized the spatial non-local information to compute the measurement, which is more generalized. Specifically, the measurement network is composed of several residual blocks and a spatial non-local network. The architecture of measurement network is illustrated in Fig. 4(c). The measurement is obtained as follows,

$$Z_t = X_t^c + \mathcal{M}(X_t^c; \theta_z) \tag{12}$$

where θ_z are the parameters for network $\mathcal{M}(\cdot)$.

This formulation also means that the existing image restoration methods could be seamless integrated into our framework as the measurement network, which demonstrates the flexibility of the proposed framework.

G. Update Step

Based on the prior state estimation $\hat{X}_t^-$ from the temporal non-local mapping network (Section III-D), the transition matrix $\tilde{A}_t$ obtained from the linearization network (Section III-E), and the measurement Z_t obtained from the measurement network (Section III-F), we follows the Kalman update state and compute the corresponding the posterior estimation in the following way [1],

$$P_t^- = \tilde{A}_t P_{t-1} \tilde{A}_t^{\mathrm{T}} + Q_{t-1}, \tag{13}$$

$$K_t = P_t^- H^{\mathrm{T}} (H P_t^- H^{\mathrm{T}} + U_t)^{-1}, \tag{14}$$

$$\hat{X}_t = \hat{X}_t^- + K_t (Z_t - H\hat{X}_t^-), \tag{15}$$

$$P_t = (I - K_t H) P_t^-, \tag{16}$$

Here, we introduce the denotations in above equations. First, $\hat{X}_t$ is the posterior estimation for the frame X_t, which is the final restoration result at current time step. In addition, we use P_t^- and P_t to represent the estimated state covariance matrixs for the prior estimation and the posterior estimation respectively. The Kalman gain K_t is used to balance the trade-off between prior estimation and measurement. H represents

[1]Eq. (13) corresponds to the covariance estimation and listed here for better presentation.

the measurement matrix that describes relationship between measurement and original frame, and we use an identity matrix in our implementation. Q_{t-1} and U_t represent the covariance matrixs for process noise and the measurement noise, and these two matrixs are assumed to be constant over time.

Discussion: In our approach, the Kalman gain updates at each time step, which provides the potential to combine the prior estimation and measurement in an adaptive way. In case the restoration error of the previous frame is generated, it means that the prior estimation is not reliable enough. However, Z_t can still provide effective information for the restoration of current frame, which improves the robustness of the proposed framework. Therefore, the error accumulation problem can be alleviated in our deep Kalman model.

H. Training Strategy

In order to build this deep non-local Kalman model, we optimize the proposed three networks in the following way. First, for the temporal non-local residual network with trainable parameters θ_f, the optimization procedure is formulated as follows,

$$\mathcal{L}_f(\theta_f) = ||X_t - \mathcal{F}(\hat{X}_{t-1}, X_t^c; \theta_f)||_2^2, \tag{17}$$

It is notable that previous restored frame $\hat{X}_{t-1}$ is required for obtaining the posterior estimation in the current time step. In order to solve this chicken-and-egg problem, a straightforward approach is to train a video clip (5 frames or more) in one iteration, then we can iteratively optimize the current frame based on the previous restored frame. However, training multiple video clips in one iteration is a huge challenge for GPU memory size. In contrast, an on-line updating scheme is proposed in our framework. First, a buffer is built to save the restored image in each iteration. Then, in each iteration, we use the restored previous frame in the buffer and decoded frame at current time step to optimize our proposed temporal non-local residual network. The restored frame of current time will be saved into the buffer.

Then we fix the trainable parameters θ_f and optimize the linearization network $\mathcal{G}(\theta_m)$ based on the following loss function,

$$\mathcal{L}_m(\theta_m) = ||\hat{X}_t^- - \mathcal{G}(\hat{X}_{t-1}, X_t^c; \theta_m)\hat{X}_{t-1}||_2^2, \tag{18}$$

In our implementation, a small patch size (4×4) is used to reduce the computational complexity when optimizing θ_m.

Then, the measurement net is optimized based on the following loss function,

$$\mathcal{L}_z(\theta_z) = ||X_t - \mathcal{M}(X_t^c; \theta_z)||_2^2 \tag{19}$$

where X_t represents the original frame, X_t^c is the compressed frame.

Implementation Details. We adopt the Adam solver [57] with the initial learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$ to train our model. The learning rate is divided by 10 after every 20 epochs. In order to stabilize the training procedure, the gradient clip technique is employed with global norm 0.001. The batch size is set to 20. The trainable parameters are initialized based on [58].

The training procedures are summarized as follows. First, the prediction network is optimized based on the loss function $\mathcal{L}_f$ in Eq. (17) for 40 epochs. Then we fix the parameters θ_f and train the linearization network by using the loss $\mathcal{L}_m$ in Eq. (18). Finally, the measurement network is optimized based on the loss function $\mathcal{L}_z$ in Eq. (19) for another 40 epochs.

IV. Experiments

In this section, we perform extensive experiments to demonstrate the effectiveness of the proposed deep non-local Kalman network. The experimental results are evaluated on the Vimeo-90K [22], HEVC test sequences [59] and MPI Sintel dataset [60]. The whole system is implemented based on the Tensorflow [61] platform. The training time on two Titan X GPUs is about 26 hours.

A. Experimental Setup

Training Dataset. In our experiments, we use the Vimeo-90K dataset [22] as the training dataset. Vimeo-90K dataset [22] is widely used for low-level vision tasks, such as video super-resolution (SR), video denoising and video artifact reduction. There are 4,278 videos with 89,800 independent clips in the dataset, the resolution of video clip is 448×256. We follow [22] and use 64,612 clips for training and 7,824 clips for performance evaluation. In this section, both PSNR and SSIM [62] are utilized as the evaluation metrics for video compression artifact reduction task.

In addition, to generate the compressed frames, we use HEVC codec (x265 [63]) with quantization parameter $qp = 32$ and $qp = 37$ to generate the compressed frames. And the video format is RGB 4:4:4. We disable the loop filters for HEVC in Table II and Table III. We also follow the setting in [22] and compress the video by JPEG2000 codec with quality $q = 20$ and $q = 40$. The tile size of JPEG2000 is 256x256. The PSNR/SSIM are evaluated on the RGB channels by using function from MATLAB. In the following experiments, we train different models for different codecs or quality levels.

In addition, in order to make a fair comparison with MFQE [46], DS-CNN [21], we also train our model by using the same codec setting and training dataset in [46]. Specifically, we collect 70 uncompressed video sequences from Xiph website [64] and JCT-VC [59]. 60 sequences are used for training and 10 sequences are used for testing. The compressed video sequences are generated by HM [59] with quantization parameter $qp = 37$. The video format is YUV 4:2:0. We use HM with low-delay P setting. The loop filters are used. The PSNR is evaluated in Y channel.

B. Experimental Results

Comparison with the State-of-the-art Methods. To demonstrate the effectiveness of our approach, we compare the non-local Kalman model with several recent image and video artifact reduction methods: ARCNN [18], DnCNN [17],V-BM4D [66], Toflow [22], Li *et al.* [65], DS-CNN [21], MFQE

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2943214, IEEE Transactions on Image Processing

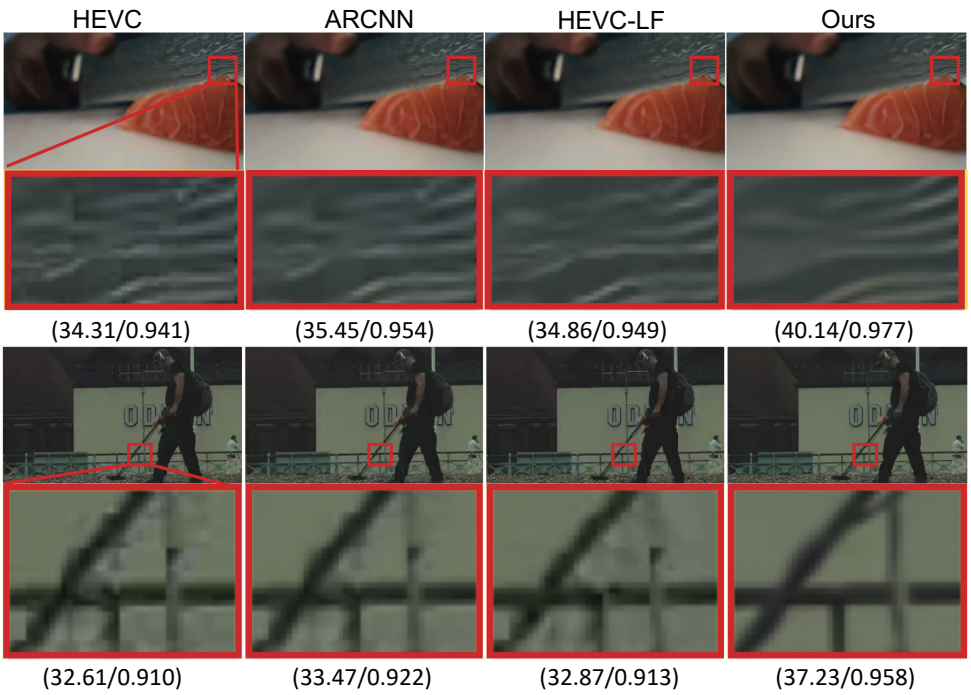


Fig. 5. Quantitative (PSNR/SSIM) and visual comparison of different methods for HEVC artifact reduction on the Vimeo dataset at qp=37.

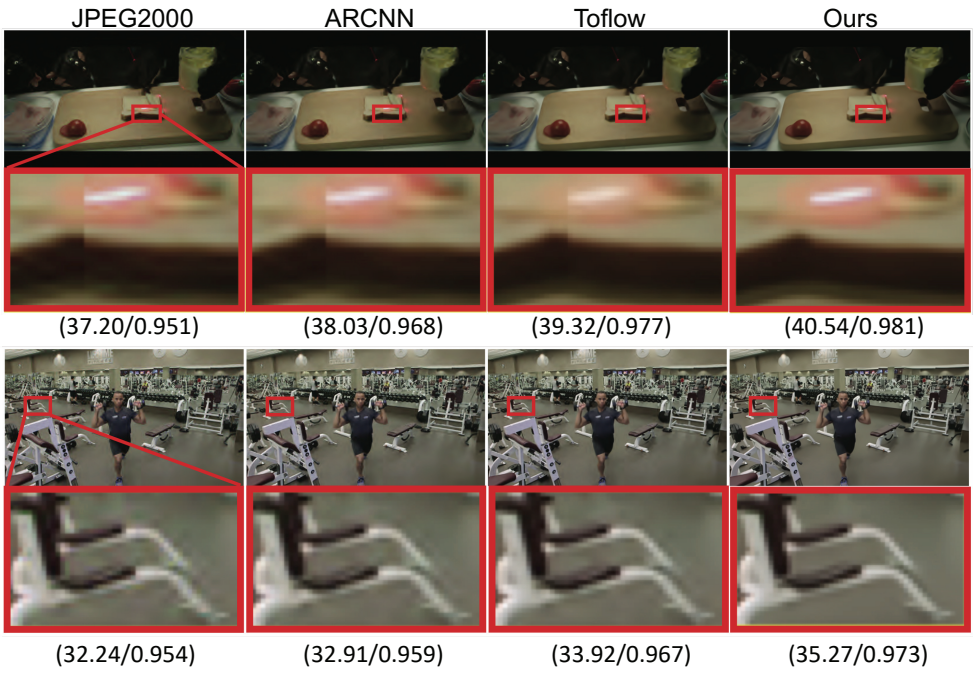


Fig. 6. Quantitative (PSNR/SSIM) and visual comparison of JPEG2000 artifact reduction on the Vimeo dataset for q=20.

[46] and DKF [27]. In addition, modern video codecs already have a default artifact reduction scheme. For example, HEVC utilizes loop filter [1] (HEVC-LF) to reduce the blocking artifacts. This technique is also included for comparison.

For ARCNN [18] and DnCNN [17] , we use the code provided by the authors and train their models on the Vimeo-90k dataset. For V-BM4D and Toflow, we directly cited their results in [22]. The results of HEVC-LF are generated by enabling loop filter and SAO [1] in HEVC codec (x265). For the results of Li *et al.* [65], DS-CNN [21] and MFQE [46], we directly cited their results in [46].

In order to evaluate the performance on the Vimeo dataset, we follow [22] and only evaluate the 4^{th} frame of each clip in the Vimeo testing dataset. The quantitative results are reported in Table II and Table III. As we can see, our proposed approach outperforms the state-of-the-art methods in term of PSNR and SSIM. For example, as shown in the first row in Table II, our method has a 0.36dB improvement when compared with

TABLE I
AVERAGE PSNR GAIN (DB) FOR THE TEST SEQUENCES IN [46]. 1: PEOPLEONSTREET 2: TUNNELFLAG 3: KIMONO 4:BARSCENE 5: VIDYO1 6: VIDYO3 7: VIDYO4 8: BASKETBALLPASS 9: RACEHORSES 10:MAD).

Seq.	ARCNN [18]	DnCNN [17]	Li *et al.* [65]	DS-CNN [21]	MFQE [46]	Ours
1	0.1287	0.1955	0.2523	0.4762	0.7716	**0.8081**
2	0.0718	0.1888	0.2857	0.4228	0.6042	**0.8715**
3	0.1095	0.1328	0.1872	0.2394	**0.4715**	0.3850
4	0.1304	0.2084	0.2170	0.3173	0.4381	**0.4618**
5	0.1900	0.2936	0.3645	0.3252	0.5496	**0.7330**
6	0.1522	0.1944	0.2630	0.3728	0.5980	**0.6210**
7	0.1455	0.2224	0.2570	0.2777	0.3898	**0.4612**
8	0.1305	0.2424	0.2939	0.2790	0.4838	**0.5202**
9	0.1573	0.2588	0.3034	0.2720	0.3935	**0.4670**
10	0.1490	0.2509	0.2926	0.2498	0.4019	**0.4740**
Avg.	0.1364	0.2188	0.2717	0.3232	0.5102	**0.5802**

the learning based image artifact reduction DnCNN [17]. In addition, our method also outperforms the DKF method [27] by 0.13dB, which demonstrates the effectiveness of non-local prior information.

In Table I, we provide the evaluation results on the testing dataset in [46]. It is observed that our DNKF model performs better than approaches in MFQE [46] and DS-CNN [21]. And our approach only uses one previous reference frame while MFQE [46] used two reference frames. Therefore, it is possible to improve the performance further by employing more reference frames.

Fig. 5 and Fig. 6 provides the qualitative comparisons of ARCNN [18], Toflow [22], HEVC-LF [1] and ours. It is obvious that compressed HEVC/JPEG2000 frames have annoying artifact, such as blockness. Based on the proposed Kalman model, our framework can remove these artifacts while other methods may fail. For example, the railing (the fourth row in Fig. 5) and the equipment (the fourth row in Fig. 6) both have complex texture and structure, our method can well restore these complex regions while other baseline methods still have visible artifact.

Ablation Study of Measurement Network(MN). In this subsection, we investigate the effectiveness of the proposed measurement network. Our measurement network is composed of several residual blocks and a spatial non-local network module. Note that the output of our measurement network itself can be readily used as the artifact reduction result. So the results in this subsection are obtained without using the prediction network. In order to validate that the non-local module can serve as important prior information for improving the performance, we train another model with the same architecture but without spatial non-local network (SNL) in the measurement network. Quantitative results on the Vimeo-90k dataset are listed in Table IV. When compared with our simplified model without spatial non-local network(see the 1^{st} row), our model with spatial non-local network (MN+SNL, see the 2^{nd} row) can boost the performance by 0.1dB in term of PSNR. It demonstrates that incorporating strong non-local prior information can improve the restoration performance.

Ablation Study on the Prediction Network (PN). We further evaluate the effectiveness of the temporal non-local residual network. Note that the output of our prediction network itself can be also readily used for the video artifact reduction. So the results in this subsection are obtained without using the measurement network.

First, in order to validate the effectiveness of the proposed temporal non-local (TNL) module, we perform another experiment by removing the TNL in prediction network (see 3^{rd} row). It is observed that TNL module improves the performance with 0.12dB from 35.60dB to 35.72dB.

In order to validate the effectiveness of the recursive filtering, we train another model, which utilizes the same network architecture as our prediction network but the input are X_t^c and $\hat{X}_{t-1}$. Namely, it restores the current frame by employing previous restored frames. The quantitative results are reported in Table IV. When compared with our simplified model without using recursive filtering (PN+TNL) (see the 4^{th} row), our model with recursive filtering (PN+TNL+RF, see the 5^{th} row) can significantly improve the quality of restored frame by 0.15dB from 35.72dB to 35.87 dB in terms of PSNR. It shows that our recursive filtering scheme can effectively leverage information from previous restored frames, which provides more accurate pixel information.

It is worth mentioning that the result in the last row of Table IV is the best as we combine the outputs from both the measurement network and the prediction network through the Kalman update process. In addition, we also find that the temporal information is crucial for the video restoration. For example, the results from prediction network (PN+TNL+RF, see the 5^{th} row) is better the measurement network (MN+SNL, see the 2^{nd} row). The results of our previous DKN model without prediction residual is provided in the 7^{th} row. It validates that our proposed spatial/temporal non-lcoal network can improve the performance for the video restoration.

Flow based image warp. In order to exploit the temporal information, existing video restoration approaches use the optical flow to warp the reference frames [22], [24]. We also compare our non-local operation with this traditional pipeline. Specifically, we warp the reference frame $\hat{X}_{t-1}$ based on the estimated flow from [67] and concatenate the warped image and decoded frame X_t^c as the input for prediction network. Our proposed temporal non-local residual network (see the 5^{th} row) performs better than the flow based approaches (PN+FLOW, see the 6^{th} row). In addition, our method is more lightweight. For example, the flow based approaches need extra optical flow network and the parameters of Spynet [67] are about 1.44M.

Cross dataset validation. To investigate the generalization ability of our model, we also perform experiments by evaluating our approach on other datasets. In this subsection, we train our model on the Vimeo dataset and evaluate it on the HEVC standard sequences [59] and MPI Sintel Flwo dataset [60]. The experimental results provided in Table V and Table VI show that our approach performs better than the state-of-the-art methods.

Video denoising. Since our approach does not rely on task specific prior, such as the prediction error in [27], our framework can be easily extended to other video restoration tasks (e.g., video denoising). We have conducted a new experiment for video denoising by using the current approach. Specifically,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TIP.2019.2943214, IEEE Transactions on Image Processing

TABLE II
AVERAGE PSNR/SSIM RESULTS ON THE VIMEO TEST SEQUENCES FOR HEVC ARTIFACT REDUCTION (QP=32,37).

Dataset	Setting	Compressed	ARCNN [18]	DnCNN [17]	HEVC-LF [1]	DKF [27]	Ours
Vimeo	qp=32	33.69/0.944	34.87/0.954	35.58/0.961	34.19/0.950	35.81/0.962	**35.94/0.963**
	qp=37	31.79/0.920	32.54/0.930	33.01/0.936	31.98/0.923	33.23/0.939	**33.41/0.940**

TABLE III
AVERAGE PSNR/SSIM RESULTS ON THE VIMEO DATASET FOR JPEG2000 ARTIFACT REDUCTION (Q=20,40).

Dataset	Setting	Compressed	ARCNN [18]	DnCNN [17]	V-BM4D [66]	Toflow [22]	DKF [27]	Ours
Vimeo	q=20	35.61/0.956	36.11/0.960	37.26/0.967	35.75/0.959	36.92/0.966	37.93/0.971	**38.14/0.971**
	q=40	33.51/0.936	34.21/0.944	35.22/0.953	33.99/0.940	34.97/0.953	35.88/0.958	**36.02/0.959**

TABLE IV
ABLATION STUDY OF THE PROPOSED DEEP NON-LOCAL KALMAN FILTERING METHOD ON THE VIMEO-90K DATASET. THE PSNR OF THE COMPRESSED VIDEO IS 33.69DB.

No.	MN	SNL	PN	TNL	RF	FLOW	PSNR
1	✓						35.47
2	✓	✓					35.57
3			✓				35.60
4			✓	✓			35.72
5			✓	✓	✓		35.87
6			✓			✓	35.64
7	✓		✓		✓		35.73
8	✓	✓	✓	✓	✓		**35.94**

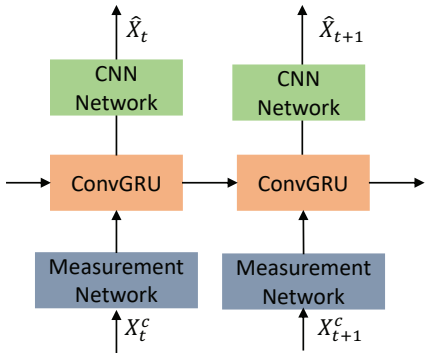


Fig. 7. The architecture of RNN based approach for video restoration. 'ConvGRU' represents the convolutional GRU module [68].

the input X_t^c and $\hat{X}_{t-1}$ represent the noisy current frame and restored previous frame. Experimental results in Table VII demonstrate that our method can improve the performance significantly. For example, compared with the learning based image denoising method DnCNN [17], our approach has 1.29dB gain.

Comparison with the RNN based approach. The traditional RNN architectures also try to utilize the information/feature from previous time step in a recursive way. This characteristic means that the RNN based approaches can also be utilized for video restoration. In this subsection, we design a new recursive approach for video restoration task based on convolutional gated recurrent unit (convGRU) [68]. The detailed architecture is shown in Fig. 7. We use the recurrent network to completely replace the Kalman filter in Fig. 2. Specifically, the same CNN architecture is used to extract the features from the distorted frames at each time step and a convGRU module is used to restore the original image based on these features. The corresponding result in Fig. 7 is 35.22dB, while ours is better (35.94dB). Our observation is that it is difficult to train the recurrent network, especially for the low-level task. Our pipeline makes it easier to learn the network by combining both measurement and prior estimation and using the non-local prior information.

Neighbouring size of non-lcoal network. We also investigate the influence of neighbouring size of non-local network in Table VIII. The proposed method achieves the best performance when size is 10. A possible explanation is that when the neighbouring size is larger, more irrelevant content is involved to refine the feature, which may degrade the performance.

Fusion stage. For the temporal non-local residual network in Fig. 4, we insert the non-local block after the third residual block (Middle Fusion). In fact, we can also insert the non-local block before the first residual block (Early Fusion) or after the last residual block (Late Fusion). Experimental results show that middle fusion (35.72dB) achieves better performance than early fusion (35.61dB) or late fusion (35.65dB).

Flickering evaluation. Flickering artifact is one of the most important factors for evaluating the quality of videos. In order to measure the visual quality of the proposed method, we employ the existing flicker metrics [69] and compare the video quality from different restoration methods (see Table IX). The smaller the score value, the smaller the flickering artifacts, In Table IX, we provide the flickering score for the MPI dataset. It validates that our method also has the advantage of reducing the flicker artifact of the compressed video. For example, the flickering score of compressed video is 0.9017, while the restores video in our approach is 0.6838.

Computational complexity. We perform video restoration on a sever with Titan X GPU. For the video frame with resolution 448x256, the inference time for TOFlow is 1.5s, while the corresponding time for our non-local Kalman model is 0.18s. The reason is that TOFlow used multiple reference frames and had to calculate multiple optical flow fields, which increases the computational complexity significantly. In addition, the inference time for DnCNN and ARCNN are 0.077s and

TABLE V
AVERAGE PSNR/SSIM RESULTS EVALUATED ON HEVC STANDARD TEST SEQUENCES (CLASS E) AND MPI SINTEL FLOW DATATSET FOR VIDEO ARTIFACT REDUCTION (HEVC, QP=32) FOR CROSS DATASET VALIDATION.

Test Dataset	Compressed	HEVC-LF [1]	DnCNN [17]	DKF [27]	Ours
HEVC Sequneces	35.62/0.974	36.13/0.975	36.60/0.970	36.72/0.973	**36.99/0.977**
MPI Sintel datast	32.87/0.944	33.30/0.951	34.20/0.959	34.21/0.960	**34.36/0.960**

TABLE VI
AVERAGE PSNR/SSIM RESULTS EVALUATED ON HEVC STANDARD TEST SEQUENCES (CLASS C) AND MPI SINTEL FLOW DATASET FOR VIDEO ARTIFACT REDUCTION (JPEG2000, Q=20) FOR CROSS DATASET VALIDATION.

Test Dataset	Compressed	Toflow [22]	DnCNN [17]	DKF [27]	Ours
HEVC Sequneces	32.17/0.948	32.37/0.948	33.19/0.953	33.83/0.958	**33.96/0.959**
MPI Sintel datast	34.91/0.959	34.78/0.959	36.40/0.969	37.01/0.973	**37.12/0.974**

TABLE VII
AVERAGE PSNR(DB) RESULTS FOR VIDEO DENOISING ON THE VIMEO DATASET.

Sigma	V-BM4D [66]	DnCNN [17]	Ours
15	35.80	37.15	38.44
20	34.39	34.95	36.00

TABLE VIII
AVERAGE PSNR(DB) RESULTS FOR NON-LOCAL NETWORK WITH DIFFERENT SIZES.

Size	3	5	10	15
PSNR	36.61	35.68	35.72	35.71

0.011s, respectively. The parameters of our model is 1.2M.

V. CONCLUSIONS

In this paper, we have proposed a deep non-local Kalman filtering network for video artifact reduction. The video compression artifact reduction has been formulated as a Kalman filtering procedure, where several deep neural networks are designed to predict the states and estimations. Therefore, the recursive nature of Kalman filtering and representation learning ability of neural network are both exploited in our framework. In addition, the non-local prior information is incorporated to obtain high quality reconstruction. Our methodology has been successfully extended to solve other low-level computer vision tasks, such as denoising. Experimental results have demonstrated the superiority of our deep Kalman filtering network over the state-of-the-art methods.

ACKNOWLEDGMENT

This work is supported in part by National Natural Science Foundation of China(61771306) Natural Science Foundation of Shanghai(18ZR1418100), Chinese National Key S&T Special Program(2013ZX01033001-002-002), Shanghai Key Laboratory of Digital Media Processing and Transmissions(STCSM 18DZ2270700). This research is also partially supported by the Australian Research Council Future Fellowship under Grant FT180100116.

TABLE IX
FLICKERING SCORE ON THE MPI DATASET. THE SMALLER VALUE REPRESENTS THE SMALLER FLICKER ARTIFACT.

Methods	Flickering Score [69]
Compressed	0.9017
HEVC-LF [1]	0.8132
DnCNN [17]	0.6932
Ours	**0.6838**

REFERENCES

[1] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[2] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the h. 264/avc standard," *IEEE Transactions on circuits and systems for video technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[3] G. Lu, X. Zhang, L. Chen, and Z. Gao, "Novel integration of frame rate up conversion and hevc coding based on rate-distortion optimization," *IEEE Transactions on Image Processing*, vol. 27, no. 2, pp. 678–691, 2018.

[4] M.-Y. Shen and C.-C. J. Kuo, "Review of postprocessing techniques for compression artifact removal," *Journal of Visual Communication and Image Representation*, vol. 9, no. 1, pp. 2–14, 1998.

[5] H. C. Reeve and J. S. Lim, "Reduction of blocking effects in image coding," *Optical Engineering*, vol. 23, no. 1, p. 230134, 1984.

[6] C. Jung, L. Jiao, H. Qi, and T. Sun, "Image deblocking via sparse representation," *Signal Processing: Image Communication*, vol. 27, no. 6, pp. 663–677, 2012.

[7] I. Choi, S. Kim, M. S. Brown, and Y.-W. Tai, "A learning-based approach to reduce jpeg artifacts in image matting," in *ICCV*, 2013, pp. 2880–2887.

[8] H. Chang, M. K. Ng, and T. Zeng, "Reducing artifacts in jpeg decompression via a learned dictionary," *IEEE Transactions on Signal Processing*, vol. 62, no. 3, pp. 718–728, 2014.

[9] X. Liu, X. Wu, J. Zhou, and D. Zhao, "Data-driven sparsity-based restoration of jpeg-compressed images in dual transform-pixel domain." in *CVPR*, vol. 1, no. 2, 2015, p. 5.

[10] W. Ouyang and X. Wang, "Joint deep learning for pedestrian detection," in *ICCV*, 2013, pp. 2056–2063.

[11] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy *et al.*, "Deepid-net: Deformable deep convolutional neural networks for object detection," in *CVPR*, 2015, pp. 2403–2412.

[12] L. Wang, W. Ouyang, X. Wang, and H. Lu, "Visual tracking with fully convolutional networks," in *ICCV*, 2015, pp. 3119–3127.

[13] R. Zhao, W. Ouyang, and X. Wang, "Unsupervised salience learning for person re-identification," in *CVPR*, 2013, pp. 3586–3593.

[14] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *ECCV*. Springer, 2014, pp. 184–199.

[15] ——, "Image super-resolution using deep convolutional networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[16] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *CVPR*, 2017, pp. 4539–4547.

[17] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[18] C. Dong, Y. Deng, C. Change Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *ICCV*, 2015, pp. 576–584.

[19] J. Guo and H. Chao, "Building dual-domain representations for compression artifacts reduction," in *ECCV*. Springer, 2016, pp. 628–644.

[20] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo, "Deep generative adversarial compression artifact removal," *arXiv preprint arXiv:1704.02518*, 2017.

[21] R. Yang, M. Xu, and Z. Wang, "Decoder-side hevc quality enhancement with scalable convolutional neural network," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2017, pp. 817–822.

[22] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *arXiv preprint arXiv:1711.09078*, 2017.

[23] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia, "Detail-revealing deep video super-resolution," in *ICCV*, 2017, pp. 22–29.

[24] D. Liu, Z. Wang, Y. Fan, X. Liu, Z. Wang, S. Chang, and T. Huang, "Robust video super-resolution with learned temporal dynamics," in *CVPR*.

[25] J. Caballero, C. Ledig, A. Aitken, A. Acosta, J. Totz, Z. Wang, and W. Shi, "Real-time video super-resolution with spatio-temporal networks and motion compensation," in *CVPR*, 2017.

[26] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *CVPR*, vol. 2. IEEE, 2005, pp. 60–65.

[27] G. Lu, W. Ouyang, D. Xu, X. Zhang, Z. Gao, and M.-T. Sun, "Deep kalman filtering network for video compression artifact reduction," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 568–584.

[28] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1395–1411, 2007.

[29] X. Zhang, R. Xiong, X. Fan, S. Ma, and W. Gao, "Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 4613–4626, 2013.

[30] Z. Wang, D. Liu, S. Chang, Q. Ling, Y. Yang, and T. S. Huang, "D3: Deep dual-domain based fast restoration of jpeg-compressed images," in *CVPR*, 2016, pp. 2764–2772.

[31] P. Svoboda, M. Hradis, D. Barina, and P. Zemcik, "Compression artifacts removal using convolutional neural networks," *arXiv preprint arXiv:1605.00366*, 2016.

[32] X.-J. Mao, C. Shen, and Y.-B. Yang, "Image denoising using very deep fully convolutional encoder-decoder networks with symmetric skip connections," *arXiv preprint*, 2016.

[33] J. Guo and H. Chao, "One-to-many network for visually pleasing compression artifacts reduction," in *CVPR*, 2017, pp. 3038–3047.

[34] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," *arXiv preprint*, 2017.

[35] J. R. Chang, C.-L. Li, B. Poczos, B. V. Kumar, and A. C. Sankaranarayanan, "One network to solve them allsolving linear inverse problems using deep projection models," *arXiv preprint*, 2017.

[36] S. A. Bigdeli, M. Zwicker, P. Favaro, and M. Jin, "Deep mean-shift priors for image restoration," in *NIPS*, 2017, pp. 763–772.

[37] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *GlobalSIP*. IEEE, 2013, pp. 945–948.

[38] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

[39] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Bm3d image denoising with shape-adaptive principal component analysis," in *SPARS'09-Signal Processing with Adaptive Sparse Structured Representations*, 2009.

[40] R. Liao, X. Tao, R. Li, Z. Ma, and J. Jia, "Video super-resolution via deep draft-ensemble learning," in *ICCV*, 2015, pp. 531–539.

[41] A. Kappeler, S. Yoo, Q. Dai, and A. K. Katsaggelos, "Video super-resolution with convolutional neural networks," *IEEE Transactions on Computational Imaging*, vol. 2, no. 2, pp. 109–122, 2016.

[42] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *NIPS*, 2015, pp. 2017–2025.

[43] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy, "Edvr: Video restoration with enhanced deformable convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.

[44] Y. Dai, D. Liu, and F. Wu, "A convolutional neural network approach for post-processing in hevc intra coding," in *International Conference on Multimedia Modeling*. Springer, 2017, pp. 28–39.

[45] T. Wang, M. Chen, and H. Chao, "A novel deep learning-based method of improving coding efficiency from the decoder-end for hevc," in *2017 Data Compression Conference (DCC)*. IEEE, 2017, pp. 410–419.

[46] R. Yang, M. Xu, Z. Wang, and T. Li, "Multi-frame quality enhancement for compressed video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6664–6673.

[47] Z. Guan, Q. Xing, M. Xu, R. Yang, T. Liu, and Z. Wang, "Mfqe 2.0: A new approach for multi-frame quality enhancement on compressed video," *arXiv preprint arXiv:1902.09707*, 2019.

[48] S. D.-C. Shashua and S. Mannor, "Deep robust kalman filter," *arXiv preprint arXiv:1703.02310*, 2017.

[49] R. G. Krishnan, U. Shalit, and D. Sontag, "Deep kalman filters," *arXiv preprint arXiv:1511.05121*, 2015.

[50] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[51] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, "Non-local recurrent network for image restoration," in *Advances in Neural Information Processing Systems*, 2018, pp. 1680–1689.

[52] A. Davy, T. Ehret, G. Facciolo, J.-M. Morel, and P. Arias, "Non-local video denoising by cnn," *arXiv preprint arXiv:1811.12758*, 2018.

[53] S. Li, D. Zeng, Z. Bian, and J. Ma Sr, "Low-dose cerebral ct perfusion restoration via non-local convolution neural network: initial study," in *15th International Meeting on Fully Three-Dimensional Image Reconstruction in Radiology and Nuclear Medicine*, vol. 11072. International Society for Optics and Photonics, 2019, p. 1107224.

[54] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Journal of Basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960.

[55] S. S. Haykin *et al.*, *Kalman filtering and neural networks*. Wiley Online Library, 2001.

[56] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *ECCV*. Springer, 2016, pp. 630–645.

[57] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[58] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[59] F. Bossen *et al.*, "Common test conditions and software reference configurations," *JCTVC-L1100*, vol. 12, 2013.

[60] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black, "A naturalistic open source movie for optical flow evaluation," in *European Conf. on Computer Vision (ECCV)*, ser. Part IV, LNCS 7577, A. Fitzgibbon et al. (Eds.), Ed. Springer-Verlag, Oct. 2012, pp. 611–625.

[61] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467*, 2016.

[62] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

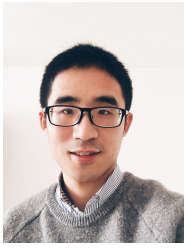
[63] "x265 hevc encoder / h.265 video codec." http://x265.org, accessed: 2018-10-30.

[64] "Xiph dataset." https://media.xiph.org/video/derf/, accessed: 2018-09-30.

[65] K. Li, B. Bare, and B. Yan, "An efficient deep convolutional neural networks model for compressed image deblocking," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2017, pp. 1320–1325.

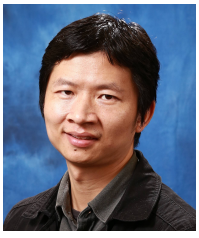
[66] M. Maggioni, G. Boracchi, A. Foi, and K. Egiazarian, "Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms," *IEEE Transactions on Image Processing*, vol. 21, no. 9, pp. 3952–3966, 2012.

[67] A. Ranjan and M. J. Black, "Optical flow estimation using a spatial pyramid network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4161–4170.
[68] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in neural information processing systems*, 2015, pp. 802–810.
[69] D. T. Vo, T. Q. Nguyen, S. Yea, and A. Vetro, "Adaptive fuzzy filtering for artifact reduction in compressed images and videos." *IEEE Trans. Image Processing*, vol. 18, no. 6, pp. 1166–1178, 2009.

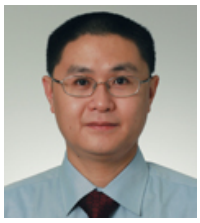


Guo Lu received his B.S. degree in electronic engineering, from Ocean University of China, Shandong, China, in 2014. He is currently pursing the Ph.D. degree from the Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, Shanghai, China. His current research interests include video coding and processing.

Xiaoyun Zhang received her B.S. and M.S. in Applied Mathematics from Xian Jiaotong University in 1998 and 2001, and Ph.D. degree in pattern recognition from Shanghai Jiao Tong University, China, in 2004. Her Ph.D thesis has been nominated as National 100 Best Ph.d. Theses of China. Since 2011, she has been an associate professor at Shanghai Jiao Tong University, and was as a visiting scholar with Harvard Medical School in 2017. Her research interests include computer vision, image and video processing, machine learning, medical image analysis and computer assisted intervention.



Wanli Ouyang received the PhD degree in the Department of Electronic Engineering, Chinese University of Hong Kong. Since 2017, he is a senior lecturer with the University of Sydney. His research interests include image processing, computer vision, and pattern recognition. He is a senior member of the IEEE.



Dong Xu received the BE and PhD degrees from University of Science and Technology of China, in 2001 and 2005, respectively. While pursuing the PhD degree, he was an intern with Microsoft Research Asia, Beijing, China, and a research assistant with the Chinese University of Hong Kong, Shatin, Hong Kong, for more than two years. He was a post-doctoral research scientist with Columbia University, New York, NY, for one year. He worked as a faculty member with Nanyang Technological University, Singapore. Currently, he is a professor and chair in Computer Engineering with the School of Electrical and Information Engineering, the University of Sydney, Australia. His current research interests include computer vision, statistical learning, and multimedia content analysis. He was the co-author of a paper that won the Best Student Paper award in the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) in 2010, and a paper that won the Prize Paper award in IEEE Transactions on Multimedia (T-MM) in 2014. He is a fellow of the IEEE.

Li Chen received his B.S. and M.S. degrees, both from Northwestern Polytechnical University at Xian of China in 1998 and 2000, and the Ph.D. Degree from Shanghai Jiao Tong University, China, in 2006, all in electrical engineering. His current research interests include Image and Video Processing, DSP and VLSI for Image and video processing.

Zhiyong Gao received the B.S. and M.S. degrees in electrical engineering from the Changsha Institute of Technology (CIT), Changsha, China, in 1981 and 1984, respectively, and the Ph.D. degree from Tsinghua University, Beijing, China, in 1989. From 1994 to 2010, he took several senior technical positions in England, including a Principal Engineer with Snell & Wilcox, Petersfield, U.K., from 1995 to 2000, a Video Architect with 3DLabs, Egham, U.K., from 2000 to 2001, a Consultant Engineer with Sony European Semiconductor Design Center, Basingstoke, U.K., from 2001 to 2004, and a Digital Video Architect with Imagination Technologies, Kings Langley, U.K., from 2004 to 2010. Since 2010, he has been a Professor with Shanghai Jiao Tong University. His research interests include video processing and its implementation, video coding, digital TV and broadcasting.