Deep Kalman Filtering Network for Video Compression Artifact Reduction

Guo Lu[1], Wanli Ouyang[2,3], Dong Xu[2](✉), Xiaoyun Zhang[1], Zhiyong Gao[1], and Ming-Ting Sun[4]

[1] Shanghai Jiao Tong University, Shanghai, China
{luguo2014,xiaoyun.zhang,zhiyong.gao}@sjtu.edu.cn
[2] The University of Sydney, Sydney, Australia
{wanli.ouyang,dong.xu}@sydney.edu.au
[3] SenseTime Computer Vision Research Group, The University of Sydney, Sydney, Australia
[4] University of Washington, Seattle, USA
mts@uw.edu

Abstract. When lossy video compression algorithms are applied, compression artifacts often appear in videos, making decoded videos unpleasant for human visual systems. In this paper, we model the video artifact reduction task as a Kalman filtering procedure and restore decoded frames through a deep Kalman filtering network. Different from the existing works using the noisy previous *decoded* frames as temporal information in the restoration problem, we utilize the less noisy previous *restored* frame and build a recursive filtering scheme based on the Kalman model. This strategy can provide more accurate and consistent temporal information, which produces higher quality restoration results. In addition, the strong prior information of prediction residual is also exploited for restoration through a well designed neural network. These two components are combined under the Kalman framework and optimized through the deep Kalman filtering network. Our approach can well bridge the gap between the model-based methods and learning-based methods by integrating the recursive nature of the Kalman model and highly non-linear transformation ability of deep neural network. Experimental results on the benchmark dataset demonstrate the effectiveness of our proposed method.

Keywords: Compression artifact reduction · Deep neural network
Kalman model · Recursive filtering · Video restoration

1 Introduction

Compression artifact reduction methods aim at generating artifact-free images from lossy decoded images. To store and transfer a large amount of images and videos on the Internet, image and video compression algorithms (e.g., JPEG,

V. Ferrari et al. (Eds.): ECCV 2018, LNCS 11218, pp. 591–608, 2018.
https://doi.org/10.1007/978-3-030-01264-9_35

H.264) are widely used [1–3]. However, these algorithms often introduce undesired compression artifacts, such as blocking, blurring and ringing artifacts. Thus, compression artifact reduction has attracted increasing attention and many methods have been developed in the past few decades.

Early works use manually designed filters [4,5] and sparse coding methods [6–9] to remove compression artifacts. Recently, convolutional neural network (CNN) based approaches have been successfully applied for a lot of computer vision tasks [10–20], such as super-resolution [15,16], denoising [17] and artifact reduction [18–20]. In particular, Dong et al. [18] firstly proposed a four-layer neural network to eliminate the JPEG compression artifacts.

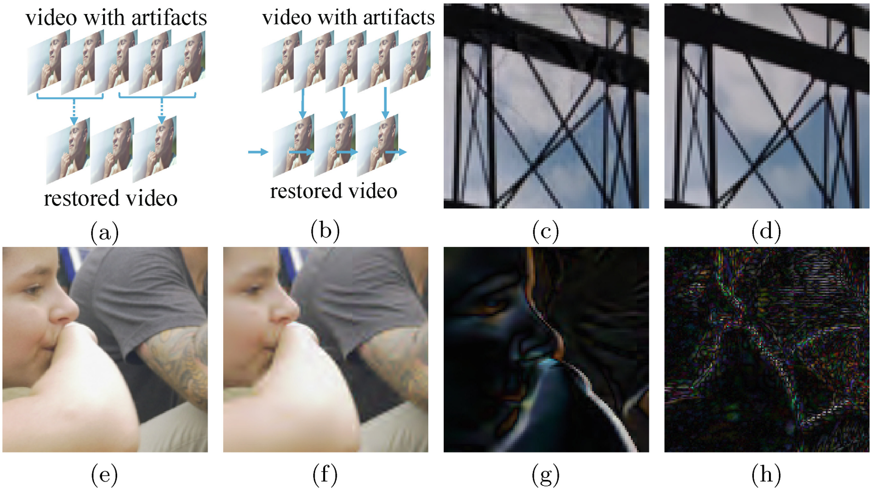


Fig. 1. Different methodologies for video artifact reduction (a) the traditional pipeline without considering previous restored frames. (b) our Kalman filtering pipeline. Artifact reduction results between (c) Xue *et al.* [21] and (d) our proposed deep Kalman filtering network. (e) original frame (f) decoded frame (g) quantized prediction residual (h) distortion between the original frame and the decoded frame.

For the video artifact reduction task, our motivations are two-fold. First, the restoration process for the current frame could benefit from the previous restored frames. A reason is that when compared to the decoded frame, the previous restored frame can provide more accurate information. Therefore temporal information (such as motion clues) from neighbouring frames is more precise and robust, and can provide the potential to further improve the performance. In addition, the dependence of previous restored frames naturally leads to a recursive pipeline for video artifact reduction. Therefore it recursively restores the current frame by potentially utilizing all past restored frames, which means we can leverage effective information propagated from previous estimations. Currently, most of the state-of-the-art deep learning approaches for compression artifact reduction are limited to remove artifacts in a single image [16–19]. Although the video artifact reduction method [21] or video super-resolution methods [22–24] try to combine temporal information for the restoration tasks, their methods ignore the previous restored frame and restore each frame separately as shown in Fig. 1(a). Therefore, the video artifact reduction performance can be further improved by using an appropriate dynamic filtering scheme.

Second, modern video compression algorithms may contain powerful prior information that can be utilized to restore the decoded frame. It has been observed that practical video compression standards are not optimal according to the information theory [9], therefore the resulting compression code streams still have redundancies. It is possible, at least theoretically, to improve the restoration results by exploiting the knowledge hidden in the code streams. For video compression algorithms, *inter prediction* is a fundamental technique used to reduce temporal redundancy. Therefore, the decoded frame consists of two components: prediction frame and quantized prediction residual. As shown in Fig. 1(g)–(h), the distortion between the original frame and the decoded frame has a strong relationship with quantized prediction residual, i.e., the region which has high distortion often corresponds to high quantized prediction residual values. Therefore, it is possible to enhance the restoration by employing this task-specific prior information.

In this paper, we propose a deep Kalman filtering network (DKFN) for video artifact reduction. The proposed approach can be used as a post-processing technique, and thus can be applied to different compression algorithms. Specifically, we model the video artifact reduction problem as a Kalman filtering procedure, which can recursively restore the decoded frame and capture information that propagated from previous restored frames. To perform Kalman filtering for decoded frames, we build two deep convolutional neural networks: prediction network and measurement network. The prediction network aims to obtain *prior estimation* based on the previous restored frame. At the same time, we investigate the quantized prediction residual in the coding algorithms and a novel measurement net incorporating this strong prior information is proposed for robust *measurement*. After that, the restored frame can be obtained by fusing the *prior estimation* and the *measurement* under the Kalman framework. Our proposed approach bridges the gap between model-based methods and learning-based methods by integrating the recursive nature of the Kalman model and highly non-linear transform ability of neural network. Therefore, our approach can restore high quality frames from a series of decoded video frames. To the best of our knowledge, we are the first to develop a new deep convolutional neural network under the Kalman filtering framework for video artifact reduction.

In summary, the main contributions of this work are two-fold. First, we formulate the video artifact reduction problem as a Kalman filtering procedure, which can recursively restore the decoded frames. In this procedure, we utilize the CNN to predict and update the state of Kalman filtering. Second, we employ the quantized prediction residual as the strong prior information for video artifact reduction through deep neural network. Experimental results show that our proposed approach outperforms the state-of-the-art methods for reducing video compression artifacts.

2 Related Work

2.1 Single Image Compression Artifact Reduction

A lot of methods have been proposed to remove the compression artifacts. Early methods [25,26] designed new filters to reduce blocking and ringing artifacts. One of the disadvantages for these methods is that such manually designed filters cannot sufficiently handle the compression degradation and may oversmooth the decoded images. Learning methods based on sparse coding were also proposed for image artifact reduction [8,9,27]. Chang *et al.* [8] proposed to learn a sparse representation from a training image set, which is used to reduce artifacts introduced by compression. Liu *et al.* [9] exploited the DCT information and built a sparsity-based dual domain approach.

Recently, deep convolutional neural network based methods have been successfully utilized for the low-level computer vision tasks. Dong *et al.* [18] proposed artifact reduction CNN (ARCNN) to remove the artifacts from JPEG compression. Inspired by ARCNN, several methods have been proposed to reduce compression artifact by using various techniques, such as residual learning [28], skip connection [28,29], batch normalization [17], perceptual loss [30], residual block [20] and generative adversarial network [20]. For example, Zhang *et al.* [17] proposed a 20-layer neural network based on batch normalization and residual learning to eliminate Gaussian noise with unknown noise level. Tai *et al.* [16] proposed a memory block, consisting of a recursive unit and a gate unit, to explicitly mine persistent memory through an adaptive learning process. In addition, a lot of methods [31–33] were proposed to learn the image prior by using CNN and achieved competitive results for the image restoration tasks.

As mentioned in [9], compression code streams still have redundancies. Therefore, it is possible to obtain a more robust estimation by exploiting the prior knowledge hidden in the encoder. However, most of the previous works do not exploit this important prior information. Although the works in [9,19,27] proposed to combine DCT information, it is not sufficient especially for the video artifact reduction task. In our work, we further exploit the prior information in the code streams and incorporate prediction residual into our framework for robust compression artifact reduction.

2.2 Deep Learning for Video Restoration

Due to the popularity of neural networks for image restoration, several CNN based methods [21,22,34–36] were also proposed for the video restoration tasks. For video super-resolution, Liao *et al.* [34] first generated an ensemble of SR draft via motion compensation, and then used a CNN model to restore the high resolution frame from all drafts. Kappeler *et al.* [35] estimated optical flow and selected the corresponding patches across frames to train a CNN model for video super-resolution. Based on the spatial transformation network (STN) [36], the works in [21–24] aligned the neighboring frames according to the estimated optical flow or transform parameters and increased the temporal coherence for the video SR task. Tao *et al.* [22] achieved sub-pixel motion compensation and resolution enhancement with high performance. Xue *et al.* [21] utilized a joint training strategy to optimize the motion estimation and video restoration tasks and achieved the state-of-the-art results for video artifact reduction.

Compared to the methods for single image compression artifact reduction (see Sect. 2.1), the video restoration methods exploit temporal information. However, these methods process noisy/low-resolution videos separately without considering the previous restored frames. Therefore, they cannot improve video restoration performance by utilizing more accurate temporal information. In our work, we recursively restore each frame in the videos by leveraging the previous restored frame for video artifact reduction. Although the work [37,38] try to combine deep neural network and Kalman filter, they are not designed for the image/video enhancement tasks.

3 Methodology

We first give a brief introduction about the basic Kalman filter and then describe our formulation for video artifact reduction and 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 consider RGB/YUV channels in our implementation. After compression, X_t^c is the decoded frame of X_t. $\hat{X}_t^-$ denotes the prior estimation and $\hat{X}_t$ denotes the posterior estimation for restoring X_t from the decoded frame X_t^c. R_t^c is the quantized prediction residual in video coding standards, such as H.264. R_t is the corresponding unquantized prediction residual.

3.1 Brief Introduction of Kalman Filter

Kalman filter [39] is an efficient recursive filter that estimates the internal state from a series of noisy measurements. In artifact reduction task, the internal state is the original image to be restored, and the noisy measurements can be considered as the images with compression artifacts.

Preliminary Formulation. The Kalman filter model assumes that the state X_t at time t is changed from the state X_{t-1} at time $t-1$ according to

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

where A_t is the transition matrix at time t and w_{t-1} is the process noise. The measurement Z_t of the true state X_t is defined as follows,

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

where H is the measurement matrix and v_t represents the measurement noise. However, the system may be non-linear in some complex scenarios. Therefore, a non-linear model for the transition process in Eq. (1) can be formulated as follows,

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

where $f(\cdot)$ is the non-linear transition model. Linear Kalman filter corresponds to Eqs. (1) and (2). Non-linear Kalman filter corresponds to Eq. (3) and Eq. (2).

Kalman Filtering. As shown in Fig. 2(a), Kalman filtering consists of two steps, prediction and update.

In the *prediction step*, it calculates the prior estimation from the posterior estimation of the previous time step. For non-linear model Eqs. (3) and (2), the prediction step is accomplished by two sub-steps as follows,

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

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

where Q_{t-1} is the covariance of the process noise w_{t-1} at time $t-1$, P_t^- is a covariance matrix used for the update step. A_t in Eq. (5) is defined as the Jacobian matrix of $f(\cdot)$, i.e., $A_t = \frac{\partial f(\hat{X}_{t-1},0)}{\partial X}$ [40]. Prediction procedure for the linear model can be considered as a special case by setting $f(\hat{X}_{t-1}, 0) = A_t \hat{X}_{t-1}$.

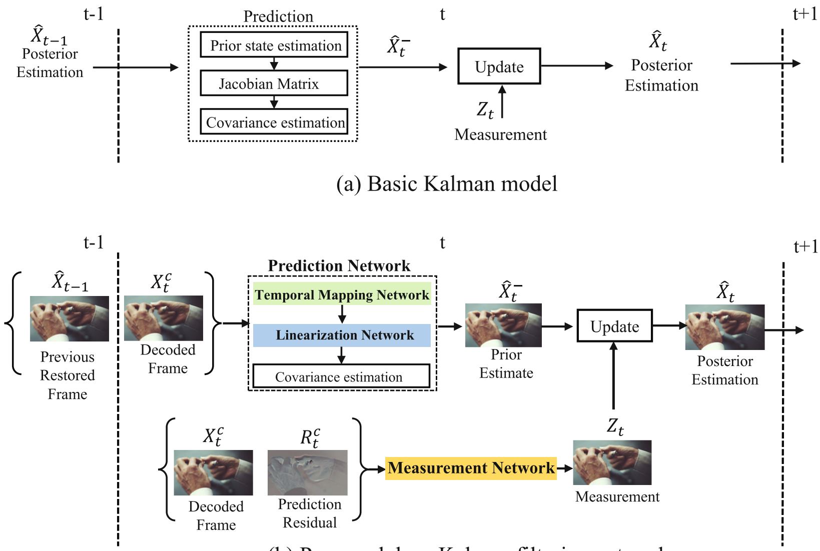
In the *update step*, Kalman filter will calculate the posterior estimate by fusing the prior estimate from the prediction step and the measurement. Details about the update step can be found in Sect. 3.6.

An overview of Kalman filtering is shown in Fig. 2(a). First, the prediction step uses the estimated state $\hat{X}_{t-1}$ at time $t-1$ to obtain a prior state estimation $\hat{X}_t^-$ at time t. Then the prior state estimation $\hat{X}_t^-$ and the measurement Z_t are used by the update step to obtain the posterior estimation of the state, denoted by $\hat{X}_t$. These two steps are performed recursively.

3.2 Overview of Our Deep Kalman Filtering Network

Figure 2(b) shows an overview of the proposed deep Kalman filtering network. In this framework, the previous restored frame $\hat{X}_{t-1}$ and the decoded frame X_t^c are used for obtaining a prior estimate $\hat{X}_t^-$. Then the measurement Z_t obtained from the measurement network and the prior estimate $\hat{X}_t^-$ are used by the update step for obtaining the posterior estimation $\hat{X}_t$.

The proposed DKFN framework follows the Kalman filtering procedure by using the prediction and update steps for obtaining the predicted state $\hat{X}_t$. The main differences to the original Kalman filtering are as follows.



(a) Basic Kalman model

(b) Proposed deep Kalman filtering network

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 uses the decoded frame X_t^c and the quantized prediction residual R_t^c to obtain an initial measurement Z_t. After that, we can build the posterior estimate by fusing the prior estimate $\hat{X}_t^-$ and the measurement Z_t.

First, in the prior estimation sub-step of the prediction step, we use the temporal mapping sub-network as the non-linear function $f(\cdot)$ in Eq. (3) to obtain the prior state estimation. Specifically, the temporal mapping sub-network takes the previous restored frame $\hat{X}_{t-1}$ and the decoded frame X_t^c as the input and generates the prior estimate $\hat{X}_t^-$ for the current frame. Details are given in Sect. 3.3.

Second, in the covariance estimation sub-step, the transition matrix A_t in Eq. (5) is approximated by a linearization sub-network. In conventional non-linear Kalman filter, the Jacobian matrix of the non-linear function $f(\cdot)$ is used to obtain A_t. For our CNN implementation of $f(\cdot)$, however, it is too complex to compute, especially when we need to compute the Jacobian matrix for each pixel location. Therefore, we approximate the calculation of Jacobian matrix by using a linearization sub-network. Details are given in Sect. 3.4.

Third, in the update procedure, we use a measurement 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 Sect. 3.5.

3.3 Temporal Mapping Network

Mathematical Formulation. Based on the temporal characteristic of video sequences, the temporal mapping sub-network is used for implementing the non-linear function $f(\cdot)$ in the prior state estimation as follows:

$$\hat{X}_t^- = \mathcal{F}(\hat{X}_{t-1}, X_t^c; \theta_f), \tag{6}$$

where θ_f are the trainable parameters. Equation(6) indicates that the prior estimation of the current frame X_t is related with its estimated temporal neighbouring frame $\hat{X}_{t-1}$ and its decoded frame X_t^c at the current time step. This formulation is based on the following assumptions. First, temporal evolution characteristic can provide a strong motion clue to predict X_t based on the previous frame $\hat{X}_{t-1}$. Second, due to the existence of complex motion scenarios with occlusion, it is necessary to exploit the information from the decoded frame X_t^c to build a more accurate estimation for X_t. Based on this assumption, our formulation in Eq. (6) adds the decoded frame X_t^c as the extra input to the transition function $f(\cdot)$ defined in Eq. (4).

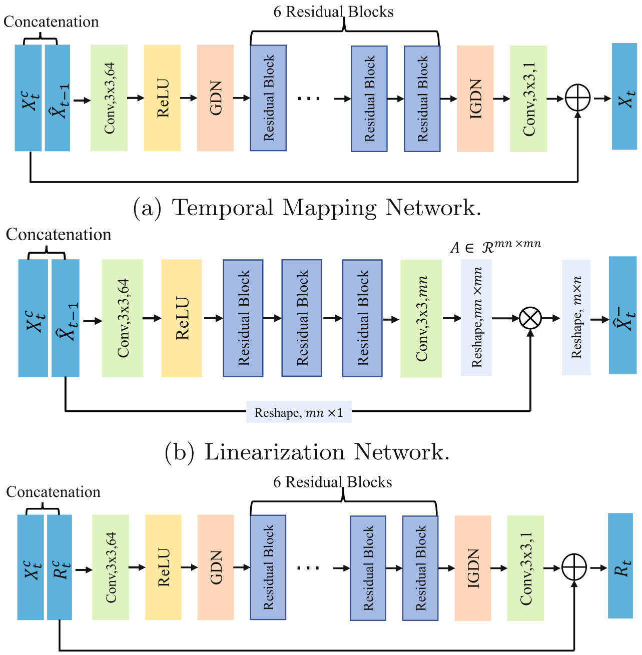
Network Implementation. The temporal mapping sub-network architecture is shown in Fig. 3(a). Specifically, each residual block contains two convolutional layers with the pre-activation structure [41]. We use convolutional layers with 3×3 kernels and 64 output channels in the whole network except the last layer. The last layer is a simple convolutional neural network with one feature map without non-linear transform. Generalized divisive normalization (GDN) and inverse GDN (IGDN) [42] are used because they are well-suited for Gaussianizing data from natural images. More training details are discussed in Sect. 3.7.

3.4 Linearization Network

Linearization network aims to learn a linear transition matrix A_t for the covariance estimation in Eq. (5) adaptively for different image regions. It is non-trivial to calculate the Jacobian matrix of transition function $\mathcal{F}(\cdot)$ and linearize it through Taylor series. Therefore, we use a simple neural network to learn a transition matrix. Specifically, given the prior estimation $\hat{X}_t^-$, previous restored frame $\hat{X}_{t-1}$ and decoded frame X_t^c, the problem is expressed by the following way,

$$\hat{X}_t^- = \mathcal{F}(\hat{X}_{t-1}, X_t^c; \theta_f) \approx \tilde{A}_t \hat{X}_{t-1}, \text{ where } \tilde{A}_t = \mathcal{G}(\hat{X}_{t-1}, X_t^c; \theta_m), \tag{7}$$

$\mathcal{G}(\hat{X}_{t-1}, X_t^c; \theta_m)$ is the linearization network with the parameters θ_m. $\tilde{A}_t$ is the output of this network. The network architecture is shown in Fig. 3(b). Given $\hat{X}_{t-1}, X_t^c \in \mathcal{R}^{mn \times 1}$, the network will generate a transition matrix $\tilde{A}_t \in \mathcal{R}^{mn \times mn}$ as an approximation to A_t in Eq. (5).



(a) Temporal Mapping Network.

(b) Linearization Network.

(c) Measurement network.

Fig. 3. Network architecture of the proposed (a) Temporal Mapping Network (b) Linearization network. (c) Measurement Network. For better illustration, we omit the matrix conversion process of $X_t, \hat{X}_t^c$ and R_t^c from $\mathcal{R}^{mn\times 1}$ to $\mathcal{R}^{m\times n}$. Here 'Conv,3 × 3,64' represents the convolution operation with the 3 × 3 kernel and 64 feature maps. 'Reshape, $m \times n$' is the operation that reshapes one matrix to $m \times n$. $\bigoplus$ and $\bigotimes$ represent element-wise addition and matrix multiplication.

3.5 Measurement Network

Since prediction based coding is one of the most important techniques for video coding standards (such as MPEG2, H.264 or H.265), we take this coding approach into consideration when designing the measurement network. In prediction based coding, the decoded frame X_t^c can be decoupled into two components, i.e., $X_t^c = X_t^p + R_t^c$, where X_t^p and R_t^c represent the prediction frame and the quantized prediction residual, respectively. Note that quantization is only used for the prediction residual and the distortion in compression only comes from R_t^c. In addition, for non-predictive video codecs, such as JPEG2000, we use an existing warp operation [23,36] from the previous decoded frame to the current decoded frame, and the difference between them is considered as the prediction residual R_t^c. For most of video codecs (e.g.,H.264 and H.265), we can directly utilize the quantized prediction residual in the code streams.

We obtain the measurement using the quantized prediction residual R_t^c as follows,

$$Z_t = X_t^p + \hat{R}_t, \text{ where } \hat{R}_t = \mathcal{M}(X_t^c, R_t^c; \theta_z), \tag{8}$$

where $\hat{R}_t$ is the restored residual to remove the effect of quantization so that Z_t is close to the original image X_t. We use a deep neural network as shown in Fig. 3(c) (with same architecture as Fig. 3(a)) for the function $\mathcal{M}(\cdot)$. This network takes the decoded frame X_t^c and the quantized prediction residual R_t^c as the input and estimates the restored residual $\hat{R}_t$. There are two advantages of our formulation for measurement. On one hand, instead of utilizing the decoded frame X_t^c as the measurement, our measurement formulation avoids explicitly modeling the complex relationship between original frames and decoded frames. On the other hand, most of the existing artifact reduction methods can be embedded into our model as the measurement method, which provides a flexible framework to obtain a more accurate measurement value.

3.6 Update Step

Given the prior state estimation $\hat{X}_t^-$ from the temporal mapping network (Sect. 3.3), the transition matrix $\tilde{A}_t$ obtained from the linearization network (Sect. 3.4), and the measurement Z_t obtained from the measurement network (Sect. 3.5), we can use the following steps[1] to obtain the posterior estimation of the restored image:

$$P_t^- = \tilde{A}_t P_{t-1} \tilde{A}_t^{\mathrm{T}} + Q_{t-1}, \tag{9}$$

$$K_t = P_t^- H^{\mathrm{T}} (H P_t^- H^{\mathrm{T}} + U_t)^{-1}, \tag{10}$$

$$\hat{X}_t = \hat{X}_t^- + K_t (Z_t - H \hat{X}_t^-), \tag{11}$$

$$P_t = (I - K_t H) P_t^-, \tag{12}$$

where $\hat{X}_t$ represents the posterior estimation for the image X_t. P_t^- and P_t are the estimated state covariance matrixs for the prior estimation and the posterior estimation respectively. K_t is the Kalman gain at time t. H is the measurement matrix defined in Eq. (2) and is assumed to be an identity matrix in this work. Q_{t-1} and U_t are the process noise covariance matrix and the measurement noise covariance matrix respectively. We assume Q_{t-1} and U_t to be constant over time. For more details about the update procedure of Kalman filtering, please refer to [40].

Discussion. Our approach can solve the error accumulation problem of the recursive pipeline through the adaptive Kalman gain. For example, when the errors accumulate in the previous restored frames, the degree of reliability for prior estimation (i.e., information from the previous frame) will be decreased and the final result will depend more on the measurement (i.e., the current frame).

[1] Eq. (9) corresponds to the covariance estimation and listed here for better presentation.

3.7 Training Strategy

There are three sets of trainable parameters θ_f, θ_m and θ_z in our approach. First, the parameters θ_f in the temporal mapping network are optimized as follows,

$$\mathcal{L}_f(\theta_f) = ||X_t - \mathcal{F}(\hat{X}_{t-1}, X_t^c; \theta_f)||_2^2, \tag{13}$$

Note that in the minimization procedure for Eq. (13), the restored frame of the previous one $\hat{X}_{t-1}$ is required. This leads to the chicken-and-egg problem. A straightforward method is to feed several frames of a clip into the network and train all the input frames in the iteration. However, this strategy increases GPU memory consumption significantly and simultaneous training multi-frames for a video clip is non-trivial. Alternatively, we adopt an on-line update strategy. Specifically, the estimation results $\hat{X}_t$ in each iteration will be saved in a buffer. In the following iterations, $\hat{X}_t$ in the buffer will be used to provide more accurate temporal information when estimating X_{t+1}. Therefore, each training sample in the buffer will be updated in an epoch. We only need to optimize one frame for a video clip in each iteration, which is more efficient.

After that, the parameters θ_f are fixed and we can optimize the linearization network $\mathcal{G}(\theta_m)$ by using the following loss function:

$$\mathcal{L}_m(\theta_m) = ||\hat{X}_t^- - \mathcal{G}(\hat{X}_{t-1}, X_t^c; \theta_m)\hat{X}_{t-1}||_2^2, \tag{14}$$

Note that we use a small patch size (4×4) to reduce the computational cost when optimizing θ_m.

Then, we will train the measurement net and optimize θ_z based on the following loss function,

$$\mathcal{L}_z(\theta_z) = ||X_t - (\mathcal{M}(X_t^c, R_t^c; \theta_z) + X_t^p)||_2^2, \tag{15}$$

Finally, we fine-tune the whole deep Kalman filtering network based on the loss $\mathcal{L}$ defined as follows,

$$\mathcal{L}(\theta) = ||X_t - \hat{X}_t||_2^2, \tag{16}$$

θ are the trainable parameters in the deep Kalman filtering network.

4 Experiments

To demonstrate the effectiveness of the proposed model for video artifact reduction, we perform the experiments on the benchmark dataset Vimeo-90K [21]. Our approach is implemented by using the Tensorflow [43] platform. It takes 22 h to train the whole model by using two Titan X GPUs.

4.1 Experimental Setup

Dataset. The Vimeo-90K dataset [21] is recently built for evaluating different video processing tasks, such as video denoising, video super-resolution (SR) and

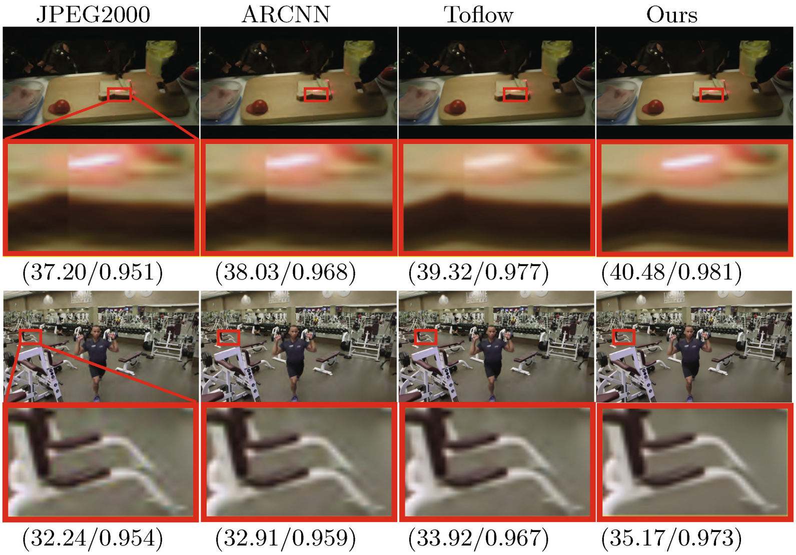


Fig. 4. Quantitative (PSNR/SSIM) and visual comparison of JPEG2000 artifact reduction on the Vimeo dataset for q = 20.

video artifact reduction. It consists of 4,278 videos with 89,800 independent clips that are different from each other in content. All frames have the resolutio of 448×256. For video compression artifact reduction, we follow [21] to use 64,612 clips for training and 7,824 clips for performance evaluation. In this section, PSNR and SSIM [44] are utilized as the evaluation metrics.

To demonstrate the effectiveness of the proposed method, we generate compressed/decoded frames through two coding settings, i.e., codec HEVC (x265) with quantization parameter $qp = 32$ and $qp = 37$ and codec JPEG2000 with quality $q = 20$ and $q = 40$.

Implementation Details. For model training, we use the Adam solver [45] with the initial learning rate of 0.001, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The learning rate is divided by 10 after every 20 epochs. We apply gradient clip with global norm 0.001 to stabilize the training process. The mini-batch size is set to 32. We use the method in [46] for weight initialization. Our approach takes 0.15 s to restore a color image with the size of 448×256.

We first train the temporal mapping network using the loss $\mathcal{L}_f$ in Eq. (13). After 40 epochs, we fix the parameters θ_f and train the linearization network by using the loss $\mathcal{L}_m$. Then we train the measurement network using the loss $\mathcal{L}_z$ in Eq. (15). After 40 epochs, the training loss will become stable. Finally, we fine-tune the whole model. In the following experiments, we train different models for different codecs or quality levels.

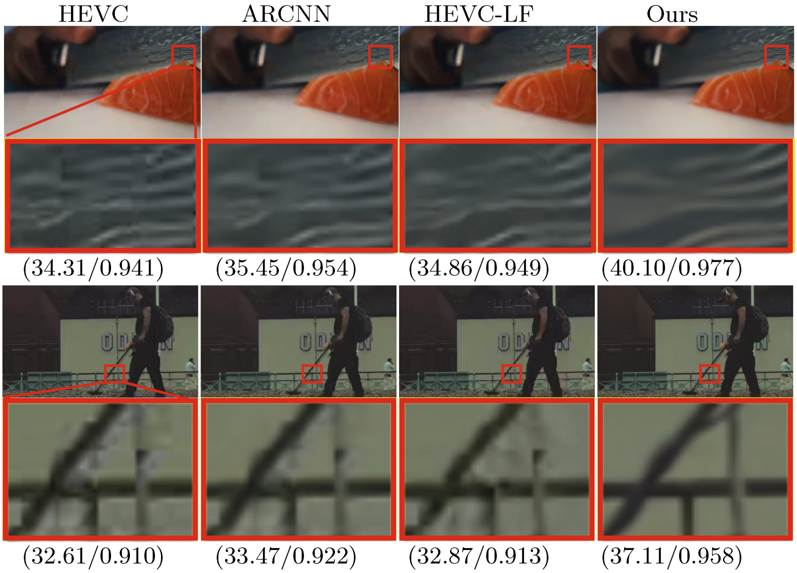


Fig. 5. Quantitative (PSNR/SSIM) and visual comparison of different methods for HEVC artifact reduction on the Vimeo dataset at qp = 37.

4.2 Experimental Results

Comparison with the State-of-the-Art Methods. To demonstrate the effectiveness of our approach, we compare it with several recent image and video artifact reduction methods: ARCNN [18], DnCNN [17], V-BM4D [47] and Toflow [21]. 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 training dataset. For V-BM4D and Toflow, we directly cited their results in [21]. The results of HEVC-LF are generated by enabling loop filter and SAO [1] in HEVC codec (x265). For fair comparison with the existing approaches, we follow [21] and only evaluate the 4^{th} frame of each clip in the Vimeo dataset. The quantitative results are reported in Tables 1 and 2. As we can see, our proposed approach outperforms the state-of-the-art methods by more than 0.6db in term of PSNR.

Qualitative comparisons of ARCNN [18], Toflow [21], HEVC-LF [1] and ours are shown in Figs. 4 and 5. In these figures, the blocking artifacts exist in JPEG2000/HEVC decoded frame, our proposed method successfully removes these artifacts while other methods still have observable artifacts. For example, the equipment (the fourth row in Fig. 4) and the railing (the fourth row in Fig. 5) both have complex texture and structure, our method can well recover these complex regions while other baseline methods may fail.

Table 1. Average PSNR/SSIM results on the Vimeo dataset for JPEG2000 artifact reduction (q = 20,40).

Dataset	Setting	ARCNN [18]	DnCNN [17]	V-BM4D [47]	Toflow [21]	Ours
Vimeo	q = 20	36.11/0.960	37.26/0.967	35.75/0.959	36.92/0.966	**37.93/0.971**
	q = 40	34.21/0.944	35.22/0.953	33.99/0.940	34.97/0.953	**35.88/0.958**

Table 2. Average PSNR/SSIM results on the Vimeo test sequences for HEVC artifact reduction (qp = 32,37).

Dataset	Setting	ARCNN [18]	DnCNN [17]	HEVC-LF [1]	Ours
Vimeo	qp = 32	34.87/0.954	35.58/0.961	34.19/0.950	**35.81/0.962**
	qp = 37	32.54/0.930	33.01/0.936	31.98/0.923	**33.23/0.939**

Table 3. Ablation study of the proposed deep Kalman filtering method on the Vimeo-90k dataset. The results with or without using the prediction residual (PR) in the measurement network (MN) are reported in the first two rows. The results with or without using the recursive filtering (RF) scheme in the temporal network (TM) are reported in the 3^{rd} and 4^{th} rows. Our full model is MN+PR+TM+RF (the 5^{th} row).

MN	PR	TM	RF	PSNR/SSIM
✓				37.15/0.967
✓	✓			37.49/0.968
		✓		37.35/0.967
		✓	✓	37.76/0.970
✓	✓	✓	✓	37.93/0.971

Ablation Study of Measurement Network (MN)**.**In this subsection, we investigate the effectiveness of the proposed measurement network. 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 temporal mapping network. In order to validate that prediction residual can serve as important prior information for improving the performance, we train another model with the same architecture but without using prediction residual (PR) as the input. Therefore, it generates restored frames by only using the decoded frames as the input. Quantitative results on the Vimeo-90k dataset are listed in Table 3. When compared with our simplified model without prediction residual(see the 1^{st} row), our simplified model with prediction residual (MN+PR, see the 2^{nd} row) can boost the performance by 0.34 dB in term of PSNR. It demonstrates that incorporating strong prior information can improve the restoration performance.

Ablation Study on the Temporal Mapping Network (TM)**.** We further evaluate the effectiveness of the temporal mapping network. Note that the output of our temporal mapping 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. For comparison, we train another model, which utilizes the same network architecture as our temporal mapping network but the input is the concatenation of X_t^c and X_{t-1}^c. Namely, it restores the current frame without considering previous restored frames. The quantitative results are reported in Table 3. When compared with our simplified model without using recursive filtering (RF) (see the 3^{rd} row), our simplified model with recursive filtering (TM+RF, see the 4^{th} row) can significantly improve the quality of restored frame by 0.41dB in term of PSNR. A possiable explanation is 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 5^{th} row is the best as we combine the outputs from both the measurement network and the temporal mapping network through the Kalman update process.

Table 4. Average PSNR/SSIM results evaluated on two new datasets for video artifact reduction (JPEG2000, q = 20) for cross dataset validation.

Test dataset	Toflow [21]	DnCNN [17]	Ours
HEVC dequneces	32.37/0.948	33.19/0.953	**33.83/0.958**
MPI Sintel datast	34.78/0.959	36.40/0.969	**37.01/0.973**

Cross Dataset Validation. The results on the HEVC standard sequences (Class D) and the MPI Sintel Flow dataset in Table 4 show that our approach performs better than the state-of-the-art methods.

Comparison with the RNN Based Approach. 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 convolutional gated recurrent unit (GRU) module is used to restore the original image based on these features. The result of our work is 37.93 dB, which outperforms the recurrent network based method (37.10 dB). One possible explanation is that it is difficult to train the recurrent network, while our pipeline makes it easier to learn the network by using the domain knowledge of prediction residual and combining both measurement and prior estimation.

5 Conclusions

In this paper, we have proposed a deep Kalman filtering network for video artifact reduction. We model the video compression artifact reduction task as a

Kalman filtering procedure and update the state function by learning deep neural networks. Our framework can take advantage of both the recursive nature of Kalman filtering and representation learning ability of neural network. Experimental results have demonstrated the superiority of our deep Kalman filtering network over the state-of-the-art methods. Our methodology can also be extended to solve other low-level computer vision tasks, such as video super-resolution or denoising, which will be studied in the future.

Acknowledgement. This work was supported by s research project from SenseTime. This work was also supported in part by National Natural Science Foundation of China (61771306, 61521062), Natural Science Foundation of Shanghai(18ZR1418100)Chinese National Key S&T Special Program(2013ZX01033001-002-002), STCSM Grant 17DZ1205602, Shanghai Key Laboratory of Digital Media Processing and Transmissions (STCSM 18DZ2270700).

References

1. Sullivan, G.J., Ohm, J., Han, W.J., Wiegand, T.: Overview of the high efficiency video coding (HEVC) standard. TCSVT **22**(12), 1649–1668 (2012)
2. Schwarz, H., Marpe, D., Wiegand, T.: Overview of the scalable video coding extension of the H. 264/AVC standard. TCSVT **17**(9), 1103–1120 (2007)
3. Lu, G., Zhang, X., Chen, L., Gao, Z.: Novel integration of frame rate up conversion and HEVC coding based on rate-distortion optimization. TIP **27**(2), 678–691 (2018)
4. Shen, M.Y., Kuo, C.C.J.: Review of postprocessing techniques for compression artifact removal. J. Vis. Commun. Image Represent. **9**(1), 2–14 (1998)
5. Reeve, H.C., Lim, J.S.: Reduction of blocking effects in image coding. Opt. Eng. **23**(1) (1984)
6. Jung, C., Jiao, L., Qi, H., Sun, T.: Image deblocking via sparse representation. Signal Process. Image Commun. **27**(6), 663–677 (2012)
7. Choi, I., Kim, S., Brown, M.S., Tai, Y.W.: A learning-based approach to reduce JPEG artifacts in image matting. In: ICCV (2013)
8. Chang, H., Ng, M.K., Zeng, T.: Reducing artifacts in JPEG decompression via a learned dictionary. IEEE Trans. Signal Process. **62**(3), 718–728 (2014)
9. Liu, X., Wu, X., Zhou, J., Zhao, D.: Data-driven sparsity-based restoration of JPEG-compressed images in dual transform-pixel domain. In: CVPR, vol. 1. p. 5 (2015)
10. Ouyang, W., Wang, X.: Joint deep learning for pedestrian detection. In: ICCV (2013)
11. Ouyang, W., et al.: Deepid-net: deformable deep convolutional neural networks for object detection. In: CVPR (2015)
12. Wang, L., Ouyang, W., Wang, X., Lu, H.: Visual tracking with fully convolutional networks. In: ICCV (2015)
13. Zhao, R., Ouyang, W., Wang, X.: Unsupervised salience learning for person re-identification. In: CVPR (2013)
14. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 184–199. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10593-2_13

15. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE Trans. Pattern Anal. Mach. Intell. **38**(2), 295–307 (2016)
16. Tai, Y., Yang, J., Liu, X., Xu, C.: Memnet: a persistent memory network for image restoration. In: CVPR (2017)
17. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. TIP **26**(7), 3142–3155 (2017)
18. Dong, C., Deng, Y., Change Loy, C., Tang, X.: Compression artifacts reduction by a deep convolutional network. In: ICCV (2015)
19. Guo, J., Chao, H.: Building dual-domain representations for compression artifacts reduction. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 628–644. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_38
20. Galteri, L., Seidenari, L., Bertini, M., Del Bimbo, A.: Deep generative adversarial compression artifact removal. arXiv preprint arXiv:1704.02518 (2017)
21. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. arXiv preprint arXiv:1711.09078 (2017)
22. Tao, X., Gao, H., Liao, R., Wang, J., Jia, J.: Detail-revealing deep video super-resolution. In: ICCV (2017)
23. Liu, D., et al.: Robust video super-resolution with learned temporal dynamics. In: CVPR (2017)
24. Caballero, J., et al.: Real-time video super-resolution with spatio-temporal networks and motion compensation. In: CVPR (2017)
25. Foi, A., Katkovnik, V., Egiazarian, K.: Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. TIP **16**(5), 1395–1411 (2007)
26. Zhang, X., Xiong, R., Fan, X., Ma, S., Gao, W.: Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity. TIP **22**(12), 4613–4626 (2013)
27. Wang, Z., Liu, D., Chang, S., Ling, Q., Yang, Y., Huang, T.S.: D3: dep dual-domain based fast restoration of JPEG-compressed images. In: CVPR (2016)
28. Svoboda, P., Hradis, M., Barina, D., Zemcik, P.: Compression artifacts removal using convolutional neural networks. arXiv preprint arXiv:1605.00366 (2016)
29. Mao, X.J., Shen, C., Yang, Y.B.: Image denoising using very deep fully convolutional encoder-decoder networks with symmetric skip connections. arXiv preprint (2016)
30. Guo, J., Chao, H.: One-to-many network for visually pleasing compression artifacts reduction. In: CVPR (2017)
31. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep CNN denoiser prior for image restoration. arXiv preprint (2017)
32. Chang, J.R., Li, C.L., Poczos, B., Kumar, B.V., Sankaranarayanan, A.C.: One network to solve them allsolving linear inverse problems using deep projection models. arXiv preprint (2017)
33. Bigdeli, S.A., Zwicker, M., Favaro, P., Jin, M.: Deep mean-shift priors for image restoration. In: NIPS (2017)
34. Liao, R., Tao, X., Li, R., Ma, Z., Jia, J.: Video super-resolution via deep draft-ensemble learning. In: ICCV (2015)
35. Kappeler, A., Yoo, S., Dai, Q., Katsaggelos, A.K.: Video super-resolution with convolutional neural networks. IEEE Trans. Comput. Imaging **2**(2), 109–122 (2016)
36. Jaderberg, M., Simonyan, K., Zisserman, A., et al.: Spatial transformer networks. In: NIPS (2015)

37. Shashua, S.D.C., Mannor, S.: Deep robust kalman filter. arXiv preprint arXiv:1703.02310 (2017)
38. Krishnan, R.G., Shalit, U., Sontag, D.: Deep Kalman filters. arXiv preprint arXiv:1511.05121 (2015)
39. Kalman, R.E.: A new approach to linear filtering and prediction problems. J. Basic Eng. **82**(1), 35–45 (1960)
40. Haykin, S.S.: Kalman Filtering and Neural Networks. Wiley Online Library, New York (2001)
41. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38
42. Ballé, J., Laparra, V., Simoncelli, E.P.: Density modeling of images using a generalized normalization transformation. arXiv preprint arXiv:1511.06281 (2015)
43. Abadi, M., et al.: Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467 (2016)
44. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. TIP **13**(4), 600–612 (2004)
45. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
46. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: International Conference on Artificial Intelligence and Statistics (2010)
47. Maggioni, M., Boracchi, G., Foi, A., Egiazarian, K.: Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. TIP **21**(9), 3952–3966 (2012)