Online Rain/Snow Removal from Surveillance Videos

Minghan Li, Xiangyong Cao, *Member, IEEE,* Qian Zhao, Lei Zhang, *Fellow, IEEE,* and Deyu Meng, *Member, IEEE*

***Abstract*—Video rain/snow removal from surveillance videos is an important task in the computer vision community since rain/snow existed in videos can severely degenerate the performance of many surveillance system. Various methods have been investigated extensively, but most only consider consistent rain/snow under stable background scenes. Rain/snow captured from practical surveillance camera, however, is always highly dynamic in time, and those videos also include occasionally transformed background scenes and background motions caused by waving leaves or water surfaces. To this issue, this paper proposes a novel rain/snow removal approach, which fully considers dynamic statistics of both rain/snow and background scenes taken from a video sequence. Specifically, the rain/snow is encoded as an online multi-scale convolutional sparse coding (OMS-CSC) model, which not only finely delivers the sparse scattering and multi-scale shapes of real rain/snow, but also well distinguish the components of background motion from rain/snow layer. The real-time ameliorated parameters in the model well encodes their temporally dynamic configurations. Furthermore, a transformation operator imposed on the background scenes is further embedded into the proposed model, which finely conveys the background transformations, such as rotations, scalings and distortions, inevitably existed in a real video sequence. The approach so constructed can naturally better adapt to the dynamic rain/snow as well as background changes, and also suitable to deal with the streaming video attributed its online learning mode. The proposed model is formulated in a concise maximum a posterior (MAP) framework and is readily solved by the alternating direction method of multipliers (ADMM) algorithm. Compared with the state-of-the-art online and offline video rain/snow removal methods, the proposed method achieves best performance on synthetic and real videos datasets both visually and quantitatively. Specifically, our method can be implemented in relatively high efficiency, showing its potential to real-time video rain/snow removal. The code page is at: https://github.com/MinghanLi/OTMSCSC_matlab_2020.**

Index Terms—**multi-scale, convolutional sparse coding, rain/snow removal, dynamic background, online learning, alignment method.**

I. Introduction

VIDEOS captured from outdoor surveillance system are often contaminated by rain or snow, which has a negative effect on the perceptual quality and tends to degrade the performance of subsequent video processing tasks, such as human detection [1], person re-identification [2], object tracking [3] and scene analysis [4]. Thus, removing rain and snow from surveillance videos is an important video pre-processing step and has attracted much attention in the computer vision community.

In recent decades, various methods have been proposed for removing rain from a video. The earliest video rain removal approach was proposed based on the photometry property of rain [5]. After that, more methods taking advantage of the essential physical characteristics of rain, such as photometric appearance [6], chromatic consistency [7], shape and brightness [8], and spatial-temporal configurations [9], were introduced to better separate rain streaks from the background of videos. However, these methods do not utilize the prior knowledge of video structure, such as spatial smoothness of foreground objects and temporal similarity of background scenes, and thus cannot always obtain satisfactory performance especially in complex scenes. In recent years, low-rank models [10] show a great potential for this task and always achieve state-of-the-art performance due to their better consideration of video structure prior knowledge both in foreground and background. Specifically, these methods not only use the low-rank structure for the background, but also fully facilitate the prior knowledge of the rain, such as sparsity and spatial smoothness [11], [12]. Very recently, deep learning based methods have also been proposed for this task. These methods address the problem of video rain removal by constructing deep recurrent convolutional networks [13][14] or deep convolutional network [15] and implement the task in a popular end-to-end learning manner.

Albeit achieving good progress, most of current methods are implemented on a pre-fixed length of videos and assume consistent rain/snow shapes under static background scenes. This, however, is evidently deviated from the real scenarios. On one hand, the rain/snow contained in a video sequence is generally with configurations changed constantly along time. On the other hand, the background scene in the video is also always dynamic, inevitably containing background motion, such as

Manuscript received May 17, 2020; revised October 15, 2020; accepted December 27, 2020. This work was supported in part by the National Natural Science Foundation of China (NSFC) projects under Grant 62076196, Grant 61721002, Grant 11690011, and Grant U1811461 and in part by the Hong Kong Research Grant Council (RGC)—General Research Fund (GRF) under Grant PolyU 15221618E. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xiaolin Wu. (Corresponding author: Deyu Meng.)

Minghan Li and Lei Zhang are with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Hong Kong. E-mail: liminghan0330@gmail.com, cslzhang@comp.polyu.edu.hk.

Xiangyong Cao and Qian Zhao are with the School of Mathematics and Statistics and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi'an Jiaotong University, Shaanxi, P.R. China. E-mail: caoxiangyong45@gmail.com, timmy.zhaoqian@xjtu.edu.cn.

Deyu Meng is with the Faculty of Information Technology, Macau University of Science and Technology, Taipa, Macau, and School of Mathematics and Statistics, Xi'an Jiaotong University, Xian, Shaan'xi, PR China. E-mail: dymeng@xjtu.edu.cn.

Digital Object Identifier 10.1109/TIP.2021.3050313

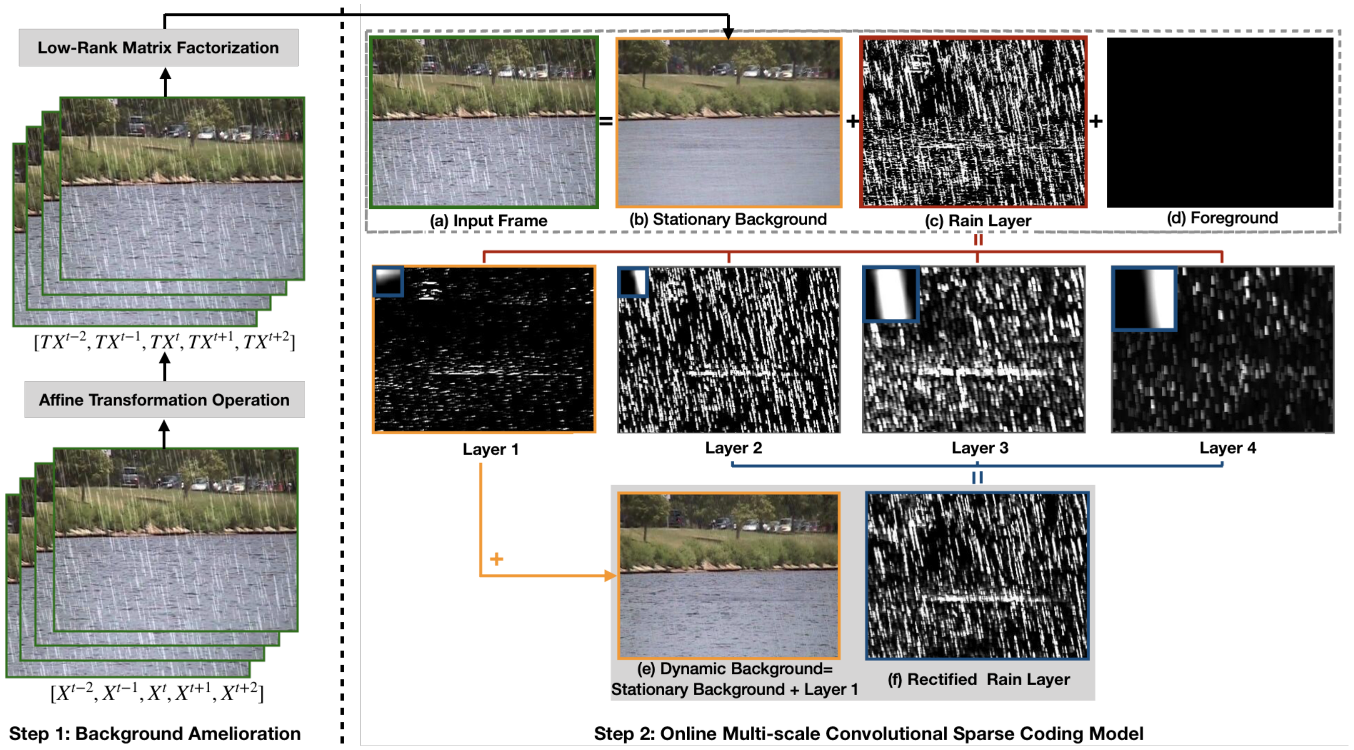

Low-Rank Matrix Factorization
$[TX^{t-2}, TX^{t-1}, TX^{t}, TX^{t+1}, TX^{t+2}]$
Affine Transformation Operation
$[X^{t-2}, X^{t-1}, X^{t}, X^{t+1}, X^{t+2}]$
Step 1: Background Amelioration
(a) Input Frame
=
(b) Stationary Background
+
(c) Rain Layer
+
(d) Foreground
Layer 1
Layer 2
Layer 3
Layer 4
+
(e) Dynamic Background=
Stationary Background + Layer 1
(f) Rectified Rain Layer
Step 2: Online Multi-scale Convolutional Sparse Coding Model

Fig. 1. The diagram of the proposed OTMS-CSC model implemented on a video with dynamic background. As shown in the left figure, the background alignment based on its adjacent frames produces an initial stationary background. The online MS-CSC model shown on the right decomposes (a) the input video frame into four parts: (b) stationary background, (c) rain layer, (d) moving objects and the background noise. The rain layer (b) can be further decomposed as four sub-layers with various filters, which encode the repetitive local patterns of both rain/snow and background motions, displayed in the top-left corner of the second row. For the video with dynamic background, the final dynamic background (e) is the combination of the stationary background and the sub-layers with background motions, and the rectified rain layer only combines those sub-layers with relatively vertical filters representing rains.

swing leaves and water waves as typically shown in Fig. 1, and timely transformations such as translation, rotation, scaling and distortion, due to camera jitters. Lacking considerations to such dynamic characteristics inclines to degenerate the performance of current methods in such real cases. Besides, as the dramatically increasing surveillance cameras installed all over the world, the real video is always coming online as a streaming format. Most current methods, however, are implemented/trained on a pre-fixed video sequence, and thus cannot finely and efficiently adapt to such kinds of streaming videos continually and endlessly coming in time. These issues have hampered the availability of existing methods in real applications and thus is worthy to be specifically investigated.

Against the aforementioned issues, this paper proposes a new online rain/snow removal method from surveillance videos by fully encoding the dynamic statistics of both rain/snow and background scenes in a video along time into the model, and realizing it with an online mode to make it potentially available to handle constantly coming streaming video sequence. Specifically, inspired by the multi-scale convolutional sparse coding (MS-CSC) model designed for video rain removal (still for static rain) previously proposed in [16], which finely delivers the sparse scattering and multi-scale shapes of real rain, this work encodes the dynamic temporal changing tendency of rain/snow and background motions as a dynamic MS-CSC framework by timely parameter amelioration for the model in an online implementation manner. Besides, a transformation operator capable of being adaptively updated along time is imposed on the background scenes to finely fit the background transformations existed in a video sequence. All these knowledge are formulated into a concise maximum a posterior (MAP) framework, which can be easily solved by alternative optimization technique.

In all, the contributions of this work can be mainly summarized as follows:

1) An online multi-scale convolutional sparse coding model is specifically designed for encoding dynamic rain/snow and background motions with temporal variations. The model is formulated as a concise probabilistic framework, where the feature maps are gradually ameliorated under regularization of a penalty for enforcing them close to those calculated from the previous frames, and the filters encode the repetitive local patterns of dynamic rain/snow and background motions in each frame of a video. In this manner, the insightful dynamic rain/snow properties and the background motions can be finely delivered.

2) An affine transformation operator is further embedded into the proposed model, and can be automatically adjusted to fit a wide range of video background transformations. This makes the method more robust to general camera movements, like rotation, translation, scaling or distortion.

3) To handle the challenging task of rain removal from videos with dynamic background, based on the sequences in the dynamic background category of the changedetection.net [17] (CDNet) dataset, we build the first new synthetic dynamic dataset whose video contains both rain streaks and background

motions such as waving leaves and water waves, called CDNet-Rain dataset. The superiority of the proposed method in robustness and efficiency are comprehensively substantiated by experiments implemented on the proposed dynamic dataset both visually and quantitatively, as compared with other state-of-the-art methods.

4) We take the video instance segmentation (VIS) task as the example to further verify whether removing rain and snow from a video can bring a positive impact on the sub-sequence video processing task. Specifically, based on the large-scale video instance segmentation valid dataset YouTube-VIS[18], we construct a video rain removal benchmark for the video instance segmentation task called YouTube-VIS-Rain dataset. The visual and quantitative experimental results on the benchmark also demonstrate that, compared with directly employing the video instance segmentation algorithm on the contaminated videos, the video rain removal pre-processing via our proposed model is evidently benefical to the final performance of the handled video processing task.

The rest of paper is organized as follows. Section 2 introduces the related works. Section 3 reviews the offline multi-scale convolutional sparse coding (offline MS-CSC) model[16] suitable for removing static rain and proposes the online transformed multi-scale convolutional sparse coding (OTMS-CSC) model as well as its solving algorithm. Section 4 demonstrates experimental results on synthetic and real rainy/snowy videos with/without dynamic background to substantiate the superiority of the proposed method and further verifies that the pre-processing of video rain removal can bring a positive impact on the video instance segmentation task. Finally, conclusions are drawn in Section 5.

II. Related Works

In this section, we give a brief review on the methods of video rain and snow removal. The related developments on single image rain and snow removal, multi-scale modeling and video alignment are also introduced for literature comprehensiveness. It should be indicated that albeit different in physical generation mechanisms, in visual imaging perspectives, both rainfall and snowfall on a digital image or a video frame have very similar geometric characteristics, which makes multiple methods, as well as ours, proposed to treat both scenarios simultaneously.

A. Video Rain and Snow Removal Methods

Garg and Nayar [5] made the earliest study on the photometric appearance of rain drops and developed a rain detection method by utilizing a linear space-time correlation model. To better reduce the effects of rain before camera shots in images/videos, Garg and Nayar [6], [19] further proposed a method by adjusting the camera parameters such as field depth and exposure time.

In the past years, more physical intrinsic properties of rain streaks have been explored and formulated in algorithm designing. For example, Zhang et al. [7] incorporated both chromatic and temporal properties and utilized K-means clustering for distinguishing background and rain streaks from videos. Later, Barnum et al. [8] first considered the impact of snow on videos. They derived a physical model for representing raindrops and snowflakes and used them to determine the general shape and brightness of a single streak. The streak model combined with the statistical properties of rain and snow can then conduct how they affect the spatial-temporal frequencies of an image sequence. To enhance the robustness of rain removal, Barnum et al. [20] employed the regular visual effects of rain and snow in global frequency information to approximate rain streaks as a motion-blurred Gaussian. Afterwards, to integrate more prior knowledge of the task, Jiang et al. [21] proposed a tensor-based video rain streak removal approach by considering the sparsity of rain streaks, smoothness along the raindrops and the rain-perpendicular direction, and global and local correlation along time direction.

In recent years, low-rank based models have drawn more research attention for the task of video rain/snow removal. Chen et al. [10] first investigated spatial-temporal correlation among local patches with rain streaks and used low-rank term to extract rain streaks from a video. Later, Kim et al. [22] proposed a rain and snow removal method based on temporal correlation and low-rank matrix completion. To further exclude false candidates, Santhaseelan et al. [23] used local phase congruency to detect rain and applied chromatic constrain. To deal with heavy rain and snow in dynamic scenes, Ren et al. [11] divided rain into sparse and dense ones based on the low-rank hypothesis of the background. Based on the low-rank background assumption, Wei et al. [12] further encoded rain streaks as a patch-based mixture of Gaussians. Such stochastic manner for encoding rain streaks could make the method deliver a wider range of rain information.

Very recently, motivated by the booming of deep learning (DL) techniques, several DL methods also appeared for the task. Liu et al. [13], [24] addressed the problem by constructing deep recurrent convolutional networks, which builds a joint recurrent rain removal and reconstruction network that seamlessly integrates rain degradation classification, spatial texture appearances based rain removal, and temporal coherence based background detail reconstruction. Meanwhile, Chen et al. [15] proposed a deep derain framework which applies superpixel segmentation to decompose the scene into depth consistent units. Alignment of scene contents are done at the superpixel level to handle the videos with highly complex and dynamic scenes. Yang et al. [14] not only proposed a two-stage recurrent network with dual-level flow regularizations to perform the inverse recovery process of the rain synthesis model for video deraining, but also developed a novel rain synthesis model to produce more visually authentic paired training and evaluation videos.

B. Single Image Rain and Snow Removal Methods

For literature comprehensiveness, we also briefly review the rain/snow removal methods for a single image. Kang et al. [25] firstly formulated the problem as an image decomposition problem based on morphological component analysis, which achieves rain component from the high frequency part of an image by using dictionary learning and sparse coding. Later, Luo et al. [26] built a nonlinear screen blend model based on discriminative sparse codes. Besides, Ding et al. [27] designed a guided L_0 smoothing filter to obtain a coarse rain-free or snow-free image, and Li et al. [28] utilized patch-based Gaussian mixture model (GMM) priors to distinguish and remove rain from background in a single image. Wang et al. [29] designed a 3-layer hierarchical scheme to classify the high-frequency part into rain/snow and non-rain/snow components. Gu et al. [30] jointly analyzed sparse representation and synthesis sparse representation to encode background scene and rain streaks. Meanwhile, Zhang et al. [31] learned a set of generic sparsity-based and low-rank representation-based convolutional filters for efficiently representing background and rain streaks in an image.

Recently, DL-based methods represent the new trend for this task. Fu et al. [32] firstly developed a deep convolutional neural network (CNN) model to extract discriminative features of rain in high frequency layer of an image. The training pairs are constructed based on the whole image. Later, Fu et al. [33] constructed the training pairs by using image patches and utilized the res-net as the classifier. Zhang et al. [34] first proposed a derain network based on generative adversarial network for single image derain. Yang et al. [35] designed a multi-task DL architecture that learns the binary rain streak map, the appearance of rain streaks and the clean background. Liu et al. [36] proposed a multistage and multi-scale network to deal with the removal of translucent and opaque snow particles. Very recently, Yang et al. [37] constructed a contextualized deep network, which incorporates a binary rain map indicating rain-streak regions, and accommodates various shapes, directions, and sizes of overlapping rain streaks as well as rain accumulation to model heavy rain. For dealing with heavy rain, Li et al. [38] proposed a two-stage network: a physics-based backbone followed by a depth-guided generative adversarial networks (GAN) refinement, which aims to estimate the rain streaks, the transmission, and the atmospheric light, and to recover the background details failed to be retrieved by the first stage. Wang et al. [39] proposed a model-driven deep neural network for the task, with fully interpretable network structures.

Although these image-based methods can also deal with rain/snow removal in a video via a rough frame-by-frame manner, the missing use of the important temporal information for such a specific task inclines to make the video-based methods perform significantly better than image-based ones.

C. Online Learning Approaches

Online learning is a method of machine learning in which data becomes available in a sequential order and is used to update the best predictor for future data at each step, as opposed to batch learning techniques which generate the best predictor by learning on the entire training data set at once. Online learning is a common technique used in areas of machine learning where it is computationally infeasible to train over the entire dataset, requiring the need of out-of-core algorithms. Online learning algorithms may be prone to catastrophic interference, a problem that can be addressed by incremental learning approaches. Recently, online learning methods have attracted increasing attention in many computer science tasks, such as background subtraction [40], [41], [42]. In video rain/snow removal task, online learning is used to calculate only one frame at a time, and gradually ameliorate rain/snow based on the real-time video variations.

D. Alignment Approaches for Videos

Since camera jitter tends to damage the low-rank background structure of a video, we always need to align the transformed videos to accurately extract the low-rank background. Many alignment methods have been attempted to this issue. For example, Zhang et al. [43] proposed an approach to directly extract certain 3D invariant structures through their 2D images by undoing the (affine or projective) domain transformations. Zhang et al. [44] further proposed a general method for recovering low-rank 3-order tensors, which introduced auxiliary variables and relaxed the hard equality constraints by the alternating direction method of multipliers (ADMM) [45]. Yong et al. [40] proposed an alignment method for aligning the video background based on optimizing a supplemental affine transformation operator, and applied it to the task of dynamic background subtraction.

III. Online Transformed MS-CSC model for Dynamic Video Rain/Snow Removal

This work is inspired by our previous conference work [16], proposing an offline multi-scale convolutional sparse coding (MS-CSC) model, specifically designed for rain removal issue (with consistent rain temporarily) in a fixed length of video sequence. We thus first introduce the formulation of this offline model.

A. Offline MS-CSC Model

Let $\mathcal{X} \in R^{h\times w\times n}$ denotes the input video, where h, w, and n represent its height, width and the number of frames, respectively. We assume that the video $\mathcal{X}$ can be decomposed as:

$$\mathcal{X} = \mathcal{B} + \mathcal{F} + \mathcal{R} + \mathcal{E}, \tag{1}$$

where $\mathcal{B}, \mathcal{F}, \mathcal{R}, \mathcal{E} \in R^{h\times w\times n}$ represent background scene, moving objects, rain layer, and background noise of the video, respectively. These parts can then be modeled separately as follows [16].

Background Modeling: For a fixed length of video sequence captured from a surveillance camera, its background tends to keep steady over the frames, and thus can be rationally assumed to be resided on a low-dimensional subspace [46], [47], [48], [49], [50], leading to its low-rank matrix factorization representation as:

$$\mathcal{B} = \mathrm{Fold}(UV^T), \tag{2}$$

where $U \in R^{d\times r}, V \in R^{n\times r},\ d = hw,\ r < \min(d, n)$. The operation 'Fold' refers to fold up each matrix column into the corresponding frame matrix, and thus the background $\mathcal{B}$ is a tensor with the same size as input $\mathcal{X}$.

Rain Layer Modeling: Since rain in a video contain repetitive local patterns sparsely scattering over different areas, and also exhibits multi-scale property due to its occurrence positions with different distances to the cameras, multi-scale convolutional sparse coding (MS-CSC) [51] is thus utilized to model rain as follows:

$$\mathcal{R}=\sum_{k=1}^{K}\sum_{s=1}^{s_k} D_{ks}\otimes\mathcal{M}_{ks}, \tag{3}$$

where $\otimes$ denotes convolutional operation, and $\mathcal{M}=\{\mathcal{M}_{ks}\}_{k,s=1}^{K,s_k}\subset R^{h\times w\times n}$ is a set of feature maps that approximate the rain streak positions, and $D=\{D_{ks}\}_{k,s=1}^{K,s_k}\subset R^{p_k\times p_k}$ denotes the filters representing the repetitive local patterns of rain streaks. K and s_k denote the numbers of entire filters and filters at the k-th scale, respectively. Considering the sparsity of feature maps, the L_1-penalty [52] is utilized to regularize them.

Moving objects Modeling: Motivated by the work [12], Markov random field (MRF) is used to explicitly detect the moving objects. Let $\mathcal{H}\in R^{h\times w\times n}$ be a binary tensor denoting the moving object support:

$$\mathcal{H}_{ijn}=\begin{cases}1, & \text{location } (i,j,n) \text{ is moving objects},\\ 0, & \text{location } (i,j,n) \text{ is background},\end{cases} \tag{4}$$

and $\mathcal{H}^{\perp}$ is the complementary of $\mathcal{H}$ (i.e., $\mathcal{H}+\mathcal{H}^{\perp}=\mathbf{1}$, $\mathbf{1}$ is a tensor with all elements as 1). Eq.(1) can be then reformulated as:

$$\mathcal{X}=\mathcal{H}^{\perp}\circ\mathcal{B}+\mathcal{H}\circ\mathcal{F}+\mathcal{R}+\mathcal{E}, \tag{5}$$

where operation $\circ$ denotes the element-wise multiplication. Since moving objects always exhibit smooth property, total variation (TV) penalty [53] is adopted to regularize them. Additionally, considering the sparse feature and continuous shapes along both space and time of moving object, L_1-penalty and weighted 3-dimensional total variation (3DTV) penalty are both employed to regularize the moving objects support $\mathcal{H}$ simultaneously.

By assuming that the background noise $\mathcal{E}$ follows an i.i.d. Gaussian, we can then integrate the aforementioned three models imposed on background, rain streak and moving objects to get the MS-CSC model for offline video rain removal as follows [16]:

$$\begin{aligned}\min_{\Theta}\mathcal{L}(\Theta)=&\parallel \mathcal{X}-\mathcal{H}^{\perp}\circ\mathcal{B}-\mathcal{H}\circ\mathcal{F}-R\parallel_F^2+\lambda\parallel\mathcal{F}\parallel_{TV}\\ &+\alpha\parallel\mathcal{H}\parallel_{3DTV}+\beta\parallel\mathcal{H}\parallel_1+b\sum_{k=1}^{K}\sum_{s=1}^{n_k}\parallel\mathcal{M}_{ks}\parallel_1\\ s.t.\qquad &\mathcal{B}=\text{Fold}(UV^T)\\ &\mathcal{R}=\sum_{k=1}^{K}\sum_{s=1}^{s_k}D_{ks}\otimes\mathcal{M}_{ks},\quad\parallel D_{ks}\parallel_F^2\leq 1,\end{aligned}$$

where $\Theta=\{D,\mathcal{M},\mathcal{H},\mathcal{F},U,V,\mathcal{R}\}$ are the variables involved in the problem to be optimized.

B. Online Transformed MS-CSC Model

The previous MS-CSC model is specifically designed for rain removal in a pre-fixed length of video under the assumption that the rain is of consistent configuration along time. Specifically, the rain feature maps $\mathcal{M}$ (as defined in Eq. (3)) of all video frames attained under fixed filters are assumed to follow a unique independent and identically distributed Laplacian. The real rain shapes, however, are always both correlated and distinctive along time, and varying from frame to frame across the entire video. The simple encoding manner of MS-CSC is thus inappropriate to real scenarios. We thus present the online MS-CSC model, which not only provides a more proper way to describe temporally dynamic rain/snow and background motions, but also makes the method more efficient and potentially applicable to streaming videos with continuously increasing frames in real time.

For symbol unification, we denote the newly coming single frame as $X^t\in R^{h\times w}$, where h and w represent the height and width of this frame, respectively, and $d=h*w$ denotes the total number of pixels in this single frame. Similar to (1), we then decompose newly coming single frame X^t as the following three parts:

$$X^t=B^t+F^t+R^t+E^t, \tag{6}$$

where $B^t,F^t,R^t,E^t\in R^{h\times w}$ represent the background scene, moving objects, rain layer and background noise of the current frame, respectively. We then put forward the schemes to model these parts based on the dynamic characteristics of rain/snow.

1) Modeling Dynamic Rain/Snow Layer: Albeit different in physical generation mechanisms, in visual imaging perspectives, both rainfall and snowfall on a digital image or a video frame have very similar geometric characteristics, i.e., with repetitive local patterns sparsely scattered over different positions of the image, and of multi-scale configurations due to their occurrence on positions with different distances to the cameras. Such two intrinsic characteristics are thus encoded into a concise probabilistic framework by the multi-scale convolutional sparse coding (MS-CSC) model [16], namely:

$$R^t=\sum_{k=1}^{K}\sum_{s=1}^{s_k}D_{ks}^t\otimes M_{ks}^t, \tag{7}$$

where $M^t=\{M_{ks}^t\}_{k,s=1}^{K,s_k}\subset R^{h\times w}$ is a set of feature maps that approximate the rain streak positions, and $D^t=\{D_{ks}^t\}_{k,s=1}^{K,s_k}\subset R^{p_k\times p_k}$ denotes the filters representing the repetitive local patterns of rain streaks. K and s_k denote the total scale number of filters and the total number of filters with k-th scale, respectively.

Similar to the MS-CSC model, the sparsity of feature map M_{ks}^t is also regularized by the Laplacian distribution:

$$M_{ks}^t\sim\text{Laplacian}(M_{ks}^t|0,b_{ks}^t), \tag{8}$$

where the scale parameter $b_{ks}^t>0$ is specified for the current frame reflecting the specific rain degreen in this frame. Furthermore, the correlation of rain between current and previous frames is represented by the following prior term imposed on b_{ks}^t:

$$b_{ks}^t\sim\ \text{Inv-Gam}(b_{ks}^t|N^{t-1}-1,N^{t-1}b_{ks}^{t-1}), \tag{9}$$

where $N^{t-1}=(t-1)d$ and b_{ks}^{t-1} are both the scale parameter learned from the previous frames. Here $\text{Inv-Gam}(\cdot)$ denotes the Inverse-Gamma distribution, a conjugate prior to b_{ks}^t, whose mode is exactly the one of previously learned (i.e., b_{ks}^{t-1}). It is then naturally delivered that the correlation of rain degreen between current frame and the learned knowledge from previous ones.

In the way as aforementioned, the dynamic characteristic of rain/snow across a video can then be rationally represented. In specific, the scale parameter in each frame is specifically learned and different from one another, finely representing the distinctiveness (i.e. 'non-identical') of rain/snow among different frames. Furthermore, the scale parameter of feature map distribution for the current frame is regularized by that of previously learned ones, well encoding the correlation (i.e., 'non-independent') across especially adjacent frames. The model is thus expected to better adapt to the variations of the dynamic rain/snow.

2) Modeling Moving Objects and Background Noise: Following the MS-CSC model, we also adopt Markov random field [54], [55] to detect the moving objects. Let $H \in R^{h\times w}$ is a binary matrix denoting the moving object support, which is defined as

$$H_{ij} = \begin{cases} 1, & \text{location } (i,j) \text{ is moving objects,} \\ 0, & \text{location } (i,j) \text{ is background.} \end{cases} \quad (10)$$

Let $H^{\perp}$ be complementary of H satisfying $H + H^{\perp} = \mathbf{1}$, $\mathbf{1}$ is a matrix with all elements as 1. Eq.(6) can then be equivalently expressed as:

$$X^t = H^{t\perp} \circ B^t + H^t \circ F^t + R^t + E^t. \quad (11)$$

Like the offline MS-CSC optimization problem, by assuming all elements of the background noise E^t follow a Gaussian distribution with zero mean and variance $(\sigma^t)^2$, we can then get the probabilistic model for the component x_{ij}^t of X^t as follows:

$$x_{ij}^t \sim \ N(x_{ij}^t | (H_{ij}^{t\perp} \circ B_{ij}^t + H_{ij}^t \circ F_{ij}^t + R_{ij}^t), (\sigma^t)^2). \quad (12)$$

Similar to the dynamic shapes of rain in practical video, the background noise embedded in the video is also with dynamic forms, and also both distinctive and correlated among video frames. We can then also represent this dynamic knowledge. Specifically, for video noise in the current frame with variance $(\sigma^t)^2$, we model it in the similar modeling manner as aforementioned, i.e., imposing conjugate prior to $(\sigma^t)^2$ as:

$$(\sigma^t)^2 \sim \ \text{Inv-Gam}((\sigma^t)^2 | \frac{N^{t-1}}{2} - 1, \frac{N^{t-1}(\sigma^{t-1})^2}{2}), \quad (13)$$

where $N^{t-1} = (t-1)d$, and $(\sigma^{t-1})^2$ denotes the variance of Gaussian noise learned from the previous frames. The mode of this prior is also the knowledge previously learned (i.e., $(\sigma^{t-1})^2$). This encoding manner is thus also able to deliver the dynamic property of noises along the video.

3) Modeling Background Transformations: To tackle transformations of background scenes in a video due to camera jitter, like translation, rotation and scaling, a flexible affine transformation operation is imposed on the background. In the decomposition form (6) for the current frame X^t, the background component B^t is expressed to be transformed from the previous one B^t as

$$B^t = B^{t-1} \odot \tau, \quad (14)$$

where τ denotes the transformed operator implemented on the initial background B^{t-1}, and can be formulated as an affine or projective transformation [40]. Then, Eq.(11) and (12) can be reformulated as:

$$X^t = H^{t\perp} \circ (B^{t-1} \odot \tau) + H^t \circ F^t + R^t + E^t., \quad (15)$$

$$x_{ij}^t \sim \ N(x_{ij}^t | ((H_{ij}^t)^{\perp} \circ (B_{ij}^{t-1} \odot \tau) + H_{ij}^t \circ F_{ij}^t + R_{ij}^t), (\sigma^t)^2). \quad (16)$$

4) Online Transformed MS-CSC Model: For convenience, we denote all involved parameters as $\Theta = \{H, \tau, D, M, F, \sigma^2, b\}$ and the parameters in the current and last frames as Θ^t and Θ^{t-1}, respectively. Based on the models provided in the last sections, given the previous parameters Θ^{t-1} and newly coming frame X^t, we can then obtain the posterior distribution of Θ as follows:

$$\begin{aligned} & p(H^t, \tau, D^t, M^t, F^t, (\sigma^t)^2, b^t | X^t, \Theta^{t-1}) \\ & \quad \propto p(X^t | H^t, \tau, F^t, D^t, M^t, (\sigma^t)^2) p((\sigma^t)^2 | \Theta^{t-1}) \\ & \quad p(M^t | b^t) p(b^t | \Theta^{t-1}) p(H^t) p(D^t) p(F^t) p(\tau). \end{aligned} \quad (17)$$

Through maximizing this posterior, the updated parameters Θ^t for the current frame can then be attained. This MAP problem can then be equivalently expressed as the following minimization problem:

$$\begin{aligned} \mathcal{L}(\Theta^t) = & -\ln p(X^t | H^t, B^{t-1}, \tau, F^t, D^t, M^t, (\sigma^t)^2) + Q_E((\sigma^t)^2) \\ & -\sum_{k,s} \ln p(M_{k,s}^t | b_{k,s}^t) + Q_R(b^t) + Q_F(F^t, H^t), \\ s.t. \quad & R^t = \sum_{k,s} D_{ks}^t \otimes M_{ks}^t, \quad \| D_{ks}^t \|_F^2 \leq 1, \end{aligned} \quad (18)$$

where

$$Q_E((\sigma^t)^2) = N^{t-1}(\ln \sigma^t + (\sigma^{t-1})^2 / 2(\sigma^t)^2), \quad (19)$$

$$Q_R(b^t) = N^{t-1} \sum_{k,s} (\ln b_{ks}^t + b_{ks}^{t-1} / b_{ks}^t), \quad (20)$$

$$Q_F(F^t, H^t) = \lambda \| F^t \|_{TV} + \alpha \| H^t \|_{\text{3DTV}} + \beta \| H^t \|_1 . \quad (21)$$

Specifically, $Q_R((\sigma^t)^2)$ and $Q_E(b^t)$ correspond to the regularization terms for the distributions of feature map M_{ks}^t and noises embedded in X^t, respectively, which can be more intuitively understood by the following equivalent forms:

$$Q_E((\sigma^t)^2) = N^{t-1} D_{\text{KL}}(N(x|0, (\sigma^{t-1})^2) \,\|\, N(x|0, (\sigma^t)^2)), \quad (22)$$

$$Q_R(b^t) = N^{t-1} \sum_{k,s} D_{\text{KL}}(L(M_{ks}^t | 0, b_{ks}^{t-1}) \,\|\, L(M_{ks}^t | 0, b_{ks}^t)) \quad (23)$$

where $D_{\text{KL}}(\cdot \parallel \cdot)$ denotes the KL divergence between two distributions. Particularly, it can be easily observed that $Q_R(b^t)$ functions to rectify the rain streaks on the current frame with parameter b_{ks}^t to approximate the previously learned rain streaks with parameter b_{ks}^{t-1}, so as to make the rain shapes in the adjacent frames correlated. Similarly, the regularization term $Q_E((\sigma^t)^2)$ inclines to enforce the background noise in the current frame close to that embedded in the previous ones. This easily explains why our method can fit dynamic rain, as well as varying background noises, in a video with evidently non-i.i.d. configurations.

The corresponding augmented Lagrangian function of Eq. (18) can be written as follows:

$$\begin{aligned}\mathcal{L}(\Theta^t) = & \frac{1}{2(\sigma^t)^2}\parallel X^t - (H^t)^{\perp}\circ(B^{t-1}\odot\tau) - H^t\circ F^t - R^t\parallel_F^2 \\ & + d\ln\sigma + N^{t-1}(\ln\sigma^t + \frac{\sigma^{t-1^2}}{2\sigma^{t^2}}) + \alpha\parallel H^t\parallel_{\mathrm{3DTV}} + \beta\parallel H^t\parallel_1 \\ & + \sum_{k,s}(d\ln b_{ks}^t + \frac{1}{b_{ks}^t}\parallel M_{ks}^t\parallel_1) + \sum_{k,s} N^{t-1}(\ln b_{ks}^t + \frac{b_{ks}^{t-1}}{b_{ks}^t}) \\ & + \lambda\parallel F^t\parallel_{TV} + \frac{\rho}{2}\parallel\sum_{k,s} D_{ks}^t\otimes M_{ks}^t - R^t + T^t\parallel_F^2, \end{aligned} \tag{24}$$

where T^t and ρ are the Lagrange variable and the penalty parameter, respectively.

C. ADMM Algorithm

We can then readily adopt the alternating direction method of multipliers (ADMM) [45], a variant of the augmented Lagrangian scheme, to iteratively optimize each variable involved in Eq. (24). To simplify the relevant subproblems, we will utilize the following equation:

$$\begin{aligned}&\parallel X^t - ((H^t)^{\perp}\circ(B^{t-1}\odot\tau) + H^t\circ F^t + R^t)\parallel_F^2 = \\ &\parallel(H^t)^{\perp}\circ(X^t - (B^{t-1}\odot\tau) - R^t)\parallel_F^2 + \parallel H^t\circ(X^t - F^t - R^t)\parallel_F^2 .\end{aligned}$$

Next, we discuss how to solve each subproblem separately.

Update H^t: The subproblem with respect to H^t is

$$\begin{aligned}\min_{H^t}\ & \frac{1}{2(\sigma^t)^2}\parallel X^t - (H^t)^{\perp}\circ(B^{t-1}\odot\tau) - H^t\circ F^t - R^t\parallel_F^2 \\ & + \alpha\parallel H^t\parallel_{3DTV} + \beta\parallel H^t\parallel_1 .\end{aligned} \tag{25}$$

This subproblem is a standard energy minimization problem, which can be efficiently solved by graph cut algorithm [56], [57].

Update F^t: The subproblem with respect to F^t is

$$\min_{F^t}\ \parallel H^t\circ(X^t - F^t - R^t)\parallel_F^2 + 2(\sigma^t)^2\lambda\parallel F^t\parallel_{TV}, \tag{26}$$

which is easily solved by the TV regularization algorithm [53].

Update τ and B^t: Since $B^{t-1}\odot\tau$ is a nonlinear geometric transform, it's hard to directly optimize it and we resort to the following linear approximation:

$$B^t = B^{t-1}\odot\tau + J\triangle\tau, \tag{27}$$

where J is the Jacobian of X^t with respect to τ. We can iteratively approximate the original nonlinear transformation with a locally linear approximation, as $\tau = \tau + \triangle\tau$. Therefore, the subproblem with respect to τ can be reformulated as:

$$\min_{\triangle\tau}\parallel(H^t)^{\perp}\circ(X^t - B^{t-1}\odot\tau - J\triangle\tau - R^t)\parallel^2 . \tag{28}$$

It can be solved in closed-form. The solution is:

$$\triangle\tau = (J'J)^{-1}J'(X^t - R^t - B^{t-1}\odot\tau). \tag{29}$$

Fixing $\triangle\tau$, we can use Eq. (27) to update the background.

Update M^t: The subproblem with respect M^t is

$$\min_{M_{ks}^t}\frac{1}{2}\parallel\sum_{k=1}^{K}\sum_{s=1}^{s_k} D_{ks}^t\otimes M_{ks}^t - R^t + T^t\parallel_F^2 + \sum_{k=1}^{K}\sum_{s=1}^{s_k}\frac{b_{ks}^t}{\rho}\parallel M_{ks}^t\parallel_1 . \tag{30}$$

This subproblem is a standard convolutional sparse coding (CSC) problem and can be readily solved by [58], which adopts the ADMM scheme and FFT to improve computation efficiency.

Update D^t: The subproblem with respect to D^t is

$$\min_{D^t}\frac{1}{2}\parallel\sum_{k=1}^{K}\sum_{s=1}^{s_k} D_{ks}^t\otimes M_{ks}^t - R^t + T^t\parallel_F^2,\ \ \text{s.t.}\parallel D_{ks}^t\parallel_F^2\leq 1. \tag{31}$$

We use online learning algorithm for sparse coding [59] to update the filters $D^t = \{D_{ks}^t\}_{k,s=1}^{K,n_k}$. The algorithm utilizes block-coordinate descent with warm restarts $D^{t-1} = \{D_{ks}^{t-1}\}_{k,s=1}^{K,n_k}$.

Update R^t: The subproblem with respect to R^t is

$$\begin{aligned}\min_{R^t}\ & \frac{1}{2(\sigma^t)^2}\parallel X^t - (H^t)^{\perp}\circ(B^{t-1}\odot\tau) - H^t\circ F^t - R^t\parallel_F^2 \\ & + \frac{\rho}{2}\parallel\sum_{k=1}^{K}\sum_{s=1}^{s_k} D_{ks}^t\otimes M_{ks}^t - R^t + T^t\parallel_F^2 .\end{aligned} \tag{32}$$

The closed-form solution is

$$R^t = (X^t - \Gamma^t)/(1 + \rho(\sigma^t)^2) \tag{33}$$

where $\Gamma^t = (H^t)^{\perp}\circ(B^{t-1}\odot\tau) + H^t\circ F^t - \rho(\sigma^t)^2(\sum_{k,s}^{K,s_k} D_{ks}^t\otimes M_{ks}^t + T^t)$.

Update T^t: Following the general ADMM setting, T^t can be updated as:

$$T^t = T^{t-1} + \sum_{k,s} D_{ks}^t\otimes M_{ks}^t - R^t. \tag{34}$$

Update $(\sigma^t)^2$: The subproblem with respect $(\sigma^t)^2$ is

$$\begin{aligned}\min_{(\sigma^t)^2}\ & \frac{1}{2(\sigma^t)^2}\parallel X^t - ((H^t)^{\perp}\circ B^t + H^t\circ F^t + R^t)\parallel_F^2 \\ & + d\ln\sigma^t + N^{t-1}(\ln\sigma^t + \frac{\sigma^{t-1^2}}{2(\sigma^t)^2}).\end{aligned} \tag{35}$$

Its closed-form solution is:

$$(\sigma^t)^2 = \frac{1}{t}(\overline{\sigma}^t)^2 + \frac{t-1}{t}\sigma^{t-1^2}, \tag{36}$$

where $(\overline{\sigma}^t)^2 = \frac{1}{d}\parallel X^t - ((H^t)^{\perp}\circ B^t + H^t\circ F^t + R^t)\parallel_F^2$.

Update b_{ks}^t: The subproblem with respect to b_{ks}^t is

$$\min_{b_{ks}^t}(d + N^{t-1})\ln b_{ks}^t + (b_{ks}^t)^{-1}(\parallel M_{ks}^t\parallel_1 + N^{t-1}b_{ks}^{t-1}). \tag{37}$$

Its closed-form solution is:

$$b_{ks}^t = \frac{1}{t}\overline{b}_{ks}^t + \frac{t-1}{t}b_{ks}^{t-1}, \tag{38}$$

where $\overline{b}_{ks}^t = \frac{1}{d}\parallel M_{ks}^t\parallel_1$.

The algorithm for solving this online transformed MS-CSC (OTMS-CSC) model can then be summarized as Algorithm 1.

Algorithm 1 Algorithm for OTMS-CSC Model

```
Input: The newly coming frame: $X^t \in \mathbb{R}^{h \times w}$; model variables
    of last frame: $\Theta^{t-1} = \{H^{t-1}, B^{t-1}, D^{t-1}\}$; the parameters
    of last frame: $\{(\sigma^{t-1})^2, b^{t-1}\}$.
Initialization: $\{H^t, D^t\} = \{H^{t-1}, D^{t-1}\}$, $\tau = 0$.
if $t/l == 0$ then
    update $B^{t-1} = \hat{B}^{t-1}$ by using the strategy suggested in
    Sec. 3.4.2.
end if
while not converge do
    Update $\triangle\tau$ by Eq. (29) and update $\tau = \tau + \triangle\tau$.
    Update aligned background $B^t$ by Eq. (27).
    Update $H^t, F^t$ by Eq.(25), (26), respectively.
    Update $M^t, D^t$ by Eq.(30), (31), respectively.
    Update $R^t, T^t$ by Eq.(33), (34), respectively.
    Update $(\sigma^t)^2, b^t$ by Eq.(36), (38), respectively.
end while
Output: $\Theta^t = \{H^t, D^t, B^t, F^t, {\sigma^t}^2, b^t\}$;
        Recovered frame = ${H^t}^{\perp} \circ B^t + H^t \circ F^t$.
```

D. Some Remarks

1) Explanation for Function of D_{KL} Regularizations: It should be noted that the D_{KL} regularization in Eq. (22) and Eq. (23) intrinsically conduct the superiority of the proposed OTMS-CSC model for removing dynamic rain/snow. Specifically, the offline MS-CSC model [16] intrinsically specifies one unique value for the parameter σ^2 as well as b to represent the background noise variance and scale parameter in feature map representing rain/snow, respectively, for all the frames of the video. The offline model is thus only suitable to be used in the video with static background and consistent rain/snow shapes. The OTMS-CSC model, however, can finely handle dynamic rain with videos with dynamic rain and varying background noises. This advantage is naturally conducted by the fact that the model assumes that each frame has its own specific noise parameter $(\sigma^t)^2$ and scale parameter b^t, by simultaneously fitting the knowledge of the current frame and being regularized by those ($(\sigma^{t-1})^2$ and b^{t-1}) obtained from the previous frames. This makes this model, implemented for each new frame in an online mode, better adapt the specific structures of rain/snow or background for the current frame, generally varied from those for previous ones.

To more intuitively clarify this point, we illustrate in Fig. 2 the changing tendencies of parameters $(\sigma^t)^2$ and b^t for a sequence of video frames, containing snow varying from heavy to light. It can be seen that both $(\sigma^t)^2$ and b^t are gradually decreasing along time, finely reflecting the dynamic changes of snow along time.

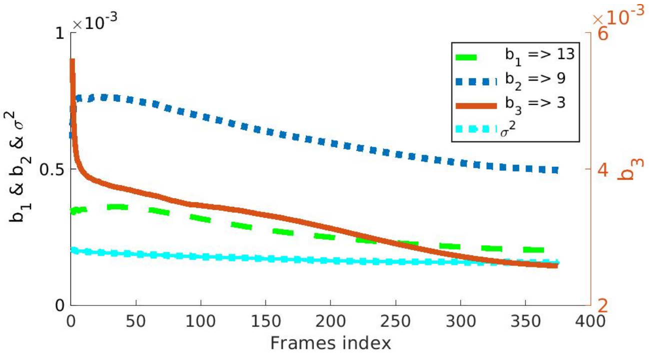


Fig. 2. The changing tendency of the noise variance $(\sigma^t)^2$ and the scale parameter b^t along a video containing dynamic snow varying from heavy to light. Since there are three different scales of filters (used for 13×13, 9×9, 3×3 patch sizes , respectively) are utilized, there are three scale parameter changing curves.

2) The Case for Videos with Rain/Snow and Dynamic Background: Given a sequence of surveillance video, if we stack the video frames as columns of a matrix, then the low-rank component naturally corresponds to the stationary background and the remaining component captures the moving objects and rain layers. Obviously, for videos with rain/snow and dynamic background, the background motions like swing leaves or water waves should also be removed from the stationary background, as shown in Fig. 1 (b), thus mixed with the moving objects and rain layers.

Actually, the filters D_{ks}^t of Eq. (7) can always help distinguish background dynamics and rain layers. Specifically, the patterns of rain streaks are relatively vertical or oblique in most cases, and the dynamic backgrounds, like water waves or swing leaves, are more often figured by relatively more horizontal filters. To make an intuitive understanding, the complete decomposition process of the OTMS-CSC model on a video with generated rain and water waves is displayed in Fig. 1. As shown in the second row of Fig. 1, the entire rain mixed with water waves can be divided into four sub-layers, the corresponding size of filters shown in the top-left corner are 5*5, 5*5, 9*9, 13*13, respectively. The first separated layer with the relatively horizontal filter appropriately extract the water waves, while the other three separated layers encode various rain layers with multiple scales and shapes. Thus, for videos with dynamic background, the final dynamic background (as shown in Fig. 1 (e)) should be a combination between stationary background and those separated sub-layers representing background motions, and the rain layer (as shown in Fig. 1 (f)) should be a combination among those other sub-layers.

3) Background Amelioration: Our method gradually updates the background B^t of the current frame from the affine transformation on that of the last frame B^{t-1} by Eq. (27). Due to constantly temporal scene shifting of the videos (especially brought by the camera moving along a certain direction in a short time) and incremental accumulation of computing errors, the recovered video background tends to be gradually deviated from the real one, which always makes the rain-removed videos look more or less blurry after a period of algorithm computing. To alleviate this issue, our algorithm needs to specifically ameliorate the background knowledge B^t after implementing certain frames by our algorithm.

Our strategy is as follows: When our algorithm is run l iterations (the current frame is denoted as the t^{th} one), we then pick up two frames before and after current frame to get a subgroup as:

$$\hat{\mathcal{X}}^t = [X^{t-2}, X^{t-1}, X^t, X^{t+1}, X^{t+2}]. \tag{39}$$

We then easily align all other frames under the reference of the current frame by using the similar manner as we introduced in Eq. (27), to obtain the aligned subgroup as (a $h \times w \times 5$ tensor):

$$\mathcal{T}\hat{\mathcal{X}} = [TX^{t-2}, TX^{t-1}, X^{t}, TX^{t+1}, TX^{t+2}], \qquad (40)$$

where $TX^j = X^t \odot \tau^j$ $(j = t-2, t-1, t+1, t+2)$, and τ^j is calculated readily by Eq. (27)-(29). Then we can easily calculate the optimal rank-one approximation $\hat{B}^{t-1}$ of the unfolded matrix $T\hat{X} \in R^{hw\times 5}$ of $\mathcal{T}\hat{\mathcal{X}}$ efficiently by SVD, and replace B^{t-1} as $\hat{B}^{t-1}$ to get the new ameliorated background initialization.

4) Potential to be Used for Streaming Videos: It is evident that the proposed OTMS-CSC algorithm is implemented in an online mode, i.e., each time run on a unique newly coming frame. This learning manner makes our method potentially applicable to practical streaming videos. In specific, in each implementation stage for a frame X^t, the algorithm only requires a fixed memory to restore related parameters $H^t, B^t, D^t, (\sigma^t)^2, b^t$. Besides, since the implementation is similar to each new frame, its time complexity is also fixed in each learning stage. This makes our method potentially feasible to the practical videos continuously coming with streaming format beyond current offline methods, which not only need increasingly more space complexity for larger length of videos, but also require increasingly larger time complexity for larger video sequence (even need to pre-implement the algorithms on the entire video again). This makes them hardly useable to this typical real video format in practice. Comparatively, our method makes the real-time execution of rain removal possible to be realized for practical streaming video. What we need to do is to improve the efficiency of our algorithm on one frame to make it gradually meet the real-time requirements. Possible regimes include further improvement on hardware power, further speed-up on algorithm implementation (like modify it distributed/parallel or transform it in faster implementation platform), or replace some of its stages with faster algorithms. This is a meaningful issue worthy of making further endeavors in future research.

IV. Experimental Results

To make a sufficiently comprehensive and diverse comparison, this section contains experiments on videos with synthetic and real rain/snow, experiments on videos with dynamic background, further vertification of video rain removal on the video instance segmentation task, and failure cases. All experiments were implemented on a PC with i7 CPU and 32G RAM.

Some state-of-the-art video rain/snow removal methods have also been implemented for comparison, including Garg et al. [5][1], Jiang et al. [21][2], Ren et al. [11][3], Wei et al. [12][4], Liu et al. [13][5], Li et al. [16][6], Chen et al. [15][7] and Yang et al. [24][8]. Note that these methods contain both model-driven MAP-based and data-driven deep learning representative state-of-the-art technologies for a comprehensive comparison. And some derain methods for surveillance system, like Wei et al. [12] and Li et al. [16], are only able to handle the videos with definitely static background, thus automatically disappeared in visual and quantitative comparison for videos with background transformations, such as Tab. II and Fig. 5.

A. Experiments on Videos with Synthetic and Real Rain/Snow

In this section, to make a sufficiently comprehensive and diverse comparison, we not only includes almost all typical data in this domain, like NTURain[15][9], but also collects some real rainy and snowy videos from real-world monitoring systems and social media platforms. Considering the limitation of paper length and inconvenience for the result exhibition in video tasks, only twelve videos including five synthetic videos and seven real videos can be displayed on the paper in both quantitative and qualitative perspectives. More video demonstrations on the obtained results by all completing video rain removal methods have been reported in our specifically constructed website[10] for easy and better observation.

All experiments were implemented on a PC with i7 CPU and 32G RAM. Three different scales of filters $(13*13, 9*9, 5*5)$ are implemented on all videos in this subsection.

1) Experiments on Videos with Synthetic Rain/Snow: The synthehtic rainy videos are generated by adding clean video and generated rain directly by pixel, where the rain/snow with various types used were synthetically generated by Photoshop on a black background. We first introduce experiments executed on videos with synthetic rain/snow, including two with static backgrounds, one of them is shown in Fig. 3, and one with evidently dynamic background with evident translations among adjacent frames, as depicted in Fig. 4 and four synthetic videos in the group a of the NTURain [15] testing dataset Fig. 5. The clean videos as shown in Fig. 3 and Fig.4 are downloaded from CDNET database[17][11] and surveillance system of Xi'an Jiaotong University respectively, and those of Fig. 5 are the synthetic testing data of the NTU-Rain dataset [15].

The video with static background as shown in Fig. 3 contains snow. From Fig. 3, we can easily observe that the compared methods proposed by (c) Garg et al., (d) Jiang et al. and (g) Liu et al. fail to completely remove the snow, and that proposed by (e) Ren et al., (f) Wei et al. and (h) Li et al. have not finely kept the shape of the moving objects when removing the rain streaks. Comparatively, our proposed OTMS-CSC method has a better visual performance in both snow removing and background/foreground detail preservation. Quantitative comparisons on two videos are also presented in Tab. I, which

[1] http://www.cs.columbia.edu/CAVE/projects/camera rain/

[2] Code is provided by the authors

[3] http://vision.sia.cn/our%20team/RenWeihong-homepage/vision-renweihong%28English%29.html

[4] http://vision.sia.cn/our%20team/RenWeihong-homepage/vision-renweihong%28English%29.html

[5] https://github.com/flyywh/J4RNet-Deep-Video-Deraining-CVPR-2018

[6] https://github.com/MinghanLi/MS-CSC-Rain-Streak-Removal

[7] https://github.com/hotndy/SPAC-SupplementaryMaterials

[8] https://github.com/flyywh/CVPR-2020-Self-Rain-Removal

[9] https://github.com/hotndy/SPAC-SupplementaryMaterials

[10] https://sites.google.com/view/onlinetmscsc/

[11] http://www.changedetection.net

TABLE I
QUANTITATIVE PERFORMANCE COMPARISON OF ALL COMPETING METHODS ON STATIC VIDEOS WITH SYNTHETIC RAIN AND SNOW. NOTE THAT ALL QUANTITATIVE RESULTS ARE THE MEAN OF ALL FRAMES IN THE VIDEO.

Types	Static videos					
Dataset	Highway			Playground (Fig. 3)		
Metrics	PSNR	VIF	SSIM	PSNR	VIF	SSIM
Input	23.82	0.766	0.929	27.93	0.595	0.831
Garg et al. [5]	24.64	0.750	0.920	35.87	0.819	0.950
Jiang et al.[21]	24.32	0.713	0.929	35.80	0.779	0.977
Ren et al. [11]	23.52	0.681	0.927	30.34	0.921	0.995
Wei et al. [12]	24.43	0.761	0.943	34.58	0.945	0.993
Liu et al. [13]	22.19	0.555	0.895	31.56	0.616	0.946
Li et al. [16]	25.37	0.790	**0.957**	42.95	0.980	0.997
OTMS-CSC	**25.91**	**0.796**	**0.957**	**46.29**	**0.988**	**0.999**

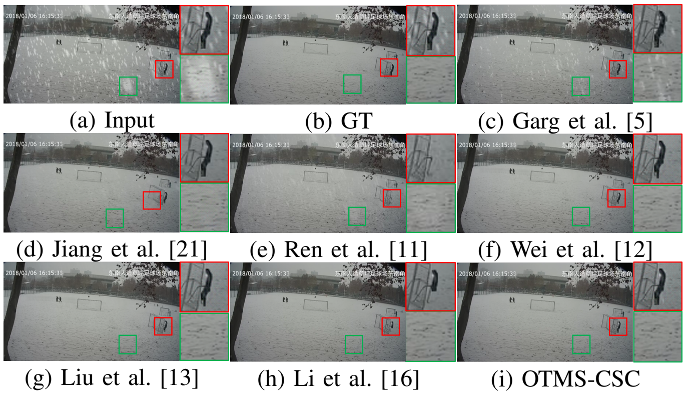

(a) Input (b) GT (c) Garg et al. [5]
(d) Jiang et al. [21] (e) Ren et al. [11] (f) Wei et al. [12]
(g) Liu et al. [13] (h) Li et al. [16] (i) OTMS-CSC

Fig. 3. Visual comparison on a static video with synthetic snow.

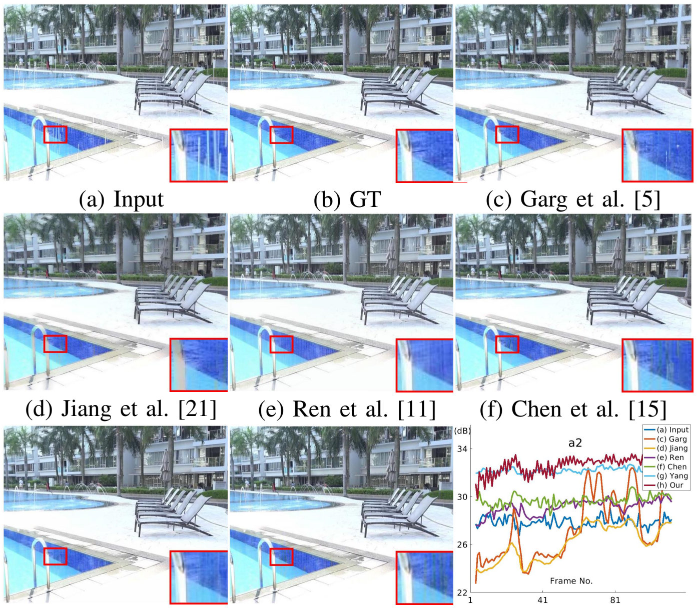

(a) Input (b) GT (c) Garg et al. [5]
(d) Jiang et al. [21] (e) Ren et al. [11] (f) Chen et al. [15]
(g) Yang et al. [24] (h) OTMSCSC (i) PSNR Comparison

Fig. 5. Visual comparison on the panning unstable video with synthetic rain (a2 in the NTURain dataset) and the PSNR evolution curves on all frames in the videos.

fully complies with the aforementioned visual observations. Specifically, we adopt three image quality assessment (IQA) metrics, called PSNR, VIF [60] and SSIM [61], to evaluate the performance of all competing methods on entire videos. Note that the all quantitative results in the table are the mean of all frames in the video. The table indicates that our proposed OTMS-CSC model can perform best in all cases in terms of all IQAs, as compared with other competing methods.

For slow panning videos as shown in Fig. 4 - 5, there are obvious rain streaks remaining on the recovered frames obtained by (c) Garg et al., (e) Ren et al. and (f) Liu et al. The method of (d) Jiang et al. has not done well in preservation of background details (like the texture of wall). Our proposed OTMS-CSC method attains a relatively better performance in both aspects. For the synthetic testing data of NTURain dataset introduced by Chen et al. [15] displayed in Fig. 5, all aforementioned methods still keep its own limitations in rain removal or texture information retention. Besides, the new added method proposed by (f) Chen et al. can hardly remove the heavy rain bars with suddenly bright forming serious occlusions to background scene as shown in red boxes. Comparatively, our proposed method still remains stable and gets expected performance on videos with heavy rain and complex background texture. The average quantitative comparisons in the entire video presented in Tab. II further verify that our proposed model can stay the highest or the second highest in all cases in terms of all IQAs, as compared with other competing methods.

Based on the above tables and figures, proposed OTMS-CSC model achieves stable and best performance on synthetic videos datasets both visually and quantitatively. Considering that all other methods are implemented on the entire video (iteratively utilizing the video multiple times) or need additionally pre-collected training data while our method is sequentially implemented in the video sequence (i.e., each frame is only iterated one time and then dropped out), it should be rational to say our method is efficient.

2) Experiments on Videos with Real Rain/Snow: We further evaluate the performance of the proposed method on videos with real rainy or snowy scenarios. Due to the limitation of paper length, only five real videos have been shown in our experiments, including a video captured under static background and four videos under backgrounds with typical transformations like random jitter, translation, and scale transformation. More video visual results by all completing methods have been reported in our specifically constructed website[10] for easy and better observation. Fig. 6 and Fig. 8 include three public rain videos used in [5] or downloaded from Youtube[12] respectively.

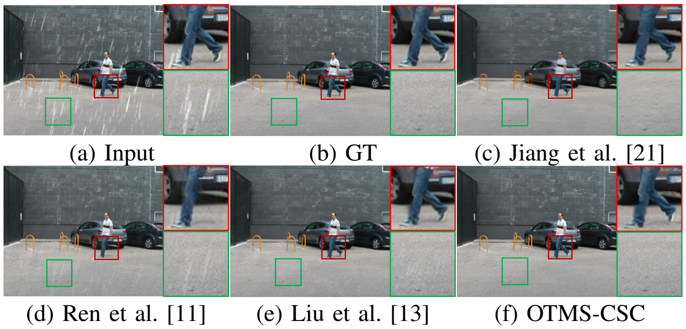

(a) Input (b) GT (c) Jiang et al. [21]
(d) Ren et al. [11] (e) Liu et al. [13] (f) OTMS-CSC

Fig. 4. Visual comparison on a slow panning video with synthetic rain.

[12] https://www.youtube.com/watch?v=kNTYEKjXqzs, HbgoKKj7TNA

TABLE II
QUANTITATIVE PERFORMANCE COMPARISON OF ALL COMPETING METHODS ON VIDEOS WITH SYNTHETIC RAIN AND BACKGROUND TRANSFORMATIONS. ALL QUANTITATIVE RESULTS ARE AVERAGED OVER ALL FRAMES IN THE VIDEOS.

Types	Dynamic video		NTURain									
Dataset	Human (Fig. 4)		$a1$		$a2$ (Fig. 5)		$a3$		$a4$		Average	
Metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Input	29.32	0.909	28.25	0.9350	27.88	0.9436	27.61	0.9193	31.24	0.9440	28.75	0.9355
Garg et al. [5]	36.11	**0.969**	23.65	0.8422	27.37	0.9096	24.14	0.8572	32.62	0.9522	26.95	0.8903
Jiang et al. [21]	32.51	0.960	22.05	0.8829	25.99	0.7458	24.13	0.6701	29.70	0.9535	25.47	0.8156
Ren et al. [11]	31.33	0.956	28.93	0.9335	29.06	0.9440	28.46	0.9243	30.43	0.9582	29.22	0.9400
Liu et al. [13]	34.69	0.965	-	-	-	-	-	-	-	-	-	-
Chen et al. [15]	-	-	29.15	0.9505	29.73	0.9533	29.13	0.9440	33.86	0.9673	30.47	0.9463
Yang et al. [24]	-	-	32.17	0.9616	32.22	0.9659	31.57	0.9534	**35.76**	**0.9736**	**32.93**	0.9636
OTMS-CSC	**37.65**	0.966	**30.88**	**0.9679**	**32.60**	**0.9708**	**31.82**	**0.9649**	32.85	0.9620	32.04	**0.9664**

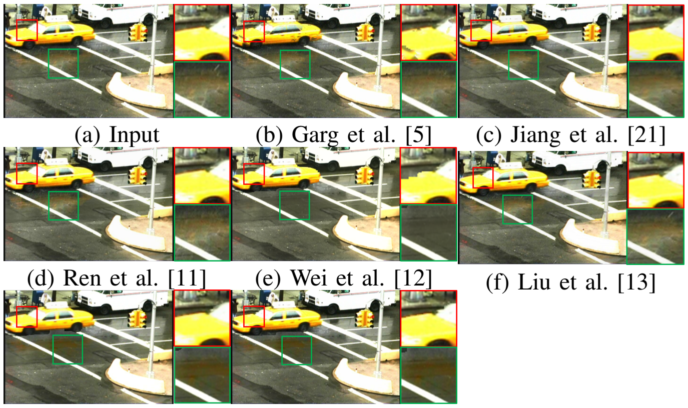

(a) Input (b) Garg et al. [5] (c) Jiang et al. [21]
(d) Ren et al. [11] (e) Wei et al. [12] (f) Liu et al. [13]
(g) Li et al. [16] (h) OTMS-CSC

Fig. 6. Visual comparison on a typical real video with dynamic rain and static background.

Fig. 7 shows two real rainy videos from the real testing data of NTURain dataset.

The video shown in Fig. 6 is captured by a surveillance equipment in street, containing dynamically varying rain along time. From the figures, we can easily observe that the derained frames of all other compared methods still contain certain rain streaks. By contrast, our proposed OTMS-CSC method is capable of better removing all the rain and snow.

Fig. 7 and 8 show rain and snow removal results on real videos with slow panning and scaling, respectively. It can be seen from above two figures that the methods proposed by (b) Garg et al. and (c) Jiang et al. cannot fully remove rain/snow and fail to recover the texture information underlying the frames, that proposed by (d) Ren et al. and (e) Liu et al. fail to detect and remove the rain streaks or snowflakes since they are not capable of dealing with video transformations. The method of (e) Chen et al. sometimes misses some heavy rain bars with obvious bright. The proposed OTMS-CSC method can obtain better visualized performance since they consider the background transformation and online multi-scale convolutional sparse coding in the modeling. This verifies that aligning video background can help to improve the final performance of rain/snow removal especially for videos with background transformation. Please refer to the website[10] for more comprehensive illustration of the video results.

3) Run Time Comparison: Although some earlier methods, such as Garg et al. and Jiang et al., run very efficiently, their performance is not comparable with recent video rain removal methodologies. Therefore, considering the balance between running time and performance, this paper only includes those methods published in recent years, which usually are comparable in performance. To show the efficiency of the proposed online method, we list the average running time per frame of each compared method in Tab. III in four representative static and dynamic videos with synthetic and real rain/snow, respectively. From the table, the speed advantage of the OTMS-CSC method is evident attributed to its online learning manner. Besides, in order to better intuitive time comparisons between offline and online learning, corresponding to MS-CSC[16] and OTMS-CSC model respectively, the time line of MS-CSC model in dynamic videos shown in Fig. 9 (c-d) also have been provided in this part. As we show in Fig. 9, this online method has a good scalability, i.e., its time cost is linearly increasing with more input video frames, naturally due to its fixed training time on each video frame. Together with its fixed space complexity along time as discussed in Sec. 3.4.4, the method is expected to be potentially useful for real streaming videos.

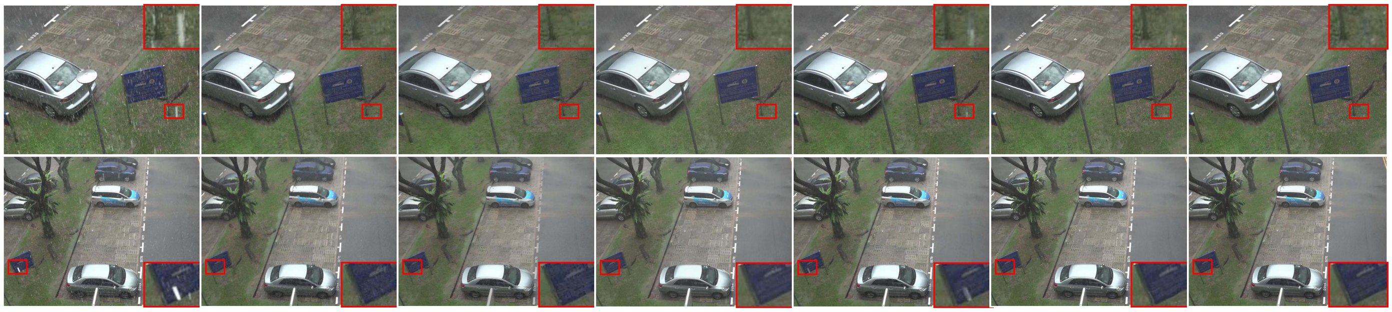
TABLE III
AVERAGE RUNNING TIME COMPARISON OF ALL COMPETING METHODS ON FOUR TYPICAL RAINY/SNOWY VIDEOS WITH STATIC OR DYNAMIC BACKGROUND. (UINT: S/FRAME)

Type	Dataset	Size	Ren [11]	Wei [12]	Liu [13]	Li [16]	Our
Static	Fig. 3	$270 * 480$	3.67	8.62	4.82	3.37	**0.96**
	Light	$360 * 480$	8.05	13.30	4.82	2.69	**0.88**
Dynamic	Fig. 4	$288 * 352$	50.3	-	4.03	-	**0.87**
	Fig. 8	$360 * 640$	80.4	-	8.55	-	**1.36**

B. Experiments on Videos with Dynamic Background

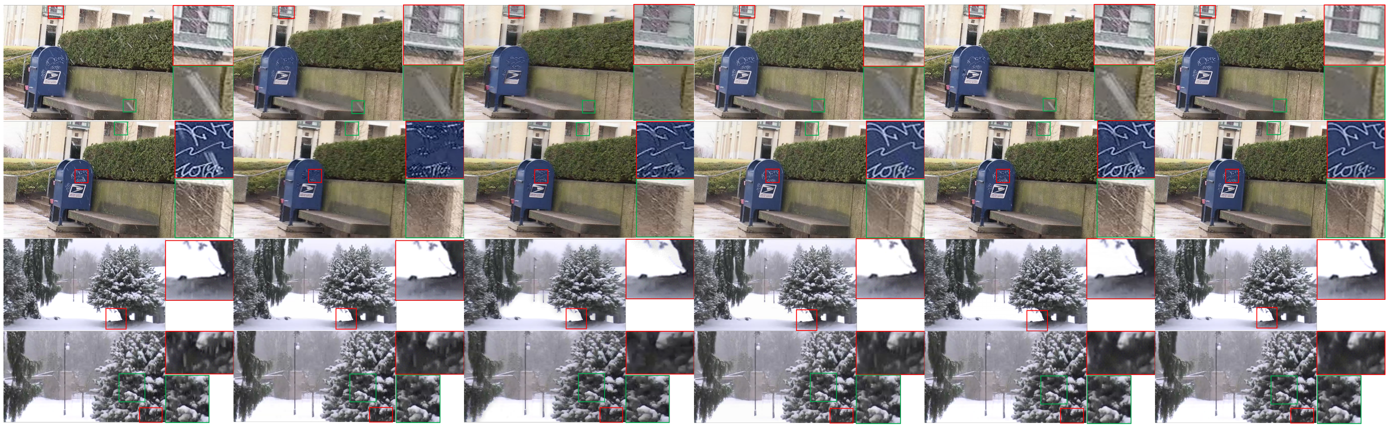
The rain removal experiments on videos with dynamic background have been performed on the proposed synthetic CDNet-Rain dataset. The rough process of generating the dataset is to add rgenerated ain streaks by Adobe After Effects [13] to the frames clipped from the dynamic background sequences of the CDNet [17] dataset. There are seven sequences in the CDNet-Rain dataset, two of which are based on the same sequence with swing leaves, named Fall01 and Fall02 respectively. The difference between them is that the former does not contain moving objects, while the latter does. The detailed introduction of CDNet-Rain dataset is listed in Tab.

[13] https://www.adobe.com/products/aftereffects.html



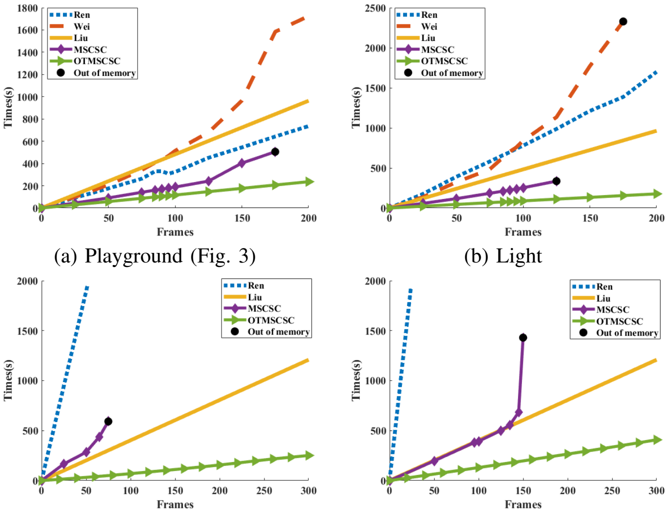
(a) Input (b) Garg et al. [5] (c) Jiang et al. [21] (d) Ren et al. [11] (e) Chen et al. [15] (f) Yang et al. [24] (g) OTMS-CSC

Fig. 7. Visual comparison on two real rainy videos extracted from panning unstable cameras (the NTURain testing dataset ra2 and ra3 respectively).



(a) Input (b) Garg et al. [5] (c) Jiang et al. [21] (d) Ren et al. [11] (e) Liu et al. [13] (f) OTMS-CSC

Fig. 8. Visual Comparison on two real snowy videos with obvious horizontal movement and scale transformation respectively.



(a) Playground (Fig. 3)

(b) Light

(c) Human (Fig. 4)

(d) Pine (Fig. 8)

Fig. 9. Run time comparison of comparable methods on four videos with static ((a) and (b)) or transformed background ((c) and (d)) respectively. The black point denotes the method over the current frames will report the error: out of memory.

IV. Four different scales of filters $(13 * 13, 9 * 9, 5 * 5, 5 * 5)$ are adopted on all videos of CDNet-Rain dataset.

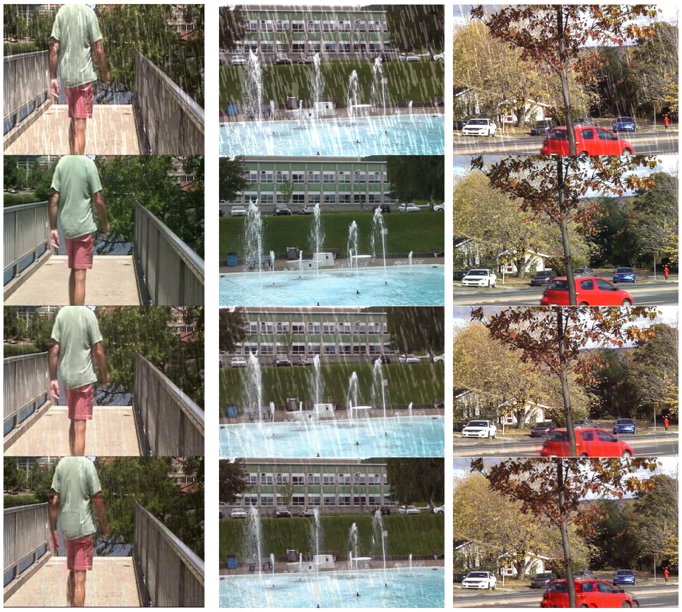
In order to test the stability and generalizable usefulness of video rain removal algorithms more fairly, we execute all seven testing video sequences on a fixed experimental setting for all competing methods. The quantitative performance comparison are listed in Tab. IV. It is seen that the proposed OTMS-CSC model achieves the best results on five out of the seven video sequences, and also stays the best average performance on the average of whole dataset. For those two video sequences Fall01 and Fall02, the performance of proposed OTMS-CSC model is sligtly lower than the SLDNet model, because the graph cut algorithm used in our algorithm for cutting moving objects mask is not accurate enough in the segmentation of the object edges. Furthermore, the higher SSIM index of OTMS-CSC model over SLDNet validates the effectiveness of multi-scale convolutional sparse coding model, which can separate background motions from the mixed rain layer. Fig. 10 shows some visual comparison of video rain removal for all competing models on synthetic CDNet-Rain dataset. As compared with the SLDNet model [24], which can preserve most of the details in the background but remain some streak residuals, the proposed OTMS-CSC model estimates a cleaner background with less rain streak residuals , which substantiates the superiority of our proposed method in generalization on dynamic videos.

C. Video Rain Removal Verification on Video Instance Segmentation Task

In order to further verify whether removing rain and snow from a video could bring positive impact on the sub-sequence video processing tasks, we take the video instance segmentation (VIS) task [18], which aims to simultaneously detect, segment and track instances in videos, as an example for

TABLE IV
QUANTITATIVE PERFORMANCE COMPARISON OF ALL COMPETING METHODS ON SYNTHETIC CDNET-RAIN DATASET WITH DYNAMIC BACKGROUND.

Methods		Input		SLDNet [24]		OTMS-CSC	
Videos	Frames clip	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Canoe	801-1100	21.52	0.6747	24.25	0.7594	**24.95**	**0.7657**
Boats	6901-7300	23.00	0.7873	28.05	0.8808	**28.58**	**0.8853**
Overpass	2201-2800	21.53	0.7594	25.29	0.8554	**27.32**	**0.8741**
Fountion01	1-400	19.29	0.7072	22.12	0.7733	**24.13**	**0.8661**
Fountaion02	1-400	22.65	0.8173	27.63	0.8988	**29.99**	**0.9120**
Fall01	1-200	22.53	0.8716	**26.95**	**0.9261**	25.04	0.9181
Fall02	3901-4000	22.52	0.8740	**27.18**	**0.9279**	25.87	0.9239
Ave. Perf.	-	21.86	0.7845	25.92	0.8602	**26.55**	**0.8779**



(a) Overpass (b) Fountain01 (c) Fall02

Fig. 10. Visual comparison on synthetic CDNet-Rain dataset with dynamic background. From upper to lower: input frame, groundtruth clean frame, results produced by SLDNet and OTMS-CSC model respectively.

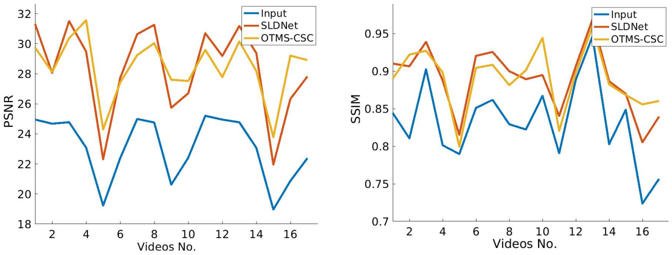
evaluation. Specifically, the video rain removal algorithms can be served as a pre-processing step to ameliorate quality of images/videos, so as to make the following processing task capable of being normally handled by off-the-shelf techniques.

To facilitate such an evaluation, based on the large-scale video instance segmentation valid dataset YouTube-VIS proposed in [18], which consists of 301 high-resolution YouTube videos, we propose a video rain removal benchmark for video instance segmentation task called YouTube-VIS-Rain. Specifically, we selected seventeen outdoor videos from YouTube-VIS valid dataset and synthesized rain over these videos with varying parameters. In the pre-processing step, the seventeen synthetic videos were implemented by video rain removal methods for removing rains. The obtained videos are then put back into the YouTube-VIS valid dataset to perform video instance segmentation task.

Quantitative performance metrics for both tasks are taken into account, including PSNR and SSIM metrics for video rain removal task, together with average precision (AP) and average recall (AR) metrics [62] for video instance segmentation task. For video rain removal task, the average quantitative performance comparison on the seventeen videos from YouTube-VIS-Rain dataset are shown in the second and third columns of Tab. V, and the PSNR and SSIM comparison on each video sequence are displayed in Fig. 11. Compared with the SLDNet model, the proposed OTMS-CSC method achieves better results on average PSNR and SSIM metrics. Visual results shown in the first row of Fig. 12 indicate that the SLDNet model fails to detect completely rain streaks and still leaves obvious rain marks in the video, while the proposed OTMS-CSC method can better remove rain streaks from the background. But affected by total variation (TV) regularization on the foreground, the OTMS-CSC model perform still not sufficiently perfect in removing rain from the foreground compared with the SLDNet model. This is why the quantitative metrics of OTMS-CSC model are slightly lower than those of the SLDNet model in some videos as shown in Fig. 11.

TABLE V
QUANTITATIVE PERFORMANCE COMPARISON FOR BOTH VIDEO RAIN REMOVAL TASK AND VIDEO INSTANCE SEGMENTATION TASKS ON YOUTUBE-VIS-RAIN DATASET.

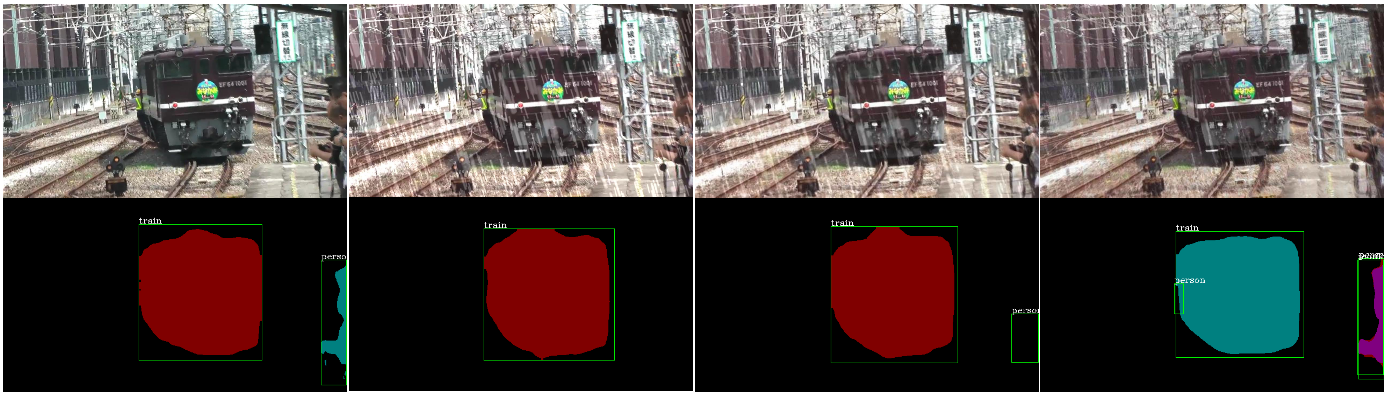
Tasks	Video Rain Removal		Video Instance Segmentation				
Metrics	PSNR	SSIM	mAP	AP50	AP75	AR1	AR10
GT	-	-	30.32	51.1	32.6	31.0	35.4
Rainy Input	23.06	0.8315	29.75	50.7	31.5	30.5	34.9
SLDNet [24]	28.30	0.8884	**29.98**	**50.7**	**31.8**	**30.7**	**35.1**
OTMS-CSC	**28.44**	**0.8894**	29.95	50.6	**31.8**	**30.7**	35.0



(a) PSNR Comparison (b) SSIM Comparison

Fig. 11. PSNR and SSIM evolution curves of video rain removal task on all seventeen synthetic videos from the YouTube-VIS-Rain dataset.

We employ the video instance segmentation algorithm proposed in [18] to further evaluate the impact of rain/snow on the performance of video instance segmentation task. There are four settings on those seventeen videos from YouTube-VIS-Rain dataset for comparison: clean videos (GT), rainy videos, rain-free videos removed by SLDNet and OTMS-CSC model respectively. The corresponding quantitative metrics are listed in the last five columns of Tab. V. It is seen that compared with taking clean videos as input, introducing seventeen dirty videos into YouTube-VIS dataset does cause obvious performance degradation in all five metrics. The mAP index decreased from 30.32 to 29.75, and the AP75 index fell 1.1. After rain removal pre-processing by SLDNet and OTMS-CSC models, all metrics of VIS task have been moderately improved. Since the VIS task pays more attention on the moving objects, the performance of the proposed OTMS-CSC method is sightly lower than those of the SLDNet model. The second row of Fig. 12 exhibits instance segmentation visualization results for four settings. As can be seen, in those rainy/snowy videos, the actual features of objects (such



(a) GT (b) Rainy Input (c) SLDNet[3] (d) OTMS-CSC

Fig. 12. Performance visual comparison for both video rain removal task and video instance segmentation task on a synthetic video of the YouTube-VIS-Rain dataset.

as person) are very likely to be destroyed by rain or snow, making it difficult for the network to classify and track the instances accurately. The rain/snow removal pre-processing does be beneficial to the final performance for this task.

D. Failure cases

The proposed method still has limitations on handling general video rain removal tasks, especially for those captured with non-surveillance cameras. Specifically, there are three limitations of our proposed OTMS-CSC method. Firstly, when camouflage effects occur (the photometric similarity of moving objects and the background), the graph cut algorithm used to obtain moving object mask in our algorithm tends to confuse the moving objects with the background, resulting in incomplete moving object mask, especially in videos with extensive moving objects. Secondly, our proposed OTMS-CSC model currently cannot handle those challenging videos with fast illumination changes because it does not meet the low-rank assumption of background extraction. Thirdly, the proposed model is with limitation for videos captured by fast moving cameras, like the videos in the Group b (with the speed range between 20 to 30 km/h) of synthetic test data of the NTURain dataset. For those videos, the affine transformation operator used to align the background of the video frame may lack sufficient overlap information between the frames to be aligned. We'll make further endeavor on these degenerated cases for the video rain removal task in our future research.

V. Conclusion

In this paper, we have proposed a new rain/snow removal method for surveillance videos containing dynamic rain/snow captured with camera jitter. Both dynamic characteristics of rain/snow variations and background scenes along time inevitably encountered in real cases, have been fully considered in our method. Especially, the method is with a natural online implementation manner, with fixed space and time complexity for handling each frame of continuously coming videos, making it potentially useful for dealing with practical streaming video sequences. In the future, we will further ameliorate the capability of the proposed method in more challenging video cases, like those captured under fast moving cameras or those under background with strong color contrast and rain/snow with large streak shapes, and try to design rational techniques or use some advanced computing equipments to further speed up the method for each unique frame to make it meet with the real-time requirements on practical streaming videos. Furthermore, we will consider the spatial heteroscedasticity property [63] of noises in our future work. We will also try to consider how to better express raindrop numbers in the rain removal tasks to more faithfully encode the feature maps of our model in our future investigations.

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *IEEE Computer Society Conference on Computer Vision*, vol. 1, no. 12, pp. 886–893, 2005.

[2] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani, "Person re-identification by symmetry-driven accumulation of local features," in *Computer Vision and Pattern Recognition*, 2010.

[3] S. Mukhopadhyay and A. K. Tripathi, "Combating bad weather part i: Rain removal from video," *Synthesis Lectures on Image, Video, and Multimedia Processing*, vol. 7, no. 2, pp. 1–93, 2014.

[4] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis." *A model of saliency-based visual attention for rapid scene analysis.*, vol. 20, no. 11, pp. 1254–1259, 1998.

[5] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2004, pp. I–I.

[6] ——, "When does a camera see rain?" in *International Conference on Computer Vision*, vol. 2. IEEE, 2005, pp. 1067–1074.

[7] X. Zhang, H. Li, Y. Qi, W. Leow, and T. Ng, "Rain removal in video by combining temporal and chromatic properties," in *IEEE International Conference on Multimedia and Expo*, 2006, pp. 461–464.

[8] P. Barnum, T. Kanade, and S. Narasimhan, "Spatio-temporal frequency analysis for removing rain and snow from videos," *Photometric Analysis For Computer Vision*.

[9] A. K. Tripathi and S. Mukhopadhyay, "A probabilistic approach for detection and removal of rain from videos," *Iete Journal of Research*, vol. 57, no. 1, p. 82, 2011.

[10] Y.-L. Chen and C.-T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *International Conference on Computer Vision*, 2013, pp. 1968–1975.

[11] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang, "Video desnowing and deraining based on matrix decomposition." in *Computer Vision and Pattern Recognition*, 2017.

[12] W. Wei, L. Yi, Q. Xie, Q. Zhao, D. Meng, and Z. Xu, "Should we encode rain streaks in video as deterministic or stochastic?" in *International Conference on Computer Vision*, 2017.

[13] J. Liu, W. Yang, s. Yang, and G. Zongming, "Erase or fill? deep joint recurrent rain removal and reconstruction in videos," *Computer Vision and Pattern Recognition*, 2018.

[14] W. Yang, J. Liu, and J. Feng, “Frame-consistent recurrent video deraining with dual-level flow,” *Computer Vision and Pattern Recognition*, pp. 1661–1670, 2019.

[15] J. Chen, C.-H. Tan, J. Hou, and chau Lap-Pui, “Robust video content alignment and compensation for clear vision through the rain,” *Computer Vision and Pattern Recognition*, 2018.

[16] M. Li, Q. Xie, Q. Zhao, W. Wei, S. Gu, J. Tao, and D. Meng, “Video rain removal by multiscale convolutional sparse coding,” *Computer Vision and Pattern Recognition*, 2018.

[17] N. Goyette, P.-M. Jodoin, F. Porikli, J. Konrad, and P. Ishwar, “Changedetection. net: A new change detection benchmark dataset,” in *Computer Vision and Pattern Recognition Workshops*. IEEE, 2012, pp. 1–8.

[18] L. Yang, Y. Fan, and N. Xu, “Video instance segmentation,” in *International Conference on Computer Vision*, 2019, pp. 5188–5197.

[19] K. Garg and S. K. Nayar, “Vision and rain,” *International Journal of Computer Vision*, vol. 75, no. 1, pp. 3–27, 2007.

[20] P. C. Barnum, S. Narasimhan, and T. Kanade, “Analysis of rain and snow in frequency space,” *Computer Vision and Pattern Recognition*, vol. 86, no. 2, p. 256, 2010.

[21] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, “A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors.” in *Computer Vision and Pattern Recognition*, 2017.

[22] J. H. Kim, J. Y. Sim, and C. S. Kim, “Video deraining and desnowing using temporal correlation and low-rank matrix completion.” *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2658–70, 2015.

[23] V. Santhaseelan and V. K. Asari, “Utilizing local phase information to remove rain from video,” *International Journal of Computer Vision*, vol. 112, no. 1, pp. 71–89, 2015.

[24] W. Yang, R. T. Tan, S. Wang, and J. Liu, “Self-learning video rain streak removal: When cyclic consistency meets temporal correspondence,” in *Computer Vision and Pattern Recognition*, 2020, pp. 1720–1729.

[25] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, “Automatic single-image-based rain streaks removal via image decomposition,” *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1742–1755, 2012.

[26] Y. Luo, Y. Xu, and H. Ji, “Removing rain from a single image via discriminative sparse coding,” in *International Conference on Computer Vision*, 2015, pp. 3397–3405.

[27] X. Ding, L. Chen, X. Zheng, Y. Huang, and D. Zeng, “Single image rain and snow removal via guided l0 smoothing filter,” *Multimedia Tools and Applications*, vol. 75, no. 5, pp. 2697–2712, 2016.

[28] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, “Rain streak removal using layer priors,” in *Computer Vision and Pattern Recognition*, 2016, pp. 2736–2744.

[29] Y. Wang, S. Liu, C. Chen, and B. Zeng, “A hierarchical approach for rain or snow removing in a single color image,” *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3936–3950, 2017.

[30] S. Gu, D. Meng, W. Zuo, and L. Zhang, “Joint convolutional analysis and synthesis sparse representation for single image layer separation.” in *International Conference on Computer Vision*, 2017.

[31] H. Zhang and V. M. Patel, “Convolutional sparse and low-rank coding-based rain streak removal,” in *IEEE Winter Conference on Applications of Computer Vision*, 2017.

[32] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, “Clearing the skies: A deep network architecture for single-image rain removal,” *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, 2017.

[33] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, “Removing rain from single images via a deep detail network,” in *Computer Vision and Pattern Recognition*, 2017, pp. 3855–3863.

[34] H. Zhang and V. M. Sindagi, V an Patel, “Image de-raining using a conditional generative adversarial network,” in *arXiv:1701.05957v3*, 2017, pp. 111–122.

[35] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, “Deep joint rain detection and removal from a single image,” in *Computer Vision and Pattern Recognition*, 2017.

[36] Y.-F. Liu, D.-W. Jaw, S.-C. Huang, and J.-N. Hwang, “Desnownet: Context-aware deep network for snow removal,” *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 3064–3073, 2018.

[37] W. Yang, R. T. Tan, J. Feng, J. Liu, S. Yan, and Z. Guo, “Joint rain detection and removal from a single image with contextualized deep networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[38] R. Li, L. F. Cheong, and R. T. Tan, “Heavy rain image restoration: Integrating physics model and conditional adversarial learning,” *Computer Vision and Pattern Recognition*, pp. 1633–1642, 2019.

[39] H. Wang, Q. Xie, Q. Zhao, and D. Meng, “A model-driven deep neural network for single image rain removal,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3103–3112.

[40] H. Yong, D. Meng, W. Zuo, and L. . Zhang, “Robust online matrix factorization for dynamic background subtraction,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[41] C. Lu, J. Feng, Y. Chen, W. Liu, Z. Lin, and S. Yan, “Tensor robust principal component analysis: Exact recovery of corrupted low-rank tensors via convex optimization,” in *Computer Vision and Pattern Recognition*, 2016, pp. 5249–5257.

[42] M. Shakeri and H. Zhang, “Moving object detection under discontinuous change in illumination using tensor low-rank and invariant sparse decomposition,” in *Computer Vision and Pattern Recognition*, 2019, pp. 7221–7230.

[43] Z. Zhang, A. Ganesh, X. Liang, and Y. Ma, “Tilt: Transform invariant low-rank textures,” *International Journal of Computer Vision*, 2012.

[44] X. Zhang, D. Wang, Z. Zhou, and Y. Ma, “Simultaneous rectification and alignment via robust recovery of low-rank tensors,” *Neural Information Processing Systems*, 2013.

[45] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc, 2011.

[46] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust principal component analysis?” *Journal of the ACM*, vol. 58, no. 3, p. 11, 2011.

[47] D. Meng and F. De La Torre, “Robust matrix factorization with unknown noise,” in *International Conference on Computer Vision*, 2013, pp. 1337–1344.

[48] Q. Zhao, D. Meng, Z. Xu, W. Zuo, and Y. Yan, “l1-norm low-rank matrix factorization by variational bayesianmethod.” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, no. 4, pp. 829–839, 2015.

[49] Q. Zhao, D. Meng, Z. Xu, W. Zuo, and L. Zhang, “Robust principal component analysis with complex noise.” in *International Conference on Machine Learning*, 2014, pp. 55–63.

[50] X. Cao, Q. Zhao, D. Meng, Y. Chen, and Z. Xu, “Robust low-rank matrix factorization under general mixture noise distributions,” *IEEE Transactions on Image Processing*, vol. 25, no. 10, pp. 4677–4690, 2016.

[51] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, “Deconvolutional networks,” in *IEEE Winter Conference on Applications of Computer Vision*.

[52] D. Meng, Q. Zhao, and Z. Xu, “Improve robustness of sparse pca by l 1-norm maximization,” in *Pattern Recognition*, 2012, pp. 487–497.

[53] J. Wang, Q. Li, S. Yang, W. Fan, P. Wonka, and J. Ye, “A highly scalable parallel algorithm for isotropic total variation models,” in *International Conference on Machine Learning*, 2014, pp. 235–243.

[54] X. Zhou, C. Yang, and W. Yu, “Moving object detection by detecting contiguous outliers in the low-rank representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 597–610, 2013.

[55] M. Shakeri and H. Zhang, “Corola: A sequential solution to moving object detection using low-rank approximation,” *Computer Vision and Image Understanding*, vol. 146, pp. 27–39, 2016.

[56] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.

[57] V. Kolmogorov and R. Zabin, “What energy functions can be minimized via graph cuts?” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 2, pp. 147–159, 2004.

[58] B. Wohlberg., “Efficient convolutional sparse coding.” in *In IEEE ICASSP*, 2014.

[59] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online learning for matrix factorization and sparse coding,” *Journal of Machine Learning Research*, vol. 11, no. 1, pp. 19–60, 2009.

[60] H. R. Sheikh and A. C. Bovik, “Image information and visual quality,” *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 430–444, 2006.

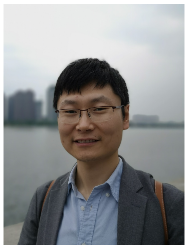
[61] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[62] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*. Springer, 2014, pp. 740–755.

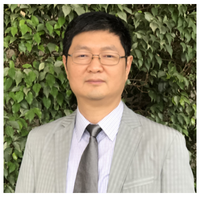
[63] Y. Chen, X. Cao, Q. Zhao, D. Meng, and Z. Xu, “Denoising hyperspectral image with non-i.i.d. noise structure,” *IEEE Transactions on Cybernetics*, 2017.

Minghan Li received the B.Sc degree in 2016 from Xinjiang University, Urumqi, China, and the M.Sc. degree in mathematics from Xi'an Jiaotong University, Xi'an, China, in 2019. She is currently pursuing the Ph.D. degree in the Department of Computing, Hong Kong Polytechnic University, Hong Kong. Her current research interests include video rain/snow removal, video detection and segementation, deep learning and graph model.

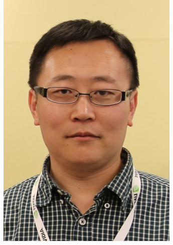
Xiangyong Cao (Member, IEEE) received the B.Sc. and Ph.D. degrees from the Xi'an Jiaotong University, Xi'an, China, in 2012 and 2018, respectively. From 2016 to 2017, he was a Visiting Scholar with Columbia University, New York, NY, USA. He is currently an Assistant Professor with the School of Mathematics and Statistics, Xi'an Jiaotong University. His research interests include statistical modeling, and image processing.



Qian Zhao received the B.Sc. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2009 and 2015, respectively. He was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA, USA, from 2013 to 2014. He is currently an Associate Professor with the School of Mathematics and Statistics, Xi'an Jiaotong University. His current research interests include low-rank matrix/tensor analysis, Bayesian modeling, and meta-learning.



Lei Zhang (M'04, SM'14, F'18) received his B.Sc. degree in 1995 from Shenyang Institute of Aeronautical Engineering, Shenyang, P.R. China, and M.Sc. and Ph.D degrees in Control Theory and Engineering from Northwestern Polytechnical University, Xi'an, P.R. China, in 1998 and 2001, respectively. From 2001 to 2002, he was a research associate in the Department of Computing, The Hong Kong Polytechnic University. From January 2003 to January 2006 he worked as a Postdoctoral Fellow in the Department of Electrical and Computer Engineering, McMaster University, Canada. In 2006, he joined the Department of Computing, The Hong Kong Polytechnic University, as an Assistant Professor. Since July 2017, he has been a Chair Professor in the same department. His research interests include Computer Vision, Image and Video Analysis, Pattern Recognition, and Biometrics, etc. Prof. Zhang has published more than 200 papers in those areas. As of 2020, his publications have been cited more than 57,000 times in literature. Prof. Zhang is a Senior Associate Editor of IEEE Trans. on Image Processing, and is/was an Associate Editor of IEEE Trans. on Pattern Analysis and Machine Intelligence, SIAM Journal of Imaging Sciences, IEEE Trans. on CSVT, and Image and Vision Computing, etc. He is a "Clarivate Analytics Highly Cited Researcher" from 2015 to 2020. More information can be found in his homepage http://www4.comp.polyu.edu.hk/~cslzhang/.



Deyu Meng (M'09) received the B.Sc., M.Sc., and Ph.D. degrees in 2001, 2004, and 2008, respectively, from Xi'an Jiaotong University, Xi'an, China. He is currently a professor with School of Mathematics and Statistics, Xi'an Jiaotong University, and adjunct professor with Macau Institute of Systems Engineering, Macau University of Science and Technology. From 2012 to 2014, he took his two-year sabbatical leave in Carnegie Mellon University. His current research interests include model-based deep learning, variational networks, and meta-learning.