Video denoising for security and privacy in fog computing

Hong Zhang1 | Yifan Yang1 | Ding Yuan1 | Daniel Sun2 | Jun Zhang3 | Guoqiang Li4 | Mingui Sun5

1 Image Processing Center, Beihang University, Beijing, China
2 CSIRO, Data61, Clayton, Australia
3 Swinburne University of Technology, Melbourne, Australia
4 School of Software, Shanghai Jiaotong University, Shanghai, China
5 University of Pittsburgh, Department of Neurosurgery, Pittsburgh, PA, USA

Correspondence
Yifan Yang, Image Processing Center, Beihang University, Beijing, China. Email: stephenyoung@buaa.edu.cn

Funding information
National Natural Science Foundation of China, Grant/Award Number: 61571026; National Key Project of Research and Development Plan, China, Grant/Award Number: 2016YFE0108100; National Institutes of Health, USA, Grant/Award Number: R01CA165255 and R21CA172864

Summary
To reduce heavy noise from degraded video in low or predictable latency and preserve privacy, a powerful and efficient video denoising algorithm is proposed based on fog computing for Visual Internet of Things. The conventional method is to remove noise in the cloud; however, this may overload computation and communication and raise security and privacy issues. The proposed denoising algorithm is distributed to heterogeneous devices at network edges to preserve privacy and avoid security risks as noise can be reduced in the fog rather than the cloud. To address the problems of latency, communication rate, and extremely heavy noise, structure registration, inter-frame and inner-frame filters, and distribution compensation are applied in the proposed algorithm. A scheme for encrypting the denoised data at network edges is provided so that security and privacy issues may be avoided during transmission and storage. Compared with other denoising approaches under extremely heavy noise conditions, the experimental results demonstrate that the proposed approach achieves superior denoising performance in terms of peak signal-noise ratio and visual quality at low computational cost, high bandwidth efficiency, and low-latency response in a fog computing manner.

KEYWORDS
bi-directional infinite impulse response filter, fog computing, iterative closest set, privacy preservation, structural similarity, video denoising

1 INTRODUCTION

Security surveillance cameras are widely used in recent Visual Internet of Things (VIoT) applications. However, video noise in extreme environments results in degraded visual quality and increased unpredictable bandwidth. If the video denoising routine is centralized and implemented in the cloud, the computation and communication load on cloud servers is inevitably high. Moreover, to denoise encrypted video in the cloud, raw sensitive data may be decrypted and exposed among cloud servers, resulting in security and privacy risks. Thus, an efficient and privacy-preserving approach is required for removing noise from video at network front-end nodes. Extensive research has been conducted on new technologies1-4 to improve the image quality (eg, sensitivity and signal-noise-ratio) of front-end CMOS and CCD devices. However, the noise cannot be eliminated thoroughly. In extreme environments such as dark night and hazy weather, a surveillance camera usually operates in high-gain amplifier mode or digital enhancement mode to enhance visual information. It is inevitable that noise increases as the image becomes more detailed. Owing to noise contamination, the compression rate may drastically decrease when a video stream is transmitted from a sensor node to a cloud server via the network, resulting in communication bandwidth bottleneck and unpredictable latency. To overcome this problem, the noise should be removed from sensor nodes at network edges. However, the performance of edge devices such as geographically distributed cameras is often limited by memory and processing capacity. Thus, a low-computation and low-latency video denoising algorithm is required. In recent security surveillance applications, low-light night vision and dehaze enhancement have been widely used, resulting in heavy noise. To reduce this noise in low or predictable latency and preserve privacy, a fog computing-based video denoising algorithm is a sensible choice.

The aim of this study is to improve the efficiency and the security of VIoT-based surveillance systems, which benefit from the concept of fog computing.5-8 Using low-computation and high-performance video denoising. The denoised video could reduce the data rate after stream compression at network edges so that the computational workload of the cloud could also be reduced. Moreover, denoising is a selectable function
FIGURE 1 The architecture of fog computing based surveillance system

for security surveillance applications, and there is no need for integration into front-end sensor nodes, thus saving hardware cost. The denoising algorithm can be implemented on heterogeneous fog devices at edges of networks connecting distributed cameras and cloud servers. These devices at fog nodes are regarded as video pre-processing platforms containing embedded system boards or industrial personal computers with video capture modules. In such a system, when denoising is enabled, the denoising process can be carried out on these devices. Furthermore, fog devices provide video compression and encryption services. The compressed and encrypted data stream is transmitted to the cloud, where remote and high-level services are provided such as data storage and log management. This scheme provides more confidential transmission and storage, thus avoiding malicious attacks on the cloud. The infrastructure of the VIoT-based surveillance system can be described as a combination of heterogeneous physical resources in a fog computing environment as shown in Figure 1. To be used on edge devices, denoising algorithms for fog computing should be designed with the following characteristics: high performance, high privacy-preserving, low latency, and low data rate.

In this study, extremely heavy noise is considered. That is, additive white Gaussian noise whose standard deviation $\sigma$ is larger than 100. To the best of the authors’ knowledge, few experimental results have been reported when $\sigma > 100$. In such extreme conditions, traditional methods cannot be applied, and pronounced distortion occurs. Moreover, the quantization problem of data clipping is considered. It is called bounded noisy model. A novel and powerful method is proposed using iterative registration and tempo-spatial filtering based on low-frequency structural features. The computational workload and denoising performance of this method is suitable to latency-sensitive, memory-sensitive, and bandwidth-sensitive edge devices in a fog computing infrastructure. The proposed algorithm can be distributed to the edge devices of a surveillance network. These devices capture original noisy video from cameras as input, and implement video denoising on-line. The denoised video can be compressed into a low data-rate stream. Then, the stream is transmitted to the cloud for further processing. The steps of the proposed method are as follows.

A low-pass filter is first used to smoothen each frame of noisy video on large scale. The filtered video retains low-frequency structural characteristics (e.g., brightness and gradient distribution, and global shape) for subsequent registration of adjacent frames. In this process, a bi-directional infinite impulse response (B-IIR) filter is implemented to realize fast smoothening.

Subsequently, a structural feature set is constructed using sparse points that are extracted from the wavelike surface of the low-pass filtered image. In this part, a registration method, called iterative closest set (ICS), is introduced, which registers the current frame with the previous frame using the low-frequency structural features. As a result, both the translation and the rotation of motion trajectories are provided. After frame-wise registration, a series of successive frames can be matched.

The adjacent matched frames are then used to implement an inter-frame temporal filter by adaptively controlling the bandwidth. To minimize buffer size and latency, a temporal low-order IIR filter is employed to coarsely reduce noise.

To further reduce weak noise, an inner-frame spatial filter is subsequently applied.

Finally, according to the bounded noisy model, a non-linear distribution compensation function is obtained to correct the grayscale of the filtered image and provide a well-denoised result.

The flowchart of the proposed method is shown in Figure 2. For illustrative purposes, a motion sequence of the Lena images was generated using simulated and regular translation and rotation, and the intermediate result is shown for each step. Video compression follows denoising to reduce transmission data rate. Furthermore, the denoised and compressed data is encrypted in a fog-computing scheme for security and privacy.

The remainder of this paper is organized as follows. In Section 2, related work on fog computing and image denoising is reviewed. In Section 3, the basic observation model is introduced. In Section 4, the robust structural similarity registration strategy for estimating the motion trajectory under heavy noise is presented. A fast approach for structure similarity matching aimed at translations is introduced, as well as an algorithm using
FIGURE 2 The flowchart of the proposed algorithm. A, The simulated video sequences with random motion and AWGN ($\sigma > 100$) based on manual motion of Lena image; B, The B-IIR filtered result; C, The structure registration; D, The coarse inter-frame denoised result; E, The fine inner-frame denoised result; F, The distribution compensation result

the iterative closest set technique for estimating rotations. In Sections 5 and 6, the coarse-to-fine denoising and grayscale correction methods are introduced. In Section 7, the proposed scheme for security and privacy preservation in data transmission and storage is introduced. In Section 8, the details of the simulation and the implementation are provided, as well as experimental results and discussion. Section 9 concludes the paper.

2 RELATED WORK

Shared computing platforms have become a major computing paradigm; however, security and privacy are more compromised than in isolated and independent computing systems, and the problem in cloud, edge, and fog computing has been widely addressed.9-18 Recently, the combination of Internet of Things (IoT), Cloud/Edge/Fog computing, and Big Data has been recognized as fundamental. In IoT, Visual Internet of Things (ViIoT) has been widely used in video security surveillance for smart city and industrial automation.5 In these fields, image processing technology is important for improving network efficiency. In the work of Memos et al,19 an efficient algorithm was proposed for a media-based surveillance system in smart cities using high-efficiency video encoding. The algorithm can significantly reduce the memory requirement of sensor nodes in a network environment. In the work of Hasan et al,20 an IoT-based algorithm for color frame transmission and generation was proposed for reducing power consumption, which is required by the energy constraints of power-sensitive IoT systems. The new generation ViIoT requires geo-distribution, heterogeneous support, low latency, and low bandwidth. Fog computing was proposed as a platform to meet these requirements. There are several real-time applications based on fog computing. In the work of Gia et al,21 a healthcare solution was provided based on fog computing. It achieved high bandwidth efficiency and low-latency real-time response at network edges. In the work of Teeapittayanon et al,22 distributed deep neural networks (DDNNs) over distributed computing hierarchies were proposed. They consisted of cloud, fog, and end devices. To the best of the authors’ knowledge, there is limited research on video denoising based on fog computing.

Security and privacy issues are involved in IoT, particularly in fog computing. In other works,23-25 the challenges of fog computing were pointed out, namely trust, authentication, secure communications, user privacy, and malicious attacks. Hu et al provided a face identification and resolution framework using fog computing with a security and privacy preservation scheme.26 They proposed an authentication and session key agreement scheme, a data encryption scheme, and a data integrity checking scheme for resolving the issues of confidentiality and availability. Zheng et al proposed privacy-preserving image denoising from external cloud databases.23 They designed a cloud hosting encrypted databases for secure query-based image denoising services and achieved satisfactory denoising performance. Fog-implemented denoising allows the encryption of privacy data at network edges and secure data transmission to cloud servers.

Several related image denoising methods have been proposed in the last decades. They are based on the assumption that noise follows the additive white Gaussian (AWG) distribution. Traditional denoising methods have been classified into “local” or “non-local” and “point-wise” or “multi-point”.27,28 Local point-wise methods employ a neighborhood filter in the spatial domain or the transform domain such as bilateral filter,29 anisotropic filter,30 wavelet-based filter,31 guided filter,32 meanshift filter,33 or variation formulations.34 They cannot retain sharp edges. The multi-point method is used to determine more relations between a pixel and its neighborhood, whereby all points in the neighborhood are estimated by successively blocking, estimating, and aggregating. The typical local multi-point algorithms are adaptive-PCA,35 K-SVD,36 MS-K-SVD,37 CT-Tri-SuS-MMAE,38 and MLPD,39 in which the structural information is used to improve denoising performance. For further improvement, it was found that the structural similarity between the current patch and its neighboring patches can provide more relevant information for distinguishing noise and signal. Buades et al proposed a non-local mean40 that denoises image point-wisely using the structural similarity of patches. NLM was the first non-local point-wise approach. Dabov et al41,42 proposed block-matching and 3D filtering (BM3D), which was a milestone in image denoising.

There are several powerful non-local video denoising methods. For instance, Dabov et al extended BM3D to video denoising and called it VBM3D.43 It groups wide non-local similar patches in temporal sequences. However, VBM3D cannot distinguish non-local spatial and temporal
Observation Model

The end node of the surveillance camera generates the video data accompanied with noise, which is primarily due to the sensor. In this study, the ideal noisy model is considered, which is defined as

\[ y = x + n, \]  

where \( x \) is a noise-free image, \( n \) is additive white Gaussian (AWG) noise, which is assumed normally distributed with standard deviation \( \sigma \) and mean zero, and \( y \) is an unbounded noisy image contaminated by \( n \). The expectation of \( y \) can be written as

\[ E(y) = E(x) + E(n). \]  

Ideally, \( E(n) \) is considered zero, and (3) is then given as follows:

\[ E(y) = E(x). \]  

Thus, the noise can be reduced by calculating the mean of these coincident regions. However, the quantized data of the digital image is bounded owing to the 8-bit quantization that is imposed by the popular digital transmitting protocol. \( y \) is considered limited within the range \([0, 255]\) and normalized in the range \([0, 1]\). The expectation of \( n \) cannot be considered zero. To extend (1), the bounded observation model \( y \) should be represented as follows:

\[ \hat{y} = \begin{cases} 
0, & x + n < 0 \\
(x + n, & 0 \leq x + n \leq 1 \\
1, & x + n > 1. 
\end{cases} \]  

Thus, the expectation of \( n \) is a function of the expectation of \( x \), namely \( E(n, E(x)) \)

\[ E(n, E(x)) = \left( \int_{-\infty}^{0} A(0 - E(x))e^{-\frac{a-n\sigma^2}{\sigma^2}} dt + \int_{0}^{1} A(t - E(x))e^{-\frac{t-n\sigma^2}{\sigma^2}} dt + \int_{1}^{\infty} A(1 - E(x))e^{-\frac{1-n\sigma^2}{\sigma^2}} dt \right) / \int_{-\infty}^{\infty} A e^{-\frac{a-n\sigma^2}{\sigma^2}} dt. \]  

where,

\[ A = \frac{1}{\sqrt{2\pi}e^{\frac{1}{2}}}, \quad B = 2(\sigma/255)^2. \]  

Then, the expectation of \( \hat{y} \) can be expressed as

\[ E(y) = E(x) + E(n, E(x)). \]
4 | REGISTRATION STRATEGY

The aim of registration is to provide robust and fast motion estimation under heavy noise contamination. Low-frequency structural similarity can be used to estimate the motion trajectory and group coincident points in temporal sequences. A fast low-pass filter, called B-IIR filter, will be first introduced. Subsequently, the structural feature set extracted from adjacent filtered frames will be considered. Finally, based on low-frequency structural similarity, translation-only (TO) registration and iterative closest set (ICS) registration will be introduced.

4.1 | Fast B-IIR filter

In a noise-free natural image, the great majority of energy and intrinsic structural information are concentrated in the low-frequency components, even if the image is contaminated by heavy noise. This implies that the low-frequency structural characteristics (e.g., brightness, shape, and gradient) can be used to measure similarity. Thus, a low-pass filter can be applied on each video sequence to obtain the low-frequency components. There are several methods for implementing a 2D image low-pass filter such as the fast Fourier transform (FFT) and the convolution. However, the computation complexity of these methods is \(O(N^*M^*(\log N + \log M)^*2)\) and \(O(N^*M^*n^*n)\), respectively, where \(N, M\) are the dimensions of the image, and \(n\) is the size of the convolution mask, which is inversely proportional to the bandwidth, resulting in unacceptable computational workload.

In this study, a novel fast low-order B-IIR filter is proposed as a zero phase shift filter. The typical IIR is a half-band filter that exhibits non-zero phase distortion, which causes hemi-ghost artifacts. To overcome this problem, two general IIR filters with double reverse time-domain are combined. Thus, the phase shift will be offset to zero. This filter is called B-directional IIR filter (B-IIR), and its diagram is shown in Figure 3. The advantage of this approach is that its computation complexity is as low as \(O(N^*M^*t^*4)\) and is irrelevant to the bandwidth. The parameter \(t\) is the order of the filter.

![Block diagram of zero-phase B-IIR filter](image)

**FIGURE 3**    Block diagram of zero-phase B-IIR filter

![Four loops of B-IIR low-pass filter](image)

**FIGURE 4**    Four loops of B-IIR low-pass filter. A, The original image; B, The left-to-right loop; C, The right-to-left loop; D, The top-to-bottom loop; E, The bottom-to-top loop and the final result
In Figure 3, $H(z)$ is a $z$-domain IIR expression, namely

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{i=0}^{t} b_i z^{-i}}{\sum_{i=0}^{t} a_i z^{-i}}. \tag{8}$$

This equation can be expressed in the discrete time-domain as follows:

$$y(n) = \sum_{i=0}^{t} b_i x(n - i) - \sum_{i=0}^{t} a_i y(n - i). \tag{9}$$

Equation (9) is a general equation for any order. As previously mentioned, the low-frequency component contains robust structural similarity, which can improve the registration accuracy in the next step. To eliminate the negative impact of registration caused by the noise of the high-frequency component, the choice of the low-pass filter depends on the bandwidth rather than the ripple or the decay. Considering computational flexibility and efficiency, the simplest first-order filter with zero-order hold (ZOH) discretization is chosen as a low-pass filter.

Thus, letting $t = 1$ for a first-order filter and using the ZOH discretization method, the equation can be simplified as

$$y(n) = bx(n) + (1 - b)y(n - 1). \tag{10}$$

where $b \in [0, 1]$. This adjusts the bandwidth of the low pass filter by a decline rate of $-20$ dB per decade. A smaller value of $b$ results in narrower bandwidth. That is, for a larger standard deviation of the heavy Gaussian noise, a smaller $b$ is required for suppressing high-frequency noise. Experiments show that $b \in [0.4, 0.6]$ is a good compromise for a noise level $\sigma = 100$.

To apply a low-pass filter on a 2D image, vertical and horizontal B-IIR filtering are implemented separately. To decompose the process, the input image should be processed by the $H(z)$ transformation in four loops, namely left-to-right, right-to-left, top-to-bottom, and bottom-to-top loops. The effects of these intermediate loops are shown in Figure 4.

### 4.2 Low-frequency structural feature set

Structural features can be extracted from the low-frequency components of the image. These features (e.g., shapes, location of edges, and luminance) can be used to measure the similarity of adjacent images. As the images have been pre-processed by the low-pass filter, the aim is to extract sparse structural features as sets and to fast-match them for registering adjacent frames.

After applying the low-pass filter, the perspective is changed, and the filtered image is observed from 3D space rather than 2D plane, where the grayscale becomes the third dimension variable. Thus, the filtered image could be regarded as a wavelike surface concentrating intrinsic low-frequency structural characteristics. These surfaces can be used to robustly register two adjacent frames.

As shown in Figure 5, adjacent previous and current frames are surface-to-surface registered by minimizing the mean squared error (MSE) of the distance.

![Surface-to-surface registration of adjacent low-pass filtered frames](image-url)
A typical surface-to-surface registration can be implemented by global searching. When the searching window is s’s and the image block size is M*N, the time complexity is O(M*N’s’s). This does not meet the requirement of fog computing; thus, a more efficient method is required.

According to entropy theory, the information of salient structural characteristics is located where the gradient values drastically change. Therefore, arbitrary gradient-based edge detection algorithms could be used to locate maximal gradient points (e.g., Canny, zero-cross) in filtered images (shown in Figure 2C). These points are distributed on the steep of the wavelike surface. A pixel is regarded as a vector with (x, y) coordinate position and gray value. The position of each edge point (x, y) and the corresponding grayscale g(x, y) in the filtered image are collected to form a set V of vectors v = (x, y, g(x, y)). Considering that dense edge vectors are redundant and affect calculation efficiency, a conditional down-sampling is arranged. The sampled set is denoted by P = {p1, p2, ..., pn} ⊆ V, where n is the number of vectors, and is called structural feature set.

The next step is to estimate the motion trajectory by registering the set P (from the previous frame) with a reference surface T (from the current frame), which contains a complete vector set of all pixels in the current frame. The size of T is larger than the size of P. The motion trajectory estimation can be reduced to point-to-surface registration.

### 4.3 Translation-only registration

For each position component (x, y) of P, its searching window is defined as an s’s neighborhood. Within this searching window, the vectors (X ik, Y ik, g X ik, Y ik) that correspond to each element in T are searched and arranged into a new set denoted by X k, where i = 1, 2, ..., n, k = 1, 2, ..., s’s, and X k ⊆ T. Then, the minimum of dk is achieved by calculating the MSE of the grayscale components between P and each X k, that is,

\[
\text{Min } d_k = \sum_i^n (g_{pi}(x_i, y_i) - g_{X_i k}(x_i, y_i))^2, \text{ } k = 1, \ldots, s^2.
\]

Assuming that d k is the minimum and k = K, the translation between two frames is denoted by t, i.e.,

\[
t = (x_i, y_i) - (x_{ik}, y_{ik}), \text{ } i = 1, 2, \ldots, n,
\]

and the MSE of similarity is denoted by ems = d k.

The translation t is used to align the original frames for the ensuing inter-frame filtering. The complexity of translation-only (TO) registration is reduced to O(n * s * s), where n is considerably smaller than N * M. Thus, this method is simple and efficient.

As TO registration is only for translation scenes and is constrained by its searching window size, to handle motion trajectories with both rotation and translation, an iterative closest set (ICS) registration is to be introduced.

### 4.4 Iterative closest set registration

The concept of registering the current frame with the previous frame using the feature set of the wavelike surface derives from 3D shape registration. In this section, a method is proposed for determining the best rotation and translation in a 2D plane registration. The aim is to minimize the distance between the target set T (from the previous frame) and the reference set T (from the current frame).

The distance of vectors v = (x, y, g(x, y), (g(x, y)} are collected to form a set V of vectors v = (x, y, g(x, y)). The motion trajectory and the image block size is s*s. The MSE of similarity is denoted by ems = d k.

The process of searching for the closest set from the reference surface is shown in Figure 6.

Subsequently, the unit quaternions are used to determine the best rotation and translation between P and X. The best translation vector t is calculated by the vector difference between the two mass center vectors \( \mu_p \) and \( \mu_x \), that is,

\[
\mu_p = \frac{1}{n} \sum_{i=1}^{n} p_i \text{ and } \mu_x = \frac{1}{n} \sum_{i=1}^{n} x_i,
\]

\[
t = \mu_x - \mu_p.
\]

A matrix of sums of products is now introduced to solve the least-squares problem for rotations as follows:

\[
M = \sum_{i=1}^{n} p_i \cdot x_i' = \begin{bmatrix} S_{xx} & S_{xy} & S_{xz} \\ S_{yx} & S_{yy} & S_{yz} \\ S_{zx} & S_{zy} & S_{zz} \end{bmatrix},
\]

where the individual elements \( S_{vw} \) are

\[
S_{vw} = \sum_{i}^{n} v_{i,v} \cdot w_{i,w}.
\]
Subsequently, we use the elements in (16), which are related to the \((x, y)\) position components in the matrix \(M\), to construct a symmetric 4\(\times\)4 matrix \(Q\) as follows:

\[
Q = \begin{bmatrix}
S_{xx} + S_{yy} & 0 & 0 & S_{xy} - S_{yx} \\
0 & S_{xx} - S_{yy} & S_{xy} + S_{yx} & 0 \\
0 & S_{xy} + S_{yx} & S_{yy} - S_{xx} & 0 \\
S_{xy} - S_{yx} & 0 & 0 & -S_{xx} - S_{yy}
\end{bmatrix}.
\] (18)

Singular value decomposition (SVD) is applied to the matrix \(Q\) to obtain its eigenvalues and eigenvectors. The eigenvector corresponding to the maximum eigenvalue is the optimal unit rotation quaternion \([q_0, q_1, q_2, q_3]\) for registering \(P\) and \(X\).

According to the unit rotation quaternion, the optimal rotation matrix \(R\) is the following:

\[
R = \begin{bmatrix}
q_0^2 + q_1^2 - q_2^2 - q_3^2 & 2(q_1q_2 + q_0q_3) & 2(q_1q_3 - q_0q_2) & 0 \\
2(q_1q_2 - q_0q_3) & q_0^2 - q_1^2 + q_2^2 - q_3^2 & 2(q_2q_3 + q_0q_1) & 0 \\
2(q_1q_3 + q_0q_2) & 2(q_2q_3 - q_0q_1) & q_0^2 - q_1^2 - q_2^2 + q_3^2 & 0 \\
0 & 0 & 0 & 1
\end{bmatrix}.
\] (19)

After the best translation \(t\) and the obtained optimal rotation \(R\) have been performed on the position components \((x, y)\) of \(P\), the new set \(P^{(i)}\) is updated; however, its grayscale component \(g\) is retained. The MSE between the grayscale components of \(X\) and \(P\) is denoted by

\[
e_{\text{ms}} = \frac{1}{n} \sum_{i=1}^{n} (g_i - p_i)^2.
\] (20)

Repeating the above process, the updated set will continuously approximate the target set through the iterations, and a better registration between them will thus be obtained. To distinguish the variables \(t, R, P, X, e_{\text{ms}}\) in different iterations, the superscripted symbols \(P^{(0)}, R^{(0)}, P^{(1)}, X^{(0)}, e_{\text{ms}}^{(0)}\), will be used, where \(r\) denotes the iteration.

Rotation \(R^{(i)}\) and translation \(t^{(i)}\) are calculated to transform the set \(P^{(i)}\) to the updated \(P^{(i+1)}\) in (21); then, the new closest set \(X^{(v+i)}\) relative to \(P^{(v+i)}\) is obtained in (22)

\[
(X^{(v+i)}) = (x_{p^{(v)}, y_{p^{(v)}}}) - (x_{p^{(v)}}, y_{p^{(v)}}) \cdot R^T + (\mu_x, \mu_y, t_x, t_y)
\] (21)

\[
X^{(v+i)} = \varphi(P^{(v+i)}, T).
\] (22)

As iterations increase, the difference of \(e_{\text{ms}}^{(v)}\) and \(e_{\text{ms}}^{(v+1)}\) decreases and converge to the minimal value \(\epsilon\), i.e.,

\[
\lim_{v \to \infty} \text{abs} (e_{\text{ms}}^{(v)} - e_{\text{ms}}^{(v+1)}) = \epsilon.
\] (23)

A terminal threshold \(\xi\) is set as an acceptable rate of convergence to stop the iteration of registering. When \(\text{abs}(e_{\text{ms}}^{(v)} - e_{\text{ms}}^{(v+1)})/(e_{\text{ms}}^{(v+1)}) < \xi\), the iteration should stop.

After \(L\) iterations, the final transformation of translation \(t\) and rotation \(R\) between the previous frame feature set \(P\) and the current frame reference surface \(T\) is as follows:

\[
t = \mu_{p^{(L)}} - \mu_{p^{(0)}}
\] (24)

\[
R = I \cdot R^{(0)} \cdot R^{(1)} \cdot \cdots \cdot R^{(L)},
\] (25)

where \(I\) is a unit matrix.
Finally, the translation and rotation are applied to register the previous frame with the current frame with low computational workload. As the adjacent video frames are successively registered frame-by-frame, a motion trajectory of the camera or the scene can be obtained by these translational vectors and rotational matrices. $e_{ms}$ can be regarded as a global registration error that measures whether the structure is successfully registered and will be used as an adaptive parameter in the next section.

5 | COARSE-TO-FINE DENOISING

Denoising is implemented in two steps. The first step is coarse denoising, which employs the inter-frame temporal filter. The second step is fine denoising, which involves the inner-frame spatial filter.

5.1 | Inter-frame temporal denoising

Under the assumption that the structure of adjacent frames has been well registered, inter-frame denoising is used to serialize the corresponding points in the same real scene and filter the noise in the temporal dimension by a fast first-order IIR.

The current input noisy frame is denoted by $F_{in}$, and the current output, which is a denoised frame, is denoted by $F_{out}$. The previous output is denoted by $F'_{out}$. The first-order IIR filter for inter-frame denoising is the following:

$$F_{out}(x, y) = w(\sigma_{dst}, e_{ms}, d_{f}(x, y)) \cdot F_{in}(x, y) + (1 - w(\sigma_{dst}, e_{ms}, d_{f}(x, y))) \cdot F'_{out}(x', y'),$$

where $(x', y')$ is a transformation of $(x, y)$ according to the registered result. The filter coefficient $w$ is a piecewise function related to the destination standard deviation of noise $\sigma_{dst}$, the global registration error $e_{ms}$, and the gray distance of the local pixel difference $d_{f}(x, y)$ between two registered frames. $\sigma_{dst}$ is a key parameter that suppresses noise. $e_{ms}$ and $d_{f}(x, y)$ are auxiliary parameters for avoiding ghost artifacts if local motion occurs. The value of the function $w(\sigma_{dst}, e_{ms}, d_{f}(x, y))$ should be within (0,1).

As the IIR filter is a closed-form function with infinite inputs, only one frame image can be cached as intermediate to realize a first-order filter. Fog computing can save hardware resources and computational time. As shown in Figure 2D, the temporal low-pass IIR filter is effective to suppress the noise from heavy to weak obviously.

5.2 | Inner-frame spatial denoising

After inter-frame denoising, the remaining noise should be further removed. To this end, a novel patch-wise local structural similarity filter (SSF) is proposed. The approach is still based on the assumption that similarity patches can be obtained by structural characteristics in the low-frequency band, and inner-frame low-pass filtering is used. Neighboring similarity patches are grouped together according to the reference patch, and then, their patch-group mean values are assigned to them. Finally, all patches are combined according to their similarity weights to generate the filtered image. The flowchart of these steps is shown in Figure 7. The steps of the SSF algorithm for inner-frame denoising are as follows.

---

**FIGURE 7** Flowchart of the structural similarity filter for inner-frame spatial denoising
SSF denoising algorithm

Step 1. Obtain a low-frequency structure estimate using the Wiener B-IIR filter.

1.1 Calculate the Wiener filter to approximate the B-IIR bandwidth:
   \[ \tau_{\text{LP}} = \text{Apprx} \left( T_{2D}(y)^2 + \sigma^2 \right). \]

1.2 Generate the low-pass filtered image:
   \[ y_{\text{LP}} = \text{B-IIR}(y, \tau_{\text{LP}}). \]

Step 2. Group and filter neighboring similarity patches.

2.1 Group the similarity patches \( Y_{\text{LP}} \) from filtered \( y_{\text{LP}} \) by structural similarity block-matching (SSBM), which is based on the distance of the normalized MSE. The similarity threshold is a fixed parameter \( \lambda \):
   \[ S_x = \text{SSBM} \left( Y_{\text{LP}}, \lambda \right). \]

2.2 Calculate the mean value \( \hat{Y}_s \) of matched patches from original \( y \) according to \( S_x \):
   \[ \hat{Y}_s = \text{mean} (Y_{s}). \]

Step 3. Aggregate \( \hat{Y}_s \) using weighted averaging with weight of similarity \( w_s \) of matched \( S_x \):
   \[ y_{\text{final}} = \sum \left( \hat{Y}_s * w_s \right) / \sum w_s. \]

Notation:

1. \( y \) denotes an input noisy image. \( \sigma \) is the transfer parameter of destination deviation resulting from inter-frame denoising. \( T_{2D}() \) is the transformation function in the frequency-domain.

2. \( \text{Apprx}() \) is an approximation transform function for the Wiener IIR filter. It is a standard approach in signal processing; thus, it will not be discussed here.

3. The bandwidth parameter \( \tau_{\text{LP}} \) of \( \text{Apprx}() \) output is a relatively stable variable for general video. Therefore, it is not necessary to be continually updated frame-by-frame for global B-IIR. Usually, it is constant for short video.

4. \( y_{\text{LP}} \) denotes a LP filtered image with structural features left. \( Y_{\text{LP}} \) is a filtered patch unit contained in \( y_{\text{LP}} \).

5. \( S_x \) denotes a set of similar coordinates.

6 | DISTRIBUTION COMPENSATION

After the coarse-to-fine denoising, a denoised video is obtained; however, its grayscale contrast is distorted owing to the intrinsic loss of the observation model mentioned in Section 3. The known denoised results are treated as the expectation of observation \( E(y) \), and the compensated result is estimated based on the true expectation \( E(x) \). The observation model is considered bounded; moreover, it is assumed that the expectation of model \( y \) has been derived as in (5). According to the statistical model, the distribution of the observation's expectation can be compensated.

As the standard deviation \( \sigma \) has been estimated, the curves mapping \( E(y) \sim E(x) \), shown in Figure 8, can be obtained by (5). In this Figure, the values have been normalized to the range \([0,1]\). A mapping curve is used to compensate the grayscale, and the final result is obtained.

![Figure 8](image-url)
7 | SCHEME FOR SECURITY AND PRIVACY

In this section, the proposed fog-computing-based video denoising is implemented for security and privacy preservation. There are several security and privacy issues involved in video surveillance on IoT such as trust, authentication, secure communications, user privacy, and malicious attacks.

Fog computing is combined with video denoising to provide an efficient service in the proposed scheme. Video denoising is distributively performed at front-end fog nodes, thus reducing computational workload and memory cost. It is possible to compress the denoising result at low data rate for low-latency communication. The encryption can be performed after the denoising and compression processes at front-end network devices. Whether the denoising procedure should be performed in the current environment is determined either by evaluating the video quality or by manual commands. On one hand, certain objective indices such as peak-signal-noise-ratio (PSNR) and MSE can be adopted to evaluate the video quality. Denoising is automatically switched on once any of the above indices exceeds a predetermined configurable threshold. On the other hand, the video quality can be judged visually by the users according to its visual appearance. A scheme consisting of sensor nodes, fog nodes, cloud nodes, and user nodes is proposed. The sensor nodes are the geo-distributed cameras in the surveillance system. Fog nodes consist of embedded system boards and industrial personal computers, which provide optional denoising and the required compression and encryption. Cloud nodes are composed of data center servers, which provide log management and remote database service. User nodes comprise personal devices (e.g., computers, pads, or smart phones), which can retrieve authorized data from the cloud and perform local decryption.

Under this scheme, the surveillance video, which is privacy-sensitive, grabbed by a fog node, is encrypted at a network edge. The encrypted data is securely transmitted between the fog and the cloud, among servers in the cloud, and between the cloud and the user, thus avoiding the insecure and redundant process of data decryption and decompression for post-denoising in the cloud, where malicious attacks may potentially occur. Compared with cloud-based image denoising, the proposed scheme provides confidentiality in data transmission and storage.

8 | EXPERIMENTS

In this section, a straightforward experimental setting is presented. The parameter setting reflects certain requirements in the fog computing environment concerning low latency, low computation cost, and high quality. The proposed algorithm was compared with the state-of-the-art methods. The experiment was conducted on a 3.0 GHz Intel Core i5 CPU with 8 GB 1600 MHz DDR3 RAM, which simulates the typical hardware specification of an industrial personal computer at a network edge.

8.1 | Robust estimation of motion trajectory

The estimation accuracy of the motion trajectory affects the inter-frame denoising performance and the final result; hence, the robustness of the estimated trajectory under heavy noise is compared with that in noise-free condition.

The experimental parameters were set as follows. In the B-IIR filter, $b$ is proportional to the standard deviation of the destination divided by the standard deviation of the source. A zero-cross edge detector was chosen with hard-threshold 0.015. Then, 5% of the edge points were resampled to construct a set for registration. The stopping criterion is any of the following: the number of iterations exceeds 10, the registration MSE $e_{\text{RM}}$ is less than 0.0001, or the convergence rate smaller than the terminal threshold $\xi = 0.0005$.

Figure 9 shows the robustness of the motion trajectory of the video Foreman under heavy noise and in noise-free condition in vertical translation, horizontal translation, and rotation. Dotted lines correspond to video sequences corrupted by white Gaussian noise with standard deviation $\sigma = 150$, whereas solid lines correspond to noise-free estimation. The root mean squared errors ($e_{\text{RM}}$) for the two conditions are shown in Figure 9. The $e_{\text{RM}}$ for the translation is smaller than 1 pixel, and the $e_{\text{RM}}$ for the rotation is smaller than 0.2. It shows that the estimated trajectories of the camera motion are robust under heavy noise.

8.2 | Denoising performance

In this section, the performance of the proposed algorithm is compared to that other state-of-the-art denoising methods using benchmark videos (i.e., Miss. America, Foreman, and Salesman) and custom videos (i.e., Desk, Hall, Office). The results are shown in Table 1 and Figure 11.

Adjacent-frame registration is considered with and without rotation. Tests for TO or ICS registration will be presented later. Patch size was set to $N_p * N_p$ ($N_p = 9$), and the search window for similar patches was restricted within a $N_s * N_s$ ($N_s = 13$) neighborhood centered at the reference patch. Considering the trade-off between running time and performance, an interval between two adjacent reference patches was set with step $N_r$ ($N_r = 9$).

The average PSNR was adopted as a principal evaluation index of denoising quality, which is defined by

$$\text{PSNR}^{(\tau)} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}^{(\tau)}} \right). \quad (27)$$

where $\text{MSE}^{(\tau)}$ is the MSE of the clean reference $x$ and the denoised result $y$ in the $r^{th}$ frame, ie,

$$\text{MSE}^{(\tau)} = \frac{1}{NM} \sum_{i}^{NM} (y_i - x_i)^2. \quad (28)$$
FIGURE 9  Frame-by-frame estimated trajectory on video sequences Foreman under heavy noise ($\sigma = 150$, dot line) and noise-free (solid line). A, vertical translation trajectory (unit:pixel, $e_{RMS} = 0.88$); B, horizontal translation trajectory (unit:pixel, $e_{RMS} = 0.74$); C, rotation trajectory (unit:degree, $e_{RMS} = 0.19$)

TABLE 1  Denoising performances of our approach compared with VBM3D and VBM4D

<table>
<thead>
<tr>
<th>Video Name</th>
<th>Approach</th>
<th>Consumed RAM (byte per frame)</th>
<th>Standard deviation</th>
<th>Average PSNR (dB) /Latency (second per frame)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>$\sigma = 100$</td>
<td>$\sigma = 150$</td>
</tr>
<tr>
<td>Miss A</td>
<td>VBM3D</td>
<td>1.86 M</td>
<td>26.81/0.217</td>
<td>24.06/0.21</td>
</tr>
<tr>
<td>(360<em>288</em>150)</td>
<td>VBM4D (ideal noise)</td>
<td>14 M</td>
<td>28.27/2.08</td>
<td>25.11/2.03</td>
</tr>
<tr>
<td></td>
<td>Our approach (TO)</td>
<td>0.49 M</td>
<td>30.37/0.15</td>
<td>28.73/0.16</td>
</tr>
<tr>
<td></td>
<td>Our approach (ICS)</td>
<td>0.50 M</td>
<td>30.47/0.18</td>
<td>28.95/0.19</td>
</tr>
<tr>
<td>Foreman</td>
<td>VBM3D</td>
<td>1.70 M</td>
<td>23.74/0.21</td>
<td>21.54/0.21</td>
</tr>
<tr>
<td>(352<em>288</em>300)</td>
<td>VBM4D (ideal noise)</td>
<td>12.3 M</td>
<td>25.99/1.78</td>
<td>23.47/2.13</td>
</tr>
<tr>
<td></td>
<td>Our approach (TO)</td>
<td>0.45 M</td>
<td>26.21/0.13</td>
<td>24.42/0.13</td>
</tr>
<tr>
<td></td>
<td>Our approach (ICS)</td>
<td>0.46 M</td>
<td>26.05/0.19</td>
<td>24.51/0.21</td>
</tr>
<tr>
<td>Salesman</td>
<td>VBM3D</td>
<td>2.10 M</td>
<td>23.07/0.22</td>
<td>21.32/0.21</td>
</tr>
<tr>
<td>(352<em>288</em>50)</td>
<td>VBM4D (ideal noise)</td>
<td>12.8 M</td>
<td>25.21/1.77</td>
<td>22.66/1.80</td>
</tr>
<tr>
<td></td>
<td>Our approach (TO)</td>
<td>0.62 M</td>
<td>23.89/0.12</td>
<td>22.68/0.14</td>
</tr>
<tr>
<td></td>
<td>Our approach (ICS)</td>
<td>0.64 M</td>
<td>23.77/0.15</td>
<td>22.55/0.18</td>
</tr>
<tr>
<td>Desk</td>
<td>VBM3D</td>
<td>4.92 M</td>
<td>23.02/0.63</td>
<td>20.53/0.61</td>
</tr>
<tr>
<td>(640<em>480</em>406)</td>
<td>VBM4D (ideal noise)</td>
<td>36 M</td>
<td>25.28/6.22</td>
<td>22.52/6.27</td>
</tr>
<tr>
<td></td>
<td>Our approach (TO)</td>
<td>1.32 M</td>
<td>25.85/0.35</td>
<td>23.92/0.37</td>
</tr>
<tr>
<td></td>
<td>Our approach (ICS)</td>
<td>1.33 M</td>
<td>25.30/0.62</td>
<td>23.74/0.51</td>
</tr>
<tr>
<td>Hall</td>
<td>VBM3D</td>
<td>5.05 M</td>
<td>23.23/0.81</td>
<td>21.4/0.80</td>
</tr>
<tr>
<td>(640<em>480</em>277)</td>
<td>VBM4D (ideal noise)</td>
<td>36 M</td>
<td>24.6/6.94</td>
<td>22.55/6.25</td>
</tr>
<tr>
<td></td>
<td>Our approach (TO)</td>
<td>1.38 M</td>
<td>24.73/0.33</td>
<td>23.64/0.38</td>
</tr>
<tr>
<td></td>
<td>Our approach (ICS)</td>
<td>1.40 M</td>
<td>24.87/0.57</td>
<td>23.81/0.56</td>
</tr>
</tbody>
</table>
where $NM$ represents the resolution of a single frame. The average PSNR of a video is defined by

$$\text{PSNR}_{\text{avg}} = \frac{1}{L} \sum_{\tau} \text{PSNR}^{(\tau)}$$

(29)

where $L$ is the total number of frames.

The proposed denoising algorithm was compared with VBM3D and VBM4D. Owing to its limitations, VBM4D can be applied only to the ideal unbounded noisy model (1). If it is applied to the bounded noisy model (4), significant distortion occurs. The restraint of VBM4D under the ideal unbounded condition was dropped so that comparison may be made with the proposed approach under a worse bounded condition. In spite of that, the proposed approach significantly outperformed both VBM3D and VBM4D in terms of average PSNR under heavy noise conditions ($\sigma > 100$).

**FIGURE 10** Frame-by-frame PSNR (dB) output of the sequences Foreman corrupted by white Gaussian noise with standard deviation $\sigma = 150$. Comparison with VBM3D (dot line), VBM4D (ideal unbounded, dash line) and our approach (ICS, solid line).

**FIGURE 11** Comparison of denoising performance of V-BM4D and Our Proposed on video Miss. America, Foreman, Desk and Hall with standard deviation $\sigma = 150$. A: Original frame; B: Noisy frame; C: VBM3D; D: VBM4D (ideal unbounded); E: Our approach.
FIGURE 12  Real-world video Office denoised by VBM3D, VBM4D and our approach from left to right. The top row shows frames with original size 640*480, and the bottom row shows the partial zoom-in of the top row

A comparison of TO and ICS rotation registration is shown in Table 1. It can be seen that the ICS registration step is better suited for scenes with rotation or diverse motions such as Miss. A, Foreman, and Hall. TO registration can handle scenes such as Salesman and Desk, which primarily contain translation motions.

A special experiment for observing frame-by-frame results is shown in Figure 10. Each approach was tested on the video Foreman, in which the objects move and scenes vary simultaneously. Less irregular local motions or frequent scene changes in the video data resulted in the best denoising performance that the proposed approach can achieve. In this experiment, the evaluation index of PSNR was improved by 1–2 dB in average compared with VBM4D as shown in Figure 10.

These approaches were compared both objectively and subjectively. In Figure 11, known artificial noise was added to the original videos, and the denoised results obtained by different methods were compared. The visual appearance of the proposed approach is more aesthetically pleasing. There is less distortion around low contrast features so that more object details can be clearly observed and distinguished. The performance of VBM4D is acceptable; however, it is presented in an ideal unbounded model. In Figure 12, the real world noisy video Office captured in enhancement mode with small aperture was denoised by each approach. This video follows the strict bounded noisy model owing to quantization. The result of the proposed approach is clearly visible with sharp edges and details. By contrast, VBM4D is not satisfactory owing to boundedness, whereas VBM3D presents mosaic artifacts in the details.

8.3 | Computational cost

Latency and memory requirements are key indicators considered in fog computing and are intuitively measured by the computational cost of each approach. The comparison of latency was performed on the same hardware platform using a single thread. As shown in Table 1, the latency of the proposed algorithm outperforms that of VBM3D; moreover, the proposed algorithm was 10 times as fast as VBM4D. The processing speed of the proposed algorithm was approximately 5–6 frames per second with a frame resolution of 360*288. In terms of random access memory (RAM) requirements, the proposed approach used the least RAM. Hence, hardware cost can be drastically reduced. The algorithm occupied approximately 0.5 Mbyte RAM per frame of resolution 360*288.

8.4 | Compression rate and transmitting bandwidth

The compression rate of a denoising method determines the bandwidth capability of fog computing when the surveillance system transmits the video stream from edges to servers. In this experiment, a video compression module was arranged after the video denoising module in cascade. Different combinations of video compression methods and input types were implemented. Two compression methods were tested, namely Motion JPEG64 and MPEG-4,65 on the standard video Miss. A. The data rate of the combination is shown in Table 2. It can be seen that the denoising result by the proposed method can be compressed at lower data rate compared with those by VBM3D and VBM4D. In MPEG-4 mode, the proposed denoising
algorithm, which is deployed in fog computing manner, can save up to 70% bandwidth at network edges. It can dramatically reduce communication latency and improve quality.

9 | CONCLUSIONS

A novel video denoising algorithm based on fog computing was proposed for preserving security and privacy. This algorithm benefits from the denoising performance and resource constraints at network edges and realizes low-latency and low data-rate transmission. Owing to the security and privacy preservation of the proposed scheme, privacy-sensitive videos can be obtained, denoised, compressed, and encrypted at network edges. It can provide confidentiality of data transmission and storage to VIoT surveillance systems.

Experimental results showed that the proposed approach outperforms VBM3D and VBM4D in terms of both objective performance and visual appearance under extremely heavy noise. It can restore fine details considerably better than other state-of-the-art methods as it uses a realistic bounded noisy model. Its computational latency is equivalent to or twice as short as that of VBM3D and is nearly 10 times as short as that of VBM4D, owing to the efficient registration strategy and the fast IIR-based filtering methods. Implementing the proposed denoising approach combined with video compression in a fog computing manner results in a 70% bandwidth reduction, which meets the low communication latency in MPEG-4 mode.

In the future, parallelizing can be developed to speed up the proposed algorithm on the distributed nodes of high-performance embedded hardware or the industrial personal computers in the fog.

ACKNOWLEDGMENTS

This work was partially supported by the National Natural Science Foundation of China (grant no. 61571026), the National Key Project of Research and Development Plan, China (grant no. 2016YFE0108100), and the National Institutes of Health, USA (grant no. R01CA165255 and R21CA172864).

ORCID

Yifan Yang http://orcid.org/0000-0003-4237-5874

REFERENCES


---

### Table 2

<table>
<thead>
<tr>
<th>Input</th>
<th>Uncompression, kbps</th>
<th>Motion JPEG, kbps</th>
<th>MPEG-4, kbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise-free source</td>
<td>24883</td>
<td>2111</td>
<td>228</td>
</tr>
<tr>
<td>Noisy source</td>
<td>24883</td>
<td>15090</td>
<td>2056</td>
</tr>
<tr>
<td>VBM3D’s result</td>
<td>24883</td>
<td>3600</td>
<td>1421</td>
</tr>
<tr>
<td>VBM4D’s result</td>
<td>24883</td>
<td>6605</td>
<td>1306</td>
</tr>
<tr>
<td>Ours result (ICS)</td>
<td>24883</td>
<td>1551</td>
<td>549</td>
</tr>
</tbody>
</table>


22. Tearthipattayanon S, McDaniel B, Kung HT. Distributed deep neural networks over the cloud, the edge and end devices. Paper presented at: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS); 2017; Atlanta, GA.


34. Selesnick IW, Graber HL, Pfeil DS, Barbour RL. Simultaneous low-pass filtering and total variation denoising.


