PAPER

A Real-Time Cascaded Video Denoising Algorithm Using Intensity and Structure Tensor

Xin TAN[†a)], **Yu LIU**[†], ***Nonmembers***, **Huaxin XIAO**[†], ***Student Member***, ***and*** **Maojun ZHANG**[†], ***Nonmember***

SUMMARY A cascaded video denoising method based on frame averaging is proposed in this paper. A novel segmentation approach using intensity and structure tensor is used for change compensation, which can effectively suppress noise while preserving the structure of an image. The cascaded framework solves the problem of noise residual caused by single-frame averaging. The classical Wiener filter is used for spatial denoising in changing areas. Our algorithm works in real-time on an FPGA, since it does not involve future frames. Experiments on standard grayscale videos for various noise levels demonstrate that the proposed method is competitive with current state-of-the-art video denoising algorithms on both peak signal-to-noise ratio and structural similarity evaluations, particularly when dealing with large-scale noise.

key words: *cascaded framework, change segmentation, real-time capability, structure tensor, video denoising*

1. Introduction

Denoising is an important issue in image and video processing. Video denoising can enhance image quality, increase compression efficiency, reduce transmission bandwidth, and improve the robustness of the subsequent processes, such as feature extraction, object detection, and pattern classification.

Some basic image denoising ideas and algorithms are also used in video denoising, since videos consist of a sequence of images associated in time. These common algorithms include Gaussian filter, bilateral filter [1], [2], domain transformation [3]–[6], similar blocks matching [6]–[10], and sparse representations [11]–[13]. Video, compared to an image, provides sufficient additional information from nearby frames; thus, a better estimate of the original signal is expected. However, in order to exploit the extra information, video denoising algorithms have to handle a critical problem - motion estimation. The presence of motion between frames makes a video denoising method more complicated than image denoising. The changes caused by illumination or shadow should also be estimated. A recent and popular change estimation method is nonlocal patch based matching [6], [8]–[10]. In VBM3D [6] for example, change information is obtained by adaptively clustering similar patches across frames (space plus time). The position of the patch indicates the motion trajectory. Besides nonlocal matching, there are several other change estimation schemes. A multi-resolution motion analysis method in the wavelet domain was proposed in [14]. In [15], the change estimation was in the 3D DCT domain. Lian et al. [16] used vector estimation of wavelet coefficients. Some other algorithms used advanced statistical models, such as [17].

In this paper, we not only propose an effective video denoising algorithm, but also ensure the ability to implement it in hardware, such as an FPGA. Thus, it can be used in digital imaging devices in the future.

Frame averaging is one of the simplest video denoising methods, which has been studied since the 1990s [18]–[21]. The direct averaging can lead to motion blurring. Thus, change compensation is needed. Frame averaging is suitable for videos with a fixed background. In real life, the applications of such videos are numerous such as recording of the teleconference, stadiums, concerts, and museums. In particular, surveillance cameras with fixed installation are ubiquitous over streets, stations, airports, buildings, and parks. In this paper, our algorithm also aims at this kind of videos with stationary background.

A simple change compensated frame averaging method is weighted averaging by inter-frame pixels' similarity. However, in the changing areas, the denoising result is worse than the still background area, because fewer frames are averaged. Spatial filtering is applied for denoising in the changing areas [22], [23]. This simple weighted frame averaging method is detailed in Sect. II. The ASTA method [2] proposed by Bennett and McMillan is based on the simple framework above. The existing pixels' similarity-weighted averaging method is simple with low resource use and easy hardware implementation. However, its denoising performance is poor in low light. The reason is that the effectiveness of this method implies a prerequisite that the flickering change of the noise is lower than motion or illumination caused changes. Under bright light, this prerequisite can be satisfied. While under low light, the prerequisite is not met, because noise is amplified. At this time the weight cannot indicate the difference between noise and motion or lighting changes. How to accurately distinguish the changing areas from the stationary areas is the key to obtaining good results in the frame averaging method.

In our previous work [24], we improve the simple change compensated frame averaging method described above by suppressing the influence of noise on change compensation with a strong Gaussian filter. Nevertheless, the edge of the motion area is highly blurred by the Gaussian

Manuscript received December 18, 2014.
Manuscript revised March 14, 2015.
Manuscript publicized April 16, 2015.

[†]The authors are with the College of Information System and Management, National University of Defense Technology, Changsha, Hunan, 410073, P. R. China.
a) E-mail: tanxin08@nudt.edu.cn

DOI: 10.1587/transinf.2014EDP7435

filter, and some details are lost.

In this paper, a structure tensor is employed to assist and enhance segmentation of the change areas. Structure tensor [25]–[27] is a powerful tool for analyzing image structures. Compared to image gradient, linear structure tensor can better extract structure information from noisy images. This structure information repairs the blurring deficiency of the strong pre-filter. Besides, in practical applications the number of frames averaged is limited, as a result of storage space limitation. Only a few frames are frequently used. In this case, even if the change segmentation is accurate, $1/N$ amount of noise still stays. Thus, we propose a cascaded framework to address this problem.

The remainder of paper is organized as follows: Section 2 introduces the simple inter-frame pixels' similarity-weighted averaging method as a background. Section 3 proposes our intensity and structure tensor combined change segmentation. Section 4 proposes a novel cascaded denoising framework and analyzes the implementation on FPGA. In Sect. 5, experiments on standard grayscale videos for various noise levels demonstrate the effectiveness of the proposed method. Finally, Sect. 6 summarizes the work and suggests future work.

2. Simple Inter-Frame Pixels' Similarity-Weighted Averaging

Supposing the simple inter-frame pixels' similarity-weighted averaging method is centered at current frame (t_0). The smaller the difference of the inter-frame pixel values is, the larger the weight is. Every pixel of each frame is weighted averaged as

$$\begin{aligned} q_i^o &= \frac{\sum_{k\in T(i)} w_k \cdot p_k}{\sum_{k\in T(i)} w_k} \\ w_k &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d^2(k,i)}{2\sigma^2}\right) \\ d(k,i) &= \begin{cases} |p_k - p_i|, k \neq i \\ 0, k = i \end{cases} \end{aligned} \tag{1}$$

where k is the temporal index of the frame; i is the current frame's index, namely, $k = \cdots, i-2, i-1, i, i+1, i+2, \cdots$; $T(i)$ is the frame set consisting of N frames centered at frame i; p_k is the pixel value at (x, y) of the frame k, in particular, p_i is the pixel value of the current frame i of the same position (x, y); $d(k, i)$ is the absolute distance of pixels p_k and p_i. The distance $d(k, i)$ is spatially variant for the different space position (x, y); w_k is the Gaussian weight with standard deviation σ controlling how quickly the Gaussian falls off; and q_i^o is the averaged pixel value for current frame (t_0) at position (x, y). Here, we do not use the spatial coordinates in (1) because every pixel at different position has the same computation formula as (1).

The Gaussian weight w_k decreases with increase of the absolute distance $d(k, i)$. Here the pixel value distance $d(k, i)$ directly acts as the change segmentation criterion. Generally, under bright light the changes caused by motion or illumination are often large enough to get a zero weight.

Suppose the brightness difference that human eyes can distinguish is above L. Generally, L is about 10 (brightness range from 0 to 255). The selection of weight controlling parameter σ should make those weights whose corresponding $d(k, i)$ is above L close to zero.

3. Intensity and Structure Tensor Based Change Segmentation

The effectiveness of the method described in Sect. 2 relies on the distance measure $d(k, i)$ can correctly discriminate the changes caused by motion or illumination from noise. However, when noise change is larger than L, the weight is also close to 0. Thus, the noise is falsely estimated as normal change area. In this section, this drawback is rectified by our intensity and structure tensor combined scheme.

3.1 Strong Pre-Filter Based Intensity Segmentation

Intensity is one of the basic attributes of an image. Its measure is explicitly the pixel value. The changes in an image are attributable to the intensity variations. However, noise also causes intensity variations, which disturbs the normal change area segmentation. Carefully observing the noisy images (see Fig. 1 (a1)-(a2)), the noise caused intensity variation is disorderly. In a local area, some pixel intensities are enlarged, while some others are reduced. By contrast, normal motion or illumination variations results in an overall change, such as the moving hand of a person.

In order to suppress the influence of noise, a strong spatial filter is used to pre-process the noisy images. Pre-filter is frequently used in many denoising algorithms. For example, VBM3D [6] and BM3D [28] use a hard-thresholding operator on the transform domain coefficients as a coarse pre-filtering to improve the accuracy of block matching. The strong pre-filter must be used to effectively suppress large-scale noise. Considering the algorithm's complexity and noise suppression ability, we employ the Gaussian filter with a large kernel size. The new intensity distance is:

$$d_I(k,i) = \begin{cases} |K_{\rho_1}(p_k) - K_{\rho_1}(p_i)|, k \neq i \\ 0, k = i \end{cases} \tag{2}$$

where K_{ρ_1} is the Gaussian filter kernel with standard deviation ρ_1. Since too large a kernel causes heavy blur at the edge and small kernel cannot effectively suppress the noise, we set the Gaussian kernel through careful experiments. In our algorithm, the choice of the filter kernel follows the noise level; i.e, the larger the noise the larger the kernel size. In Fig. 1, the noise standard deviation is $\sigma_n = 50$. A 10×10 Gaussian filter with $\rho_1 = 5$ is used for pre-processing the noisy images. The filtered result is shown in (b1) and (b2) which are quite blurred. The intensity distance is shown in (b3). Compared to no pre-filtered intensity distance in (a3),

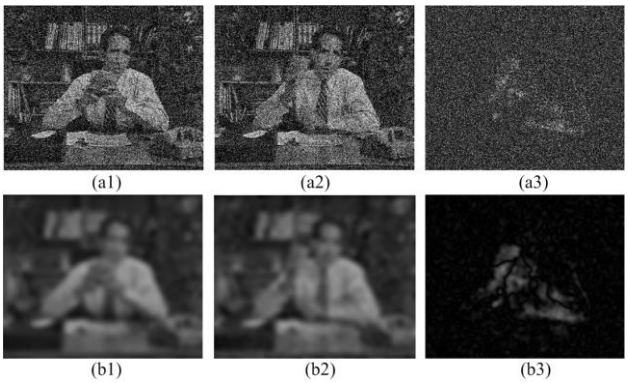

Fig. 1 Strong pre-filter on image intensity. (a1) and (a2) are the past and current frame with additive Gaussian white noise ($\sigma_n = 50$). (a3) is the intensity distance of (a1) and (a2). (b1) and (b2) are the pre-filtered results of (a1) and (a2) with a 10×10 Gaussian filter whose $\rho_1 = 5$. (b3) is the intensity distance of (b1) and (b2).

the strong filter is more powerful in suppressing noise.

3.2 Structure Tensor Based Segmentation

Although the strong pre-filter effectively suppresses large-scale noise, it also destroys the edges of the change area. Some detail variations are also damaged and even lost. Shown in Fig. 2 (a3) is the original intensity distance without noise. The outlines of the head, eyes, and arm of the salesman are clear. This indicates the motion of the salesman's body. But the strong filtered result in Fig. 2 (b3) nearly loses these outlines. In order to fix this drawback, we introduce structure tensor.

Structure tensor [25]–[27] is proposed by Weickert et al. It is a tool for analyzing the image structure by extracting the geometric features. Here, we use the simple linear structure tensor to analyze an image. It is defined as:

$$\begin{aligned}\mathbf{J}_{\rho_2}(p) &= K_{\rho 2} * (\nabla p_{\sigma'} \otimes \nabla p_{\sigma'}^T) \\ &= K_{\rho_2} * \begin{pmatrix} (I_x(p_{\sigma'}))^2 & I_x(p_{\sigma'})I_y(p_{\sigma'}) \\ I_x(p_{\sigma'})I_y(p_{\sigma'}) & (I_y(p_{\sigma'}))^2 \end{pmatrix} \end{aligned} \quad (3)$$

where ∇ is the image gradient operator; $p_{\sigma'}$ is the Gaussian filtered image of input p with Gaussian standard deviation σ'; $\otimes$ is the structure tensor product, its computation uses the image gradients $I_x(p_{\sigma'})$ and $I_y(p_{\sigma'})$ in x and y directions; $*$ is the convolution of Gaussian filter K_{ρ_2} with standard deviation ρ_2 and the structure tensor product. In general, $\rho_2 > \sigma'$. The Gaussian filter σ' before gradient operation and the filter K_{ρ_2} play the role of the strong pre-filter too. The Gaussian filter K_{ρ_2} isotropically synthesizes the local neighborhood structure tensor information, so it is called a linear structure tensor.

$\mathbf{J}_{\rho_2}$ contains the image geometric structure information. Orthogonally decomposing $\mathbf{J}_{\rho_2}$, the eigenvalues and eigenvectors can be obtained. Note $\mathbf{J}_{\rho_2} = \begin{pmatrix} a & b \\ b & c \end{pmatrix}$, the eigenvalues $\lambda_{1,2} = (a + c \pm \sqrt{(a-c)^2 + 4b^2})/2$, eigenvectors $\vec{e}_1 = [\cos(\theta)\ \sin(\theta)]^T$, and $\vec{e}_2 = [-\sin(\theta)\ \cos(\theta)]^T$ where

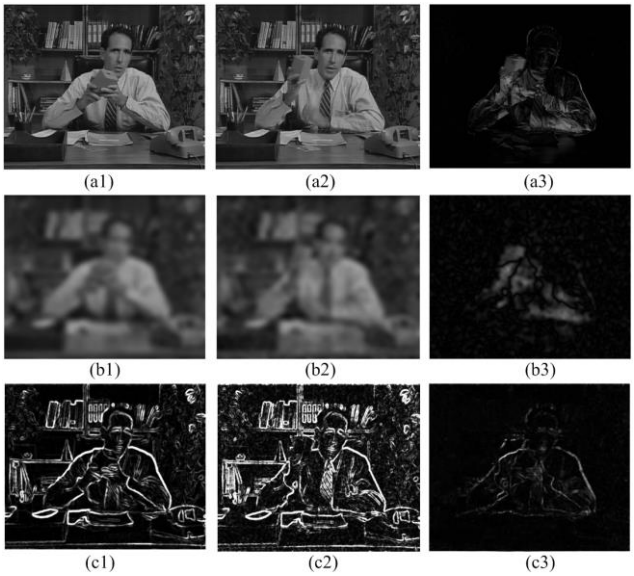

Fig. 2 Structure tensor based segmentation. (a1) and (a2) are the original images without noise pollution. (a3) is the original intensity distance of (a1) and (a2) which functions as the ground truth. (b3) is the strong pre-filtered intensity segmentation of (b1) and (b2) like that in Fig. 1. (c1) and (c2) is the maps of the structure tensor strength $\lambda_1 + \lambda_2$ of the noisy images. (c3) is the Log-Euclidean metric distance of (c1) and (c2).

$\theta = \arctan((\sqrt{(a-c)^2 + 4b^2} + (a-c))/2b)$. The maximum eigenvalue λ_1's corresponding eigenvector $\vec{e}_1$ points to the maximum gradient contrast direction, namely normal direction. The eigenvalue λ_2's corresponding eigenvector $\vec{e}_2$ is the tangential direction. $\lambda_1 + \lambda_2$ shows the strength of the structure. In Fig. 2, (c1) and (c2) are the maps of the structure strength extracted from the noisy images in Fig. 1 (a1) and (a2).

If changes occur, the variation of structure tensor is unavoidable. Thus, we can use structure tensor to detect the change. Similar to intensity distance, the structure tensor distance is also measured. Since structure tensor resides in a non-Euclidean space, a new metric called Log-Euclidean metric [29] is used. It is computed as:

$$d_{ST}(k,i) = \begin{cases} \sqrt{Trace\left\{\left[\log\left(\mathbf{J}_{\rho_2}(p_k)\right) - \log\left(\mathbf{J}_{\rho_2}(p_i)\right)\right]^2\right\}}, k \neq i \\ 0, k = i \end{cases} \quad (4)$$

where $\sqrt{Trace\left((\cdot)^2\right)}$ is the 'Frobenius norm'; $\log(\cdot)$ is structure tensor logarithmic operator defined in [29]. Suppose the matrix $A = (a_{ij})_{n\times n}$, thus Frobenius distance is:

$$\sqrt{Trace\left((A)^2\right)} = \left(\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2\right)^{1/2}. \quad (5)$$

The logarithm of a tensor is defined as a vector: $\log(\mathbf{J}_{\rho_2}) = \left(\log(a), \log(c), \sqrt{2}\log(b)\right)^T$. Figure 2 (c3) shows the Log-Euclidean metric distance of (c1) and (c2). The outlines of the head, eyes, and arm are extracted. This is

Fig. 3 Strong pre-filtered intensity and structure tensor combined change segmentation.

a good supplement for strong filtered intensity segmentation (b3). The intensity and structure tensor combined change segmentation is shown in Fig. 3. The combination is as follows:

$$d_{IST}(k,i) = \alpha \cdot d_{ST}(k,i) + d_I(k,i) \tag{6}$$

where α is the scale factor. Although Gaussian filter σ' and $K_{\rho 2}$ are used, noise still brings about more or less disturbance on structure tensor distance d_{ST}. In the stationary areas, d_{ST} does not exactly equal zero. Since the structure tensor Log-Euclidean metric distance d_{ST} does not have the same range as intensity distance d_I, the scale factor α is employed to make noise caused d_{ST} less than the brightness sensitivity L o f human eyes. In Fig. 3, $\alpha = 0.1$.

4. Cascaded Video Denoising with Frame Averaging

Now, we come back to Eq. (1). The final denoising result, in fact, is related to two factors. One is the weight w_k based on change segmentation; the other is the noisy image p_k. The new change segmentation scheme in Sect. 3 aims to acquire more accurate weight. As for noisy images p_k, how can we improve their qualities?

4.1 Use of Denoised Images

The pre-filter idea is utilized during the computation of $d_{ST}(k,i)$ and $d_I(k,i)$. Can noisy images p_k be pre-filtered too? Obviously, as for the past frames, each frame's denoising result can be directly used as the pre-filtered result. However, there is no such readily pre-filtered result for future frames. If a good pre-filtered result for future frames is required, then additional filter is needed. Considering the complexity and real-time capability for FPGA, we choose the strategy that avoids pre-filtering on the future frames and does not even use them for weighted averaging. Now, the proposed method can be described as:

$$\begin{aligned}
q_i^t &= \frac{\sum\limits_{k\in T'(i),k\neq i} w_k \cdot q_k + \sum\limits_{k\in T'(i),k=i} w_k \cdot p_i}{\sum\limits_{k\in T'(i)} w_k} \\
w_k &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{d_{IST}^2(k,i)}{2\sigma^2}\right) \\
d_{IST}(k,i) &= \alpha \cdot d_{ST}(k,i) + d_I(k,i) \\
d_I(k,i) &= \begin{cases} |K_{\rho_1}(q_k) - K_{\rho_1}(p_i)|, k \neq i \\ 0, k = i \end{cases} \\
d_{ST}(k,i) &= \\
&\begin{cases} \sqrt{Trace\left\{\left[\log\left(\mathbf{J}_{\rho_2}(q_k)\right) - \log\left(\mathbf{J}_{\rho_2}(p_i)\right)\right]^2\right\}}, k \neq i \\ 0, k = i \end{cases}
\end{aligned} \tag{7}$$

where $T'(i)$ is the frame set consisting of the past and current frames, totally N frames; $k = i - N + 1, \ldots, i - 1, i$; q_k is the denoising result of frame k; and q_i^t is the temporal denoising result.

In addition, in changing areas, due to the small weight of past frames, the denoising is worse than the stationary background. Spatial filtering is required for helping denoising in these areas. In our method, classical and simple Wiener filter [30] is chosen, which estimates the local mean and variance around each pixel. Note that the input image is $p = (a_{mn})_{W\times H}$ where m and n are the spatial coordinates, and $W \times H$ is the image resolution. For each pixel a_{mn}, the local mean $\mu_{wie} = 1/(MN) \sum\limits_{m,n\in\eta} a_{mn}$, where η is the $M \times N$ local neighborhood, and the variance $\sigma_{wie}^2 = 1/(MN) \sum\limits_{m,n\in\eta} a_{mn}^2 - \mu_{wie}^2$. Thus, the denoised pixel is $b_{mn} = \mu_{wie} + (\sigma_{wie}^2 - \nu^2)/\sigma_{wie}^2 \cdot (a_{mn} - \mu_{wie})$ where ν^2 is the noise variance. If the noise variance is not given, the average of all the local estimated variances is used. In our experiment in Sect. 5, it equals the given noise variance σ_n^2.

Note that the spatial filtering result for input image p_i is q_i^s. Weighted averaging is also used for the combination of the spatial filtering result q_i^s and the temporal filtering result q_i^t.

Suppose that the past $N - 1$ frames are all copies of the current frame. Namely, the scene is completely static. The weighted sum S_w of the past $N - 1$ frames reaches its maximum value $(N-1)\cdot w_k(d_{IST}(k,i) = 0)$. When the scene is changing, S_w is less than this maximum value. The smaller the S_w is, the more intense the change is; thus, the larger the weight of q_i^s is. Let us consider a threshold of thr, which is larger than 0, and less than the maximum sum weight $(N - 1) \cdot w_k(d_{IST}(k,i) = 0)$. When $S_w \leq thr$, the denoising quality of the temporal filter is not enough and the spatial filtering result q_i^s is added in. Therefore, the final spatio-temporal result is:

$$\begin{aligned}
q_i &= \begin{cases} q_i^t, S_w > thr \\ \dfrac{S_w \cdot q_i^t + (thr - S_w) \cdot q_i^s}{thr}, S_w \leq thr. \end{cases} \\
S_w &= \sum_{k\in T'(i),k\neq i} w_k\left(d_{IST}(k,i)\right)
\end{aligned} \tag{8}$$

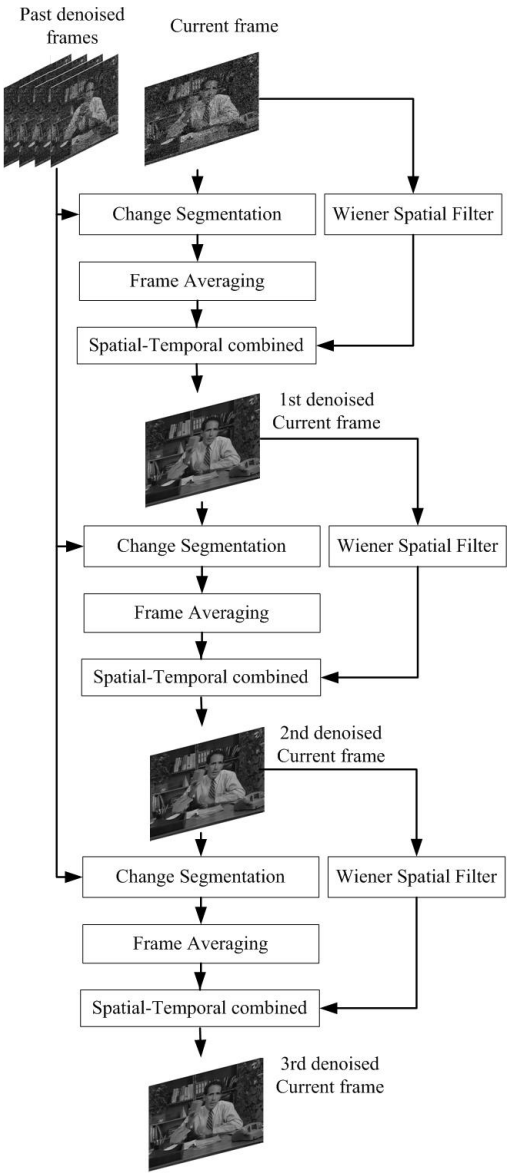

Fig. 4 Flow chart of our cascaded video denoising algorithm.

4.2 Cascaded Framework

In practical applications, the denoising resulting from (7) and (8) is limited, because of restrictions on image memory. Only several frames can be used; usually $N = 5$, 7 or 9. Suppose all past frames are denoised so well that they equal the original one and the weight calculation is also very accurate making it equal to $1/N$ for the stationary background. The noise standard deviation is σ_n for the current frame. After frame averaging, the standard deviation is reduced to σ_n/N. This residual is still not negligible. For example, when $N = 5$, and $\sigma_n = 50$, the noise residual reaches about 10.

Associated to the above pre-filter idea, we design a cascaded framework, as shown in Fig. 4. Through the cascaded framework, the denoising result of the first stage is the pre-filtered result for the next stage. Noise can be suppressed to $1/N^3$ in theory for a three-stage framework. If the number of frames $N = 5$, noise residual is less than 1%, which is acceptable. As for parameters selection, the parameters of the next stage are different from the first stage, because the noise level is different. Consider the intensity strong pre-filter ρ_1 for example. The kernel size and its standard deviation ρ_1 should both be decreased so as to adapt to the low noise level.

Fig. 5 The hardware implementation platform.

4.3 FPGA Implementation

To embed our algorithm into digital imaging devices, we have implemented our algorithm on an FPGA processor for evaluation. The implementation is carried out on an FPGA development board 'GENESIS' from DIGILENT company, as shown in Fig. 5. The FPGA processor is Xilinx Virtex-5 XC5VLX50T. The image sensor is a SONY imx122 CMOS.

Our algorithm is real-time and suitable for FPGA implementation. Since it does not involve the future frames, the time latency is less than one frame. Moreover, each step of the proposed algorithm can be easily implemented on an FPGA. These implementation steps are described below.

The Gaussian filter, in fact, is spatial weighted averaging in a local neighborhood. A 5×5 kernel only needs 25 multiplications, 25 additions, and one division. Storage of 5 image lines in Block RAM on FPGA or memory chip is also required. For structure tensor calculation, the image gradient is a convolution operation, and its implementation is the same as the Gaussian filter but with different multiplying factors. Logarithmic and square root operators in Log-Euclidean metric can be designed as look-up tables, since they are monotonic functions. The Gaussian weight w_k calculation is only related to the brightness sensitivity L of the human eyes. It is fixed after the initialization. So, it can be implemented as a look-up table too. For Wiener filter, it is a local operator, which needs several lines of storage. Other computations, such as mean and variance, only need several additions, divisions and subtractions. Similarly, the weight averaging of (7) and (8) is not complex either and only involves several multiplications, additions, subtractions and divisions.

The working of FPGA has a pipeline mode like CMOS

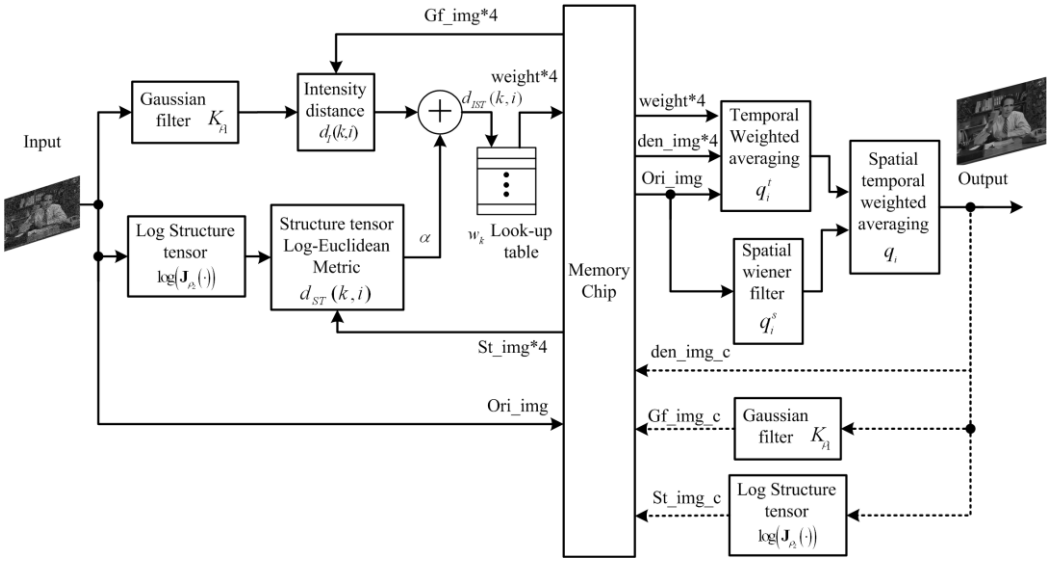

Fig. 6 The pipeline structure and DDR read/write design on FPGA. Here only one stage is presented. The calculations on the dash lines only work at the final stage.

or CCD sensors. The output of an image is pixel by pixel and line by line. The video forms a pixel stream. The processing unit is a pixel. The pipeline structure and DDR read/write design of one stage is shown in Fig. 6. Here we use four past frames, which is also the number in our experiment in Sect. 5.

At first, the input image 'Ori_img' is written to memory. Meanwhile, the intensity distance and the Log structure tensor metric are calculated. The past 4 Gaussian prefiltered images 'Gf_img' and 4 Log structure tensor images 'St_img' are read. After the weight look-up, 4 frames' 'weight' is written back to the memory. Here, the weight of the current frame is constant, for $\Delta q_{ik} = 0$ when $k = i$. In the weighted averaging of the current and past frames, the 'weight', the denoised past 4 frames 'den_img', and the current frame 'Ori_img' are read from memory. After Stage 3, the final denoised result 'den_img_c' and its Gaussian prefiltered result 'small_img_c' and Log structure tensor result 'St_img_c' are written into the memory for the denoising of next frame.

The working performance of our algorithm on FPGA is shown in Table 1. The FPGA working frequency is 90 MHz, whereas the maximum frequency of our algorithm is 116.3 MHz. The frame rate is 30 fps. The physical power consumption is 1.216 W measured by the Xpower Analyzer of the Xilinx ISE tool. The horizontal blank is 580 pixels, and the vertical blank is 120 lines. So the full Resolution of our 1080P (1920 × 1080) video is 2500 × 1200. Thus, one line consumes $1/90 \times 2500 = 27.78\,\mu s$. The time latency comes from the two aspects. One is the delay of the large size Gaussian filter, since the operation is executed until the data is ready. For a 20×20 kernel, which is the largest size in our algorithm, 20-line delay is required. The other aspect is the temporal weighted averaging. Since the weight is read from memory chip, it should be prepared in DDR before the temporal weighted averaging calculation. So some more ready weights in DDR lead to the stability of system. In our algorithm, the temporal weighted averaging starts work until 20 weight lines are ready. Thus, the total time latency is 40 lines, namely, $40 \times 27.78 = 1111.11\,\mu s$ (1.11 *ms*). It is the maximum time latency for one stage. For three stages, the maximum time latency is 3.33 *ms*. The time of one frame is 33.33 *ms* for 30 fps. Thus, our time latency is approximately 1/10 of one frame. Other time consumptions caused by hundreds of calculations, such as multiplications, divisions, subtractions, additions and looking up tables can be ignored. Since the working frequency is as high as 90 MHz, in general, their time latency is less than $1\,\mu s$. Since no extra future frame is needed, our algorithm attains real-time capability.

Table 1 Working performances of the proposed algorithm on FPGA.

Frame size	FPGA working frequency	Frame rate	Power consumption	Max time latency
1920×1080	90 MHz	30 fps	1.216W	3.33 *ms*

5. Experiments and Analysis

In this section, the performance of the proposed algorithm is evaluated. Standard test videos are downloaded from the video sequence database [31]. There are two types of videos in this database: ones with stationary background and the other ones with moving background. Since our method is aimed at videos with stationary background, we choose four of the former type of videos in our experiment: Salesman, Bridge_close, Hall_monitor, and Paris. The frame resolution of these videos is 288 × 352, and the duration is 300 frames. The experiments are carried out on the luminance channel of the video.

The noise is additive Gaussian white noise with stan-

dard deviation σ_n. We choose two state-of-the-art video denoising methods for comparison. They are VBM3D [6] and SURE-LET [5], which can be downloaded from authors' websites: [32] and [33]. In addition, the method with only pre-filtered intensity based change segmentation is chosen so as to show the structure tensor's effect.

In the experiments, there is only one parameter in VBM3D. It is the noise level estimator, which equals σ_n. The same parameter exists in SURE-LET. Besides, two more parameters are used in SURE-LET. One is the Number of Cycle-spin (Ncs). Increasing Ncs makes the denoising quality higher with a longer computation time. Similar to [5], the experiment chooses Ncs = 5. The other parameter is the Number of adjacent frames, Naf, which is used for multi-frame denoising. Here, we choose Naf = 5 like that in our algorithm.

For our algorithm, there are three sizes for strong Gaussian pre-filter at the first stage: 3×3 with standard deviation $\rho_1 = 1$ for $\sigma_n = 5, 15$; 10×10 with $\rho_1 = 3$ for $\sigma_n = 30, 50$; and 20×20 with $\rho_1 = 5$ for $\sigma_n = 75, 100$. The second and third stages have smaller kernel sizes. The parameters for structure tensor are fixed for all noise levels. In the first stage, the Gaussian filter σ' is 5×5 with standard deviation $\sigma' = 1.5$, and the filter $K_{\rho 2}$ is 5×5 with standard deviation $\rho_2 = 2$. In the second and third stages, the Gaussian filter σ' is 3×3 with $\sigma' = 0.5$, and $K_{\rho 2}$ is 3×3 with standard deviation $\rho_2 = 1$. The scale factor α is fixed at 0.1. The Wiener filter size is 5×5 for the first stage and 3×3 for the second and third stages. The threshold *thr* is set to $thr = 0.8\times(S_w)_{\max} = 0.8\times4w_k(0)$.

Two objective criteria, PSNR (Peak Signal to Noise Ratio) and SSIM (Structure SIMilarity), are employed to provide quantitative quality evaluations. PSNR is defined as:

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{R^2}{\mathrm{MSE}}\right), \tag{9}$$

where R is the dynamic range of the image. In our experiments, R equals 255 for 8 bits images. MSE is the mean squared error between the original and distorted images. SSIM is first calculated in the local windows with:

$$\mathrm{SSIM}(\mathbf{x},\mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}, \tag{10}$$

where $\mathbf{x}$ and $\mathbf{y}$ are respectively the image blocks from the original and distorted images at the same location. μ_x, μ_y, σ_x^2, σ_y^2 and σ_{xy} are the mean, variance, and cross-correlation within the block, respectively. C_1 and C_2 are constants to keep the calculations stable. In our experiments, $C_1 = 6.5$ and $C_2 = 58.5$. The SSIM of one frame is the average of each block's SSIM values. The larger the SSIM is, the better the structure integrity is. Its maximum value is 1. SSIM is believed to be a better criterion for evaluating the image quality [34]. In our experiments, the final PSNR and SSIM for a video is the average of each frame's score. The denoised pixels are clipped to the range 0-255.

Table 2 shows the comparison results using PSNR on the test videos. Table 3 shows the comparison results using SSIM. From the tables we can see that the proposed algorithm is slightly inferior to VBM3D at low noise levels. The reason is that although our Wiener spatial filtering method can be implemented on FPGA, it only operates on the single frame. However, VBM3D and SURE-LET both use block-matching across frames for motion compensation which can exploit the information from nearby frames. As a result, the motion compensation performance of our algorithm is inferior to those conventional methods. The temporal filters of them are also works well at the low noise levels. Thus, our method is not just equally effective to them. The proposed algorithm, however, is better at high

Table 2 PSNR(dB) comparison result.

σ_n	5	15	30	50	75	100
Salesman						
Input PSNR	34.15	24.61	18.59	14.15	10.63	8.13
VBM3D	**40.48**	**35.64**	**31.78**	**28.39**	24.51	23.16
SURE-LET	39.15	33.37	30.01	27.39	25.52	24.32
Proposed (no ST [a])	38.36	32.83	29.44	27.58	25.33	24.06
Proposed	38.74	33.80	30.47	27.98	**25.77**	**24.44**
bridge-close						
Input PSNR	34.15	24.61	18.59	14.15	10.63	8.13
VBM3D	**40.02**	**35.25**	**31.70**	28.92	25.87	24.04
SURE-LET	39.26	33.83	30.55	28.35	26.52	25.11
Proposed (no ST [a])	37.80	33.36	31.15	29.13	26.98	25.74
Proposed	39.11	34.13	31.27	**29.31**	**27.53**	**26.21**
Paris						
Input PSNR	34.15	24.61	18.59	14.15	10.63	8.13
VBM3D	**41.40**	**35.42**	**31.29**	**27.67**	21.91	20.11
SURE-LET	39.06	32.57	28.41	25.48	23.40	22.07
Proposed (no ST [a])	38.21	30.66	26.58	25.04	22.96	21.74
Proposed	39.63	32.96	28.79	25.86	**23.61**	**22.30**
Hall-Monitor						
Input PSNR	34.15	24.61	18.59	14.15	10.63	8.13
VBM3D	**41.54**	**37.93**	**34.69**	**31.24**	25.20	23.10
SURE-LET	40.29	34.89	31.05	28.18	26.05	24.43
Proposed (no ST [a])	40.08	35.28	31.43	28.64	25.90	24.11
Proposed	40.28	35.71	32.14	28.97	**26.43**	**24.64**

[a]ST is structure tensor based segmentation.

Table 3 SSIM comparison result.

σ_n	5	15	30	50	75	100
Salesman						
Input SSIM	0.895	0.571	0.313	0.167	0.090	0.054
VBM3D	**0.975**	**0.939**	**0.870**	0.768	0.608	0.529
SURE-LET	0.967	0.900	0.812	0.721	0.631	0.560
Proposed (no ST [a])	0.964	0.908	0.837	0.757	0.665	0.599
Proposed	0.965	0.919	0.854	**0.769**	**0.671**	**0.603**
bridge-close						
Input SSIM	0.855	0.471	0.231	0.118	0.063	0.039
VBM3D	**0.968**	**0.918**	0.838	0.755	0.656	0.614
SURE-LET	0.958	0.880	0.791	0.711	0.631	0.562
Proposed (no ST [a])	0.955	0.907	**0.849**	**0.770**	0.713	0.660
Proposed	0.960	0.912	**0.849**	**0.770**	**0.716**	**0.663**
Paris						
Input SSIM	0.897	0.619	0.409	0.265	0.167	0.112
VBM3D	**0.986**	**0.960**	**0.913**	**0.851**	0.652	0.543
SURE-LET	0.978	0.919	0.832	0.737	0.646	0.572
Proposed (no ST [a])	0.974	0.918	0.835	0.755	0.702	0.634
Proposed	0.977	0.938	0.872	0.774	**0.715**	**0.646**
Hall-Monitor						
Input SSIM	0.835	0.463	0.258	0.152	0.091	0.059
VBM3D	**0.972**	**0.955**	**0.930**	**0.892**	0.752	0.690
SURE-LET	0.966	0.925	0.863	0.791	0.716	0.647
Proposed (no ST [a])	0.966	0.939	0.893	0.786	0.792	0.723
Proposed	0.967	0.942	0.897	0.789	**0.795**	**0.726**

[a]ST is structure tensor based segmentation.

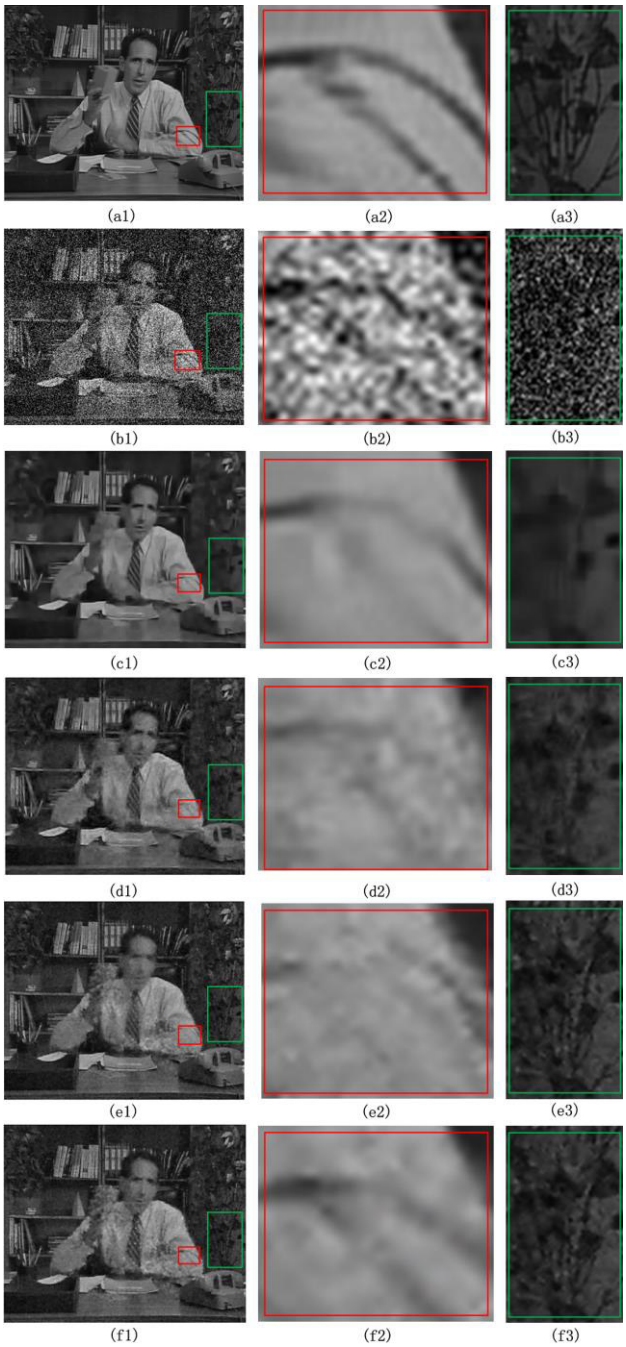

Fig. 7 Denoising result of a frame in Salesman video with noise level σ_n = 50. (a1)-(a3) Original image. (b1)-(b3) noisy image (PSNR = 14.14, SSIM = 0.165). (c)-(f) denoising result of VBM3D (PSNR = **28.10**, SSIM = 0.758), SURE-LET (PSNR = 27.23, SSIM = 0.715), proposed method but without structure tensor based segmentation (PSNR = 27.35, SSIM = 0.747), proposed method (PSNR = 27.58, SSIM = **0.761**). The red box enlarges the drape of the clothes at the arm; the green box enlarges the branches at the background.

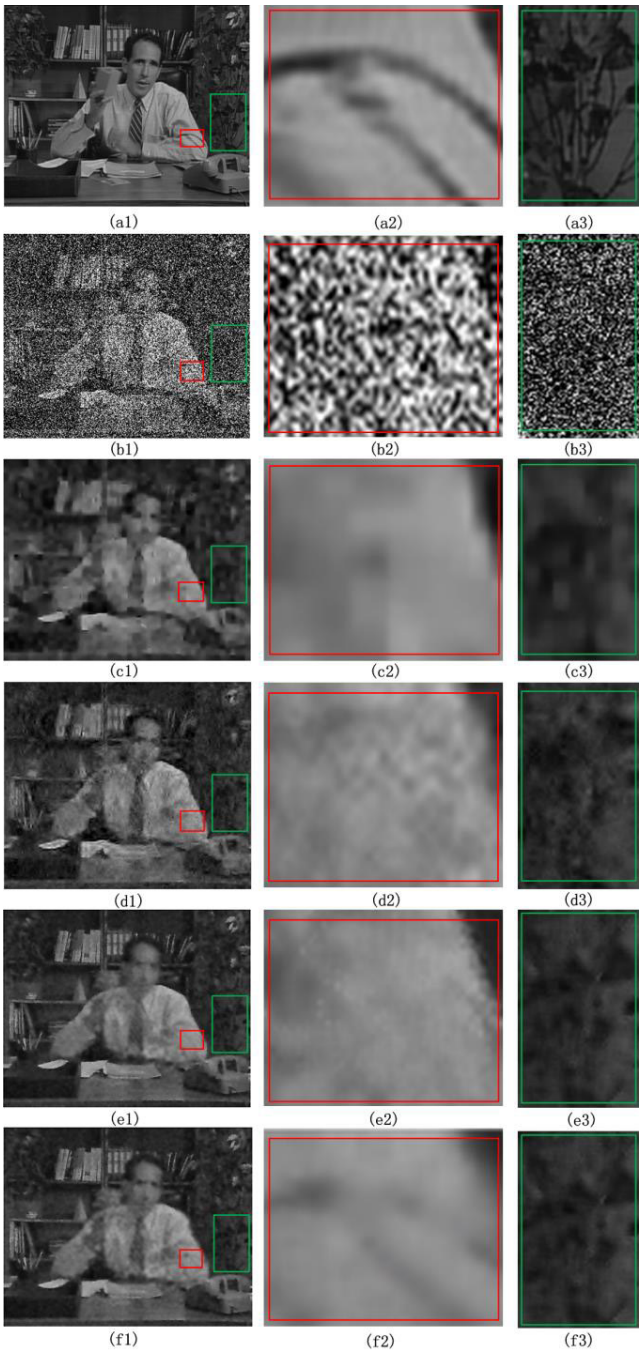

Fig. 8 Denoising result of a frame in Salesman video with noise level σ_n = 100. (a1)-(a3) Original image. (b1)-(b3) noisy image (PSNR = 8.12, SSIM = 0.053). (c)-(f) denoising result of VBM3D (PSNR = 23.03, SSIM = 0.524), SURE-LET (PSNR = 23.56, SSIM = 0.544), proposed method but without structure tensor based segmentation (PSNR = 23.31, SSIM = 0.578), proposed method (PSNR = **23.78**, SSIM = **0.587**). The red box enlarges the drape of the clothes at the arm; the green box enlarges the branches at the background.

noise levels. This is due to that block-matching lose its effectiveness under large-scale noise environment. Consequently, the motion estimation and compensation which are based on block-matching also do not work effectively. The proposed intensity and structure tensor based change segmentation, however, can do a better job when dealing with the high level noise. It leads to a better denoising performance of our method. Therefore, our method is particularly suitable for high level noise at low light illumination. It should be noted that in practical applications the most im-

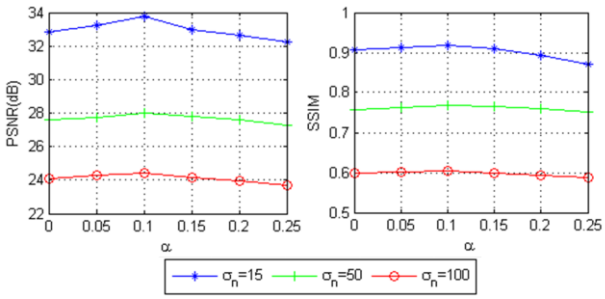

Fig. 9 Comparison for different scale factor α on Salesman video.

portant role of denoising is under low light conditions with large noise.

Figure 7 shows the denoising results of some frames in the Salesman video with $\sigma_n = 50$. From the figure we can see that VBM3D has the best smoothing result but with blurring effect. As for detail structure preservation, our method is the relatively better one. Take branches with green box in the background for example. After denoising, the branches are highly blurred by VBM3D, while our method still keeps the branches clear. SURE-LET is superior to VBM3D and inferior to ours at the branches. The effect of structure tensor is reflected at the edges of the motion area. Looking at the drape of the clothes at the elbow in the red box, the result of no structure tensor segmentation method (see Fig. 7 (e2)) loses this detail, while the complete proposed method restores this drape. VBM3D also restores this detail. The result of SURE-LET is in the middle position at this point. As for the denoising result of the whole image, the PSNR and SSIM indices indicate that our method is the best for structure preservation (SSIM), while VBM3D is the best in PSNR index for smoothness.

Figure 8 shows the denoising results of the frame in Fig. 7 with larger noise $\sigma_n = 100$. Our method is the best considering both PSNR and SSIM indices. The prominent effect comes from the details of the background, such as branches and contour of the book. VBM3D is the worst with extensive blurring of the image. The effect of structure tensor is still obvious, such as the salesman's eyes and the drape of the clothes at the elbow in the red box. Only our method retains the drape to a great extent.

In the proposed algorithm, the scale factor α is important. The influence of this parameter on denoising performance is shown in Fig. 9. The test video is Salesman here and three noise levels ($\sigma_n = 15,\ 50,\ 100$) are chosen. The variable α begins from 0 to 0.25 with an interval value of 0.05. It demonstrates that the best result is obtained when $\alpha = 0.1$. The performance degrades with smaller or larger α values. Thus, an appropriate setting of this scale factor is required.

6. Conclusion

In this paper, focusing on videos with stationary background, we propose a cascaded video denoising method based on frame-weighted averaging. The weight calculation depends on our intensity and structure tensor based change segmentation algorithm. We found that the direct intensity distance cannot differentiate motion or illumination change from large noise. A strong pre-filter is used to solve this problem. However, it leads to blurring at the edges or details in changed areas. To address this drawback, structure tensor was employed to aid change segmentation. On the other hand, because the average is computed over a limited number of frames, at

least $1/N$ noise is left behind by a single-stage frame averaging, where N is the total frame number. A cascaded framework was introduced to deal with this noise residual. The first stage acts as the pre-filter of the next stage. Our experimental comparisons with state-of-the-art algorithms demonstrated that the proposed method is competitive in both PSNR and SSIM evaluations, especially when dealing with the large-scale noise. Furthermore, our method is a real-time algorithm that has been implemented on FPGA for evaluation. Therefore, it is a suitable option for hardware devices.

Acknowledgments

This work was partially supported by National Natural Science Foundation (NSFC) of China under Grant No.61175006, No.61175015.

References

[1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," Proc. IEEE Int. Conf. Computer Vision (ICCV'98), Bombay, India, pp.839–846, 1998.
[2] E.P. Bennett and L. McMillan, "Video enhancement using per-pixel virtual exposures," Proc. ACM SIGGRAPH '05 Conference, pp.845–852, 2005.
[3] G. Varghese and Z. Wang, "Video denoising based on a spatiotemporal Gaussian scale mixture model," IEEE Trans. Circuits Syst. Video Technol., vol.20, no.7, pp.1032–1040, 2010.
[4] S. Yu, M.O. Ahmad, and M.N.S. Swamy, "Video denoising using motion compensated 3-D wavelet Transform with integrated recursive temporal filtering," IEEE Trans. Circuits Syst. Video Technol., vol.20, no.6, pp.780–791, 2010.
[5] F. Luiser, T. Blu, and M. Unser, "SURE-LET for Orthonormal Wavelet-Domain Video Denoising," IEEE Trans. Circuits Syst. Video Technol., vol.20, no.6, pp.913–919, 2010.
[6] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3-D transform-domain collaborative filtering," Proc. Conf. (EUSIPCO), pp.1257–1260, Poznan, Poland, Sept. 2007.
[7] H. Xu, Q. Sun, N. Luo, G. Cao, and D. Xia, "Iterative nonlocal total variation regularization method for image restoration," PLoS ONE, vol.8, no.6, e65865, 2013.
[8] Y. Han, and R. Chen, "Efficient video denoising based on dynamic nonlocal means," Image and Vision Computing, vol.30, no.2, pp.78–85, 2012.
[9] X. Li and Y. Zheng, "Patch-based video processing: A variational Bayesian approach," IEEE Trans. Circuits Syst. Video Technol., vol.19, no.1, pp.27–40, 2009.
[10] A. Buades, B. Coll, and J.-M. Morel, "Nonlocal image and movie denoising," Int. J. Comput. Vision, vol.76, no.2, pp.123–139, 2008.
[11] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations," IEEE Trans. Image Process., vol.18,

no.1, pp.27–35, 2009.
[12] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," IEEE Trans. Image Process., vol.15, no.12, pp.3736–3745, 2006.
[13] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," IEEE Trans. Image Process., vol.17, no.1, pp.53–69, 2008.
[14] F. Jin, P. Fieguth, and L. Winger, "Wavelet video denoising with regularized multiresolution motion estimation," EURASIP J. Appl. Singal Process., vol.2006, pp.1–11, 2006.
[15] D. Rusanovskyy and K. Egiazarian, "Video denoising algorithm in sliding 3D DCT domain," Proc. ACIVS, pp.618–625, Sept. 2005.
[16] N.-X. Lian, V. Zagorodnov, and Y.-P. Tan, "Video denoising using vector estimation of wavelet coefficients," Proc. IEEE Int. Sym. Circuits Syst., pp.2673–2676, May 2006.
[17] S.M.M. Rahman, M.O. Ahmad, and M.N.S. Swamy, "Video denoising based on inter-frame statistical modeling of wavelet coefficients," IEEE Trans. Circuits Syst. Video Tech., vol.17, no.2, pp.187–198, 2007.
[18] J.M. Boyce, "Noise reduction of image sequences using adaptive motion compensated frame averaging," Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, ICASSP, pp.461–464, March 1992.
[19] K.M. Kavanaugh, I.M.F. Pinto, M.J. McGillem, S.F. DeBoe, and G.B.J. Mancini, "Effects of video frame averaging, smoothing and edge enhancement on the accuracy and precision of quantitative coronary arteriography," Int. J. Cardiac Imaging, vol.5, no.4, pp.233–239, 1990.
[20] J.A. Smith, M. Ellis, R.F. Saunders, and A.L. Hall, "Color adaptive frame averaging," US Patent, no.5467770, Nov. 1995.
[21] B.A. Blair, and M. Parvini, "Frame averaging for use in processing video data," US Patent, no.6005638, Dec. 1999.
[22] S.R. Reeja and N.P. Kavya, "Real time video denoising," Proc. 2012 IEEE Intl. Conf. Engineering Education: Innovative Practices and Future Trends (AICERA), pp.1–5, July 2012.
[23] T. Portz, L. Zhang, and H. Jiang, "High-quality video denoising for motion-based exposure control," Proc. IEEE Intl. Conf. Computer Vision Workshops (ICCV Workshops), pp.9–16, Nov. 2011.
[24] X. Tan, Y. Liu, C. Zuo, and M. Zhang, "A real-time video denoising algorithm with FPGA implementation for Poisson-Gaussian noise," J. Real-Time Image Proc., DOI: 10.1007/s11554-014-0405-2, online, Feb. 2014.
[25] J. Weickert and H. Scharr, "A scheme for coherence-enhancing diffusion filtering with optimized rotation invariance," J. Visual Comm. Imag. Repres, vol.13, no.1-2, pp.103–118, 2002.
[26] J. Weickert, Anisotropic Diffusion in Image Processing, Teubner-Verlag, Stuttgart, Germany, 1998.
[27] J. Weickert, "Coherence-enhancing diffusion filltering," Int. J. Computer Vision, vol.31, no.2-3, pp.111–127, 1999.
[28] K. Dabov, A. Foi, V. Katkovnik, and K.O. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," IEEE Trans. Image Process., vol.16, no.8, pp.2080–2095, 2007.
[29] V. Arsigny, P. Fillard, X. Pennec, and N. Ayache, "Log-Euclidean metrics for fast and simple calculus on diffusion tensors," Magnetic Resonance in Medicine, vol.56, no.2, pp.411–421, 2006.
[30] J.S. Lim, Two-Dimensional Signal and Image Processing, Prentice Hall, Englewood Cliffs, NJ, 1990.
[31] xiph.org, "Xiph.org video test media," http://media.xiph.org/video/derf/, accessed Aug. 3, 2014.
[32] A. Foi, "Image and video denoising by sparse 3D transform-domain collaborative filtering," Transforms and spectral methods group, Department of signal processing, Tampere university, http://www.cs.tut.fi/~foi/GCF-BM3D/, accessed Aug. 3, 2014.
[33] F. Luisier, "Some SURE-LET video denoising results," Biomedical Imaging Group, École Polytechnique Fédérale de Lausanne, http://bigwww.epfl.ch/luisier/videodenoising/, accessed Aug. 3, 2014.
[34] Z. Wang, and A.C. Bovik, "Mean squared error: Love it or leave it? A new look at signal fidelity measures," IEEE Signal Process. Mag., vol.26, no.1, pp.98–117, 2009.

Xin Tan received his B.S. degree in Electronics Information from Sichuan University in 2008, and M.S. degrees in Systems Engineering at National University of Defense Technology (NUDT) in 2010. He is currently working towards the Ph. D. degree at College of Information System and Management, NUDT. His research interests focus on image/video processing and multimedia information systems.

Yu Liu received his B.S. from Northwestern Polytechnical University, Xi'an, China in 2005. He received his MSc in image processing and PhD in computer graphics from the University of East Anglia, Norwich, United Kingdom, in 2007 and 2011, respectively. He is currently an Associate Professor in the College of Information System and Management, National University of Defense Technology. His research interests include image/video processing, computer graphics, and visual-haptic technology.

Huaxin Xiao received his BS degree in automation from University of Electronic Science and Technology of China in 2012, and M.S. degrees in Systems Engineering at National University of Defense Technology (NUDT) in 2014. He is currently pursuing his ph.D. degree in control science and engineering from NUDT. His research interests include sparse representation and computer vision.

Maojun Zhang received his B.S. and Ph.D. degrees in Systems Engineering at National University of Defense Technology (NUDT) at 1992 and 1997, respectively. From 1997 to 2003, he was an Assistant Professor with Department of Systems Engineering, NUDT. Since 2003, He has been a Professor of College of Information System and Management at NUDT, Changsha, China. His major interests include computer vision, image/video processing, multimedia information systems, and virtual reality technology.