*International Conference on Image Analysis and Recognition (ICIAR11), Burnaby, BC, Canada, June, 2011*

# Enhancing Video Denoising Algorithms by Fusion from Multiple Views

Kai Zeng and Zhou Wang

Department of Electrical and Computer Engineering, University of Waterloo
Waterloo, ON, N2L 3G1, Canada
kzeng@engmail.uwaterloo.ca, zhouwang@ieee.org

**Abstract.** Video denoising is highly desirable in many real world applications. It can enhance the perceived quality of video signals, and can also help improve the performance of subsequent processes such as compression, segmentation, and object recognition. In this paper, we propose a method to enhance existing video denoising algorithms by denoising a video signal from multiple views (front-, top-, and side-views). A fusion scheme is then proposed to optimally combine the denoised videos from multiple views into one. We show that such a conceptually simple and easy-to-use strategy, which we call multiple view fusion (MVF), leads to a computationally efficient algorithm that can significantly improve video denoising results upon state-of-the-art algorithms. The effect is especially strong at high noise levels, where the gain over the best video denoising results reported in the literature, can be as high as 2-3 dB in PSNR. Significant visual quality enhancement is also observed and evidenced by improvement in terms of SSIM evaluations.

**Keywords:** video denoising, image quality enhancement, image fusion, multiple views

## 1 Introduction

Digital video or image sequence has become ubiquitous in our everyday lives. It is critically important to maintain the quality of video at an acceptable level in various application environments such as network visual communications. However, video signals are subject to noise contaminations during acquisition and transmission. Effective *video denoising* algorithms that can remove or reduce the noise is often desired. They not only supply video signals that have better perceptual quality, but also help improve the performance of the subsequent processes such as compression, segmentation, resizing, de-interlacing, and object detection, recognition, and tracking [1].

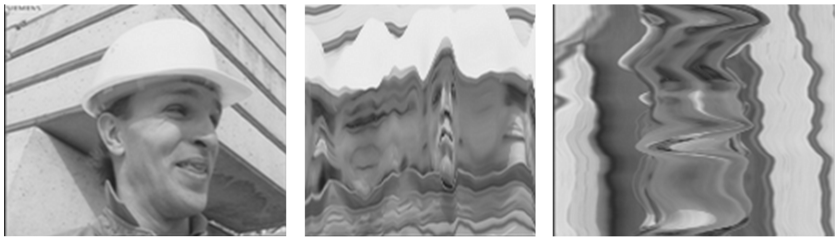
Existing video denoising algorithms may be roughly classified into three categories. In the first category, the video signal is denoised on a frame-by-frame basis, where all that is needed is a 2D still image denoising algorithm applied to each frame of the video sequence independently. Well-known and state-of-the-art still image denoising algorithms include the Matlab Wiener2D function, Bayes

least square estimation based on Gaussian scale mixture model (BLS-GSM) [2], nonlocal means denoising (NLM) [3], K-SVD method [4], Stein's unbiased risk estimator-linear expansion of threshold algorithm(SURE-LET) [5], and block matching and 3D transform shrinkage method (BM3D) [6]. For the purpose of video denoising, the major advantage of these approaches is memory efficiency, as no storage of previous frames are necessary in order to denoise the current frame. However, since the correlation between neighboring frames is completely ignored, the denoising process does not make use of all available information and thus cannot achieve the best denoising performance.

In natural video signals, there exists strong correlation between adjacent frames. The second category of video denoising approaches exploited such correlation by incorporating both intra- and inter-frame information. It was found that motion estimation and compensation could further enhance inter-frame correlation [7–9]. In [7], a motion estimation algorithm was employed for recursive temporal denoising along estimated motion trajectory. Motion compensation processes had also been incorporated into BLS-GSM and SURE-LET methods, leading to the ST-GSM [8] and video SURE-LET algorithms [9]. In [10], it was claimed that finding single motion trajectory may not be the best choice for video denoising. Instead, multiple similar patches in neighboring frames are found that may not reside along a single trajectory. This is followed by transform and shrinkage based denoising procedures. Perhaps one of the most successful video denoising methods in recent years is the extension of BM3D method for video, namely VBM3D [11], which searches similar patches in both intra- and inter-frames and uses 3D bilateral filtering for noise removal after aggregating the similar patches together.

The third category of denoising algorithms treat video sequences as 3D volumes. The algorithms can operate in the space-time domain by adaptive weighted local averaging [12], 3D order-statistic filtering [13], 3D Kalman filtering [14], or 3D Markov model based filtering [15]. They may also be applied in 3D transform domain, where soft/hard thresholding or Bayesian estimation are employed to eliminate noise, followed by an inverse 3D transform that brings the signal back to the space-time domain. The method in [16] is one such example, where 3D dual-tree complex wavelet transform was employed that demonstrates some interesting and desired properties. Recently, several authors investigated 3D-patch based methods and achieved highly competitive denoising performance [17, 18].

Ideally, to make the best use of all available information, the best video denoising algorithms would need to operate in 3D (Category 3). However, when there exists significant motion in the video, direct space-time 3D filtering or 3D transform based approaches are difficult to effectively cover all motion-related video content within local region. Meanwhile, 3D-patch based methods are expensive in finding similar 3D-patches in the 3D volume. By contrast, 2D denoising algorithms that use intra- and/or inter-frame information (Categories 1 and 2) can be made much more efficient, but their performance is restricted by not fully making use of the neighboring pixels in all three dimensions simultaneously.



(a) (b) (c)

**Fig. 1.** A video signal observed from (a) front view; (b) side view; and (c) top view.

In this paper, we propose a simple strategy, called multiple view fusion (MVF), that provides a useful compromise between 2D (Categories 1 and 2) and 3D (Category 3) approaches. In particular, we denoise the same video volume data with 2D approaches but from three different views, i.e., front view, top view, and side view. An optimal fusion scheme is then employed to combine the three denoised versions of the video. By doing so, the advantage of 2D denoising methods is utilized. Meanwhile, each pixel is denoised by its neighboring pixels from all three dimensions. We show that this simple strategy leads to significant gain of video denoising performance over different base denoising algorithms, especially at high noise levels.

## 2 Proposed Method

A video signal can be expressed as a 3D function $f(u, v, t)$, where $u$ and $v$ are the horizontal and vertical spatial indices and $t$ is the time index, respectively. A video is typically played along the time axis. At any time instance $t = t_0$, the video is displayed as a 2D front-view image $g_{FV}^{(t_0)}(u, v) = f(u, v, t_0)$ and the image changes over time $t$. If we think of a video signal as 3D volume data, then it can also be viewed from the side or the top. This gives two other ways to play the same video − a sequence of 2D top-view images $g_{TV}^{(u_0)}(v, t) = f(u_0, v, t)$ for different values of $u_0$ and a sequence of 2D side-view images $g_{SV}^{(v_0)}(u, t) = f(u, v_0, t)$ for different values of $v_0$. An example is given in Fig. 1, where the rarely observed side- and top-view images demonstrate some interesting regularized spatiotemporal structures.

Let $x$ be an original noise-free video signal, which is contaminated by additive noise $n$, resulting in a noisy signal

$$y = x + n\,. \tag{1}$$

A video denoising operator $D$ takes the noisy observation $y$ and maps it to an estimator of $x$:

$$\hat{x} = D(y)\,, \tag{2}$$

such that the difference between $x$ and $\hat{x}$ is as small as possible. How to quantify the difference between $x$ and $\hat{x}$ is another subject of study. The most typically used ones are the mean squared error (MSE) and equivalently the peak-signal-to-noise ratio (PSNR). However, recent studies showed that the structural similarity index (SSIM) [19] may be a better measure in predicting perceived image distortion.

The proposed MVF method relies on a base video denoising algorithm (which could be as simple as frame-by-frame Winer2D, or as complicated as VBM3D [11]). The base denoiser is applied to the same noisy signal $y$ multiple times but from different views, which gives multiple versions of denoised signal

$$\begin{aligned} z_1 &= D_1(y)\,, \\ z_2 &= D_2(y)\,, \\ &\cdots\cdots\,, \\ z_N &= D_N(y)\,. \end{aligned} \tag{3}$$

In this paper $N = 3$, as we have three different views, but in principle the general approach also applies to the cases of less or more views, or multiple denoising algorithms. Let $\mathbf{z} = [z_1, z_2, ..., z_N]^T$ be a vector that contains all denoised results, then the final denoised signal $\hat{x}$ is given by applying a fusion operator $F$ to $\mathbf{z}$:

$$\hat{x} = D(y) = F(\mathbf{z}) = F(D_1(y), D_2(y), ..., D_N(y))\,. \tag{4}$$

In the case that the base denoisers are predetermined, all the remaining task is to define the fusion rule $F$, which would be desired to achieve certain optimality. Here we employ a weighted average fusion method given by

$$\hat{x} = \mathbf{w}^T(\mathbf{z} - \boldsymbol{\mu}_\mathbf{z}) + \mu_x\,, \tag{5}$$

where $\mu_x = \mathbb{E}(x)$ (we use $\mathbb{E}$ to denote the expectation operator), $\boldsymbol{\mu}_\mathbf{z}$ is a column vector of expected values $[\mathbb{E}(z_1), \mathbb{E}(z_2), ..., \mathbb{E}(z_N)]^T$, and $\mathbf{w}$ is a column vector $[w_1, w_2, ...w_N]^T$ that defines the weight assigned to each denoised signal. To find the optimal weights $\mathbf{w}$ in the least-square sense, we define the following error energy function

$$E = \mathbb{E}[(x - \hat{x})^2] + \lambda\|\mathbf{w} - \frac{1}{N}\mathbf{1}\|^2\,, \tag{6}$$

where $\mathbf{1}$ is a length-$N$ column vector with all entries equaling 1. The second term is to regularize the weighting vector towards all equal weights, and the parameter $\lambda$ is used to control the strength of regularization. Taking the derivative of $E$ with respect to $\mathbf{w}$ and setting it zero, we obtain

$$(\mathbf{C}_\mathbf{z} + \lambda\mathbf{I})\mathbf{w} = \mathbf{b} + \frac{\lambda}{N}\mathbf{1}\,, \tag{7}$$

where $\mathbf{I}$ denotes the $N \times N$ identity matrix, $\mathbf{C}_\mathbf{z}$ is the covariance matrix

$$\mathbf{C}_\mathbf{z} = \mathbb{E}[(\mathbf{z} - \boldsymbol{\mu}_\mathbf{z})(\mathbf{z} - \boldsymbol{\mu}_\mathbf{z})^T]\,, \tag{8}$$

and $\mathbf{b}$ is a column vector given by

$$\mathbf{b} = \mathbb{E}[(x - \mu_x)(\mathbf{z} - \boldsymbol{\mu}_\mathbf{z})] \,. \tag{9}$$

We can then solve for optimal $\mathbf{w}$, which gives

$$\mathbf{w}_{opt} = (\mathbf{C}_\mathbf{z} + \lambda \mathbf{I})^{-1} \left( \mathbf{b} + \frac{\lambda}{N} \mathbf{1} \right) . \tag{10}$$

Here the $\lambda \mathbf{I}$ term plays an important role in stabilizing the solution, especially when $\mathbf{C}_\mathbf{z}$ is close to singular. The computation of $\mathbf{b}$ requires the original signal $x$, which is not available. But by assuming $n$ to be zero-mean and independent of $\mathbf{z}$, we have

$$\mathbf{b} = \mathbb{E}[(y - n - \mu_x)(\mathbf{z} - \boldsymbol{\mu}_\mathbf{z})] = \mathbb{E}[(y - \mu_y)(\mathbf{z} - \boldsymbol{\mu}_\mathbf{z})] \,. \tag{11}$$

When applying the above approach to real signals, the expectation operators would need to be replaced by sample means. In our implementation, we apply the weight calculation to individual non-overlapping $8 \times 8 \times 8$ blocks, resulting in block-wise space-time adaptive weights in the 3D volume. Eq. (5) is then applied to each block to obtain the final denoised signal.

## 3 Experimental Result

We use publicly available video sequences to test the proposed algorithm, which include "Akiyo", "Carphone", "Forman", "Miss America", "News", and "Salesman". The size of all sequences is $144 \times 176 \times 144$. Independent white Gaussian noise was added to the original video sequences, where the noise standard deviation, $\sigma$, covers a wide range between 10 and 100. All sequences are in YCrCb 4:2:0 format, but only the denoising results of the luma channel was reported here to validate the algorithm. Two objective criteria, namely PSNR and SSIM [19], were employed to evaluate the quality of denoised video quantitatively. PSNR is the most widely used method in the literature, but SSIM has been recognized as a much better measure to predict subjective quality measurement.

Many state-of-the-art denoising algorithms are publicly available that facilitate direct comparisons. Due to space limit, here we report our comparison results for 5 noise levels ($\sigma$ equals 10, 15, 20, 50, and 100, respectively) using three base denoising methods with and without using our MVF approach. The base algorithms are Matlab Wiener2D, BLS-GSM [2] and VBM3D [11]. We have also applied our MVF approach to a list of other highly competitive algorithms, including NLM [10], K-SVD [4], and SURE-LET [9]. Similar results were obtained but are not reported here.

Table 1 shows the comparison results using PSNR and SSIM measures, which were computed frame-by-frame and then averaged over all frames. It can be seen that the proposed MVF approach consistently leads to performance gain over all base denoising algorithms, for all test video sequences, and at all noise levels.

**Table 1.** PSNR and SSIM comparisons for three video denoising algorithms with and without MVF

| Video Sequence | *Akiyo* | | | | | *Carphone* | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Noise std ($\sigma$) | 10 | 15 | 20 | 50 | 100 | 10 | 15 | 20 | 50 | 100 |
| PSNR Results (dB) | | | | | | | | | | |
| Wiener-2D | 33.22 | 30.38 | 28.33 | 21.58 | 15.94 | 32.66 | 29.84 | 27.86 | 21.35 | 15.86 |
| with MVF | 34.69 | 31.91 | 29.89 | 23.15 | 17.52 | 33.90 | 31.20 | 29.29 | 22.87 | 17.42 |
| BLG-GSM | 36.12 | 33.73 | 32.09 | 27.32 | 24.36 | 35.34 | 33.00 | 31.40 | 26.47 | 23.15 |
| with MVF | 39.95 | 37.58 | 35.88 | 30.78 | 27.43 | 37.01 | 34.92 | 33.50 | 29.02 | 25.81 |
| VBM3D | 42.01 | 39.76 | 37.91 | 30.79 | 24.39 | 38.50 | 36.64 | 35.35 | 29.82 | 23.30 |
| with MVF | 42.33 | 40.08 | 38.36 | 32.64 | 26.93 | 38.50 | 36.71 | 35.46 | 30.97 | 25.76 |
| SSIM Results | | | | | | | | | | |
| Wiener-2D | 0.876 | 0.788 | 0.700 | 0.364 | 0.164 | 0.885 | 0.803 | 0.722 | 0.408 | 0.205 |
| with MVF | 0.906 | 0.833 | 0.757 | 0.432 | 0.213 | 0.909 | 0.840 | 0.771 | 0.472 | 0.255 |
| BLG-GSM | 0.952 | 0.924 | 0.898 | 0.765 | 0.636 | 0.951 | 0.927 | 0.902 | 0.773 | 0.627 |
| with MVF | 0.977 | 0.964 | 0.949 | 0.866 | 0.749 | 0.964 | 0.947 | 0.930 | 0.839 | 0.718 |
| VBM3D | 0.983 | 0.976 | 0.965 | 0.874 | 0.616 | 0.972 | 0.961 | 0.951 | 0.874 | 0.628 |
| with MVF | 0.986 | 0.978 | 0.967 | 0.903 | 0.684 | 0.972 | 0.961 | 0.952 | 0.892 | 0.691 |
| Video Sequence | *Foreman* | | | | | *Miss America* | | | | |
| PSNR Results (dB) | | | | | | | | | | |
| Wiener-2D | 32.22 | 29.49 | 27.55 | 21.17 | 15.77 | 34.36 | 31.35 | 29.17 | 21.91 | 16.07 |
| with MVF | 33.11 | 30.53 | 28.70 | 22.59 | 17.30 | 35.74 | 32.80 | 30.67 | 23.47 | 17.65 |
| BLG-GSM | 34.22 | 31.92 | 30.32 | 25.44 | 22.21 | 38.69 | 36.54 | 35.09 | 30.61 | 27.52 |
| with MVF | 35.83 | 33.65 | 32.12 | 27.36 | 24.05 | 41.03 | 38.99 | 37.59 | 33.16 | 30.02 |
| VBM3D | 37.37 | 35.50 | 34.12 | 28.47 | 22.46 | 41.93 | 40.19 | 38.81 | 33.55 | 26.57 |
| with MVF | 37.68 | 35.80 | 34.44 | 29.28 | 24.14 | 42.34 | 40.57 | 39.24 | 34.69 | 28.93 |
| SSIM Results | | | | | | | | | | |
| Wiener-2D | 0.887 | 0.812 | 0.738 | 0.432 | 0.220 | 0.848 | 0.737 | 0.633 | 0.275 | 0.107 |
| with MVF | 0.906 | 0.843 | 0.778 | 0.488 | 0.267 | 0.879 | 0.785 | 0.692 | 0.331 | 0.138 |
| BLG-GSM | 0.938 | 0.910 | 0.884 | 0.746 | 0.591 | 0.958 | 0.939 | 0.922 | 0.841 | 0.751 |
| with MVF | 0.952 | 0.930 | 0.908 | 0.792 | 0.646 | 0.972 | 0.960 | 0.948 | 0.884 | 0.791 |
| VBM3D | 0.961 | 0.947 | 0.933 | 0.844 | 0.601 | 0.976 | 0.968 | 0.959 | 0.901 | 0.669 |
| with MVF | 0.962 | 0.948 | 0.934 | 0.857 | 0.643 | 0.978 | 0.970 | 0.962 | 0.915 | 0.685 |
| Video Sequence | *News* | | | | | *Salesman* | | | | |
| PSNR Results (dB) | | | | | | | | | | |
| Wiener-2D | 31.95 | 29.11 | 27.14 | 20.83 | 15.65 | 31.48 | 28.97 | 27.23 | 21.28 | 15.90 |
| with MVF | 33.34 | 30.58 | 28.66 | 22.44 | 17.26 | 33.07 | 30.65 | 28.94 | 22.92 | 17.50 |
| BLG-GSM | 34.34 | 31.86 | 30.11 | 24.90 | 21.42 | 33.16 | 30.89 | 29.37 | 25.35 | 23.01 |
| with MVF | 37.72 | 35.30 | 33.57 | 28.22 | 24.58 | 36.82 | 34.43 | 32.82 | 28.34 | 25.71 |
| VBM3D | 39.76 | 37.47 | 35.73 | 28.50 | 21.69 | 38.93 | 36.49 | 34.57 | 27.92 | 23.18 |
| with MVF | 40.04 | 37.73 | 36.06 | 30.18 | 24.67 | 39.27 | 36.84 | 35.06 | 29.58 | 25.52 |
| SSIM Results | | | | | | | | | | |
| Wiener-2D | 0.887 | 0.807 | 0.731 | 0.431 | 0.231 | 0.876 | 0.798 | 0.724 | 0.415 | 0.194 |
| with MVF | 0.915 | 0.851 | 0.787 | 0.503 | 0.292 | 0.912 | 0.854 | 0.796 | 0.511 | 0.265 |
| BLG-GSM | 0.950 | 0.923 | 0.894 | 0.737 | 0.564 | 0.908 | 0.854 | 0.804 | 0.613 | 0.478 |
| with MVF | 0.973 | 0.958 | 0.942 | 0.844 | 0.712 | 0.958 | 0.930 | 0.902 | 0.769 | 0.643 |
| VBM3D | 0.981 | 0.971 | 0.960 | 0.860 | 0.581 | 0.975 | 0.956 | 0.929 | 0.739 | 0.488 |
| with MVF | 0.982 | 0.973 | 0.963 | 0.895 | 0.684 | 0.976 | 0.958 | 0.936 | 0.803 | 0.618 |

The gain is especially significant at high noise levels, where the improvement can be as high as 2-3 dB in terms of PSNR over state-of-the-art algorithms such as VBM3D, which is among the best algorithms ever reported in the literature. To demonstrate the performance improvement for individual video frames, Fig. 2 depicts PSNR and SSIM comparisons as functions of frame number for "Foreman" sequence. Again, consistent improvement is observed for almost all frames, indicating the robustness of the proposed MVF approach.

Figure 3 provides visual comparisons of the denosing results of one frame extracted from the "Salesman" sequence. For each denoised frame, the SSIM quality map is also given, where brighter pixels indicate higher SSIM values and thus better quality. Visual quality improvement by the proposed MVF approach can be perceived in various locations in the denoised frames, for example, the bookshelf region. Such improvement is also clearly indicated by the SSIM maps.

## 4 Conclusion

We propose an MVF approach that can improve video denoising performance of existing algorithms by fusing the denoising results from multiple views. Our experimental results demonstrate consistent improvement over some of the best video denoising algorithms in the literature. The proposed method is conceptually simple, easy-to-use, and computationally efficient. The complexity of the whole algorithm mainly depends on that of the base denoising method, but not the MVF procedure. In principle, the MVF strategy could be applied to any existing video denoising algorithm, but our major intension here is to apply it to 2D approaches (Categories 1 and 2 described in Section 1). The reason is that the denoising results obtained by applying 2D approaches from different views tend to be complementary to each other. By contrast, 3D approaches (Category 3) such as those using 3D patches have already considered the dependencies between neighboring pixels from all directions, and thus applying them from different views may lead to similar results that would not complement each other to a significant extent.

The video denoising performance may be further improved by adopting better base denoising algorithms or by improving the fusion method. One could also attempt to fuse the denoising results not only from multiple views but also by multiple algorithms. It is also interesting to look into novel algorithms for denoising from side- and top-views, where we have observed special regularities (that are quite different from what has been observed from front-view) that are worth deeper investigations.

## Acknowledgment

This research was supported in part by Natural Sciences and Engineering Research Council of Canada in the forms of Discovery, Strategic and CRD Grants, and by an Ontario Early Researcher Award, which are gratefully acknowledged.

## References


1. Bovik, Alan C.: Handbook of Image and Video Processing (Communications, Networking and Multimedia). Academic Press, Inc., Orlando, FL, USA, (2005)
2. Portilla, J., Strela, V., Wainwright, M.J., and Simoncelli, E.P.: Image Denoising Using Scale Mixtures of Gaussians in the Wavelet Domain. IEEE Trans. on Image Processing. 12, 1338–1351 (2003)
3. Buades, A., Coll, B., and Morel, J.M.: Nonlocal Image and Movie Denoising. Int. J. of Computer Vision, 76, 123–139 (2008)
4. Aharon, M., Elad, M., and Bruckstein, A.: K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. IEEE Trans. on Sinal Processing, 11, 4311–4322 (2006)
5. Blu, T., and Luisier, F.: The SURE-LET Approach to Image Denoising. IEEE Trans. on Image Processing, 16, 2778–2786 (2007)
6. Dabov, K., Foi, A., Katkovnik, V., and Egiazarian, K.: Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. IEEE Trans. on Image Processing. 16, 2080–2095 (2007)
7. Zlokolica, V., Pizurica, A., and Philips, W.: Wavelet-Domain Video Denoising Based on Reliability Measures. IEEE Trans. on Cir. and Sys. for Video Tech., 16, 993–1007 (2006)
8. Varghese, G., and Wang, Z.: Video Denoising Based on a Spatiotemporal Gaussian Scale Mixture Model. IEEE Trans. on Cir. and Sys. for Video Tech., 20, 1032–1040 (2010)
9. Luisier, F. and Blu, T. and Unser, M.: SURE-LET for Orthonormal Wavelet-Domain Video Denoising. IEEE Trans. on Cir. and Sys. for Video Tech., 20, 913–919 (2010)
10. Buades, A., Coll, B., Morel J.M., and Matemàtiques D.: Denoising Image Sequences does not Require Motion Estimation. Proc. of the IEEE Conf. on Advanced Video and Signal Based Surveillance, 70–74 (2005)
11. Dabov, K., Foi, A., and Egiazarian, K.: Video Denoising by Sparse 3D Transform-Domain Collaborative Filtering. Proc. of the 15-th Euro. Signal Proc. Conf., Poland, Sep. (2007)
12. Ozkan, M.K., Sezan, M.I., and Tekalp, A.M.: Adaptive Motion-compensated Filtering of Noisy Image Sequences. IEEE Trans. on Cir. and Sys. for Video Tech., 3, 277–290 (1993)
13. Arce, G.R.: Multistage Order Statistic Filters for Image Sequence Processing. IEEE Trans. on Signal Processing, 39, 1146–1163 (1991)
14. Kim J., and Woods J.W.: Spatiotemporal Adaptive 3-D Kalman Filter for Video. IEEE Trans. on Image Processing, 6, 414–424 (1997)
15. Brailean, J.C., and Katsaggelos, A.K.: Recursive Displacement Estimation and Restoration of Noisy-blurred Image Sequences. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, 5, 273–276 Apr. (1993)
16. Selesnick W.I., and Li K.Y.: Video Denoising using 2D and 3D Dualtree Complex Wavelet Transforms. Proc. SPIE, Wavelets: Applications in Signal and Image Processing X, 5207, 607–618 Nov. (2003)
17. Protter, M., and Elad, M.: Image Sequence Denoising via Sparse and Redundant Representations. IEEE Trans. on Image Processing, 18, 27–35 (2009)
18. Li X., and Yunfei Z.: Patch-based video processing: a variational Bayesian approach. IEEE Trans. on Cir. and Sys. for Video Tech., 19, 27–40 (2009)
19. Wang Z., Bovik A.C, Sheikh H.R., and Simoncelli E.P.: Image Quality Assessment: From Error Visibility to Structural Similarity. IEEE Trans. on Image Processing, 13, 600–612 (2004)

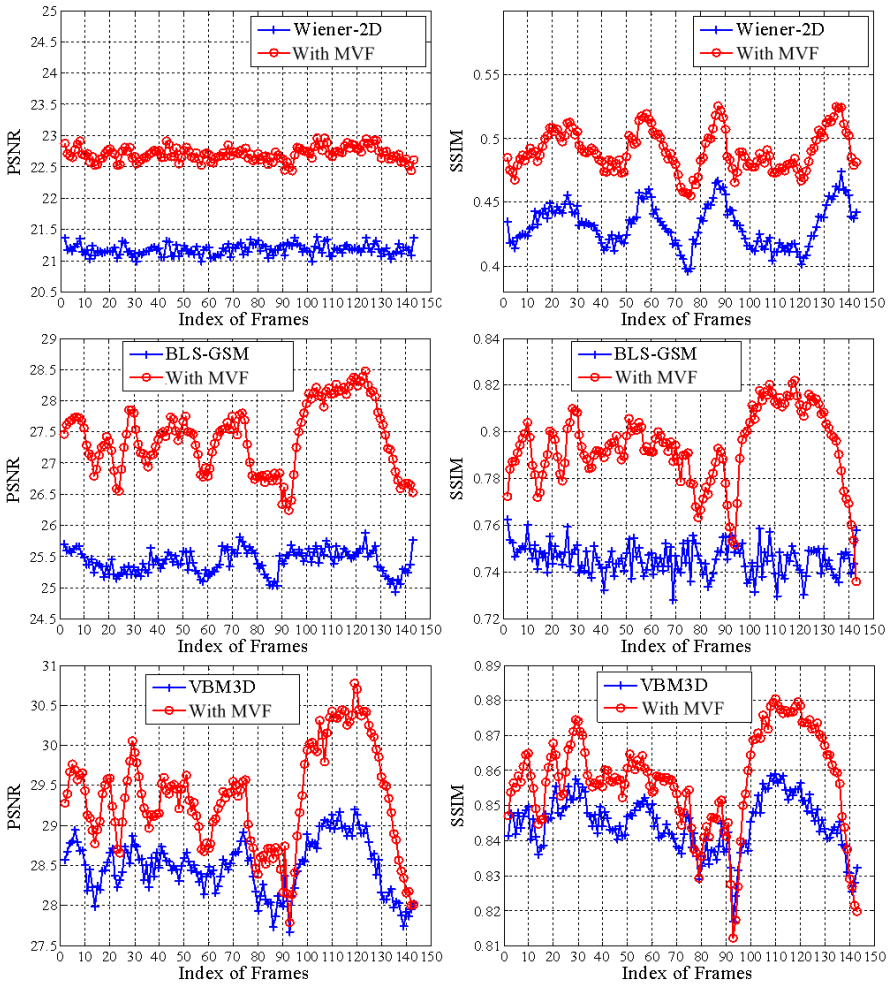



**Fig. 2.** PSNR and SSIM comparisons as functions of frame number for "Foreman" sequence. Noise level $\sigma = 50$.

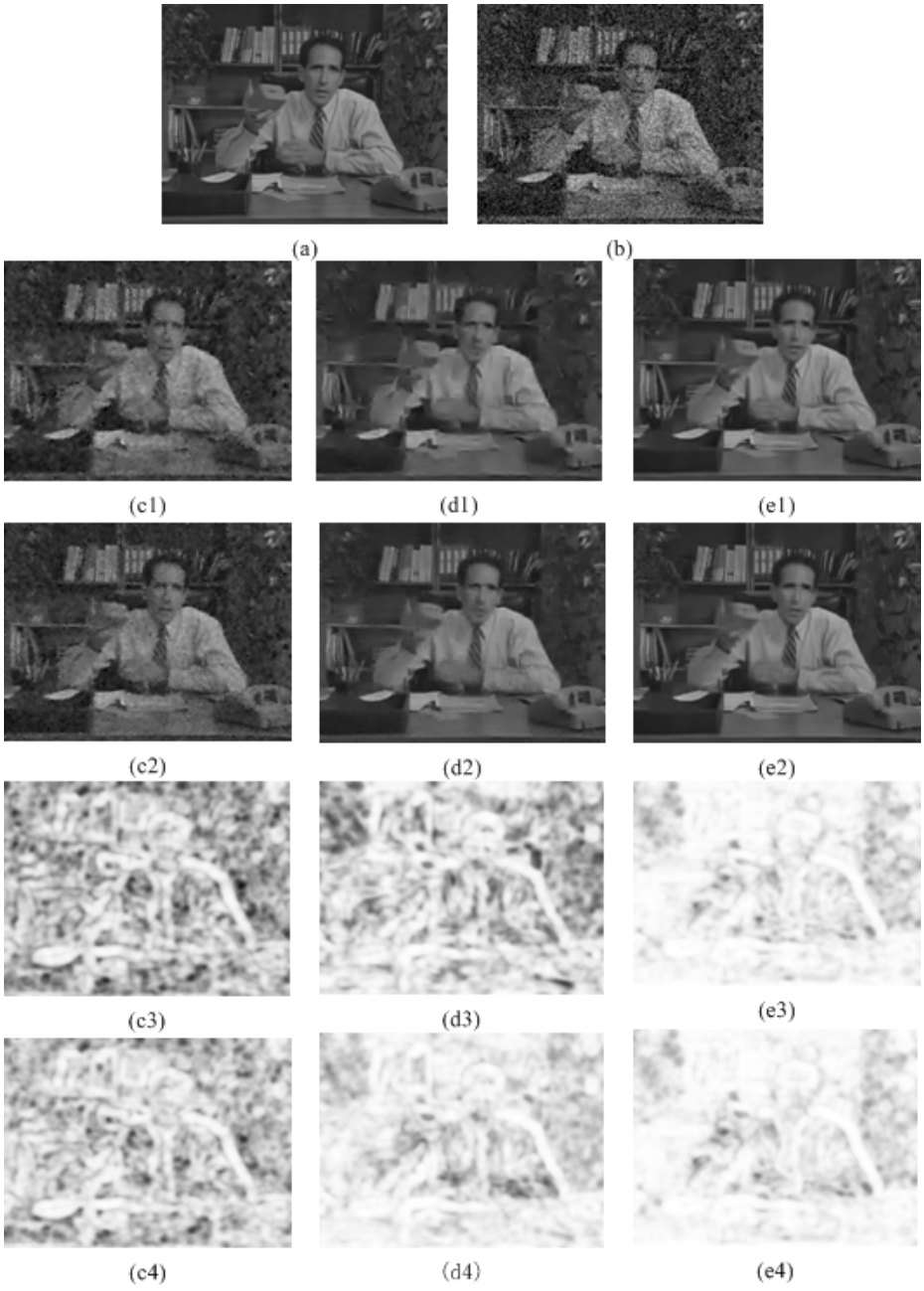



**Fig. 3.** (a): One frame extracted from original "Salesman" sequence; (b): Corresponding noisy frame with $\sigma = 50$; (c1) to (e1): Wiener2D, BLS-GSM, and VBM3D denoised frames; (c2) to (e2): Wiener2D, BLS-GSM, and VBM3D denoised frames with optimal MVF; (c3) to (e3): SSIM quality maps for (c1) to (e1); (c4) to (e4): SSIM quality maps for (c2) to (e2).