Polyview Fusion: A Strategy to Enhance Video-Denoising Algorithms

Kai Zeng, *Student Member, IEEE*, and Zhou Wang, *Member, IEEE*

***Abstract*—We propose a simple but effective strategy that aims to enhance the performance of existing video denoising algorithms, i.e., polyview fusion (PVF). The idea is to denoise the noisy video as a 3-D volume using a given base 2-D denoising algorithm but applied from multiple views (front, top, and side views). A fusion algorithm is then designed to merge the resulting multiple denoised videos into one, so that the visual quality of the fused video is improved. Extensive tests using a variety of base video-denoising algorithms show that the proposed PVF method leads to surprisingly significant and consistent gain in terms of both peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) performance, particularly at high noise levels, where the improvement over state-of-the-art denoising algorithms is often more than 2 dB in PSNR.**

***Index Terms*—Image fusion, polyview, video denoising, video quality enhancement.**

I. Introduction

Digital video has become ubiquitous and indispensable in our everyday lives. Video signals are subject to noise contaminations during acquisition and transmission. It is highly desirable to remove/reduce the noise in the video signal (or "denoise" the video), to enhance perceived video quality, and to help improve the performance of subsequent processes, such as compression; segmentation; and object detection, recognition, and tracking [1].

Existing video denoising algorithms may be classified into 2-D and 3-D approaches. The simplest 2-D approaches denoise the video frame by frame by employing 2-D still-image denoising algorithms, for which well-known and state-of-the-art algorithms include spatially adaptive 2-D Wiener filtering (Wiener-2-D) [2], Bayes' least-square estimation based on the Gaussian scale mixture model (BLS-GSM) [3], nonlocal means [4], K-SVD [5], Stein's unbiased risk estimator-linear expansion of threshold (SURE-LET) [6], and block matching and 3-D transform shrinkage (BM3D) [7]. Since the correlation between neighboring frames is completely ignored, these methods do not make use of all available information. Advanced 2-D approaches explore the correlation between adjacent frames. By incorporating motion compensation processes, state-of-the-art image denoising algorithms were extended to video, leading to the ST-GSM [8], and video SURE-LET [9] algorithms. In [10], multiple similar patches in neighboring frames that may not reside along a single trajectory are found. This is followed by transform- and shrinkage-based denoising procedures. In the video BM3D (VBM3D) method [11], similar patches in both intra- and interframes are aggregated before a two-stage 3-D collaborative filtering algorithm is employed for noise removal. Three-dimensional video-denoising schemes treat video sequences as 3-D volumes. These methods may operate in the space-time domain by adaptive weighted local averaging [12], 3-D order-statistic filtering [13], 3-D Kalman filtering [14], or 3-D Markov-model-based filtering [15]. They may also be applied in the 3-D transform domain, where soft/hard thresholding or Bayesian estimation is employed to eliminate noise, followed by an inverse 3-D transform that brings the signal back to the space-time domain [16]. Recently, 3-D-patch-based methods that achieved highly competitive denoising performance have also been investigated [17], [18].

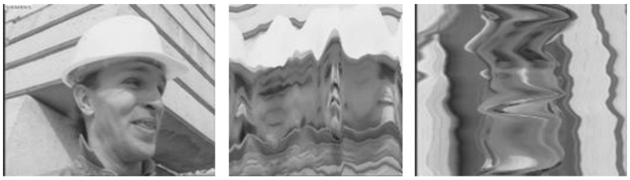


Fig. 1. Video signal observed from (a) front view, (b) side view, and (c) top view.

To make best use of all available information, an ideal video-denoising algorithm would need to operate in 3-D. However, in the presence of significant motion, direct space-time 3-D filtering or 3-D transform-based approaches are difficult to effectively cover all motion-associated image content within local regions. On the other hand, 2-D denoising algorithms that use intra- and/or interframe information may be more efficient, but their performance is restricted by not taking full advantage of the neighboring pixels in all three dimensions simultaneously.

Here, we propose to a polyview fusion (PVF) scheme, where the same noisy video volume is denoised using 2-D approaches but from three different views, i.e., front, top, and side views. This is followed by a normalization procedure inspired by the structural similarity (SSIM) measure [19] and a fusion process based on local variance. By doing so, the advantage of 2-D approaches is utilized, whereas each pixel is denoised by its neighboring pixels from all three dimensions, thus providing a compromise between 2-D and 3-D approaches.

II. Proposed Method

A digital video signal can be expressed as a 3-D function $f(u,v,t)$ discrete in both space and time, where u and v are the horizontal and vertical spatial indices, respectively, and t is the time index. A video is typically played along the time axis. At any time instance $t=t_0$, the video is displayed as a 2-D front-view image $f(u,v,t_0)$, and the image changes for different values of t_0. If we consider a video signal as 3-D volume data, then it can also be viewed from the side or the top. This gives two other ways to play the same video, i.e., a sequence of 2-D top-view images $f(u_0,v,t)$ for different values of u_0 and a sequence of 2-D side-view images $f(u,v_0,t)$ for different values of v_0. An example is shown 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 that is contaminated by additive independent zero-mean noise n with standard deviation σ_n, resulting in a noisy signal

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

A video-denoising operator $D(\cdot)$ takes the noisy observation y and maps it to an estimator of x, i.e.,

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

Manuscript received April 06, 2011; revised July 11, 2011 and September 14, 2011; accepted September 24, 2011. Date of publication October 06, 2011; date of current version March 21, 2012. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada and the Ontario Early Researcher Award Program. Some preliminary results (with different fusion approaches) of this work were presented at the International Conference on Image Analysis and Recognition, Burnaby, BC, Canada, June 2011. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Jose M. Bioucas-Dias.

The authors are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: kzeng@engmail.uwaterloo.ca; zhouwang@ieee.org).

Digital Object Identifier 10.1109/TIP.2011.2170699

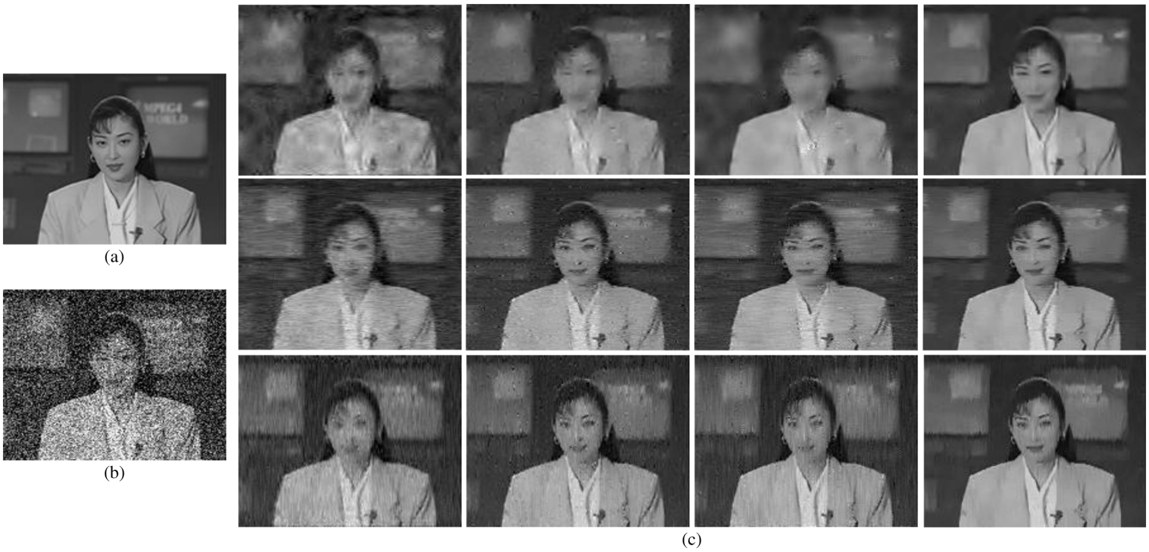


Fig. 2. Denoised frames from three different views using different denoising algorithms. (a) Original frame. (b) Noisy frame with $\sigma_n = 50$. (c) (Left to right) Denoised frames by SURE-LET, BLS-GSM, K-SVD, and VBM3D. (Top to bottom) Denoised frames from front, top, and side views, respectively.

so that the difference between x and $\hat{x}$ is as small as possible.

The proposed PVF method relies on a base video-denoising algorithm. The base denoiser is applied to the same noisy signal y but from different views, resulting in multiple versions of denoised signals, i.e.,

$$z_i = D_i(y), \qquad i = 1, \dots, N. \tag{3}$$

In our current work, $N = 3$ because 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. Fig. 2 shows sample denoised frames created by applying different denoising algorithms from three different views. It can be observed that the denoised frames have quite different appearances, even when the same denoising method is applied (from different views). Some image structures preserved in one of the views may be missing in the other views, and some artifacts that appear in one view may also be absent from another view. This suggests that the denoised frames from different views could complement each other, and fusing them (in appropriate ways) could potentially improve the denoising result. Let $\mathbf{z} = [z_1, z_2, \dots, z_N]^T$ be a vector that contains all denoised results. Then, the final denoised signal $\hat{x}$ is obtained by applying a fusion operator $F(\cdot)$ to $\mathbf{z}$, i.e.,

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

In the case that the base denoisers D_is are predetermined, the remaining task is to define fusion rule F.

Before the fusion step, however, we first apply a normalization process to each z_i. This is inspired by the SSIM index [19], which has been shown to be a much better predictor of the perceived image quality than the mean squared error (MSE). Given two image patches, the SSIM index separates the similarity measure into the luminance, contrast, and structure components. Since the luminance and contrast (measured by mean intensity and standard deviation, respectively) of an image patch can be adjusted freely without changing its structure, we can improve the SSIM measure by adapting the luminance and contrast of each z_i to match those of x while maintaining its structure. Specifically, we compute

$$\hat{z}_i = \frac{\sigma_x}{\sigma_{z_i}}(z_i - \mu_{z_i}) + \mu_x \tag{5}$$

where μ_x and μ_{z_i}, and σ_x and σ_{z_i}, denote the means and standard deviations of x and z_i, respectively. The computation in (5) requires the mean and standard deviation of x, which is not available. Fortunately, we can estimate them from noisy signal y using (1) and known noise properties (independence, zero mean, and known standard deviation) by

$$\mu_x = \mu_y \quad \text{and} \quad \sigma_x = \sqrt{\sigma_y^2 - \sigma_n^2} \tag{6}$$

where μ_y and σ_y^2 are the mean and variance of y, respectively.

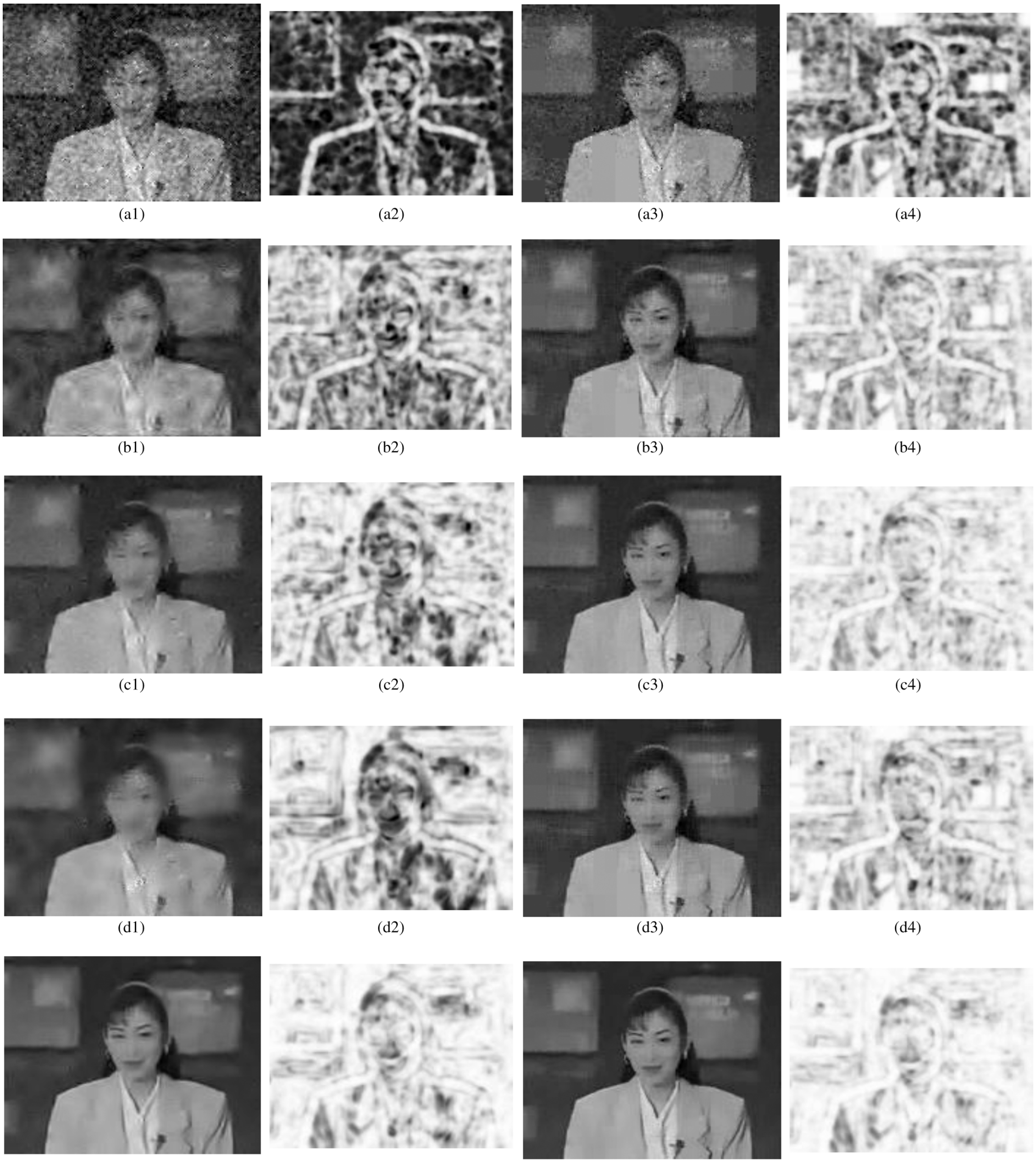
TABLE I
SRCC BETWEEN LOCAL VARIANCE AND PSNR FOR $\sigma_n = 50$

	SURE-LET	BLS-GSM	K-SVD	VBM3D
Akiyo	0.436	0.658	0.718	0.747
Carphone	0.316	0.498	0.596	0.559
Mobile	0.645	0.882	0.891	0.748
Foreman	0.321	0.579	0.537	0.590
Miss America	0.288	0.418	0.470	0.581
Mother Daughter	0.439	0.721	0.746	0.820
News	0.566	0.767	0.779	0.772
Salesman	0.734	0.769	0.788	0.820
Suzie	0.291	0.458	0.531	0.420

Our fusion rule is based on variance weighted averaging, which can be expressed as

$$\hat{x} = \frac{\sum_{i=1}^{N} \sigma_{z_i}^2 \hat{z}_i}{\sum_{i=1}^{N} \sigma_{z_i}^2}. \tag{7}$$

This is determined by our empirical studies on the relationship between the variance and the quality of denoised video patches using state-of-the-art video-denoising algorithms. Specifically, for three given 3-D patches denoised by the same video denoising algorithm but from three different views, we compute their corresponding variances and PSNR values between the denoised and original patches. We then calculate the Spearman rank-order correlation coefficient (SRCC) between the three variance and three PSNR values. Table I shows the average SRCC values (over all patches) for nine video sequences denoised with four denoising algorithms. It can be seen that, although a fairly large variations are observed (depending on both denoising algorithm and video sequence), the correlations are all positive. This



(a1) (a2) (a3) (a4)

(b1) (b2) (b3) (b4)

(c1) (c2) (c3) (c4)

(d1) (d2) (d3) (d4)

(e1) (e2) (e3) (e4)

Fig. 3. Comparison of one denoised frame from the "Akiyo" sequence with and without PVF. In the SSIM quality maps, brighter pixels indicate higher SSIM values and, thus, better quality. (a1)–(e1) Wiener2-D, SURE-LET, BLS-GSM, K-SVD, and VBM3D denoised frames without PVF. (a2)–(e2) SSIM quality maps for (a1)–(e1). (a3)–(e3) Wiener2-D, SURE-LET, BLS-GSM, K-SVD, and VBM3D denoised frames with PVF. (a4)–(e4): SSIM quality maps for (a3)–(e3).

suggests that the patches of larger variances tend to have better image quality, thus justifying variance-based weighting.

III. Experimental Results

The proposed approach is tested on publicly available video sequences, which contain various content and rich motion styles. The sequences are of size 144 × 176 × 144 and are contaminated by independent zero-mean white Gaussian noise, where the standard deviation of the noise covers a wide range between 10 and 100. After the noisy sequences are denoised using a base denoiser along three different views, the noisy and denoised sequences are divided into 16 × 16 × 16 nonoverlap 3-D patches, within which sample means and variances are computed and employed in the normalization and fusion processes described in Section II. The choices of nonoverlapping patches and size 16 are based on compromises between the denoising performance and complexity. In our simulations, there is no clipping of our-of-range values in the noise contamination and denoising processes.

All test sequences are in YCbCr 4:2:0 format, and only the denoising results of the luma channel are reported here. Two objective criteria PSNR and SSIM are employed to evaluate the quality of the denoised video. Assume that x and y are the noise-free and denoised images, respectively, and L is the dynamic range of intensity values. Then

$$\text{PSNR}(x, y) = 10 \log_{10} \left(\frac{L^2}{\text{MSE}(x, y)} \right). \quad (8)$$

TABLE II
PSNR AND SSIM COMPARISONS FOR SIX VIDEO-DENOISING ALGORITHMS WITH AND WITHOUT PVF

Video Sequence	*Akiyo*					*Carphone*					*Mobile*				
Noise std (σ_n)	10	15	20	50	100	10	15	20	50	100	10	15	20	50	100
PSNR Results (dB)															
Wiener-2D	33.22	30.38	28.34	21.56	15.95	32.67	29.85	27.86	21.34	15.86	29.79	26.92	25.00	19.50	15.11
with PVF	35.02	32.51	30.80	25.82	22.58	34.20	31.70	29.99	24.87	21.38	30.42	27.71	25.91	21.03	18.03
SURE-LET	34.38	31.95	30.32	25.53	22.06	33.70	31.25	29.60	24.68	21.12	29.68	26.94	25.17	20.54	17.93
with PVF	38.20	35.88	34.19	29.33	25.83	35.73	33.63	32.23	28.01	24.74	30.56	27.99	26.30	21.84	19.34
BLS-GSM	36.11	33.70	32.07	27.29	24.33	35.32	32.99	31.38	26.43	23.14	30.53	27.91	26.14	21.19	18.45
with PVF	40.13	37.81	36.20	31.22	27.76	37.11	35.05	33.63	29.33	25.26	31.61	29.18	27.55	22.86	20.04
K-SVD	36.41	33.96	32.22	26.77	23.60	35.84	33.67	32.04	26.07	22.24	30.29	27.72	26.01	20.99	17.92
with PVF	40.17	37.72	35.95	29.91	25.95	37.46	35.44	33.96	28.72	24.95	32.02	29.58	27.93	22.73	19.14
ST-GSM	40.58	38.25	36.51	30.83	26.67	37.66	35.70	34.29	29.64	26.00	32.58	30.24	28.64	23.91	20.69
with PVF	42.04	39.79	38.18	33.07	29.28	37.93	36.00	34.66	30.62	27.60	32.95	30.65	29.09	24.48	21.41
VBM3D	42.00	39.73	37.87	30.76	24.38	38.52	36.65	35.35	29.81	23.32	33.19	31.02	29.47	22.60	18.44
with PVF	42.32	40.06	38.35	32.66	27.13	38.52	36.66	35.38	30.99	26.00	33.57	31.48	29.98	23.65	19.75
SSIM Results															
Wiener-2D	0.877	0.788	0.701	0.363	0.165	0.885	0.803	0.723	0.407	0.205	0.934	0.883	0.831	0.583	0.360
with PVF	0.917	0.864	0.814	0.615	0.470	0.923	0.876	0.830	0.634	0.477	0.945	0.905	0.864	0.664	0.470
SURE-LET	0.920	0.879	0.841	0.665	0.474	0.921	0.881	0.845	0.673	0.488	0.926	0.875	0.826	0.603	0.404
with PVF	0.964	0.944	0.924	0.809	0.656	0.950	0.927	0.906	0.801	0.701	0.942	0.903	0.865	0.694	0.526
BLS-GSM	0.952	0.924	0.898	0.765	0.636	0.952	0.927	0.903	0.773	0.630	0.944	0.904	0.860	0.624	0.410
with PVF	0.978	0.965	0.952	0.872	0.753	0.965	0.948	0.932	0.844	0.732	0.958	0.930	0.901	0.737	0.556
K-SVD	0.954	0.926	0.899	0.748	0.607	0.954	0.933	0.911	0.766	0.599	0.940	0.901	0.859	0.603	0.347
with PVF	0.975	0.959	0.942	0.825	0.665	0.964	0.947	0.930	0.820	0.677	0.960	0.934	0.909	0.733	0.506
ST-GSM	0.980	0.969	0.957	0.882	0.766	0.966	0.953	0.940	0.873	0.775	0.964	0.942	0.920	0.791	0.609
with PVF	0.984	0.976	0.968	0.913	0.816	0.968	0.955	0.943	0.882	0.792	0.966	0.946	0.926	0.810	0.652
VBM3D	0.985	0.976	0.965	0.874	0.616	0.972	0.961	0.951	0.875	0.630	0.970	0.951	0.931	0.715	0.404
with PVF	0.986	0.978	0.968	0.904	0.697	0.972	0.962	0.952	0.893	0.703	0.973	0.956	0.938	0.772	0.526
Video Sequence	*Foreman*					*Miss America*					*Football*				
PSNR Results (dB)															
Wiener-2D	32.22	29.49	27.56	21.18	15.78	34.35	31.35	29.17	21.92	16.08	31.21	28.68	26.87	20.77	15.64
with PVF	33.16	30.65	28.93	23.79	20.41	37.49	35.23	33.63	28.59	24.98	31.22	28.71	26.97	21.48	16.88
SURE-LET	32.88	30.53	28.93	24.24	20.92	37.11	34.94	33.41	28.33	23.89	31.71	29.33	27.71	23.08	20.25
with PVF	34.64	32.43	30.93	26.55	23.48	39.88	37.93	36.55	32.03	28.26	31.74	29.38	27.80	23.47	21.20
BLS-GSM	34.20	31.90	30.31	25.43	22.19	38.68	36.56	35.09	30.60	27.43	32.33	30.19	28.77	24.38	21.65
with PVF	35.89	33.73	32.24	27.66	24.35	41.14	39.16	37.76	33.30	29.82	32.38	30.20	28.79	24.44	22.00
K-SVD	35.32	33.13	31.50	25.58	21.44	38.56	36.34	34.73	29.56	25.82	32.35	30.13	28.69	24.18	21.00
with PVF	36.56	34.42	32.84	27.16	23.13	40.72	38.60	37.03	31.88	27.97	32.33	30.09	28.60	24.12	21.51
ST-GSM	37.01	34.92	33.47	28.74	25.29	41.38	39.33	37.87	32.78	28.80	32.09	29.91	28.51	24.72	22.57
with PVF	37.27	35.20	33.76	29.22	26.00	42.25	40.28	38.87	34.25	30.74	32.35	30.19	28.80	25.07	22.68
VBM3D	37.37	35.51	34.13	28.46	22.44	41.93	40.18	38.83	33.50	26.56	32.90	30.72	29.32	25.01	21.39
with PVF	37.70	35.84	34.49	29.41	24.38	42.37	40.60	39.28	34.62	29.08	32.90	30.73	29.35	25.05	21.94
SSIM Results															
Wiener-2D	0.888	0.813	0.739	0.432	0.220	0.848	0.737	0.633	0.275	0.107	0.838	0.755	0.680	0.380	0.184
with PVF	0.911	0.856	0.802	0.578	0.414	0.935	0.899	0.865	0.709	0.567	0.843	0.764	0.692	0.399	0.199
SURE-LET	0.909	0.867	0.829	0.667	0.501	0.941	0.914	0.889	0.735	0.502	0.870	0.806	0.754	0.563	0.402
with PVF	0.936	0.906	0.879	0.749	0.599	0.965	0.950	0.934	0.841	0.705	0.871	0.808	0.756	0.575	0.429
BLS-GSM	0.938	0.910	0.884	0.746	0.592	0.958	0.940	0.922	0.840	0.746	0.869	0.817	0.777	0.619	0.481
with PVF	0.953	0.931	0.910	0.796	0.649	0.973	0.961	0.949	0.885	0.793	0.875	0.823	0.780	0.624	0.495
K-SVD	0.944	0.921	0.897	0.752	0.576	0.957	0.937	0.918	0.817	0.688	0.874	0.813	0.769	0.606	0.471
with PVF	0.954	0.933	0.911	0.778	0.594	0.968	0.952	0.936	0.837	0.701	0.879	0.823	0.778	0.595	0.439
ST-GSM	0.960	0.942	0.925	0.861	0.744	0.977	0.967	0.958	0.903	0.824	0.878	0.824	0.781	0.633	0.516
with PVF	0.960	0.942	0.926	0.865	0.750	0.978	0.970	0.961	0.907	0.830	0.883	0.826	0.781	0.638	0.522
VBM3D	0.961	0.947	0.934	0.844	0.601	0.976	0.968	0.959	0.901	0.670	0.887	0.829	0.787	0.639	0.458
with PVF	0.962	0.948	0.935	0.858	0.648	0.978	0.970	0.962	0.915	0.703	0.888	0.832	0.790	0.641	0.460

The SSIM value between two image patches is computed as

$$\text{SSIM}(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \tag{9}$$

where C_1 and C_2 are small positive constants to avoid instability when the means and variances are close to zero. This computation is applied at each location in the image using a sliding window that moves pixel by pixel across the image, resulting in an SSIM quality map, as demonstrated in Fig. 3. The SSIM value between two images is then computed as the mean of the SSIM map. Both PSNR and SSIM were computed on a frame-by-frame basis along the temporal direction and then averaged over all frames to yield the PSNR and SSIM values of the whole sequence.

We test the proposed PVF method with diverse types of based denoisers, including the Wiener-2-D (using Matlab Wiener2 function), SURE-LET [9], BLS-GSM [3], K-SVD [5], ST-GSM [8], and VBM3D [11] algorithms. The denoising computations are conducted using the default parameter settings of the code available to the public at [2] and [20]–[24], respectively. Table II shows PSNR and SSIM comparisons. Due to space limit, here, we only report the results of six sequences at five noise levels using six base denoising methods with and without PVF. The average improvement over nine test sequences is given in Table III. It can be observed that the proposed PVF approach leads to consistent performance gain over all base denoising algorithms, for all test video sequences, and at all noise levels. The gain is particularly significant at high noise levels, where the PSNR improvement could be 2 dB or higher upon the best video-denoising algorithms reported in the literature. We also observe that the gain is reduced for video sequences with significant amount of large motion.

Fig. 3 provides visual comparisons of the denoising results of one frame extracted from "Akiyo" sequence, for which the original and noisy frames are given in Fig. 2(a) and (b), respectively. Visual quality improvement by the proposed PVF approach can be easily discerned at various locations in the denoised frames. The observation is also verified by the SSIM quality map, which provides a useful indicator of local image quality variations.

Furthermore, another experiment has been conducted to measure the computational complexity of the PVF operation and how it compares

TABLE III
AVERAGE PSNR AND SSIM IMPROVEMENT OVER ALL TEST SEQUENCES

Noise std (σ_n)	10	15	20	50	100
PSNR Improvement (dB)					
Wiener-2D	1.428	1.473	1.905	3.218	4.973
SURE-LET	1.882	2.050	2.143	2.472	2.780
BLS-GSM	1.848	1.980	2.068	2.248	2.007
K-SVD	1.748	1.817	1.853	1.895	1.772
ST-GSM	0.582	0.643	0.678	1.015	1.268
VBM3D	0.245	0.259	0.311	1.038	1.958
SSIM Improvement					
Wiener-2D	0.034	0.064	0.093	0.193	0.226
SURE-LET	0.024	0.036	0.047	0.094	0.141
BLS-GSM	0.048	0.023	0.030	0.065	0.081
K-SVD	0.013	0.020	0.026	0.048	0.049
ST-GSM	0.002	0.003	0.010	0.012	0.021
VBM3D	0.001	0.002	0.003	0.023	0.061

TABLE IV
COMPUTATIONAL COMPLEXITY ANALYSIS

Base denoiser	One view denoising time (second)	PVF time (second)	PVF (%)
Wiener-2D	1.353	0.1813	4.276
SURE-LET	27.13		0.222
BLS-GSM	140.3		0.043
K-SVD	684.5		0.009
ST-GSM	1748		0.004
VBM3D	8.791		0.683

with the complexity of the base denoisers. The results are reported in Table IV, where the speed is measured in seconds based on Matlab implementations of the algorithms on a computer with Intel Core2 Duo CPU E8600 processor at 3.33 GHz. Although the implementations are not speed optimal, they give us a general idea about the amount of added complexities due to the PVF process. As can be observed, generally, the PVF procedure is of low complexity relative to the base denoising algorithms. The percentage of time spent on PVF ranges from 0.004% to 4.276% of the overall denoising process (where a base denoiser needs to be run three times and, thus, the overall process increases the computational cost by a factor of 3 or more). In conclusion, the complexity of the overall denoising algorithm mainly depends on the complexity of the base denoiser, and the PVF portion is mostly negligible.

IV. CONCLUSION AND DISCUSSION

A PVF approach is proposed to enhance video-denoising algorithms by fusing denoising results from multiple views. Our experiments demonstrate significant and consistent improvement over existing video-denoising methods. In practice, to apply PVF, one would need to store all video frames involved in the denoising and fusion processes in the memory. This may be a problem in practical systems, particularly when the video sequence is long. It is therefore preferable to divide long sequences into segments along the temporal direction and then denoise each segment independently. By adjusting the length of the segments, the memory requirement can be controlled.

In the future, better denoising results may be obtained by incorporating more advanced denoising algorithms or by improving the fusion method. Although our current implementation only fuses the denoising results by the same base denoiser applied along three views, the general PVF approach facilitates fusing the results of any finite number of denoising algorithms. Two issues are critical to the success of this approach. First, the denoising algorithms need to be complementary to each other. Second, the fusion algorithm needs to select the best denoising result among many or optimally assign weights to multiple denoising results. In our current experiment, we observe that 2-D approaches from different views tend to be more complementary to each other than 3-D approaches, which have already considered the dependencies between neighboring pixels from all directions. Since the structural regularities exhibited in the top and side views are substantially different from those in the front view (as can be observed in Fig. 2), it is preferable to use different denoising methods that are best suited to the corresponding views before fusing the results. Currently, no denoising algorithm specifically tuned to denoise from top and side views has been developed. This gives us another interesting topic for future study.

REFERENCES

[1] A. C. Bovik, *Handbook of Image and Video Processing (Communications, Networking and Multimedia)*. Orlando, FL: Academic, 2005.

[2] [Online]. Available: http://www.mathworks.com/help/toolbox/images/ref/wiener2.html

[3] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Trans. Image Process.*, vol. 12, no. 11, pp. 1338–1351, Nov. 2003.

[4] A. Buades, B. Coll, and J. M. Morel, "Nonlocal image and movie denoising," *Int. J. Comput. Vis.*, vol. 76, no. 2, pp. 123–139, Feb. 2008.

[5] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[6] T. Blu and F. Luisier, "The SURE-LET approach to image denoising," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2778–2786, Nov. 2007.

[7] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-D transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, Aug. 2007.

[8] G. Varghese and Z. Wang, "Video denoising based on a spatiotemporal Gaussian scale mixture model," *IEEE Trans. Circuits Syst. Video Tech.*, vol. 20, no. 7, pp. 1032–1040, Jul. 2010.

[9] F. Luisier, 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, Jun. 2010.

[10] A. Buades, B. Coll, J. M. Morel, and D. Matemàtiques, "Denoising image sequences does not require motion estimation," in *Proc. IEEE Conf. AVSS*, 2005, pp. 70–74.

[11] K. Dabov, A. Foi, and K. Egiazarian, "Video denoising by sparse 3-D transform-domain collaborative filtering," in *Proc. 15th Eur. Signal Process. Conf.*, Poznan, Poland, Sep. 2007, pp. 145–149.

[12] M. Ozkan, M. Sezan, and A. Tekalp, "Adaptive motion-compensated filtering of noisy image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, no. 4, pp. 277–290, Aug. 1993.

[13] G. Arce, "Multistage order statistic filters for image sequence processing," *IEEE Trans. Signal Process.*, vol. 39, no. 5, pp. 1146–1163, May 1991.

[14] J. Kim and J. W. Woods, "Spatio-temporal adaptive 3-D Kalman filter for video," *IEEE Trans. Image Process.*, vol. 6, no. 3, pp. 414–424, Mar. 1997.

[15] J. Brailean and A. Katsaggelos, "Simultaneous recursive displacement estimation and restoration of noisy-blurred image sequences," *IEEE Trans. Image Process.*, vol. 4, no. 9, pp. 1236–1251, Sep. 1995.

[16] I. W. Selesnick and K. Y. Li, "Video denoising using 2-D and 3-D dualtree complex wavelet transforms," in *Proc. SPIE, Wave.: Appl. Signal Image Process. X*, San Diego, CA, Nov. 2003, vol. 5207, pp. 607–618.

[17] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 27–35, Jan. 2009.

[18] 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, Jan. 2009.

[19] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[20] [Online]. Available: http://bigwww.epfl.ch/demo/suredenoising/index.html

[21] [Online]. Available: http://www4.io.csic.es/PagsPers/JPortilla/portada/software

[22] [Online]. Available: http://www.cs.technion.ac.il/~elad/software/

[23] [Online]. Available: https://ece.uwaterloo.ca/~z70wang/research/stgsm/STGSM.zip

[24] [Online]. Available: http://www.cs.tut.fi/~foi/GCF–BM3D/BM3D.zip