Video Denoising by Combining Kalman and Wiener Estimates

Rakesh Dugad and Narendra Ahuja*
Department of Electrical and Computer Engineering
Beckman Institute, University of Illinois, Urbana, IL 61801.
dugad@vision.ai.uiuc.edu

Abstract

This paper proposes a computationally fast scheme for denoising a video sequence. Temporal processing is done separately from spatial processing and the two are then combined to get the denoised frame. The temporal redundancy is exploited using a scalar state 1-D Kalman filter. A novel way is proposed to estimate the variance of the state noise from the *noisy* frames. The spatial redundancy is exploited using an adaptive edge-preserving Wiener filter. These two estimates are then combined using simple averaging to get the final denoised frame. Simulation results for the foreman, trevor and susie sequences show an improvement of 6 to 8 dB in PSNR over the noisy frames at PSNR of 28 and 24 dB.

1 Introduction

Noise gets added to video in the process of recording it. This problem is even more acute when converting from video on analog tapes to video in digital format. Noise is undesirable not only because it degrades the visual quality of the video but also because it degrades the performance of subsequent processing such as compression.

In [1] a spatiotemporal Kalman filter based approach to denoising video is described. Their approach requires a 3-D AR model for the the image sequence. Results are presented with the parameters of this model estimated from the original clean video sequence. Also, the dimensionality of the state vector at each pixel is high (consisting of a causal spatiotemporal support around each pixel). This increases the amount of computation as well as storage required for processing each frame.

In [2] a temporal motion compensated adaptive linear MMSE filter is proposed. The motion estimates are obtained using a robust algorithm proposed by Fogel [3]. The noisy pixel value and two pixels each in the past and future along the motion trajectory of this pixel are used to compute the local statistics. The denoised estimate is obtained using the same approach as used by Kuan [7] for spatial adaptive "linear" MMSE filtering (the final estimate is actually non-linear after incorporating these statistics). The motion estimation algorithm has high computational cost and four motion vectors (8 values) need to be stored for each pixel in addition to storing the frames.

In [4] an adaptive weighted averaging filter is proposed. This approach relies on the fact that averaging pixels in a spatiotemporal neighborhood (after motion compensation) of a given pixel gives good estimates if these pixels have nearly same intensity (except the variation due to noise). A pixel differing in intensity by more than a threshold (decided by noise variance) from the pixel of interest is considered an outlier and is not used in the averaging. Other pixels are weighed according to how much they differ from the current pixel value. This scheme is especially suited for low SNRs and abruptly changing scene content.

In [5] the minimum mean absolute difference (MAD1) block or the block at the same location (MAD0) in the previous (or next) frame is chosen as the temporal neighbor of current block depending on a threshold on MAD0/MAD1 and also on MAD0. The idea is to make sure that the match is not due to matching of noises in the two blocks in which case averaging (which is used to get the denoised block) would be ineffective.

A comprehensive review of video denoising techniques can be found in [6].

In our scheme, temporal and spatial processing for the current frame are done separately and then combined. The temporal redundancy is exploited using a 1-D Kalman filter. The state of the Kalman filter at each pixel is a scalar viz. pixel intensity value. The pixel at the previous instant in the motion trajectory of the current pixel is taken as the previous state. We describe a novel way of adaptively estimating the variance of state noise without resorting to the original clean frames. We use simple integer pixel block

*The support of the office of Naval Research under grant N00014-96-1-0502 is greatfully acknowledged.

0-7803-5467-2/99/ $ 10.00 © 1999 IEEE

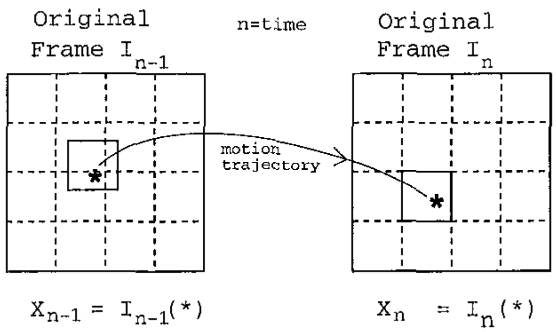


Figure 1: System Model for establishing the Kalman Filtering Equations.

matching for estimating the motion vectors. The spatial redundancy is exploited using the adaptive edge-preserving Wiener filter proposed by Kuan [7]. These two estimates are then combined using simple averaging to get the final denoised frame. Separation of temporal and spatial processing, simple block matching for motion estimation and scalar state of the Kalman filter make our scheme computationally fast. Also only the previous frame and two values per pixel of current frame (for Kalman filter) are required to be stored apart from the current frame. PSNR performance of our scheme for the foreman, trevor and susie sequences is comparable to the results in [1, 2, 4] at much less computational and memory requirements.

2 Temporal Kalman Filtering

Figure 1 shows the setup of our system. n denotes the discrete instants of time at which the video frames arrive. Consider tracking the motion of a point object denoted by an asterisk in the figure. The intensity of this object in the current (original) frame I_n is denoted by X_n and that in the previous (original) frame I_{n-1} is denoted by X_{n-1}. Ideally $X_n = X_{n-1}$ but due to error in estimating the motion trajectory or due to other reasons like change in illumination we have

$$X_n = X_{n-1} + U_n \tag{1}$$

where U_n represents the error or innovation in X_n compared to X_{n-1}. We shall call U_n as the motion noise and model it as zero mean Gaussian and independent from one pixel location to another and also independent over time. We shall see soon how the motion trajectory and statistics of U_n are computed.

Usually the original intensity values get corrupted by noise in the process of recording the video and hence our observation of the intensity X_n is given by

$$Y_n = X_n + V_n \tag{2}$$

where V_n represents the undesirable noise that we are interested in removing. V_n is also modeled as zero mean Gaussian and independent over space and time (AWGN) and independent of U_n. We note that Eqs. (1) and (2) can be set up at each pixel location in the current frame I_n.

Eqs. (1) and (2) are ideally suited for application of a Kalman filter to estimate the actual intensity (state) X_n given all of the past and current observation $Y^{(n)} = \{Y_0, \ldots, Y_n\}$. The Kalman filter actually gives an iterative procedure to compute the best (in MSE sense) linear estimate of X_n given $Y^{(n)}$. However to be able to apply the Kalman filter we need to know the motion trajectories and the following statistics for all pixels in the current frame: (a) $E[X_0]$ and $\mathrm{var}(X_0)$. For practical implementation we can take X_0 as Y_0 and $\mathrm{var}(X_0)$ as zero. However it is found experimentally that after about 20 frames the initial estimates are inconsequential : the Kalman filter reaches about the same PSNR irrespective of the initial estimates. (b) $\mathrm{var}(V_n)$. This is the variance of the undesirable recording noise. For the purpose of this paper we shall assume that this is known but it can be easily estimated. (c) $\mathrm{var}(U_n)$. This is computed as described next.

Before describing how the variance of motion noise is estimated, consider how the motion trajectories are determined. Divide the current frame into 8×8 blocks as shown in Fig. 1. Consider one such block say B_n in the current frame and find its motion compensated block (integer pixel) in the previous frame i.e. find the 8×8 block B_{n-1} in the previous frame that is closest to B_n in the mean square sense. Then a given pixel in B_n corresponds to the corresponding pixel in B_{n-1}. This gives the motion trajectory for all pixels in the current frame.

The motion noise arises due to error in estimating the true motion or due to change in illumination or scene change etc. From Eq. (1) we see that the motion noise represents the change in the intensity of the point object as it moves. We do not have access to the original frames and hence to alleviate the effect of recording noise we shall estimate $\mathrm{var}(U_n)$ as $(\mu_n - \mu_{n-1})^2$ where $\mu_n = \mathrm{mean}(B_n)$ is the maximum likelihood estimate of X_n assuming all X_n are same in block B_n and similarly for $\mu_{n-1} = \mathrm{mean}(B_{n-1})$. Then under some conditions $(\mu_n - \mu_{n-1})^2$ is the best estimate of $\mathrm{var}(U_n)$.

Now we have completely defined our system. To estimate X_n we apply the Kalman filtering equations. A further improvement is obtained by using the denoised version of previous frame (which is available when processing the current frame) for calculating the motion vectors for current frame.

3 Combining the Kalman and Wiener Estimates

We use the spatial domain adaptive edge-preserving Wiener filter proposed by Kuan [7] to exploit the spatial correlation in each video frame. The Wiener-denoised frames are then combined (as described below) with Kalman-denoised frames obtained as described in the previous section. We use a 3×3 window for implementing the Wiener filter. In most cases the Wiener filter does better than the Kalman filter. However one can outsmart both by combining them as described next.

Since the Kalman and Wiener filters employ independent algorithms and also work along spatial and temporal directions respectively it would reasonable to assume that the error due to the two filters are uncorrelated. The errors are zero mean since each is an unbiased estimator. We know that averaging two uncorrelated unbiased estimates of a signal gives an unbiased estimate which has 3 dB less MSE compared to the average of the MSEs of individual estimators.

Hence one way to combine the two estimates is to simply average the denoised frames obtained from Wiener and Kalman filters working independently. This simple scheme typically yields more than 1 dB improvement in the PSNR compared to either filter. This scheme will be referred to as "average" scheme.

Another way to combine the two estimates is to use Wiener denoised version of the current frame and denoised (according to this scheme) version of previous frame for estimating the motion vectors. The Kalman filtering scheme using these motion vectors will be referred to as "kalman-joint" scheme. These kalman-joint estimate and the Wiener estimate for the current frame are averaged to get the final denoised frame. This scheme will be referred to as the "joint" scheme.

4 Results

The following table gives the IPSNR (improvement in PSNR compared to the PSNR of corresponding noisy frame) values averaged over 10 frames for the foreman (FR), trevor (TR) and susie (SU) sequences at 28 dB ($\equiv$ std. dev. 10) and 24 dB ($\equiv$ std. dev. 16) PSNR of noisy frames. Kal, wnr, avg, jnt and kal-jnt refer to the kalman filtering scheme of Section 2, and the Wiener, average, joint and the kalman-joint schemes of Section 3. Note (1) The avg scheme does about 1 dB better over the best of kal and wnr (2) jnt does much better compared to avg especially at low PSNR (3) kal-jnt performs better compared to wnr at low PSNR. Figure 1 shows the IPSNR variation for the foreman sequence. Note that the kal-jnt scheme borrows knowledge from the wiener filter only to estimate the motion vectors; the updating is done using kalman filter. This supports the argument that a video sequence normally has much more temporal correlation compared to spatial correlation. How much we can exploit the temporal correlation depends largely on how accurate motion estimates we can get.

Figures 2(a)-2(f) show the results for the trevor sequence. Note that the Kalman solution retains much of the texture of the shirt-strips and the background compared to the Wiener solution even though it has much lower PSNR.

Figure 3 shows results for the susie sequence. Comparing figures 3(c) and 3(d) we see that the Kalman solution is visually better compared to Wiener solution. Also it retains more texture of the hair region compared to the Wiener solution.

Detailed results can also be found at *http://vision.ai.uiuc.edu/~dugad/draft/icip99denoise.html.*

Acknowledgement

We would like to thank Dr. Pierre Moulin for his help and advice regarding this work.

References

[1] J. W. Woods and J. Kim, *Motion Analysis and Image Sequence Processing*, ch. 12. 1993.

[2] M. I. Sezan, M. K. Ozkan, and S. V. Fogel, "Temporally adaptive filtering of noisy image sequences using a robust motion estimation algorithm," in ICASSP-91, pp. 2429–2432.

[3] S. Fogel, "Estimation of velocity vector field from time-varying image sequences," *CVGIP: Image Understanding*, vol. 53, pp. 253–287, May 1991.

[4] M. K. Ozkan, M. I. Sezan, and A. M. Tekalp, "Adaptive motion-compensated filtering of noisy image sequences," *IEEE Trans. CSVT*, vol. 3, pp. 277–290, Aug. 1993.

[5] J. M. Boyce, "Noise reduction of image sequences using adaptive motion compensated frame averaging," in *ICASSP-92*, vol. III, pp. 461–464.

[6] J. C. Brailean, R. P. Kleihorst, S. Efstratiadis, A. K. Katsaggelos, and R. L. Lagendijk, "Noise reduction filters for dynamic image sequences : A review," *Proc. IEEE*, vol. 83, pp. 1272–1292, Sept. 1995.

[7] D. T. Kuan, A. A. Sawchuk, T. C. Strand, and P. Chavel, "Adaptive noise smoothing filter for images with signal-dependent noise," *IEEE Trans. PAMI*, vol. 7, pp. 165–177, Mar. 1985.

Seq.	frames	PSNR	kal	wnr	avg	jnt	kal-jnt
FR	141-150	28	3.7	4.2	5.5	5.6	3.9
FR	141-150	24	4.9	5.0	6.5	6.9	5.8
TR	31-40	28	3.8	4.6	5.5	5.7	4.3
TR	31-40	24	4.7	5.8	6.6	7.2	6.1
SU	75-84	28	4.0	5.3	6.1	6.5	4.9
SU	75-84	24	4.8	6.1	6.9	7.7	6.4

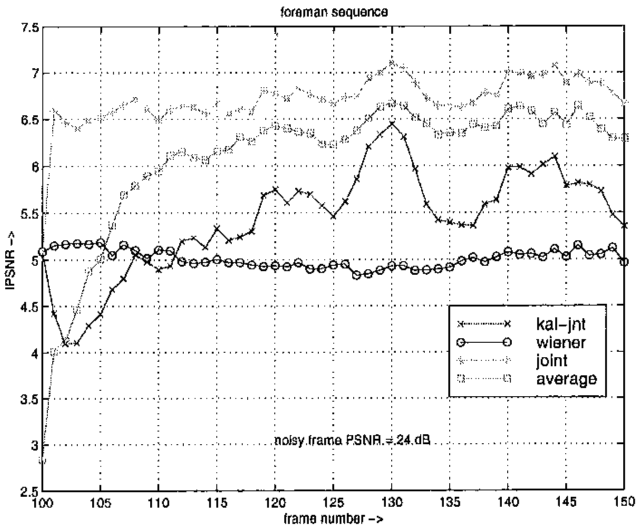
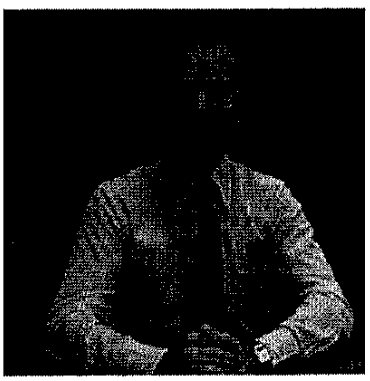
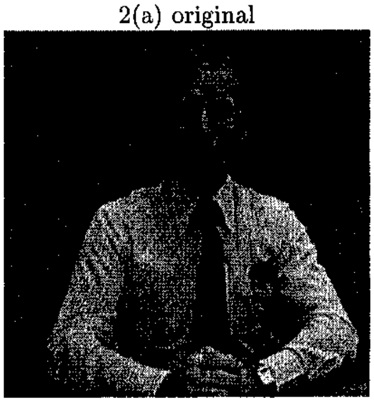


Figure 1: Foreman sequence at 24 dB PSNR.

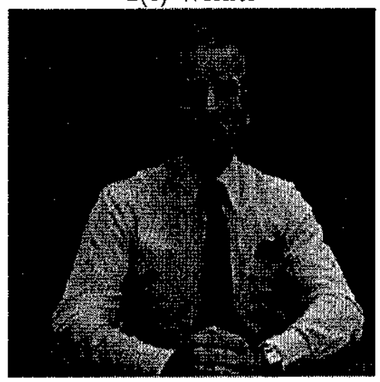


2(a) original

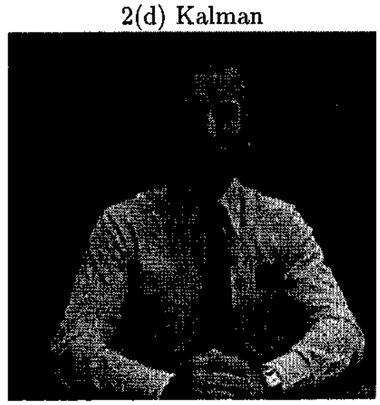


2(b) noisy (PSNR 28 dB)

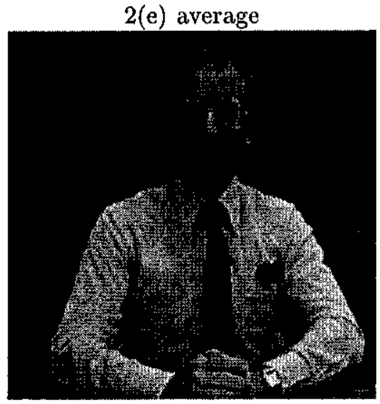
2(c) Wiener



2(d) Kalman



2(e) average



2(f) joint

(a) original (b) noisy (PSNR 24 dB)

(c) Wiener (d) Kalman

(e) average (f) joint

(g) original (h) Kalman (joint)

Figure 3: Frame 84 of susie sequence with 24 dB PSNR. (a) original (b) noise added (c) Wiener solution using a 3×3 window. (d) Kalman solution working independently of the Wiener solution. (e) average of (c) and (d). (f) using Kalman and Wiener filter jointly. (g) original (h) Kalman solution for the joint case.