PREMIER COLLOQUE IMAGE

Traitement, Synthèse, Technologie et Applications

BIARRITZ – Mai 1984 –

EDGE-IMPROVED KALMAN FILTERING FOR IMAGE RESTORATION

RESTAURATION D'IMAGE PAR FILTRAGE DE KALMAN TENANT COMPTE DES CONTOURS

H.J.TORK and A.N.VENETSANOPOULOS

Dept. of Electrical Engineering, University of Toronto, Toronto, Canada M5S 1A4

RESUME

SUMMARY

The use of Kalman-type filters for noise-degraded image restoration has been largely motivated by the possibility of on-line processing of two-dimensional (2-D) data, due to the recursive nature of such filters. Fast implementation methods are generally based on the assumption of image stationarity although most real images are in fact non-stationary in nature. As a result, Kalman filters tend to reduce image contrast and smooth the edges and the processed image may be visually inferior to the noisy image, although the noise content is effectively reduced. Adaptive techniques, while providing better results, generally require too much computation for suitability in an on-line environment and higher-order filters also demand excessive computation.

In this paper we consider first-order 2-D Kalman filters, both vector and scalar, in conjunction with simple block-type edge detectors. The edge detector output is added to the Kalman filter input in such a way that the filter response near any edges is boosted, while in relatively flat areas the filter is unaffected. Thus with little additional computational cost the subjective quality of the processed image is improved, particularly when the amount of noise in the edge weights is small.

All filter parameters are determined from small-dimensional matrices, requiring relatively little storage and computation and allowing efficient filter implementation. Results are presented for vector and scalar filters based on both non-symmetric half-plane and quarter-plane image models.

EDGE-IMPROVED KALMAN FILTERING FOR IMAGE RESTORATION

Restauration d'images par filtrage de Kalman tenant compte des contours

H.J.Tork and A.N.Venetsanopoulos

I. INTRODUCTION

The reduction of observation noise in image data is a problem which has received considerable attention in recent years. The filtering techniques considered may be broadly classified as recursive or non-recursive, of which recursive filters have the advantages of requiring less storage and having greater potential for on-line processing. Because of its proven utility in one-dimensional (1-D) applications, much of the research has been devoted to extending linear Kalman filtering concepts to 2-D.

For the efficient implementation of a 2-D Kalman filter (KF), the image statistics are generally assumed to be stationary, greatly simplifying the image modeling procedure. However, most images are in fact non-stationary [1] and are only partially described by the second-order statistics on which the KF is based. Consequently, such filters tend to smooth the image, blurring edges and reducing contrast such that the subjective quality is degraded. Ingle et al. [1] have achieved better results by segmenting the image into blocks and filtering each block independently of the others. While allowing some adaptation to the local image statistics, the segmentation is somewhat arbitrary and smoothing of edges still occurs. Earlier work by Nahi and Habibi [2] involves edge detection techniques to separate the image into two regions, object and background, and filtering each region independently. Such a method becomes quite complex for more general images in which many different regions can be distinguished. Biemond and Gerbrands [3] apply edge information extracted from the noisy image to the control input of a 1-D row-wise scan-ordered KF such that the filter responds more quickly to the presence of an edge. This method is most useful for images having mostly vertical edges and subjectively improved results have been achieved. In this paper we follow the same approach in the 2-D context, with a view towards improving the edge-response of an arbitrary first-order 2-D KF. The first-order restriction allows relatively fast and possibly on-line implementation.

In part II we briefly consider the edge-detection methods used, and in part III we present the edge-improved filtering algorithms for both vector and scalar filters. Results of processing are shown in part IV, followed by conclusions.

II. EDGE-DETECTION

For our edge-improvement scheme, we must first apply a suitable edge-detector to the noisy image. We define an edge as an abrupt change in average grey-level and require that the detector be computationally simple (i.e. low order) and insensitive to noise. These considerations lead to block-type convolution operators [3], which have less resolution but more noise suppression than other choices. The size of the operator is chosen to be as small as possible (depending on the observation noise level) such that the edge picture noise content remains within reasonable bounds. Our image is processed from left to right, column by column, and therefore we consider operators oriented towards detection of vertical edges, in particular the two operators [3]:

$$E_1 = \frac{1}{3}\begin{bmatrix} -1 & 0 & +1 \\ -1 & 0 & +1 \\ -1 & 0 & +1 \end{bmatrix} \quad , \quad E_2 = \frac{1}{6}\begin{bmatrix} -1 & -1 & +1 & +1 \\ -1 & -1 & +1 & +1 \\ -1 & -1 & +1 & +1 \end{bmatrix} \qquad (1)$$

The implementation of E_2 requires $\sim$ twice the computation of E_1 but results in half the noise variance and is therefore preferred in noisier situations. This noise variance is from the difference between the edge pictures obtained by applying the detector to the noisy versus the original image. The noise content is further reduced by thresholding.

In order to minimize the slight vertical spreading of edge information caused by both detectors, we also use a continuity criterion which requires that the points vertically adjacent to a detected edge point are also detected as an edge. If not, then the edge weight of the point in question is set to zero. This again reduces the noise variance, but edges of very small vertical spread may become obscured. See Figs. 3 and 4 for edge-picture examples.

III. THE EDGE-IMPROVED FILTERS

We require first that the degraded image process be expressed by the dynamical model [4]

$$\underline{s}(k) = F\underline{s}(k-1) + G\underline{u}(k) \qquad (2)$$

$$\underline{z}(k) = H\underline{s}(k) + \underline{v}(k) \qquad (3)$$

$\underline{s}(k)$ is the signal state vector, $\underline{u}(k)$ the generating white noise, $\underline{v}(k)$ the white Gaussian observation noise and $\underline{z}(k)$ the observation vector. F, G and H are the system matrices, depending on the image model and filter type (scalar or vector). The optimal (MMSE) estimate of $\underline{s}(k)$ given the set of observations $\{\underline{z}(j) : j=1,2,\ldots,k\}$ is found recursively from

$$\hat{\underline{s}}(k) = [I - K(k)H]F\hat{\underline{s}}(k-1) + K(k)\underline{z}(k) \qquad (4)$$

and K(k) is determined from the coupled recursive equations constituting the discrete Riccati equation (e.g. [4], [5]). For a stable model, K(k) reaches a steady state K independent of the initial conditions [5]. Then (4) can be rewritten as

$$\hat{\underline{s}}(k) = A\hat{\underline{s}}(k-1) + K\underline{z}(k) \qquad (5)$$

where $A = [I - KH]F$.
There is little loss in performance and significant computational savings in using (5) rather than (4) to filter the entire image. Efficient methods for determining and implementing K are necessary.

a) The Vector Filters

The system matrices for vector filters based on a non-symmetric half-plane (NSHP) image model are derived in [4], the quarter-plane (QP) model based on the well-known separable exponential correlation function is considered in [6] and the identification of a vector autoregressive image model is described in [7], as examples of different vector models. The image column estimates are equal to the state vector estimates of (5). For an efficient implementation with no image segmentation we choose to implement the (arbitrary) vector filter as described by Panda and Kak [6] for the QP model. We perform the relatively trivial computation required to find the steady-state K' matrix for a small-dimensional system, say 16x16, based on the same image model, and extrapolate to the full-size (image dimension M) matrix as in (6) below.

$$K_M = \begin{bmatrix} K'(1,1) \;\ldots\; K'(1,16) & & & \\ \vdots \qquad\quad \vdots & & 0 & \\ K'(8,1) \;\ldots\; K'(8,16) & 0 & & \\ 0 \quad K'(8,1) \;\ldots\; K'(8,16) & & & \\ & \ddots & \ddots & \\ & & K'(8,1) \;\ldots\; K'(8,16) & \\ & & K'(9,1) \;\ldots\; K'(9,16) & \\ 0 & & \vdots \qquad\quad \vdots & \\ & & K'(16,1) \ldots K'(16,16) & \end{bmatrix} \qquad (6)$$

In practice, and particularly for the NSHP model, most of the matrix energy is concentrated around the main diagonal and we can filter the entire image using only the values K'(8,6), K'(8,7), ..., K'(8,10) for example [8].

We now assume an arbitrary vector filter of the form (5) and try to improve the step-response in the direction of processing. Let the filter input be

$$\underline{z}(k) = \begin{cases} \underline{0} & k \leq 0 \\ \underline{z}_1 & k>0 \end{cases} \qquad (7)$$

for an arbitrary k=0. Then the filter output with zero initial conditions is

$$\hat{\underline{s}}(k) = \sum_{i=0}^{k-1} A^i K \underline{z}_1 \qquad k \geq 1 \qquad (8)$$

Assuming stability, the steady-state response is

$$\hat{\underline{s}}(\infty) = [I-A]^{-1} K \underline{z}_1 \qquad (9)$$

We now consider a modified filter of the form

$$\hat{\underline{s}}(k) = A\hat{\underline{s}}(k-1) + K[\underline{z}(k) + C\underline{e}(k)] \qquad (10)$$

where $\underline{e}(k)$ is the current vector of edge weights and C is a matrix to be determined. When $\underline{z}_1$ is the vector of all 'ones' (i.e. a unit step on every line) then the input $\underline{e}_1(k)$ corresponding to the operator E_1 in (1) may be written as

$$\underline{e}_1(k) = \begin{cases} \underline{z}_1 & k=0,1 \\ \underline{0} & \text{otherwise} \end{cases} \qquad (11)$$

ignoring slight boundary effects at the image borders. The first two non-zero terms of $\hat{\underline{s}}(k)$ are now

$$\hat{\underline{s}}(0) = KC\underline{z}_1 \qquad (12a)$$

$$\hat{\underline{s}}(1) = [I+A]KC\underline{z}_1 + K\underline{z}_1 \qquad (12b)$$

We choose C such that $\hat{\underline{s}}(1) = \hat{\underline{s}}(\infty)$ from (9), leading to

$$C = \{[I-A^2]K\}^{-1} AK \qquad (13)$$

With such a C matrix and any input for which (11) is (approximately) true, steady-state conditions are reached in two iterations.

Note that the utility of achieving steady state is somewhat less than for the 1-D case of [3]. Given a step input spanning only a small portion of the observation vector, by the time steady state is reached, the horizontal edges become blurred. An improved scheme should also take horizontal edges into account. We attempt to minimize these spreading effects by localizing the area affected by the edge weights, partially by use of the continuity criterion previously mentioned. Various possibilities for implementing the filter (10) have been considered with the best results achieved as follows:

First rewrite (10) as

$$\hat{\underline{s}}(k) = A\hat{\underline{s}}(k-1) + K\underline{z}(k) + D\underline{e}(k) \qquad (14)$$

$$\text{with } D = KC = [I-A^2]^{-1} AK \qquad (15)$$

The matrix D calculated from a small-scale system can be extrapolated similarly as in (6), and we also restrict it to being tridiagonal. Then

i) calculate the 16x16 matrix D from eq. (15)
ii) let $d_1 = D(8,7)$, $d_2 = D(8,8)$, $d_3 = D(8,9)$
iii) perform vector filtering according to (5)
iv) to the i^{th} element of $\underline{s}(k)$, add the quantity

$$[d_1 \underline{e}_k(i-1) + d_2 \underline{e}_k(i) + d_3 \underline{e}_k(i+1)] \qquad (16)$$

(denoting the i^{th} element of $\underline{e}(k)$ as $\underline{e}_k(i)$)

This method works best on vertical edges spanning more than three horizontal lines. With a real image having more random edge characteristics, we expect no steady-state values but the filter output is still boosted at vertical edges, sharpening the edges and increasing image contrast. The matrix D corresponding to edge-detector E_2 is

$$D = 2\{[I-A^2][I+A]\}^{-1} A^2 K \qquad (17)$$

b) The Scalar Filters

For these filters (also of the form (5)), the state vector dimension is still essentially the same as the image dimension, but the observations are scalar and K is now a vector as opposed to a matrix. Matrix definition with an NSHP model is discussed in [4]. Our implementation is similar, using the reduced-update recursion but with a small-scale system for faster convergence. With a steady-state vector K' from a size 16 system, we extrapolate the full-sized vector as

$$K_{M+1} = [K'(1) \ldots K'(8); 0 \ldots 0; K'(9) \ldots K'(16)]^T \qquad (18)$$

In practice, good results have been obtained with only 5 non-zero elements in K_{M+1}. If $\{x(k)\}$ represents the scan-ordered image process, an M-lag smoothed estimate can be determined by using

$$\hat{x}(k) = [0,0,\ldots,0,1]\ \hat{\underline{s}}(k+M) \qquad (19)$$

and these filters generally perform better in terms of signal-to-noise ratio (SNR) than the corresponding vector filters.

With a horizontal step input analogous to (7), the scalar filter also reaches a steady state in the sense that the output on any row of the image becomes constant after enough columns have been processed. We again assume a unit step spanning the entire image (vertical dimension M), scan-ordered edge weights from E_1 and now the filter has the form

$$\hat{\underline{s}}(k) = A\hat{\underline{s}}(k-1) + K[z(k) + ce(k)] \qquad (20)$$

On each row, a steady-state vector $\hat{\underline{s}}$ can be calculated for any specific input step. For our assumed input

$$\hat{\underline{s}}(\infty) = [I-A]^{-1} K \qquad (21)$$

Such an input produces non-zero edge weights only for $1 \leq k \leq 2M$. Thus, we set $\hat{\underline{s}}(2M+1) = \hat{\underline{s}}(\infty)$ and solve for the scalar c, resulting in the equation

$$\underbrace{cA|I-A^{2M}||I-A|^{-1}K}_{X} = \underbrace{A^{M+1}|I-A|^{-1}K}_{Y} \qquad (22)$$

We can find M estimates of c

$$\hat{c}_i = Y_i / X_i \qquad 1 \leq i \leq M \qquad (23)$$

The filter parameter c is determined by finding M' estimates using a small-dimensional system and taking the average of the estimates. This value is found to be reasonable for step inputs of vertical spread greater than or equal to the number of non-zero elements of the vector K [8]. For E_2 we use

$$cA[I+A^M-A^{2M}-A^{3M}][I-A]^{-1}K = 2A^{2M+1}[I-A]^{-1}K \qquad (24)$$

IV RESULTS

We now present processing examples for the 256x256 test image of Fig.1, which is quantized in 6 bits and has a variance of 365. Fig.2 shows the degraded image with white noise of variance 290 added for an SNR of 1 db. The edge magnitude pictures obtained via the operator E_2 of (1) from the original and degraded image respectively are shown in Figs. 3 and 4. The processing results from 4 different 2-D filters and, for comparison, the 1-D filter of [3] are presented. For each filter we show the result without using any edge information, then with the edge weights of Fig.3 and finally using the noisy edge weights of Fig.4. Not including the edge detection, the edge-improved vector filter requires about 12 multiply-and-adds per point and the scalar filter 2 less.

Figs.5-7 are obtained from a vector filter based on an NSHP image model estimated via least squares methods and the corresponding scalar filter results are presented in Figs.8-10. Figs.11-13 and 14-16 are, respectively, the vector and scalar filter results obtained using a QP model and separable exponential correlation function. The 1-D results are seen in Figs. 17-19.

In all cases, the inclusion of edge information can be seen to sharpen the edges and increase the contrast of the processed image. Whether the visual quality is improved or not depends largely on the objectionability of the increased noise in the flat areas. The results using the original image edge weights are generally very good but the edge picture of Fig.4 contains too much noise to provide a good visual result. Best results have been obtained for degraded images having SNR of $\sim$5 db or greater, where the edges can be fairly reliably detected. The 1-D filter can produce good visual results (Fig. 18) even though the noise level remains relatively high. In experiments it was found that the QP-model filters were capable of better SNR performance than the NSHP ones, but they were much more sensitive to noise in the edge weights (see Fig.13), possibly related to the pixel "correlation distance" being significantly higher for the QP model. The QP results also tended to have an objectionable criss-cross noise pattern. The scalar filter generally outperforms the corresponding vector filter because of the 256-lag smoothing possible with no extra computation (providing a delay is permitted).

V CONCLUSIONS

Methods have been presented for improving the response of an arbitrary 2-D scalar or vector Kalman filter in the presence of image edges oriented perpendicular to the direction of processing. The methods are particularly useful when most of the image edges are so oriented. Results show that even in the very noisy case image contrast is increased and edges are sharpened. Subjectively, the result may not be better due to increased noise in the flat areas. However, in lower noise cases where the edges can be reliably detected, the visual quality is significantly improved by using the (noisy) edge weights. Further improvements are being considered by using horizontal edge detectors (transposes of those in (1)) as well as vertical. Then different actions could be taken depending on the relative magnitudes of the horizontal and vertical edge weights at each point.

REFERENCES

[1]V.K.Ingle,A.Radpour,J.W.Woods and H.Kaufman, "Recursive estimation with non-homogeneous image models", in Proc. 1978 IEEE Conf. Pattern Recognition and Image Processing,Chicago,pp.105-108,May 1978.

[2]N.E.Nahi and A.Habibi,"Decision-directed recursive image enhancement",IEEE Trans. Circuits and Systems,vol.CAS-22,pp.286-293,March 1975.

[3]J.Biemond and J.J.Gerbrands,"An edge-preserving recursive noise-smoothing algorithm for image data", Tech.Rep. No. IT-79-06,Delft University of Technology, Delft,The Netherlands,Feb. 1979.

[4]J.W.Woods and C.H.Radewan,"Kalman filtering in two dimensions",IEEE Trans. Information Theory, vol.IT-23,pp.473-482,July 1977.

[5]B.D.O.Anderson and J.B.Moore,Optimal Filtering, Prentice-Hall Inc.,Englewood Cliffs,N.J.,1979.

[6]D.P.Panda and A.C.Kak,"Recursive least squares smoothing of noise in images",IEEE Trans. Acoustics Speech and Signal Processing,vol.ASSP-29,pp.188-196, Dec. 1977.

[7]B.R.Suresh and B.A.Shenoi,"New results in two-dimensional Kalman filtering with applications to image restoration",IEEE Trans. Circuits and Systems, vol. CAS-28,pp.307-319,April 1981.

[8]H.J.Tork,"First-order Kalman-type filtering for image restoration", M.A.Sc. Thesis, University of Toronto,Toronto,1983.

FIGURES

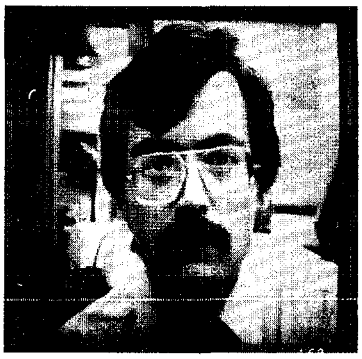

Fig. 1 Original image, variance = 365

Fig. 2 Corrupted image, SNR = 1 db

Fig.3 Edge-weight picture from original image

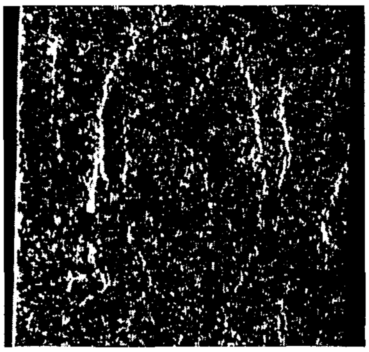

Fig. 4 Edge-weight picture from Fig.2, with threshold and continuity criteria.

Fig. 8 Scalar NSHP → imp. = 9.85 db

Fig. 5 Vector NSHP filter → improvement=9.00 db

Fig. 9 Scalar NSHP + good edge → imp.=11.7 db

Fig. 6 Vector NSHP + good edge → imp. = 11.0 db

Fig. 10 Scalar NSHP + noisy edge → imp.=10.4 db

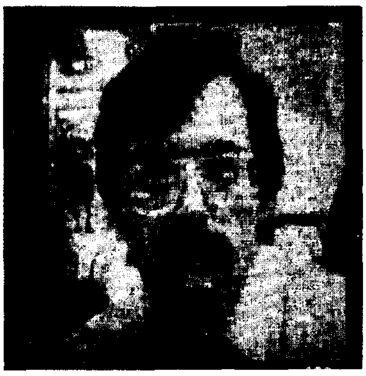

Fig. 7 Vector NSHP + noisy edge → imp.=9.73 db

Fig. 11 Vector QP → imp. = 8.51 db

Fig. 12 Vector QP + good edge → imp. = 12.1 db

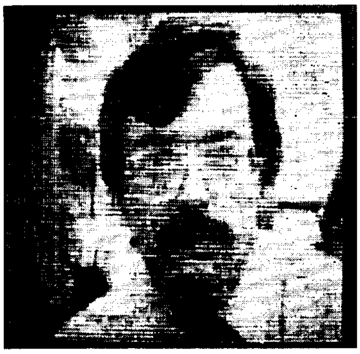

Fig. 13 Vector QP + noisy edge → imp. = 7.66 db

Fig. 14 Scalar QP → imp. = 9.85 db

Fig. 15 Scalar QP + good edge → imp. = 11.0 db

Fig. 16 Scalar QP + noisy edge → imp. = 10.1 db

Fig. 17 1-D algorithm → imp. = 6.21 db

Fig. 18 1-D alg. + good edge → imp. = 8.1 db

Fig. 19 1-D alg. + noisy edge → imp. = 6.8 db