



Fig. 7. Decoded frame 60 of salesman using VBS scheme.

quadtree is also counted as part of the coding bits. To compute the bits for the DCT coefficients and the motion vectors, we used the MPEG Huffman tables.

Figs. 2 and 3 show SNR and bit rate results at 64 kb/s for the salesman sequence. The VBS scheme improves on the FBS scheme by 1.7 dB. The average SNR's for the VBS and the FBS scheme are 32.54 dB and 30.84 dB. There are two reasons for the fluctuation of the VBS bits in Fig. 3. One reason is that the quadtree bit is not included in the bit computation in the tree decision step. This is why the actual VBS bits sometimes exceeded the target bit rate. Another reason is that the granularity of the bit rate is not fine enough to match the target bit rate exactly, which is due to the discreteness of the quantizer step size. This is why the actual VBS bit rate is sometimes smaller than the target bit rate.

Fig. 4 shows the bit rate curves of the luminance DCT coefficients, the motion vectors, and the quadtree representation. Fig. 5 shows the optimal quadtree that was used by the VBS algorithm for the frame 60. The decoded images (frame 60) for the FBS and the VBS schemes are compared in Fig. 6 (30.19 dB) and Fig. 7 (32.01 dB). The computation time of the VBS scheme was about 3 min/frame on a Sun Sparc 20 workstation.

## VI. CONCLUSION

We have presented a fast optimal algorithm for quadtree based variable block size motion estimation for video coding. The algorithm was based on a Lagrange multiplier approach and a fast quadtree pruning technique. Unlike previous VBS algorithms, the proposed algorithm achieves the optimal bit allocation between the motion vector coding and the residual error coding. The results show that the new algorithm works better than conventional fixed block size motion estimation.

The computational complexity of the optimal algorithm is not too prohibitive to be implemented, even in a realtime hardware. The algorithm will also be useful as a benchmark for other suboptimal

methods. The new approach based on the Lagrangian optimization appears to be promising especially for very low bit rate coding applications because it uses an efficient bit allocation strategy between motion vector and residual error coding.

## REFERENCES

- [1] A. Puri, H. M. Hang, and D. L. Schilling, "Interframe coding with variable block size motion compensation," in *Proc. GLOBECOM'87*, pp. 65–69.
- [2] M. H. Chan, Y. B. Yu, and A. G. Constantinides, "Variable size block matching motion compensation with application to video coding," in *Proc. Inst. Elect. Eng.*, pt. 1, pp. 205–212, Aug. 1990.
- [3] "Draft recommend. H.263," Doc. LBC-95-83, Mar. 1995.
- [4] H. Everett, "Generalized Lagrange multiplier method for solving problems of optimum allocation of resources," *Oper. Res.*, vol. 11, pp. 399–417, 1963.
- [5] J. Lee, "Optimal quadtree for variable block size motion estimation," in *Proc. IEEE Int. Conf. Image Processing*, Washington, DC, Oct. 1995, vol. 3, pp. 480–483.
- [6] J. Lee and B. W. Dickinson, "Joint optimization of frame type selection and bit allocation for MPEG video encoders," in *Proc. 1st IEEE Int. Conf. Image Processing*, Austin, TX, Nov. 1994, vol. 2, pp. 962–966.
- [7] B. Girod, "Rate-constrained motion estimation," in *Proc. SPIE VCIP'94*, Chicago, IL, pp. 1026–1034.
- [8] G. J. Sullivan and R. L. Baker, "Rate-distortion optimized motion compensation for video compression using fixed or variable size blocks," in *Proc. GLOBECOM'91*, Dec. 1991, pp. 85–90.
- [9] Y. Shoham and A. Gersho, "Efficient bit allocation for an arbitrary set of quantizers," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, pp. 1445–1453, Sept. 1988.
- [10] K. Ramchandran and M. Vetterli, "Best wavelet packet bases in a rate-distortion sense," *IEEE Trans. Image Processing*, vol. 2, pp. 160–175, Apr. 1993.

## Robust B-Spline Image Modeling with Application to Image Processing

Marta Karczewicz and Moncef Gabbouj

**Abstract**— In this correspondence, we present a new approach to two-dimensional (2-D) robust spline image smoothing based on the  $M$ -estimator algorithm. Unlike in other  $M$ -estimator based image processing algorithms, the new algorithm takes into consideration spatial relations between picture elements. The contribution of the sample to the model depends not only on the current residual of that sample, but also on the neighboring residuals. A smoothing parameter is estimated separately for each processing window and it adapts to the local structure of the image. The proposed algorithm is applied to image filtering. The resulting filter preserves details and suppresses additive Gaussian and impulsive noise efficiently.

**Index Terms**—B-splines, image filtering, image modeling.

## I. INTRODUCTION

Spline functions are frequently used to model images [1], [9], [10], [15], [18]. The segmented nature of spline functions gives them flexibility and allows them to adjust very effectively to local

Manuscript received March 25, 1996; revised May 7, 1997. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Henri Maitre.

M. Karczewicz is with Nokia Research Center, FIN-33721 Tampere, Finland (e-mail: marta.karczewicz@research.nokia.com).

M. Gabbouj is with the Signal Processing Laboratory, Tampere University of Technology, FIN-33101 Tampere, Finland (e-mail: moncef@cs.tut.fi).

Publisher Item Identifier S 1057-7149(98)03996-7.

characteristics of the data. However, most spline approximation algorithms developed for image processing assume the Gaussian noise model. Real sensor data and digital images transmitted over non-Gaussian channels, to cite only a few, do not fulfill this assumption used in classical statistical procedures. In many image processing applications, such as the ones just mentioned, the presence of outliers (e.g., a heavy tailed noise density) is unavoidable. Such outliers may result from particular atmospheric phenomena or transmission errors in a faulty channel [11], [13]. Unfortunately, least squares or maximum likelihood estimators developed under the Gaussian model are very sensitive to minor deviations from the Gaussian noise assumption. Even a small portion of outliers in the form of impulsive noise can drastically deteriorate the quality of the estimate. The main purpose of this work is to extend the recent results in spline signal approximation in the presence of a mixture of Gaussian and impulsive noise. A two-stage algorithm for discontinuity preserving surface reconstruction was proposed by Sinha and Schunk [15]. The first stage, which is based on robust regression (the least median of squares (LMedS) estimator), removes outliers; whereas the second stage, based on approximation using weighted bicubic splines, reduces normal noise. This approximation technique attenuates noise efficiently; however it tends to remove important image details such as lines. This is the inherent problem of using LMedS estimator.

The proposed method is based on the iterative  $M$ -estimation algorithm with modified residuals. At each stage, the contribution of each data sample to the model depends on the residual (difference between the data and the model) of that sample and the influence function. Two well-known influence functions are the *soft-limiter* and the *re-descending* function [6]. The currently used  $M$ -estimators have severe drawbacks when applied in image approximation. In the case of an estimator based on a soft limiter, even a small fraction of impulsive noise may have a large effect on the estimate (they have a small breakdown point).  $M$ -estimators based on the re-descending function can reach a much higher breakdown point, especially, when the initial fit is very robust. At the same time they can remove important image details, e.g., lines, since they are unable to distinguished them from outliers.

The changes introduced in the proposed method are empirical in nature. However, they allow the achievement of good approximation of the image in the presence of Gaussian and impulsive noise. The proposed algorithm takes into consideration spatial relations between picture elements. The contribution of a sample to the model depends not only on the current residual of that sample, but also on the neighboring residuals. The interaction between the picture elements has been previously used in modeling of spatial discontinuities for problems such as surface recovery [3], [16] or outlier detection [2].

## II. SPLINE APPROXIMATION

Surfaces are represented using  $B$ -splines [4]. Let  $\mathbf{t} = \{t_{-k}, \dots, t_{g+k+1}\}$  and  $\mathbf{s} = \{s_{-k}, \dots, s_{l+k+1}\}$  be two knot vectors defined in the  $x$  and  $y$  directions, respectively. The  $B$ -spline surface of degree  $k$  in the  $x$  and  $y$  directions is defined as

$$f(x, y) = \sum_{i=-k}^g \sum_{j=-k}^l \theta_{i,j} B_{i,k+1,\mathbf{t}}(x) B_{j,k+1,\mathbf{s}}(y) \quad (1)$$

where  $B_{i,k+1,\mathbf{t}}$  and  $B_{j,k+1,\mathbf{s}}$  are  $B$ -spline elements defined on the knot sequences  $\mathbf{t}$  and  $\mathbf{s}$ , respectively.

Spline approximations can be obtained by imposing some smoothness constraints on the function—smoothing splines [5]. Let us assume that a set of data values  $z_{q,r}$  and a set of weights  $w_{q,r} > 0$  corresponding to data locations  $(x_q, y_r)$ ,  $q = 1, \dots, N_1$ ,  $r = 1, \dots, N_2$  are given. The two-dimensional (2-D) smoothing spline

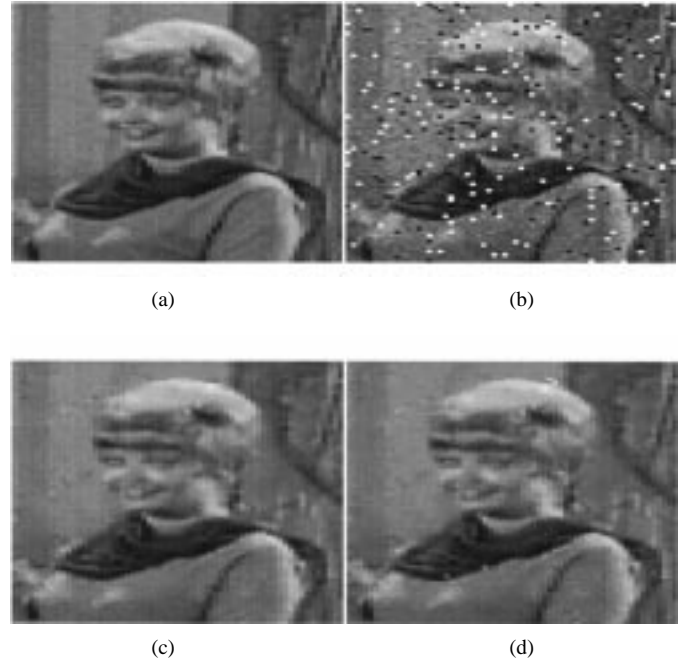


Fig. 1. (a) Original image. (b) Noisy image. (c)–(d) Restored image by the proposed spline approximation algorithm with and without additional weights, respectively.

$\hat{f}_\lambda$  is defined as the spline function  $f$  which minimizes

$$\sum_{q=1}^{N_1} \sum_{r=1}^{N_2} w_{q,r}^2 [z_{q,r} - f(x_q, y_r)]^2 + \lambda \iint \left( \frac{\partial^2 f}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 f}{\partial x \partial y} \right)^2 + \left( \frac{\partial^2 f}{\partial y^2} \right)^2 dx dy. \quad (2)$$

The first term is the residual error in the fit of the surface to the data. The second term is the second-order Laplacian smoothing functional, which is a measure of smoothness. The smoothing parameter  $\lambda$  governs the trade-off between smoothness and goodness-of-fit.

The  $B$ -spline coefficients  $\theta_{ij}$  are calculated as the solution of the linear system of equations

$$[(\mathbf{WP})^T (\mathbf{WP}) + \lambda \mathbf{\Sigma}] \boldsymbol{\theta} = (\mathbf{WP})^T \mathbf{Wz} \quad (3)$$

where

$$\boldsymbol{\theta} = \mathbf{cs} \left( \begin{bmatrix} \theta_{-k,-k} & \cdots & \theta_{g,-k} \\ \vdots & & \vdots \\ \theta_{-k,l} & \cdots & \theta_{g,l} \end{bmatrix} \right), \quad (4)$$

$$\mathbf{z} = \mathbf{cs} \left( \begin{bmatrix} z_{1,1} & \cdots & z_{N_1,1} \\ \vdots & & \vdots \\ z_{1,N_2} & \cdots & z_{N_1,N_2} \end{bmatrix} \right)$$

and

$$\mathbf{W} = \text{diag}[w_{1,1}, \dots, w_{N_1,N_2}]. \quad (5)$$

$\mathbf{cs}(\mathbf{A})$  denotes the column vector obtained by putting the columns of matrix  $\mathbf{A}$  underneath each other in their natural order. Matrices  $\mathbf{P}$  and  $\mathbf{\Sigma}$  are conveniently expressed using Kronecker product notation. Let  $\mathbf{A} \otimes \mathbf{B}$  be the Kronecker product of matrices  $\mathbf{A}$  and  $\mathbf{B}$ , then

$$\mathbf{P} = \mathbf{P}_1 \otimes \mathbf{P}_2, \quad (6)$$

where

$$\mathbf{P}_1 = \begin{bmatrix} B_{-k,k+1,\mathbf{t}}(x_1) & \cdots & B_{g,k+1,\mathbf{t}}(x_1) \\ \vdots & & \vdots \\ B_{-k,k+1,\mathbf{t}}(x_{N_1}) & \cdots & B_{g,k+1,\mathbf{t}}(x_{N_1}) \end{bmatrix},$$

$$\mathbf{P}_2 = \begin{bmatrix} B_{-k,k+1,s}(y_1) & \cdots & B_{l,k+1,s}(y_1) \\ \vdots & & \vdots \\ B_{-k,k+1,s}(y_{N_2}) & \cdots & B_{l,k+1,s}(y_{N_2}) \end{bmatrix} \quad (7)$$

and

$$\mathbf{\Sigma} = \mathbf{B}^{(2)} \otimes \mathbf{C}^{(0)} + 2\mathbf{B}^{(1)} \otimes \mathbf{C}^{(1)} + \mathbf{B}^{(0)} \otimes \mathbf{C}^{(2)}. \quad (8)$$

The elements of matrices  $\mathbf{B}^{(m)}$  and  $\mathbf{C}^{(m)}$  for  $m = 0, 1, 2$ , are, respectively,

$$\mathbf{B}_{ij}^{(m)} = \int B_{i-k-1,k+1,t}^{(m)}(x) B_{j-k-1,k+1,t}^{(m)}(x) dx, \quad (9)$$

$$i, j = 1, \dots, g + k + 1$$

and

$$\mathbf{C}_{ij}^{(m)} = \int B_{i-k-1,k+1,s}^{(m)}(y) B_{j-k-1,k+1,s}^{(m)}(y) dy, \quad (10)$$

$$i, j = 1, \dots, l + k + 1.$$

The vector  $\bar{\mathbf{z}}$  containing the approximated values is given by

$$\bar{\mathbf{z}} = \mathbf{P}\boldsymbol{\theta} = \mathbf{P}[(\mathbf{W}\mathbf{P})^T(\mathbf{W}\mathbf{P}) + \lambda\mathbf{\Sigma}]^{-1}(\mathbf{W}\mathbf{P})^T\mathbf{W}\mathbf{z}. \quad (11)$$

### A. Choosing the Smoothing Parameter

A variety of methods can be used to choose the smoothing parameter  $\lambda$  [17]. We will use the method called *equivalent degrees of freedom choice* for  $\lambda$  [17]. The parameter  $\lambda = \lambda_{\text{EDF}}$  is estimated as the solution to

$$F_{\text{EDF}}(\lambda) = \frac{\sum_{i=1}^{N_1} \sum_{j=1}^{N_2} (w_{i,j} f(x_i, y_j) - w_{i,j} z_{i,j})^2}{N - \text{df}(\lambda)} = \sigma^2 \quad (12)$$

where  $\text{df}(\lambda)$  denotes the number of degrees of freedom, which is equal to the trace of the hat matrix  $\mathbf{A}(\lambda)$ , i.e.,  $\text{df}(\lambda) = \text{trace}\{\mathbf{A}(\lambda)\}$ . The hat matrix  $\mathbf{A}(\lambda)$  is the matrix which maps data into predictions  $\bar{\mathbf{z}} = \mathbf{A}(\lambda)\mathbf{z}$ . To compute the value of  $\lambda_{\text{EDF}}$ , it is more efficient to use  $F_{\text{EDF}}(\lambda)$  in the form obtained with the help of eigenanalysis of the underlying matrices

$$F_{\text{EDF}}(\lambda) = \frac{\sum_{i=1}^N \frac{\lambda^2 v_i^2 d_i^2}{(1+\lambda d_i)^2}}{N - \sum_{i=1}^N \frac{1}{1+\lambda d_i}} = \sigma^2. \quad (13)$$

The values  $v_i$  are the elements of the vector  $\mathbf{v} = \mathbf{U}\mathbf{z}$ , where matrix  $\mathbf{U}$  is the result of the Schur decomposition of the matrix  $\mathbf{W}^{-1}(\mathbf{P}^T)^{-1}\mathbf{\Sigma}\mathbf{P}^{-1}\mathbf{W}^{-1}$ :

$$\mathbf{W}^{-1}(\mathbf{P}^T)^{-1}\mathbf{\Sigma}\mathbf{P}^{-1}\mathbf{W}^{-1} = \mathbf{U}^T \mathbf{D} \mathbf{U}$$

and  $d_1, \dots, d_M$  are the eigenvalues of the same matrix  $\mathbf{W}^{-1}(\mathbf{P}^T)^{-1}\mathbf{\Sigma}\mathbf{P}^{-1}\mathbf{W}^{-1}$  (diagonal elements of  $\mathbf{D}$ ).

### III. ITERATIVE ALGORITHM FOR ROBUST SPLINE FITTING

The proposed method of robust spline image approximation is based on the  $M$ -estimator algorithm with modified residuals (Huber [7]). Given the data values  $z_{q,r}$  at locations  $(x_q, y_r)$ ,  $q = 1, \dots, N_1$ ;  $r = 1, \dots, N_2$ , inside the processing window, the goal of robust spline fitting is to determine the estimate  $\bar{z}_{q,r}$  of the ideal image.

*Algorithm:*

- 1) Consider the  $m$ th iteration ( $m > 0$ ). Let  $\bar{z}_{q,r}^{(m)}$  be the trial values for the estimate  $\bar{z}_{q,r}$ . Denote by  $r_{q,r}^{(m)}$  and  $\hat{r}_{q,r}^{(m)}$  the residuals and modified residuals, respectively, as follows:

$$r_{q,r}^{(m)} = z_{q,r} - \bar{z}_{q,r}^{(m-1)}, \quad \hat{r}_{q,r}^{(m)} = \psi\left(\frac{r_{q,r}^{(m)}}{\sigma}\right) \sigma \quad (14)$$

where the function  $\psi$  is the classical minimax function [7]:  $\psi(u) = \min\{c, \max\{u, -c\}\}$ .

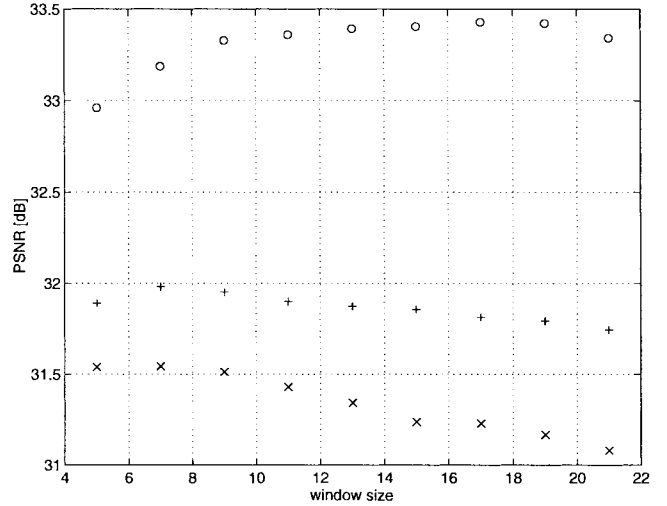


Fig. 2. PSNR obtained for three images Lenna (marked as 'o'), home (marked as '+'), and harbor (marked as 'x'), as a function of the length of the side of the processing window.

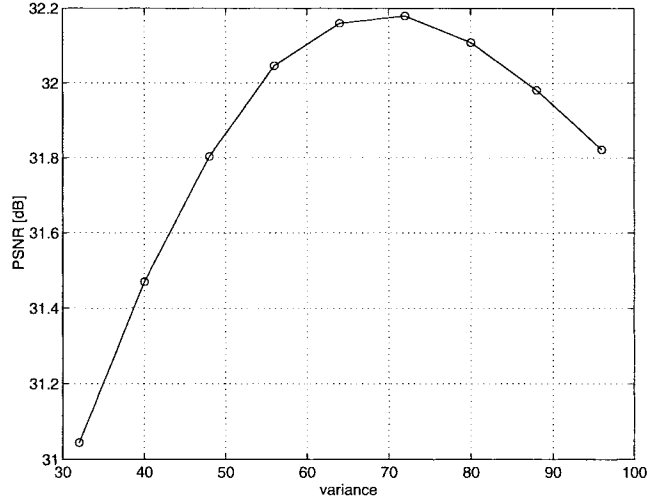


Fig. 3. Average PSNR obtained for three images, Lenna, home, and harbor, as a function of the variance of the Gaussian noise used by the proposed method. The images are corrupted by adding zero mean i.i.d. Gaussian noise of variance 64 and 5% of impulsive noise.

- 2) Set the modified residual  $\hat{r}_{q,r}^{(m)}$  to 0 if  $r_{q,r}^{(m)} > k_2\sigma$  (respectively,  $r_{q,r}^{(m)} < -k_2\sigma$ ) and the  $(x_q, y_r)$ th pixel does not create, with the neighboring pixels, a structure from the given set of detail preserving requirements, i.e.,  $r_{s,t}^{(m)} > k_1\sigma$  (respectively,  $r_{s,t}^{(m)} < -k_1\sigma$ ) for all neighboring  $(x_s, y_t)$ th pixels forming such a structure ( $k_1$  and  $k_2$  are fixed *a priori*). In our experiments, such structures are just a cluster of five or more connected pixels or four pixels creating vertical, horizontal or diagonal lines. In addition, assign a weighting factor  $w_{q,r}^{(m)}$  to each observation. If the sample has a modified residual equal to zero, its corresponding weight is equal to  $w_{\text{const}}$ , where  $0 < w_{\text{const}} \ll 1$ ; and 1, otherwise.
- 3) The spline is fitted to the auxiliary variables  $\hat{z}_{q,r}^{(m)}$  which are created by adding the modified residuals to the estimates from the previous iteration

$$\hat{z}_{q,r}^{(m)} = \bar{z}_{q,r}^{(m-1)} + \hat{r}_{q,r}^{(m)}. \quad (15)$$

First, the weighted planar least square solution (corresponding to the smoothing spline for  $\lambda \rightarrow \infty$ ) is found. If the value of the criterion function  $F_{\text{EDF}}(\lambda)$  for the resulting residuals is smaller than the variance of the noise  $\sigma^2$ , this solution is retained. Otherwise the smoothed data is computed using (11), in which the vector of observations  $\mathbf{z}$  is replaced by the vector of auxiliary variables  $\hat{\mathbf{z}}^{(m)}$

$$\bar{\mathbf{z}}^{(m)} = \mathbf{P}[(\mathbf{W}\mathbf{P})^T(\mathbf{W}\mathbf{P}) + \lambda^{(m)}\boldsymbol{\Sigma}]^{-1}\mathbf{P}^T\mathbf{W}^2\hat{\mathbf{z}}^{(m)} \quad (16)$$

where

$$\bar{\mathbf{z}}^{(m)} = \mathbf{cs} \left( \begin{bmatrix} \bar{z}_{1,1}^{(m)} & \cdots & \bar{z}_{N_1,1}^{(m)} \\ \vdots & & \vdots \\ \bar{z}_{1,N_2}^{(m)} & \cdots & \bar{z}_{N_1,N_2}^{(m)} \end{bmatrix} \right)$$

and

$$\hat{\mathbf{z}}^{(m)} = \mathbf{cs} \left( \begin{bmatrix} \hat{z}_{1,1}^{(m)} & \cdots & \hat{z}_{N_1,1}^{(m)} \\ \vdots & & \vdots \\ \hat{z}_{1,N_2}^{(m)} & \cdots & \hat{z}_{N_1,N_2}^{(m)} \end{bmatrix} \right). \quad (17)$$

The smoothing parameter  $\lambda^{(m)}$  for which the value of  $F_{\text{EDF}}(\lambda)$  and the variance  $\sigma^2$  are equal is obtained by solving (13). Also in the definition of  $v_i$  and  $d_i$  used in this equation, the vector of observations  $\mathbf{z}$  is replaced by the vector of auxiliary variables  $\hat{\mathbf{z}}^{(m)}$ .

- 4) Steps 1–3 are repeated for successive values of integer  $m$  until the norm  $\|\bar{\mathbf{z}}^{(m+1)} - \bar{\mathbf{z}}^{(m)}\|$  becomes smaller than some predefined threshold, or a maximum number of iterations is reached.

The starting values  $\bar{z}_{q,r}^{(0)}$  should be very robust, thus

$$\bar{z}_{q,r}^{(0)} = \text{median}(z_{1,1}, \dots, z_{N_1,N_2})$$

for  $q = 1, \dots, N_1; r = 1, \dots, N_2$  (18)

is a convenient choice. The following ranges for  $k_1$  and  $k_2$  are found empirically:  $1 \leq k_1 \leq 2$  and  $4.86 \leq k_2 \leq 5.86$ . The value  $c$  used in the definition of the minimax  $\psi$ -function regulates the degree of robustness; good choices are between 1 and 2 (Huber [8]) for  $m > 1$  and between 3.5 and 4 for  $m = 1$ . At the first iteration  $m = 1$ , the planar surface is fitted to the auxiliary variables which are created by adding to the median of the samples in the processing window the modified residuals  $\hat{r}_{q,r}^{(1)}$ . The modified residuals  $\hat{r}_{q,r}^{(1)}$  corresponding to the samples creating the detail usually have values  $\pm c\sigma$ . Thus, large enough  $c$  can help avoid selecting the planar surface if there is a certain detail in the processing window (e.g., line) that one wishes to preserve. If the planar fit is chosen, the approximating spline will not start changing its shape into the direction of this feature and the iterations will stop.

In the proposed algorithm, a modified residual can be set to zero as in algorithms using the redescending  $\psi$ -function; however, the decision does not depend only on the distance of the corresponding sample from the current model estimate. Furthermore, the probability that this sample belongs to the given set of structures, which one wishes to preserve, is taken into consideration. In our experiments, such structures are clusters of five or more connected pixels or set of four pixels, creating a vertical, horizontal or diagonal line (window  $5 \times 5$  is used). This provides a good trade-off between removing clusters of impulses and the minimum size of objects or details which are of interest to the observer. Setting modified residuals to zero is enough to improve detail preservation and to bound the influence of

outliers. Outliers, however, will still contribute to the final model. This can cause some artifacts especially near the boundaries of regions with different gray-level values [note the edge of the dress collar in Fig. 1(d)]. Such artifacts can be eliminated by introducing additional weights. Recall that if a sample has a zero modified residual, it gets a small nonzero weight  $w_{\text{const}}$ ,  $0 < w_{\text{const}} \ll 1$ ; otherwise, it gets a weight of 1. Samples with zero weights are not used to avoid singularities in matrices in (3). Although the weights of outlying observations are nonzero, they are much smaller than the weights of the remaining observations; and since the spline is fitted to the auxiliary variables which are already bounded, these observations will be ignored almost completely. The “smoothest” solution at these points will be preferred. An example is given in Fig. 1(c).

#### A. Local Processing of the Image

The proposed algorithm is applied locally over a  $5 \times 5$  sliding window. The knot sequence for the tensor-product bicubic  $B$ -splines is  $(-2, -2, -2, -2, 0, 2, 2, 2, 2)$  in both the  $x$  and  $y$  directions. A sliding window is passed over the input observations. The sample at the center of the window is replaced by its estimate obtained by robust spline fitting. The main reason behind using local processing window is the smoothing parameter. The smoothing parameter  $\lambda$  is inversely related to the signal-to-noise ratio (SNR). Generally, images contain projection of objects with different sizes and at different distances from the camera. Even when the noise level is stationary through the image, the local SNR will vary. Thus, it is very difficult to identify one value of the smoothing parameter for the whole image that gives the best noise suppression without smoothing out significant features. Therefore, unlike in [1] and [17], where one smoothing parameter is estimated and used for the whole image, the smoothing parameter in the proposed approach is estimated locally, separately for each processing window. The space-varying regularization parameter allows us to impose different degrees of smoothness in different image areas. In homogeneous regions, large values of the smoothing parameter are used in order to smooth out noise; while small values are used at discontinuities in order to preserve details.

In this experiment, we investigate the dependency of the quality of the restored image on the size of the processing window. The size of the processing window influence the results implicitly, through the choice of the smoothing parameter. The noisy images are generated by adding zero mean i.i.d. Gaussian noise of variance 64 and 5% of impulsive noise. The images are restored using the proposed method. The measure of the quality of the reconstructed image is the peak SNR (PSNR). The PSNR obtained for three images—Lenna, home, and harbor—as a function of used size of the processing window is given in Fig. 2. It can be seen the best results are obtained when the size of the window is equal to  $17 \times 17$  for the image Lenna, and to  $7 \times 7$  for the images harbor and home. The computational complexity of the proposed method is approximately proportional to the square of the window size. Thus, we decided to use window of size  $5 \times 5$  since, it gives good trade-off between the computational complexity and the reconstruction quality.

#### B. Sensitivity to the Accuracy of the Noise Variance Measurement

The presented method requires knowledge of the variance of the Gaussian noise. Thus the source of the noise has to be known so the variance can be measured directly or calculated using the received corrupted image. In the latter case, the method presented in [14] can be used. In both cases one will not obtain the exact value of the noise variance, just some approximation. Thus we tested the performance of the proposed method when the variance used by the algorithm is

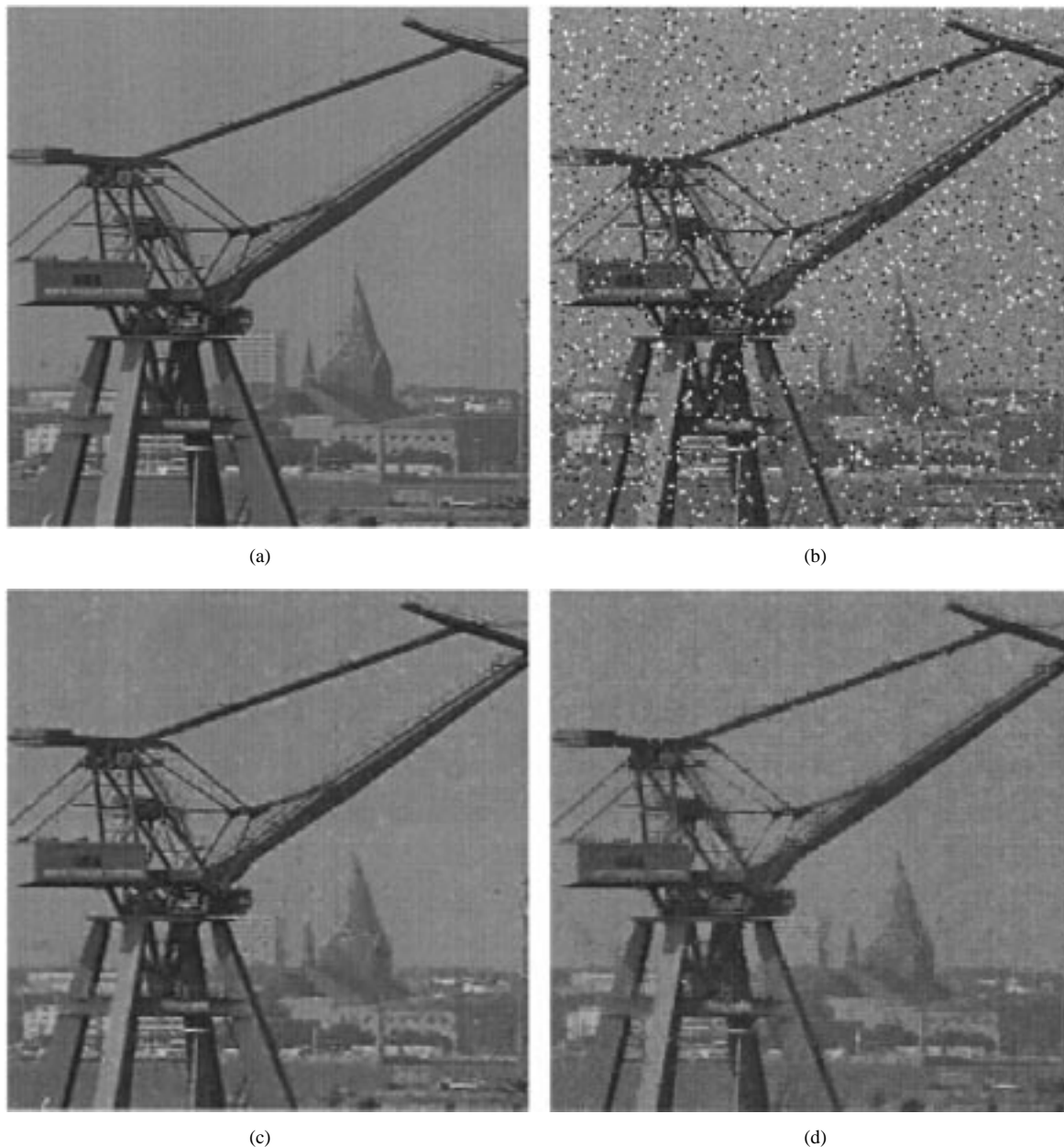


Fig. 4. (a) Original image harbor; (b) noisy image harbor; image filtered with (c) the proposed algorithm and (d) bidirectional multistage median.

different from the true variance of the noise. The noisy images are generated by adding zero mean i.i.d. Gaussian noise of variance 64 and 5% of impulsive noise. The proposed method uses the following variance values: 32, 40, 48, 56, 64, 72, 80, 88, and 96. The average PSNR obtained for three images, Lenna, home, and harbor, as a function of the variance is given in Fig. 3. As it can be seen from this figure, the proposed method is not very sensitive to the accuracy with which the estimated variance of the Gaussian noise is given. It also can be observed that the best results are obtained when the value of the variance of the Gaussian noise is slightly overestimated.

#### IV. IMAGE RECONSTRUCTION

The noisy images are generated by adding zero mean i.i.d. Gaussian noise of variance 64 and 5% of impulsive noise. The proposed method is compared against the median, the bidirectional multistage median [12] and the 2-D optimal weighted median (WM) under given structural constraints [19]. Three different  $3 \times 3$  optimal WM filters

are used: preserving horizontal and vertical lines (WM1), preserving horizontal, vertical and diagonal lines (WM2) and preserving corners (WM3). Table I summarizes the filtering results. The values of PSNR indicate that the proposed method attenuates noise significantly better than the other filters we compared against. The visual result of image restoration by spline fitting is presented in Fig. 4. The original image ( $180 \times 180$  part of image harbor) is shown in Fig. 4(a). The noisy image was generated by adding zero mean Gaussian noise of variance 64 and 5% of impulses to the original image [Fig. 4(b)]. The output image of the spline filter is shown in Fig. 4(c), and that of the bidirectional multistage filter in Fig. 4(d).

The main disadvantage of the proposed method is its computational complexity. Currently, on a HP9000 series computer, the C implementation of the proposed algorithm runs at approximately 20 s per image for easy images such as Lenna and up to about 1 min per frame for difficult images such as harbor. The approximate time required by the methods that we compared against varies from 1 to 5

TABLE I  
PSNR (dB) RESULTS FOR IMAGE CORRUPTED BY GAUSSIAN AND IMPULSE NOISE

Filter Type	Image		
	Lenna	Home	Harbor
Median (3x3)	31.36	26.67	24.65
Bidirectional multistage median (5x5)	31.58	28.73	28.02
WM1	31.58	28.87	27.84
WM2	29.85	27.92	27.71
WM3	32.00	28.08	26.29
Spline filter	32.96	31.89	31.54

s. It should be noted, however, that the code of the proposed method and the other methods has not been optimized.

The convergence of the proposed method has been studied experimentally. The algorithm converged for each block of data in the sliding window during the image restoration.

## V. CONCLUSION

A new image approximation scheme is proposed. The structural constraints are incorporated in an iterative  $M$ -estimator algorithm. As a result, an image modeling method is obtained that is not influenced by outliers and reduces Gaussian and heavy-tailed noise efficiently; and, at the same time it retains important details. The images are modeled as tensor product bicubic  $B$ -splines. The smoothing parameter  $\lambda$  is estimated separately for each processing window, thus allowing it to adapt to local structures of the image. As a result, one can expect excellent Gaussian noise removal in smooth and slowly varying areas where  $\lambda$  is large and at the same time very good preservation of important details (small values of  $\lambda$ ). Results obtained by applying the filter based on the presented approximation algorithm to real scenes indicate that this method is robust with respect to variations in the statistics of both the noise and the image.

## REFERENCES

- [1] M. Berman, "Automated smoothing of images and other regularly spaced data," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 460–468, 1994.
- [2] M. J. Black and A. Rangarajan, "The outlier process: Unifying line processes and robust statistics," in *Proc. 1994 IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, Seattle, WA, 1994, pp. 15–22.
- [3] A. Blake and A. Zisserman, *Visual Reconstruction*. Cambridge, MA: MIT Press, 1987.
- [4] C. de Boor, *A Practical Guide to Splines*, vol. 27. Berlin, Germany: Springer-Verlag, 1978.
- [5] R. L. Eubank, *Spline Smoothing and Nonparametric Regression*. New York: Dekker, 1988.
- [6] F. R. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: An Approach Based on Influence Functions*. New York: Wiley, 1986.
- [7] P. J. Huber, *Robust Statistics*. New York: Wiley, 1981.
- [8] , "Robust smoothing," in *Robustness in Statistics*. New York: Academic, 1979.
- [9] M. Karczewicz, M. Gabbouj, and J. Astola, "Robust B-Spline image smoothing," in *Proc. IEEE Int. Conf. Image Processing*, Austin, TX, Nov. 1994, pp. 540–544.
- [10] M. Karczewicz, M. Gabbouj, and I. Defée, "Edge detection using robust B-spline," in *Proc. NORSIG-94, Nordic Signal Processing Symp.*, Ålesund, Norway, June 2–4, 1994, pp. 190–194.
- [11] K. J. Kerpez, "Minimum mean square error impulse noise estimation and cancellation," *IEEE Trans. Signal Processing*, vol. 43, pp. 1651–1662, 1995.

- [12] A. Nieminen, P. Heinonen, and Y. Neuvo, "A new class of detail-preserving filters for image processing," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-9, pp. 74–90, 1987.
- [13] I. Pitas and A. N. Venetsanopoulos, *Nonlinear Digital Filters*. Boston, MA: Kluwer, 1991, pp. 49–51.
- [14] W. H. Pun and B. D. Jeffs, "Adaptive image restoration using a generalized Gaussian model for unknown noise," *IEEE Trans. Image Processing*, vol. 4, pp. 1451–1456, 1995.
- [15] S. S. Sinha and B. G. Schunck, "A two-stage algorithm for discontinuity preserving surface reconstruction," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 36–55, 1992.
- [16] D. Terzopoulos, "The computation of visible-surface representations," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 10, pp. 417–438, 1988.
- [17] A. Thompson, J. Brown, and D. Titterton, "A study of methods of choosing the smoothing parameters in image restoration by regularization," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 326–339, 1991.
- [18] V. Torre and T. A. Poggio, "On edge detection," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 147–163, 1986.
- [19] R. Yang *et al.*, "Optimal weighted median filtering under structural constraints," *IEEE Trans. Signal Processing*, vol. 43, pp. 591–604, 1995.

## On "The Convergence of Mean Field Procedures for MRF's"

Jeffrey A. Fessler

In [1], Zhang attempts to establish convergence of a mean-field iteration for an Ising Markov random field (MRF) for large values of the hyperparameter  $\beta$ . Unfortunately, (16b), which states

$$|T_\delta(u_1) - T_\delta(u_2)| \leq |u_1 - u_2|$$

is not correct for the function  $T_\delta(u)$  defined in (15) and Fig. 2. In fact, any function that satisfies (16b) for all  $u_1$  and  $u_2$  is necessarily a continuous function, unlike the particular  $T_\delta(u)$  defined in (15).

Thus, the convergence of the mean field iteration remains an important open question for large  $\beta$ .

## REFERENCES

- [1] J. Zhang, "The convergence of mean field procedures for MRF's," *IEEE Trans Image Processing*, vol. 5, pp. 1662–1665, Dec. 1996.

Manuscript received February 6, 1997; revised August 19, 1997. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stephen E. Reichenbach.

J. A. Fessler is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48109-2122 USA (e-mail: fessler@umich.edu).

Publisher Item Identifier S 1057-7149(98)04004-4.