2–D ADAPTIVE PIECEWISE–LINEAR FILTER FOR IMAGE ENHANCEMENT

V. Pahor, G. Ramponi, G.L. Sicuranza
D.E.E.I., University of Trieste
via A. Valerio, 10, 34127 Trieste, Italy
Tel: +39 40 6767140; fax: +39 40 6763460
e-mail: pahor@imagets.univ.trieste.it

ABSTRACT

A two-dimensional adaptive nonlinear filter, called 2–D FIR–PWL filter is introduced for noise cancellation from images. It is based on the cascade of a linear FIR filter and a piecewise–linear interpolating function. Experimental results show a very good behaviour of the filter, which outperforms in many application examples the Sigma filter both in terms of visual quality and numerical results.

1 INTRODUCTION

Recently, some efforts have been made to employ conventional adaptive filters in noise cancellation from images. Due to their simplicity, linear 2–D LMS adaptive filters have been investigated first [1], which have shown however some limitations in suppressing broadband noise within natural images (image blurring, modest noise reduction). Indeed, in such applications it is a special requirement to preserve the edges and details of an image, performing in the meantime a significant noise smoothing.

In this paper, we present a nonlinear adaptive filter, called FIR–PWL filter, realized in the form of the Two-Dimensional Adaptive Line Enhancer (TDALE) [1]. The originality of this model is the introduction of an adaptive piecewise-linear (PWL) function, which permits to combine a strong smoothing action with a good edge preservation.

2 The 2–D FIR–PWL Adaptive Line Enhancer

The basic scheme of the 2–D FIR–PWL Adaptive Line Enhancer (ALE) is illustrated in Fig.1, where $\{x(m,n)\}$ is the $M \times M$ input image, given by the sum of the original image $\{s(m,n)\}$ and a wideband noise $\{v(m,n)\}$. This scheme is a 2–D nonlinear extension of the well-known one dimensional ALE, a noise canceller in which the function of the adaptive filter is to perform the prediction of the present value of the input signal, so as to give at the output the best estimate of the noiseless signal s. For this purpose, the operator D is provided in the ALE scheme, which is a delay in 1-D case, whereas in 2-D case is an operator that excludes the sample to be predicted from the input data window of the filter. This operator is also called decorrelation operator, since it decorrelates the output of the filter from the wideband noise in $x(m,n)$.

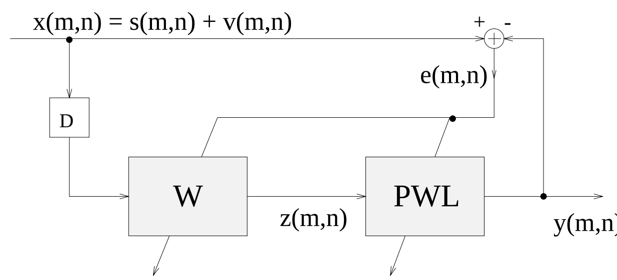

Figure 1: 2–D FIR–PWL Adaptive Line Enhancer

2.1 The I/O relation

So let $\boldsymbol{x}(m,n) = \{x(m-l,n-k), (l,k) \in S-G\}$ be the tensor of the input data and $\boldsymbol{w} = \{w(l,k), (l,k) \in S-G\}$ the tensor of the weights of the linear filter, where $S = \{(l,k), -\frac{N-1}{2} \le l,k \le \frac{N-1}{2}\}$ is the support region ($N \times N$-size whole-plane window) of the filter and G the element $\{0,0\}$, which is excluded by the decorrelation operator D from the window S. The output of the linear filter is therefore given by

$$\begin{aligned} z(m,n) &= \sum_{(l,k)\in S-G} w(l,k)x(m-l,n-k) \\ &= \boldsymbol{w} * \boldsymbol{x}(m,n) \end{aligned} \quad (1)$$

where $*$ denotes the inner product.

If the input image possesses 256 gray levels, we define a set of equally spaced breakpoints $\{\tau_1, \tau_2, \cdots, \tau_L\}$ in the range $[-\Delta\tau, 255 + \Delta\tau]$, where $\Delta\tau$ is the distance between two adjacent breakpoints. Then we may define the following continuous piecewise-linear function $f(z)$, which will map the samples $z(m,n)$:

$$y(z) = f(z) = \sum_{j=1}^{L} a_j h_j(z) \quad \forall\ z \in [\tau_1, \tau_L] \quad (2)$$

where $\{h_1, h_2, \cdots, h_L\}$ is the set of the basis functions, weighted in (2) by L coefficients $\{a_1, a_2, \cdots, a_L\}$. The generic basis function h_j has the following form (let us define $\tau_0 = \tau_1$ and $\tau_{L+1} = \tau_L$ for the sake of generality):

$$h_j(z) = \begin{cases} (z - \tau_{j-1})/\Delta\tau & if\ \tau_{j-1} \le z < \tau_j \\ (\tau_{j+1} - z)/\Delta\tau & if\ \tau_j \le z < \tau_{j+1} \\ 0 & elsewhere \end{cases} \quad (3)$$

for $j = 1, \cdots, L$. Note that the h_j's are triangular–shaped functions which are nonzero only in the interval $[\tau_{j-1}, \tau_{j+1}]$, except for h_1 and h_L which are nonzero respectively in $[\tau_1, \tau_2]$ and $[\tau_{L-1}, \tau_L]$; outside the defined input range the PWL function returns 0 for $z < \tau_1$ and 255 for $z > \tau_L$.

The output of the FIR–PWL filter is obtained substituting in (2) the variable z with the expression (1) of $z(m, n)$:

$$y(m, n) = \sum_{j=1}^{L} a_j h_j(\boldsymbol{w} * \boldsymbol{x}(m, n)) \quad (4)$$

which is the estimate of $s(m, n)$.

2.2 Adaptation algorithms

A fundamental requirement for deriving a common LMS training algorithm for a set of adaptive parameters is that the output of the filter be formed as a linear combination of those parameters. That requirement is not achieved by the FIR–PWL model, since in (4) there are products between the elements of $\{a_k\}$ and the weights of the filter $\boldsymbol{w}$. Therefore, the LMS adaptation algorithm is derived separately for each stage [3].

2.2.1 PWL function

The adaptive procedure for the PWL function minimizes the following mean square error (MSE):

$$\min_{\{a_k\}} E[e^2(m, n)] = \min_{\{a_k\}} E[(x(m, n) - \sum_{j=1}^{L} a_j h_j(\boldsymbol{w} * \boldsymbol{x}(m, n)))^2] \quad (5)$$

Considering that for a given argument $z(m, n) = \boldsymbol{w} * \boldsymbol{x}(m, n)$ only two basis functions $h_{\bar{\imath}}, h_{\bar{\imath}+1}$ can take nonzero values, we get the LMS algorithm which updates at each iteration only two coefficients $a_{\bar{\imath}}, a_{\bar{\imath}+1}$:

$$a_{\bar{\imath}}^{\beta+1} = a_{\bar{\imath}}^{\beta} + \alpha_S e(m, n) h_{\bar{\imath}}(z(m, n)) \quad (6)$$

$$a_{\bar{\imath}+1}^{\beta+1} = a_{\bar{\imath}+1}^{\beta} + \alpha_S e(m, n) h_{\bar{\imath}+1}(z(m, n)) \quad (7)$$

where α_S is a suitable step–size, β is the iteration index and the indices $\bar{\imath}, \bar{\imath}+1$ identify the two active basis functions at the current value of $z(m, n)$.

The coefficients a_j are estimated better if a variable step–size is considered. Indeed, it has been noticed that the final MSE may be reduced up to 10% by making the adaptation of the PWL function faster when the local statistics of the image change, while slowing it down in uniform areas (refer to [4] for more details).

2.2.2 FIR filter

For constructing the adaptation algorithm for the FIR filter we should use, in principle, a reference signal $r(m, n)$ for the signal $z(m, n)$, so as to be able to define the intermediate error $e^{'}(m, n) = r(m, n) - z(m, n)$. Since the error $e^{'}(m, n)$ cannot be determined, the approach which has been adopted is to replace $e^{'}(m, n)$ with $e(m, n)$ [3]. Under this assumption, the updating LMS algorithm for the linear stage may be written as follows:

$$\boldsymbol{w}^{\beta+1} = \boldsymbol{w}^{\beta} + \alpha_F e(m, n) \boldsymbol{x}(m, n) \quad (8)$$

where $e(m, n)$ is the output error defined in (5). Note that (8) is a rigorously derived LMS algorithm which minimizes the mean square error $E[e^2(m, n)]$ if the PWL function is a linearity with unitary slope.

Images are generally nonzero mean signals and therefore they are not suitable to train the filter, since the nonzero mean degrades the convergence property of an adaptive procedure. Therefore the local mean has actually to be known or estimated and then subtracted from the input signal [2]. We propose the following adaptation scheme that performs the mean estimation and subtraction [4]:

$$\boldsymbol{w}^{\beta+1} = \boldsymbol{w}^{\beta} + \alpha_F e(m, n)[\boldsymbol{x}(m, n) - \overline{x}(m, n) \cdot \boldsymbol{i}] \quad (9)$$

where $\boldsymbol{i}$ is the identity tensor and $\overline{x}(m, n)$ is the average of the elements of $\boldsymbol{x}(m, n)$, so that the sum of the elements of $[\boldsymbol{x}(m, n) - \overline{x}(m, n) \cdot \boldsymbol{i}]$ in Eq.(9) is always null. The FIR filter must be initialized, at step $\beta = 0$, with the following constraint:

$$\sum_{l,k \in S-G} w^{\beta}(i, j) = \boldsymbol{w}^{\beta} * \boldsymbol{i} = 1 \quad (10)$$

If the filter is trained using (9), it is easy to verify that (10) is assured $\forall \beta$. The constraint (10) allows to preserve the local mean of the input signal, so we may rewrite (1) as follows:

$$\begin{aligned} z(m, n) &= \boldsymbol{w} * [\boldsymbol{x}(m, n) - \overline{x}(m, n) \cdot \boldsymbol{i}] + \boldsymbol{w} * \boldsymbol{i} \cdot \overline{x}(m, n) \\ &= \boldsymbol{w} * [\boldsymbol{x}(m, n) - \overline{x}(m, n) \cdot \boldsymbol{i}] + \overline{x}(m, n) \quad (11) \end{aligned}$$

Equation (11) shows that the local mean is formally subtracted at the input and then added at the output of the filter, and consequently it proves that the I/O relation defined in (1) is still valid. If applied in a FIR–PWL scheme, the proposed method gives better results with respect to a conventional approach [2], where the local mean (estimated in an equivalent manner) is actually subtracted before applying the filtering.

2.3 Training and filtering phase

The measure of how closely the adaptive process tracks the optimal solution which minimizes the mean square error depends on several factors. First of all, it depends on both the degree of nonstationarity of the input signal (which is, in case of natural images, generally very high) and the tracking ability of the filtering scheme. The tracking ability of a FIR filter is inversely proportional to the number of its weights. In our case, relatively large masks are employed (up to 7×7, with 48 weights), in order to perform an adequate edge preservation. As far as the PWL function is concerned, actually it is not able to track the sudden variations of the statistics in an image. Therefore, the parameters of the FIR–PWL filter need to be estimated in more stationary environments through a training phase.

For this purpose, we decompose the input image in $B \times B$ subimages, each of which has generally more stationary statistics than the whole image. Before filtering each subimage, the locally optimal parameters of the filter are estimated by preprocessing the current subimage (training phase). Of course, during the filtering phase the parameters keep on being updated.

Filtering by subimages can give rise to an undesirable blocking effect in the output images. This drawback may be easily overcome by partially overlapping the input subimages and averaging the output values in the overlapped regions. The width of the overlapped regions may vary from a minimum of 1 to a maximum of $B/2$ pixels (in both dimensions).

3 Experimental results

Computer simulations have been performed using a set of standard images with different species of noise. The most interesting results are presented here, compared to the ones obtained from a Sigma filter [5]. They concern the images 'Peppers 512×512', 'Lena 256×256', 'Airfield 512×512' and 'Barbara 512×512'. The Sigma filter has been chosen for its good smoothing and edge–preserving properties, which are, especially in case of additive and multiplicative Gaussian noise, superior to those of many other nonlinear methods. The quality of the enhancement process is estimated in terms of MSE of the output image with respect to the original image, excluding from the image a border of $(N-1)/2$ pixels.

The presented algorithm has a wide set of parameters which have to be defined. Most of these parameters are left fixed or are varied in a restrict range of values: $L = 20$, $B = 16$ and $\alpha_F = 70 \div 150 \cdot 10^{-8}$, $\alpha_S = 0.003 \div 0.007$. Before filtering each subimage, the PWL function is initialized as a linearity of unitary slope, whereas the FIR filter is initialized as a low–pass filter only before filtering the first subimage, according to the constraint (10). The filter is trained in two scannings and the overlapping is performed on B/2=8 pixels.

The figures show the experimental results on image 'Peppers' (Fig.2-a), corrupted by Gaussian multiplicative noise of MSE=1000 (Fig.2-b). Fig.3-a shows the output of a 5×5 FIR–PWL filter, which yields in two sweeps a MSE of 85.1. It can be seen that the noise is strongly smoothed, details and edges appear quite intact and the sharpness is preserved, so that the overall aspect is pleasant. The smoothing action of the PWL function is evident in the uniform areas of the image, which is more effective than the one of the Sigma filter (Fig.3-b, MSE=113).

A similar effect is obtained, e.g., on the very noisy 'Airfield' image, corrupted by Gaussian additive noise ($SNR = 3dB$) [6], even if the PWL function tends to decrease the contrast of the details which lie on the uniform background. Such a behaviour slightly affects the quality of the image, whereas it influences significantly the MSE (refer to Table 1).

(a)

(b)

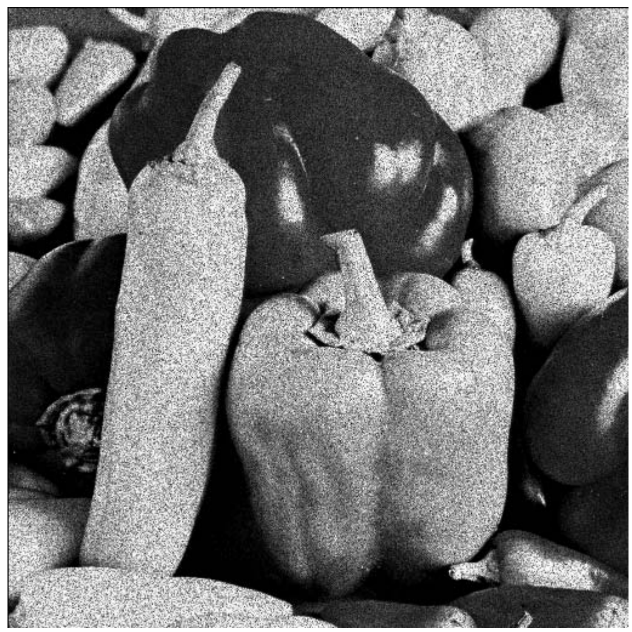

Figure 2: Original and noisy images

A 7×7 mask is suited for images rich of details and textures, like images "Lena" and "Barbara". The former image is degraded by medium strong additive Gaussian noise, the latter by contaminated Gaussian noise (of impulsive kind) for an SNR of 3dB. In both cases the filter preserves textural details, which are on the contrary lost by the Sigma filter. The superior behaviour of the FIR–PWL scheme is also assessed by the numerical results, which are presented in the following table:

Image	Input MSE	Filter type	N	S	Output MSE
Airfield	1791	FIR–PWL	5	2	279
		Sigma	5	2	308
Lena	379	FIR–PWL	7	2	81
		Sigma	5	2	108
Barbara	1490	FIR–PWL	7	2	125
		Sigma	5	2	331

Table 1: Numerical results

where N defines the size of the $N \times N$ mask and S is the number of sweeps. In the case of "Barbara" image, the FIR filter is initialized as a low–pass filter before each subimage is processed.

4 Conclusions

We have shown in this work that a conventional adaptive filtering technique may be very effective in image enhancement by introducing a simple nonlinear no-memory mapping together with an appropriate training of its parameters. On one side, the linear filter provides for a preliminary noise smoothing, which is performed by keeping sharp the processed image and discerning its local features (edges and textures). On the other side, the effect of the nonlinearity is a further noise cancellation with a good preservation of details, despite the absence of a real edge–detection mechanism.

Acknowledgements

This work has been partially supported by the European Project ESPRIT 20229 Noblesse.

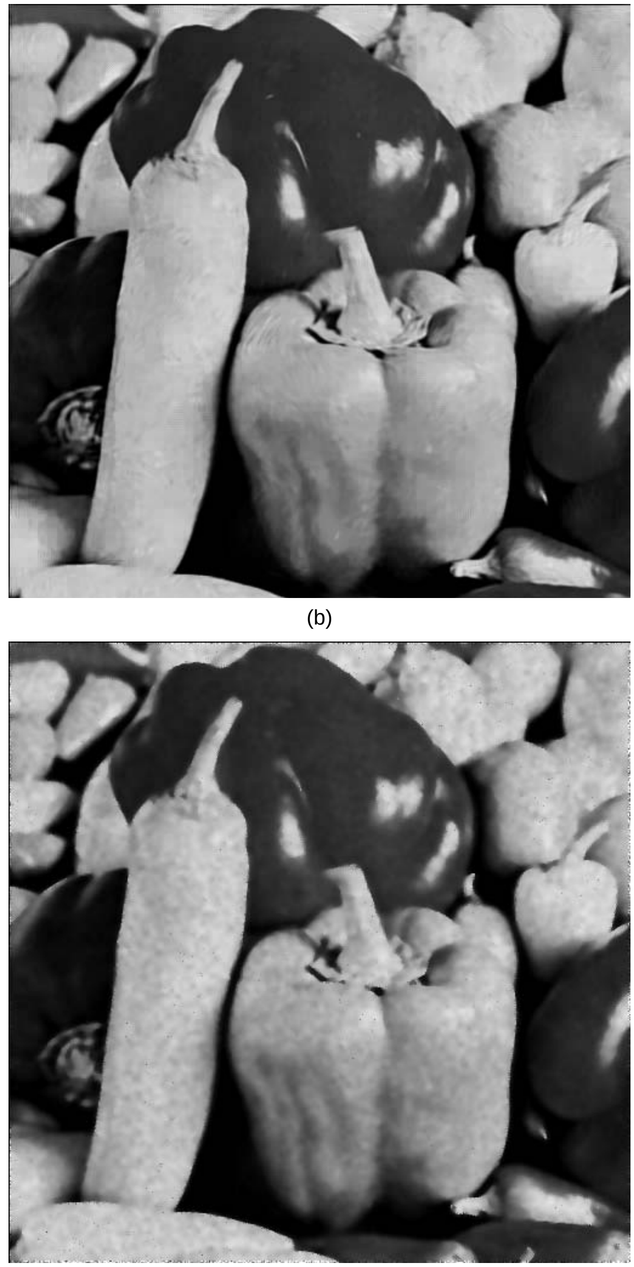

Figure 3: Processed images

References

[1] M. Hadhoud, D.W. Thomas, "The Two–Dimensional Adaptive LMS (TDLMS) Algorithm," *IEEE Trans. on Circuits and Systems*, Vol. 35, NO.5, pp.485-494, May 1988.

[2] J.N. Lin, R. Unbehauen, "Two–Dimensional LMS Adaptive Filter Incorporating a Local–Mean Estimator for Image Processing," *IEEE Trans. on Circuits and Systems-II: Analog and Digital SP*, Vol. 40, NO.7, pp.417-428, July 1993.

[3] G. Ramponi, A. Premoli, G.L. Sicuranza, "A New Piecewise–linear Model for Adaptive Nonlinear Digital Filtering," *Proc. Europ. Conf. on Circuit Theory and Design*, Copenhagen (DK), pp.595-604, Sept. 1991.

[4] V. Pahor, "Piecewise–linear adaptive models in one and two dimensions," *Technical Report*, ESPRIT 7130 NAT Project, September 1995 (PostScript file */pub/nat/pahor_nat7130.ps.Z* available via anonymous ftp at *imagets.univ.trieste.it*).

[5] J-S.Lee, "Digital Image Smoothing and the Sigma Filter," *Computer Vision, Graphics and Image Processing*, Vol.24, pp.255-269, 1983.

[6] M. Gabbouj, I. Tabus, "TUT Noisy Image Database," *Tampere University of Technology*, anonymous ftp at *sigftp.cs.tut.fi*.