

Moment-Preserving Methods for Signal Piecewise Approximation and Denoising

Amr Goneid

Department of Computer Science & Engineering,
American University in Cairo, Egypt
goneid@aucegypt.edu

Abstract

Approximations of signals in signal space are always necessary for noise removal and for lossy compression of the signals. Current techniques address the approximation process either in the signal space or in its transform space but not in both. A moment-preserving constraint can couple both spaces for better evaluation of approximations at nodal signal points. We present methods for applying moment-preserving piecewise approximations of signals using Linear, Quadratic and Cubic Spline polynomials. Results are given for experiments made to approximate 1-D signals, 2-D noisy boundaries and digitized images. In addition, we present a moment-preserving approach to the problem of denoising independent components derived from the Independent Component Analysis (ICA) of mixtures of noisy source signals.

Results demonstrate higher accuracy of the method compared to approximations obtained without moment preservation.

Keywords: *Signal Processing, Image Processing, Pattern Analysis, Piecewise approximations, Noisy ICA*

1. Introduction

Approximations of signals in signal space are always necessary for noise removal and for lossy compression of the signals. In the 1-D case, the noisy signal is considered to be a function $f(x)$ sampled at a set of distinct points $\{x_i, i = 0, 1, \dots, n\}$. The objective of an approximation method is to find an approximating function $g(x)$ defined at a set of distinct nodal points $\{z_j, j = 0, 1, \dots, m\}, m < n$, subject to a certain error minimization criterion. The approximated values can be joined using some interpolation technique. Obviously, this methodology also applies to 2-D signals or images.

The approximation problem is relevant to many applications in signal processing, pattern analysis and image processing, thus advancing a considerable amount of research in this area. Existing well known approaches derive the approximations either through constraints in the signal domain or in its transform domain but not in both. In the signal domain, several simple sub-sampling techniques have been commonly used [1] beside the more complex least squares polynomial methodology. Other approaches utilize piecewise linear approximations [2] as well as the smoother polynomial splines [3]. In the transform domain, common approaches are the FFT, DCT, and KLT transforms [e.g. 4, 5, 6].

More recently, a method developed in [7] approaches the problem by deriving the approximation in the signal domain while preserving a finite number of geometric moments that are related to its Fourier domain. The method has been applied to piecewise linear

approximations. In [8], we have examined how to extend the moment-preserving method for 1-D and 2-D signals to higher order polynomials.

In the present paper, we present the derivation of the moment preserving method for linear, quadratic and cubic spline polynomials. We show results of applying this technique to 1-D signals, closed binary boundaries and images. In addition, we present a moment-preserving approach to the problem of denoising independent components derived from the Independent Component Analysis (ICA) of mixtures of noisy source signals.

2. Theory

2.1 General

A random signal with a PDF $p(x)$ has a characteristic function that is the Fourier transform of its density function, i.e.

$$\phi(j\nu) = E_x[\exp(j\nu x)] = \int \exp(j\nu x) p(x) dx = \tau\{p(x)\} \tag{1}$$

If S_k represents the k^{th} geometric moment of the PDF then

$$S_k = E_x[x^k] = \int x^k p(x) dx = (-j)^k [d^k \phi(j\nu) / d\nu^k]_{\nu=0} \tag{2}$$

Suppose that the characteristic function has a Taylor-series expansion, then

$$\phi(j\nu) = \sum_k [d^k \phi(j\nu) / d\nu^k]_{\nu=0} \nu^k / k! = \sum_k S_k (j\nu)^k / k! \tag{3}$$

Therefore, if the characteristic function has a Taylor-series expansion valid in some region about the origin, it is uniquely determined in this interval by the geometric moments. If the moments do uniquely determine the characteristic function (and hence the Fourier transform of the density function) then they also uniquely determine the density function. The consequence of this uniqueness is that a moment-preserving approximation to the function $p(x)$ in the x -domain will also serve as an approximation constraint in the ν -domain.

2.2 Moment-Preserving approximation

Consider the k^{th} moment of the variable x over the finite interval (i,j) of the function $f(x)$ to be $S_k(i,j) = E_x[x^k]_{i,j}$. Let α be a scale reduction factor so that $x = \alpha y$ and hence we define a scaled moment as

$$\sigma_k(i,j) = \alpha^{-(k+1)} S_k(i,j) = \int_{i,j} y^k f(\alpha y) dy = E_{\alpha y}[y^k]_{i,j} \tag{4}$$

With the function $f(x)$ specified by a finite set of discrete points $\{x_i, i = 0, 1, \dots, n\}$, the scaled moment σ_k is the sum over all (n) segments $(i,i+1)$ covering the above domain:

$$\sigma_k = \sum_i \sigma_k(i,i+1) \quad , \quad i = 0, 1, \dots, n-1 \tag{5}$$

On the other hand, if we seek an approximating function $g(x)$ defined at a set of distinct nodal points $\{z_j, j = 0, 1, \dots, m\}$, then over the interval between two nodal points (p,q) we

obtain scaled moments $\mu_k(p,q)$ whose sum over the nodal intervals gives the scaled moments μ_k . For the moment preserving approximation, we require that

$$\sigma_k = \mu_k \quad \text{for } k = 0,1, \dots, m \tag{6}$$

The above moment-preserving constraint leads to a system of $m+1$ equations:

$$\sigma = E \cdot G \tag{7}$$

where E is an $m+1$ by $m+1$ square matrix of coefficients depending on the approximating polynomial, and G is a column vector representing the approximations $g(z_j)$ to the function $f(x)$ at the nodal points $\{z_j, j = 0, 1, \dots, m\}$.

3. Piecewise linear approximation

Assuming that between the nodal points z_p and z_q the function is piecewise linear, then we can use Lagrange's classical formula

$$g_{pq}(\alpha y) = (y_q - y)/(y_q - y_p) g(\alpha y_p) + (y - y_p)/(y_q - y_p) g(\alpha y_q) \tag{8}$$

to compute the k^{th} scaled moment over that region:

$$\mu_k(p,q) = \alpha^{-(k+1)} S_k(p,q) = \int_{p,q} y^k g_{pq}(\alpha y) dy = E_{\alpha y} [y^k]_{p,q}$$

With $D_{pq}(t,n) = \int_{p,q} t^n dt = (t_q^{n+1} - t_p^{n+1})/(n+1)$ then evaluation of the integral gives

$$\mu_k(p,q) = g(\alpha y_p) B_q(q,k,y) - g(\alpha y_q) B_q(p,k,y) \tag{9}$$

where $B_q(b,k,t) = \{t_b D_{pq}(t,k) - D_{pq}(t,k+1)\} / D_{pq}(t,0)$

For nodal points $\{z_j, j = 0, 1, \dots, m\}$, then the scaled moments for the m segments will be:

$$\mu_k(0,1) = g(\alpha y_0) B_1(1,k,y) - g(\alpha y_1) B_1(0,k,y)$$

$$\mu_k(1,2) = g(\alpha y_1) B_2(2,k,y) - g(\alpha y_2) B_2(1,k,y)$$

.....

$$\mu_k(m-1,m) = g(\alpha y_{m-1}) B_m(m,k,y) - g(\alpha y_m) B_m(m-1,k,y)$$

Let us define a function

$$C_j(k,t) = \begin{matrix} B_j(1,k,t) & \text{for } j = 0 \\ B_{j+1}(j+1,k,t) - B_j(j-1,k,t) & \text{for } j = 1, \dots, m-1 \\ - B_m(m-1,k,t) & \text{for } j = m \end{matrix} \tag{10}$$

Hence, the total scaled moment can be expressed as a vector μ with elements

$$\mu_k = \sum_j C_j(k,y) g(\alpha y_j) \quad , \quad j, k = 0,1,\dots,m \quad (11)$$

Notice that in the above equation, the values of y_j represent the scaled coordinates of the nodal points. When the scaled coordinates of the actual function points are used, then we obtain the actual scaled moments vector σ . Accordingly, moment preservation ($\mu = \sigma$) leads to the system

$$G = E^{-1} \cdot \sigma \quad (12)$$

where the elements of the square matrix E are given by $e(k,j) = C_j(k,y)$

4. Piecewise quadratic approximation

Here, we use equally spaced nodal points with an internal point z_r between the points z_p and z_q . With $\Delta = (z_r - z_p) = (z_q - z_r)$, Lagrange's formula can be written in the form:

$$2\Delta^2 g_{pq}(x) = (x - z_r)(x - z_q)g(z_p) - 2(x - z_p)(x - z_q)g(z_r) + (x - z_p)(x - z_r)g(z_q) \quad (13)$$

With $\mu_k(p,q) = \int_{p,q} y^k g_{pq}(\alpha y) dy$, one obtains

$$\mu_k(p,q) = g(\alpha y_p) B_{p,q}(r,q,k,y) - 2g(\alpha y_r) B_{p,q}(p,q,k,y) + g(\alpha y_q) B_{p,q}(p,r,k,y) \quad (14)$$

where

$$B_{p,q}(i,j,k,t) = [2/D_{pq}^2(t,0)] \{ D_{pq}(t,k+2) - (t_i + t_j) D_{pq}(t,k+1) + t_i t_j D_{pq}(t,k) \}$$

Similar to the method used for the piecewise linear approximation, we may write

$$\mu_k = \sum_j C_j(k,y) g(\alpha y_j) \quad , \quad j, k = 0,1,\dots,m, \quad m \text{ even}$$

where

$$C_j(k,y) = \begin{cases} B_{0,2}(1,2,k,y) & \text{for } j = 0, \\ -2 B_{j-1,j+1}(j-1,j+1,k,y) & \text{for } j \text{ odd,} \\ B_{j-2,j}(j-2,j-1,k,y) + B_{j,j+2}(j+1,j+2,k,y) & \text{for } j \text{ even,} \\ B_{m-2,m}(m-2,m-1,k,y) & \text{for } j = m \end{cases} \quad (15)$$

As before, the elements of the matrix E are $e(k,j) = C_j(k,y)$

5. Piecewise cubic spline approximation

Lagrange’s formula for the piecewise linear interpolation in the interval between z_p and z_q may be written in the form:

$$g_{pq}(z) = a g(z_p) + b g(z_q)$$

with $a = (z_q - z)/(z_q - z_p)$ and $b = (z - z_p)/(z_q - z_p) = 1 - a$

In a cubic spline approximation, we add a cubic polynomial whose second derivative varies linearly over the (p,q) interval and with zero values at z_p and z_q leading to

$$g_{pq}(z) = a g(z_p) + b g(z_q) + c g''(z_p) + d g''(z_q)$$

where

$$c = (1/6) (a^3 - a)(z_q - z_p)^2 \quad \text{and} \quad d = (1/6) (b^3 - b)(z_q - z_p)^2$$

Hence, the interpolating polynomial can be expressed as:

$$g_{pq}(\alpha y) = [(y_q - y)/\Delta_y] g(\alpha y_p) + [(y - y_p)/\Delta_y] g(\alpha y_q) + P(\alpha y) + Q(\alpha y) \tag{16}$$

where $\Delta_y = y_q - y_p$,

$$P(\alpha y) = [(y_q - y)^3 - \Delta_y^2 (y_q - y)] (\alpha^2/6 \Delta_y) \varphi_p$$

$$Q(\alpha y) = [(y - y_p)^3 - \Delta_y^2 (y - y_p)] (\alpha^2/6 \Delta_y) \varphi_q$$

φ_p and φ_q are the second derivatives at the two nodal points.

Let $U_k(p,q) = \int_{p,q} y^k [P(\alpha y) + Q(\alpha y)] dy$, so that

$$\mu_k(p,q) - U_k(p,q) = g(\alpha y_p) B_q(q,k,y) - g(\alpha y_q) B_q(p,k,y)$$

where $B_q(b,k,t)$ is as defined for the linear case. Therefore, the problem is similar to the linear case except for the term $U_k(p,q)$. Evaluation of this term gives

$$U_k(p,q) = D_{pq}(y,k) [\gamma_p \beta_0(q) - \gamma_q \beta_0(p)] + D_{pq}(y,k+1) [\gamma_p \beta_1(q) - \gamma_q \beta_1(p)] + D_{pq}(y,k+2) [\gamma_p \beta_2(q) - \gamma_q \beta_2(p)] - D_{pq}(y,k+3) [\gamma_p - \gamma_q] \tag{17}$$

where $\gamma_p = (\alpha^2/6 \Delta_y) \varphi_p$, $\beta_0(i) = y_i^3 - \Delta_y^2 y_i$, $\beta_1(i) = \Delta_y^2 - 3 y_i^2$,
 $\beta_2(i) = 3 y_i$

It follows that

$$\mu_k - U_k = \sum_j C_j(k,y) g(\alpha y_j) \quad , \quad j, k = 0,1, \dots, m \tag{18}$$

where $U_k = \sum_j U_k(j-1,j)$, $j = 1,2, \dots, m$

Accordingly, moment preservation ($\mu = \sigma$) leads to the system

$$G = E^{-1} \cdot (\sigma - U_{\mu} + U_{\sigma}) \tag{19}$$

In the above equation, E and σ are respectively the coefficient matrix and moments vector for the piecewise linear approximation, while the vectors U_{μ} and U_{σ} are computed using the nodal points and the actual function points, respectively.

6. Experimental results on signal approximation

6.1 1-D signal approximation

The above moment-preserving (MP) method has been applied to obtain piecewise approximations for various 1-D signals $f(x)$. For a given approximation, nodal points $\{z_j, j = 0, 1, \dots, m\}$ were chosen to be evenly spaced across the x -space. The vector of approximants G at those points was computed using the linear, quadratic or cubic spline methods outlined above and an approximation $g(x)$ to the function is obtained by the respective interpolation method. For comparison with usual interpolation techniques, an approximation $h(x)$ was also obtained using the function values $f(z_j)$. We have used the mean-squared error (MSE) as a measure of the error norm between $f(x)$ and each of the approximations $g(x)$ and $h(x)$.

As an example, we show here the results for the function

$$f(x) = 2\sin(0.2 x) + 5\cos(0.3 x) + w r \tag{20}$$

where r represents additive Laplacian random noise and w is an amplitude factor. For the above example, we have used an x -domain covering 10 blocks with 1601 function points and 5 nodal points in each block (i.e. one nodal point every 40 function points). For more accuracy and to reduce the need for reconditioning the matrices in the inversion process, we have used a scale factor $\alpha =$ standard deviation of x over the block.

As an example, Figure (1) compares the obtained piecewise approximations without and with moment-preserving constraint for the linear, quadratic and cubic spline approximations.

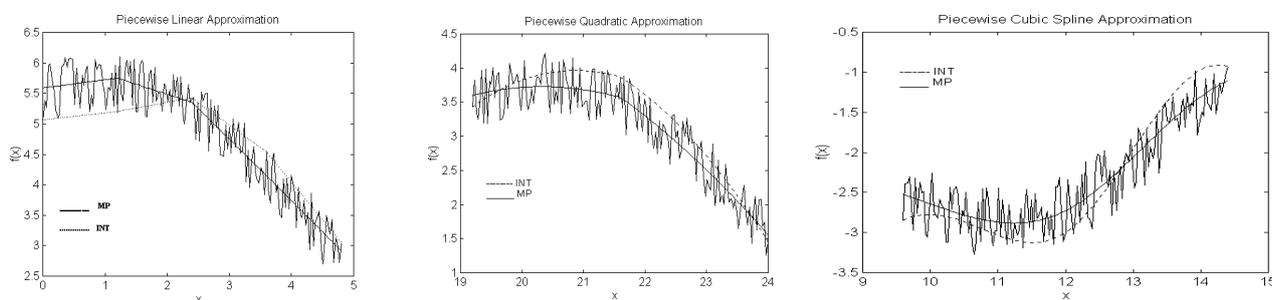


Figure (1): Piecewise linear, quadratic and cubic spline approximations.

To examine the effect of noise on piecewise approximations, we have computed the Mean Square Error (MSE) between the original noisy signals and the corresponding ones derived by appropriate interpolation from the values at the knot points. We have also included for comparison those function values derived by interpolation of knot point values obtained from a filtered set using a 5-point Gaussian filter.

Figure (2) shows the MSE as a function of signal-to-noise ratio (SNR) for the linear approximation. The solid and dotted curves relate to moment-preserving and usual interpolation MSE, respectively, while the dashed curve represents the results from the Gaussian filter.

Figure (2) clearly shows the expected decrease of the MSE by increase of the SNR level and also clearly illustrates how the MSE is significantly decreased, even at high noise level, by imposing moment preservation. It also shows that the MP approximation is superior to the usual interpolation methods and also to Gaussian filtering of the signals. Similar results are obtained for linear and cubic spline approximations.

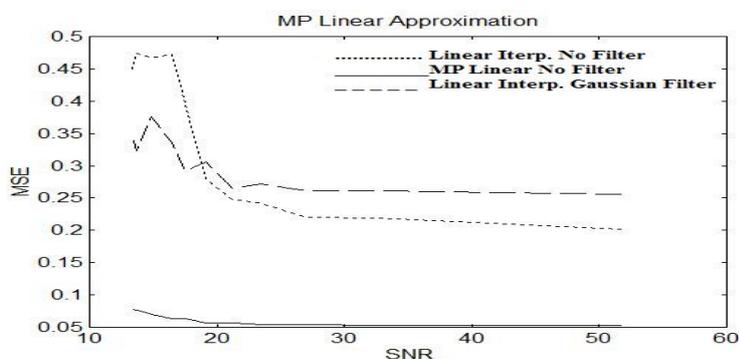


Figure (2): MP and usual linear approximations.

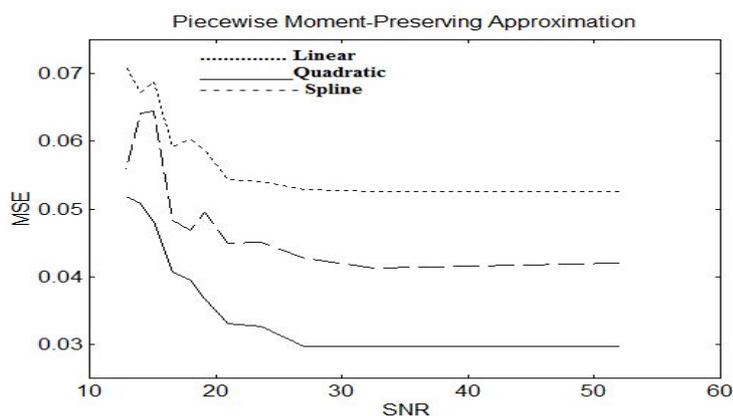


Figure (3): Comparison between the three MP approximations

In Figure (3), we show the results for the MSE obtained for the three different moment-preserving approximations at different noise levels. The results indicate that the quadratic approximation is better than the other two. Although cubic splines produce smoother

approximations on smaller x -scales, higher accuracy is obtained with the quadratic approximation on larger scales.

In general, the moment preserving methods given here prove to be able to achieve high packing ratios (e.g. 1/40) with high reconstruction accuracy and also to denoise the signals.

6.2 2-D closed boundaries

We have also applied the present methodology to noisy closed binary boundaries by obtaining their $r(\theta)$ signatures and computing the approximations for the resulting 1-D signatures. The θ -space, normalized to $\{0,2\pi\}$, is divided into 8 sectors. In each sector, one nodal θ point is selected every 40 points.

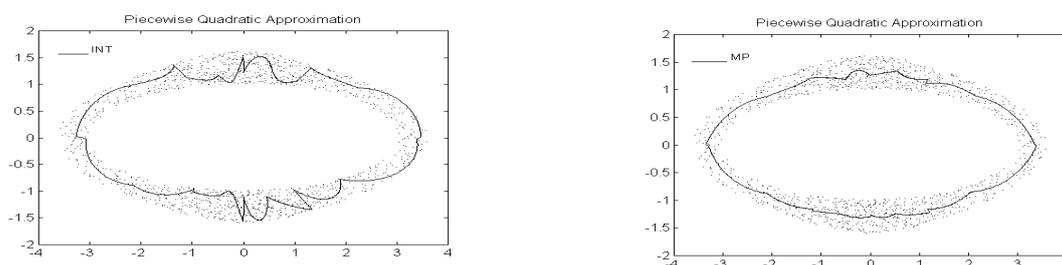


Figure (4): Samples of approximations of noisy closed boundaries

A sample of the results obtained for a noisy closed boundary using usual interpolation and moment-preserving quadratic approximation is shown in Figure (4). The problem with using a piecewise approximation without moment preservation is quite clear in the presence of significant noise levels. Values of the function at the nodal points could well be extremum points of noise amplitude leading to fluctuations between these points. The results shown in the figure indicate that using MP approximations is clearly superior to the usual interpolation methods.

A typical dependence of the MSE on the noise level in a boundary is shown in Figure (5). It can also be seen from this figure that imposing the moment-preserving criteria leads to a significant reduction in the MSE resulting from a high level of noise on the boundary.

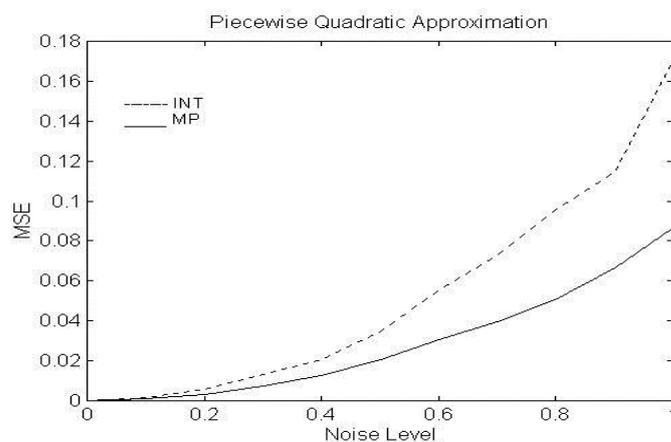


Figure (5): Typical MSE graph for boundary approximation.

It should be noted that the degree of accuracy of moment-preserving approximation for noisy boundaries does not depend strongly on the degree of the approximating polynomial. The results we obtained show that the MSE for the same noise level will not differ significantly between linear, quadratic and cubic spline approximations. Therefore, a low order polynomial with moment preservation can serve to define the skeleton of a noisy shape to a degree of accuracy significantly higher than the usual piecewise interpolation techniques.

Figure (6) shows some examples of skeleton detection using quadratic moment-preserving piecewise approximations. Again MP methods prove to be able to achieve high packing ratios (e.g. 1/40) with high reconstruction accuracy and significant denoising of the noisy closed boundaries.

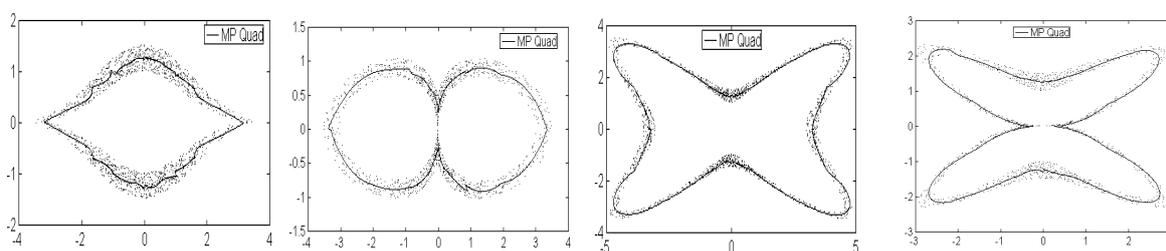


Figure (6): Examples of skeleton detection from noisy boundaries using quadratic moment-preserving approximations.

7. Approximation of Digitized Images

An image, considered as a matrix of pixel values, can be approximated to achieve lossy compression. The most efficient technique used so far is the Discrete Cosine Transform (DCT) which operates in the frequency domain rather than the spatial domain of the image. The usual method for applying the DCT is to process image blocks of size 8 x 8 pixels using a 2-D DCT. A zigzag mapping is used to preserve DCT coefficients of maximum variance while setting the rest of the 64 coefficients to zero.

Beside its low computational complexity, the advantages of using the DCT is that it packs the information in the maximum variance coefficients and that it minimizes the boundary discontinuities between the blocks.

In order to apply moment-preserving techniques to a digitised image, we use a piecewise approximation in the spatial domain. Block processing can be used after mapping the block pixels into a 1-D vector from which nodal points can be selected for the approximation process using the methods developed in the present work.

We have experimented with different sub-image geometries and have selected column processing as the block processing method. In this case, a block is chosen to be a sub-column with a number of pixels depending on the degree of variance in the column. As an example, we have used a block size of 17 pixels with 5 nodal points (one nodal pixel every 4 pixels). The sample image whose results are show here is a 256 grey level image of size 256 x 256 pixels (Figure (7a)).

Using usual quadratic interpolation and moment-preserving quadratic approximation, the resulting images are shown in Figure (7b) and Figure (7c), respectively. The corresponding MSE obtained for the two approximations are 0.0037 and 0.0023, respectively.

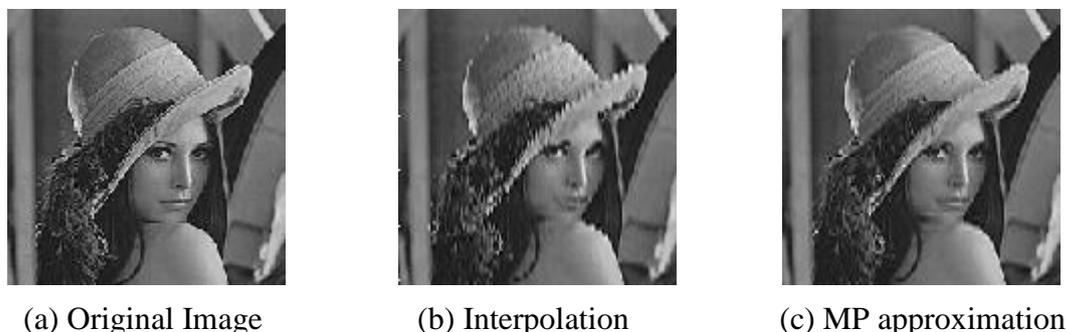


Figure (7): Image quadratic approximations

This corresponds to a reduction of about 38% in the MSE and is also apparent from comparing image 7(b) with image 7(c). It is therefore evident that the use of the moment-preserving approximation significantly enhances the quality of the image relative to the usual interpolation method. Similar conclusions have been obtained from the use of the lower order (linear) approximating polynomial.

In order to compare the present spatial approximation method with frequency domain processing, we have used the DCT with a block size of 8 x 8 pixels and preserving the highest variance K coefficients of the zigzag mapped 64 coefficients for each block. We have also processed the blocks after adding N nodal coefficients from the remaining (64 - K). These nodal points have been samples every 4 coefficients and their values have been determined by a moment-preserving quadratic method. With K = 12, and N = 3, the packing ratio is approximately comparable to the case of spatial processing shown above.

Figure (8) shows the reconstructed images using the above parameters, and Table (1) gives the MSE for different values of K and N.

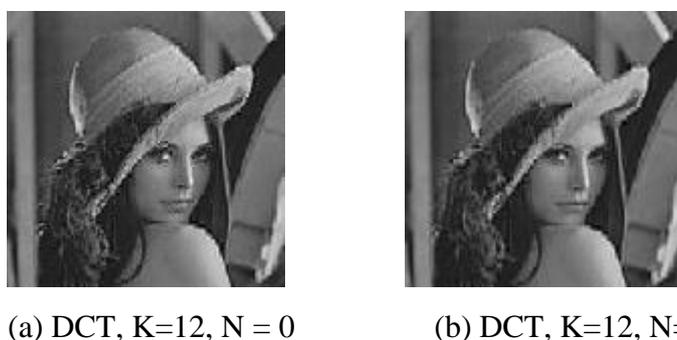


Figure (8) DCT Processing without and with N moment-preserving quadratic nodal points.

It is to be noted that the MSE of 0.0023 obtained for quadratic spatial processing (corresponding to image 7(c)) is quite close to the value of 0.0022 obtained from the DCT with $K = 12$ and $N = 3$ moment preserving quadratic nodal points. This indicates that moment-preserving approximations of images in the spatial domain can compete with transform methods such as the DCT as far as packing efficiency is concerned. However, we must recognize the advantage of low computational complexity offered by the DCT relative to moment-preserving spatial processing. This is in view of the latter methods being dependent on matrix inversion computations that increase their computational cost for image approximations.

Table (1) MSE of DCT Processing for Different Packing Parameters ($\times 10^{-4}$)

K	8	8	12	12	16	16
N	0	3	0	3	0	3
MSE	30	26	25	22	19	19

8. Denoising Independent Components in noisy ICA

8.1 Noisy ICA

In order to generate Independent Components (IC's) from a matrix X representing the mixture of independent sources, we consider the ICA instantaneous linear noiseless mixing model represented by:

$$X = A S, \tag{21}$$

where S is a random matrix of hidden sources with mutually independent components, and A is a non-singular mixing matrix. Given X , the basic problem is to find an estimate Y of S and the mixing matrix A such that:

$$Y = W X = W A S = G S \approx S, \tag{22}$$

where $W = A^{-1}$ is the *unmixing* matrix, and $G = W A$ is usually called the Global Transfer Function or Global Separating-Mixing (GSM) Matrix. The linear mapping W is such that the unmixed signals Y are statistically independent. However, the sources are recovered only up to scaling and permutation. In practice, the estimate of the unmixing matrix W is not exactly the inverse of the mixing matrix A . Hence, the departure of G from the identity matrix I can be a measure of the error in achieving complete separation of sources.

The case of noisy ICA where source noise is additive can be modelled in a way similar to the noiseless model. In this case we may write:

$$X = A (S + n) = A U \tag{23}$$

In this model we assume that the noise \mathbf{n} is independent from the sources \mathbf{S} and we may consider the special but not uncommon case of a noise covariance of the form $\sigma^2 \mathbf{I}$ so that the noisy sources \mathbf{U} are still non-gaussian and independent. In this case, the \mathbf{U} matrix can be estimated by ordinary ICA methods. Since noise is independent of the sources, it is then possible to use a method for denoising the noisy sources \mathbf{U} . In our case, we propose to use a moment preserving (MP) approximation to achieve such denoising process.

The estimation of the unmixing matrix \mathbf{W} cannot be done in closed form. Instead, solution methods are based on finding maxima or minima of some objective function [9, 10, 11]. The most two famous methods seek an estimate of \mathbf{W} either based on maximizing the negentropy (negative entropy) or by using Maximum Likelihood Estimation (MLE). Such approaches require that the solution advances iteratively in steps starting from some initial estimate until it converges to the final solution. Learning from the data is required in each of these steps leading to essentially neural unsupervised learning algorithms.

8.2 The neural learning algorithm

For computing the noisy independent components \mathbf{U} from the observed mixtures \mathbf{X} , we adopt the modified algorithm given by [12] which is based on the Fast ICA algorithm originally given by [13]. Basically, the algorithm uses a fixed-point iteration method to maximize the negentropy using a Newton iteration method. We assume that the mixture matrix \mathbf{X} is of m separate sequences each of length n samples. Such matrix is to be pre-processed by centering followed by whitening or sphering to remove correlations.

Centering removes means via the transformation $\mathbf{X} \leftarrow \mathbf{X} - E\{\mathbf{X}\}$ and whitening is done using a linear transform (PCA like) $\mathbf{Z} = \mathbf{V}\mathbf{X}$ where \mathbf{V} is a whitening matrix. A popular whitening matrix is $\mathbf{V} = \mathbf{D}^{-1/2} \mathbf{E}^T$, where \mathbf{E} and \mathbf{D} are the eigenvector and eigenvalue matrices of the covariance matrix of \mathbf{X} , respectively. The resulting new matrix \mathbf{Z} is therefore characterized by $E\{\mathbf{Z}\mathbf{Z}^T\} = \mathbf{I}$ and $E\{\mathbf{Z}\} = 0$. After obtaining the unmixing matrix \mathbf{W} from whitened data, the total unmixing matrix is then $\mathbf{W} \leftarrow \mathbf{W}\mathbf{V}$. The algorithm estimates several or all components in parallel using symmetric orthogonalization by setting $\mathbf{W} \leftarrow (\mathbf{W}\mathbf{W}^T)^{-1/2} \mathbf{W}$ in every iteration.

In this modified version of the algorithm, the performance during the iterative learning process is measured using the matrix $\mathbf{G} = \mathbf{W}\mathbf{A}$, which is supposed to converge to a permutation of the scaled identity matrix at complete separation of the IC's. This is done by decomposing $\mathbf{G} = \mathbf{Q}\mathbf{P}$, where \mathbf{P} is a positive definite *stretching* matrix and \mathbf{Q} is an orthogonal *rotational* matrix. The cosine of the rotation angle is to be found on the diagonal of \mathbf{Q} so that a convergence criterion is taken as $\Delta |diag(\mathbf{Q})|_{min} < \epsilon$, where ϵ is a threshold value. Also, In this algorithm, we use the performance (error) measure, $E3$ introduced in [12]:

$$E3 = \frac{1}{2m(m-1)} \sum_{i=1}^m \left\{ \sum_{j=1}^m |g_{ij}| - M_i + |M_i - 1| \right\} + \sum_{j=1}^m \left\{ \sum_{i=1}^m |g_{ij}| - M_j + |M_j - 1| \right\} \quad (24)$$

where g_{ij} is the ij^{th} element of the matrix \mathbf{G} of dimensions $m \times m$, $M_i = \max_k |g_{ik}|$ is the absolute value of the maximum element in row (i) and $M_j = \max_k |g_{kj}|$ is the corresponding quantity for column (j). It is shown in [12] that the index $E3$ is more precise than the commonly used $E1$ and $E2$ indices [e.g., 12] and is independent of the matrix dimensions. It is also normalized to the interval $\{0,1\}$, the greater the value of $E3$, the worse is the performance.

The algorithm is summarized in the following steps:

- Preprocess mixtures \mathbf{X} to get \mathbf{Z}
- Choose random initial orthonormal vectors \mathbf{w}_i to form initial \mathbf{W} and random \mathbf{A}
- Set $\mathbf{W}_{old} \leftarrow \mathbf{W}$
- Iterate:
 1. Do Symmetric orthogonalization of \mathbf{W} by setting $\mathbf{W} \leftarrow (\mathbf{W} \mathbf{W}^T)^{-1/2} \mathbf{W}$
 2. Compute dewhitened matrix \mathbf{A} and new $\mathbf{G} = \mathbf{W} \mathbf{A}$ and do polar decomposition of $\mathbf{G} = \mathbf{Q} \mathbf{P}$
 3. Compute error $E3$
 4. If not the first iteration, test for convergence: $\Delta | \text{diag}(\mathbf{Q}) |_{min} < \varepsilon$
 5. If converged, break.
 6. Set $\mathbf{W}_{old} \leftarrow \mathbf{W}$
 7. For each component \mathbf{w}_i of \mathbf{W} , update using learning rule

$$\mathbf{w}_i \leftarrow E\{\mathbf{z} f(\mathbf{w}_i \mathbf{z})\} - E\{f'(\mathbf{w}_i \mathbf{z})\} \mathbf{w}_i$$
- After convergence, dewhiten using $\mathbf{W} \leftarrow \mathbf{W} \mathbf{V}$
- Compute independent noisy components $\mathbf{U} = \mathbf{W} \mathbf{X}$

Algorithm (1): ICA algorithm for many components

In step 7 in the iteration loop, \mathbf{z} is a column vector representing one sample from the whitened matrix \mathbf{Z} , $f(y)$ is a non-linearity function, $f'(y)$ is its derivative and the expectation is taken as the average over the n samples in \mathbf{Z} .

The non-linearity $f(y)$ is essential in the optimization process and for the learning rule that updates the estimates of the unmixing matrix \mathbf{W} and, overall, it is important for the stability and robustness of the convergence process. It is common to use non-linearities (NL's) $f(x)$ that are derived from assumed source models such that:

$$f(x) = -\frac{\partial p(x) / \partial x}{p(x)} \tag{25}$$

where $p(x)$ is the PDF of the source [14]. Representative source models with symmetrical unimodal PDF's are simple to analyze statistically and lead to computationally efficient algorithms. As an example, the source PDF $p(x) = 1/\cosh x$ leads to the general purpose function $f(x) = \tanh(x)$. Such unimodal source distributions provide acceptable performances in the case of Blind Source Separation (BSS) of super-gaussian sources.

8.3 Experiments on denoising independent components

We have conducted a number of experiments to test the process of obtaining the noisy independent components (IC's) from a noisy mixture of signals and then denoising these IC's using moment preserving (MP) approximations as outlined before. For the purpose of these experiments, we have chosen noisy source signals U_i , $i = 1..m$, with $m = 5$ sources each of length $N = 1201$ samples as follows:

$$\begin{aligned}
 U_1(x) &= \sin(0.02x) + 3\cos(0.03x) + w r \\
 U_2(x) &= 3\sin(0.005x)\cos^2(0.01x) + w r \\
 U_3(x) &= 3\sin(0.02x)\cos^3(0.04x) + w r \\
 U_4(x) &= w r \\
 U_5(x) &= \text{Arbitrary noiseless periodic function}
 \end{aligned}
 \tag{26}$$

where $x = 0, 1, \dots, N-1$

For the above signals, the noise vector $n_i = w r$, where w is a noise weight factor and r is a random Laplacian noise component with probability density function (PDF):

$$p(x) = \frac{1}{2b} \exp\left(-\frac{|x - \mu|}{b}\right)
 \tag{27}$$

where b is the scale parameter and μ is the location parameter. The random Laplacian variates r have been generated using the relation:

$$r = \mu - b \operatorname{sgn}(R) \ln(1 - |R|)
 \tag{28}$$

where R is a uniform random variate over the interval $\{-1/2, +1/2\}$.

The noisy signals (26) are all non-gaussian, and their normalized kurtosis are calculated using the relation:

$$k_n(S) = \frac{E\{S^4\}}{E^2\{S^2\}} - 3
 \tag{29}$$

From the kurtosis, it is found that all generated sources are super-gaussian except $S_1(x)$ which is sub-gaussian. Hence it is found convenient to use $f(x) = \tanh(x)$ as a non-linearity in the ICA algorithm.

The mixed noisy signals $X = A U$ have been obtained from the noisy sources (26) using a random initial mixing matrix A . Figure (9) shows these mixtures using a noise weight $w = 1.0$.

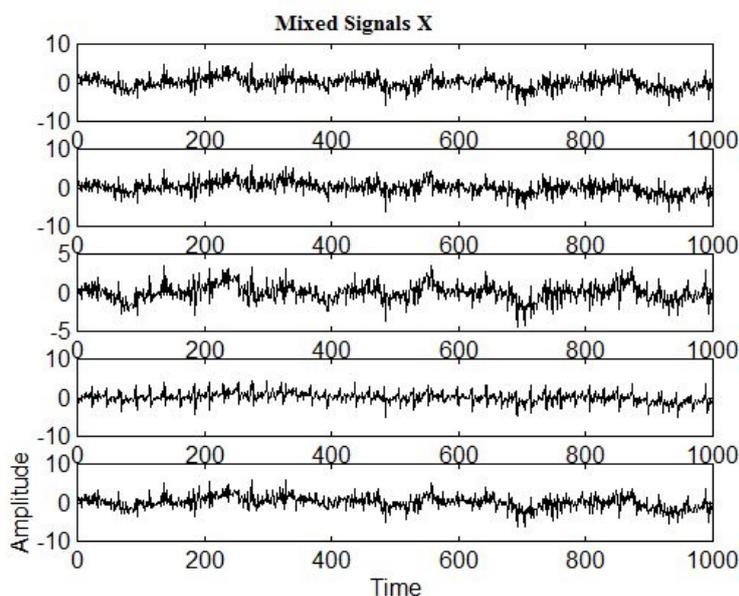


Figure (9): The mixed signals X

Using the ICA algorithm given above, we obtained estimates of the noisy IC's U as shown in Figure (10).

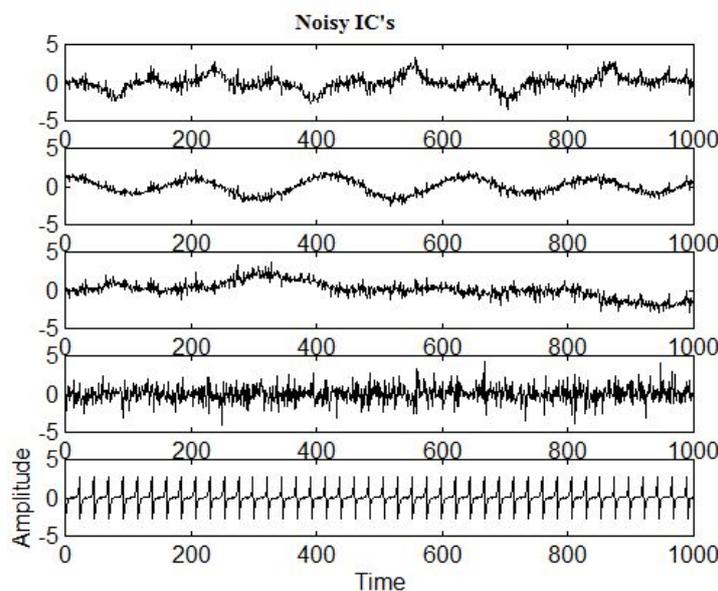


Figure (10): The noisy IC's U

We have performed the denoising process on the above shown IC's for U_1 , U_2 and U_3 using a quadratic moment preserving (MP) approximation with 40 blocks for U_1 and 30 blocks for each of U_2 and U_3 and 11 knot points per block for all of them. In each case, the denoised signals are obtained by quadratic interpolation between the knot points followed by a 5-point Gaussian filtering to remove discontinuities between the different blocks. The denoised IC's obtained by this method are shown in Figure (11).

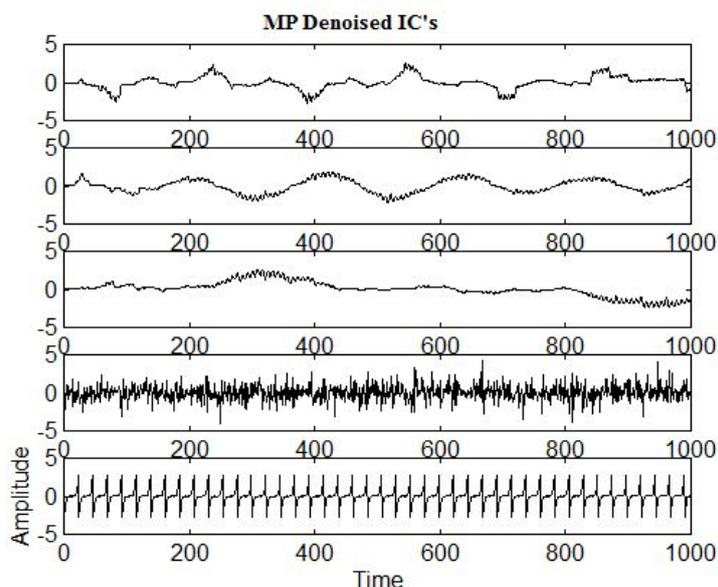


Figure (11): Denoised IC's S

It can be seen from the above figure that the MP approximation has achieved a satisfactory denoising of the IC's.

9. Conclusions

Moment-preserving piecewise approximations have been derived for linear, quadratic and cubic spline polynomials. The application of such approximations to noisy 1-D signals and to noisy 2-D boundaries has proven to be superior to the use of ordinary interpolation methods, especially for high level noise content. In case of 1-D signals, the quadratic moment-preserving approximation is more accurate than the other two polynomial approximations. For 2-D boundaries, the accuracy does not depend significantly on the degree of the polynomial used. High packing ratios (e.g. 1/40) can be achieved with high reconstruction accuracy.

For image approximation, spatial moment-preserving methods can compete with the efficient frequency domain DCT processing at comparable packing ratios. However, DCT methods have the advantage of lower computational complexity.

We have applied the moment-preserving piecewise approximation methods to the problem of noisy ICA. Results of experiments also prove that such methods are efficient in denoising independent components obtained from mixtures of noisy sources.

References

1. Tanimoto, S.L., "Image Transmission with Gross Information First", *Computer Graphics and Image Processing*, 9, 72-76, 1979
2. Stone, H., "Approximation of Curves by Line Segments", *Mathematics of Computation*, 15, 40-47, 1961
3. De Boor, C., *A Practical Guide to Splines*. New York: Springer Verlag, 1978

4. Rosin, P.L., and West, G.A.W., "Techniques for Segmenting Image Curves into Meaningful Descriptions", *Pattern Recognition*, 24(7), 643-652, 1991
5. Rao, K.R., and Yip, P., *Discrete Cosine Transform- Algorithms, Advantages, Applications*. Academic Press, Inc., 1990
6. Ahmed, H., et al., *Discrete Cosine Transform, Discrete Transforms and their Applications*. Van Nostrand Reinhold Co., Inc., pp. 9-12, 1985
7. Nguyen, T.B. and Oommen, B.J., "Moment-Preserving Piecewise Linear Approximations of Signals and Images", *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(1), 84-91, 1997
8. Goneid, A., AbuSeif, S., "Moment-Preserving Piecewise Approximations for 1-D and 2-D Signals", *Proc. Int. Conf. on Computer, Communication and Control Technologies (CCCT'03)*, Orlando, Florida, USA, July 31 – Aug 2, 2003, Vol. 4
9. Everson, R. and Roberts, S., "ICA: A flexible non-linearity and decorrelating manifold approach", *Neural Computation*. 11, 1957-1983, 1999
10. Amari, S. and Cichocki, A. "Adaptive blind signal processing- Neural Network approaches". *Proc. of the IEEE*, 86 (10), 2026 - 2048, 1998
11. Giannakopoulos, X., Karhunen, J., and Oja, E., "Experimental comparison of neural algorithms for independent component analysis and blind separation", *Int. J. of Neural Systems*, 9 (2), 651 - 656, 1999
12. Goneid, A., Kamel, A., and Farag, I., "New convergence and performance measures for Blind Source Separation algorithms", *Egyptian Computer Science Journal*, 31 (2), 13 - 24, 2009
13. Hyvarinen, A., "Fast and robust fixed-point algorithms for independent component analysis", *IEEE Trans. on Neural Networks*, 10(3), 626-634, 1999.
14. Goneid, A., Kamel, A., and Farag, I., "Generalized Mixture Models for Blind Source Separation", *Egyptian Computer Science Journal*, 34 (1), 1 - 14, 2010