by equalization with no need to transfer the already computed parts of the list from the slaves to the master processor.

Both histogram list calculation and equalization algorithms have been implemented on a TELMAT T-NODE machine with 4, 8, or 16 transputers T800. In the case of 16 transputers, the speedup obtained over the serial computation ranges from 4.01 ($L = 64$) to 14.87 ($L = 256$). As expected, the speedup is best for 256 levels/component because the computational load is much larger than the communication load between processors. The same observations hold for parallel equalization using the color histogram list, where a speedup of 10.15 has been obtained for $L = 256$ and a farm of 16 transputers.

REFERENCES

[1] A. K. Jain, *Fundamentals of Digital Image Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1989.
[2] C. W. Therrien, "Multichannel filtering methods for color image segmentation," in *Proc. IEEE Conf. Comput. Vision Patt. Recogn.*, 1985, pp. 637–639.
[3] K. L. Rangasami and C. Ramalingan, "Estimation and choice of neighbors in spatial interaction models of images," *IEEE Trans. Inform. Theory*, vol. IT-29, pp. 60–72, Jan. 1983.
[4] W. Niblack, *An Introduction to Digital Image Processing.* Englewood Cliffs, NJ: Prentice-Hall, 1986.
[5] A. Papoulis, *Probability, Random Variables and Stochastic Processes.* New York: McGraw-Hill, 1984.

A Fast Learning Algorithm for Gabor Transformation

Ayman Ibrahim and Mahmood R. Azimi-Sadjadi

Abstract— An adaptive learning approach for the computation of the coefficients of the generalized nonorthogonal 2-D Gabor transform representation is introduced in this correspondence. The algorithm uses a recursive least squares (RLS) type algorithm. The aim is to achieve minimum mean squared error for the reconstructed image from the set of the Gabor coefficients. The proposed RLS learning offers better accuracy and faster convergence behavior when compared with the least mean squares (LMS)-based algorithms. Applications of this scheme in image data reduction are also demonstrated.

I. INTRODUCTION

The Gabor transform [1]-[11] is viewed as the optimum case of the short time Fourier transform (STFT) in which the window function is chosen to have a Gaussian shape. This choice of the window function in the 2-D Gabor elementary functions guarantees the lower bound of the joint uncertainty, i.e., the 2-D Heisenberg inequality, in the two conjoint spatial-frequency domains. The Gabor analysis is based on projecting a given signal/image onto a family of shifted and modulated Gaussian window functions, which are called the "Gabor elementary functions" or the "Gabor basis functions," and the corresponding projection coefficients are called the "Gabor transform

coefficients." The use of such a transform is motivated by the fact that Gabor elementary functions have optimal localization property [2], [3] in the joint time (or spatial) and frequency domains. This leads to optimal extraction of the textural information from the images, which is an important feature for pattern recognition, segmentation, and image analysis applications. Beside the optimal localization property, other benefits of the Gabor transform include compatibility with mammalian visual systems [2], [3] and energy packing capability, which leads to lower entropy in the transform domain [4]. The deficiency of the Gabor transform, however, is that the elementary functions are not orthogonal. As a result, there is no straightforward method available for extracting these transform coefficients. If they were orthogonal, the extraction of these coefficients could have been done easily by the simple inner product formula [4].

Many approaches have been proposed to find a method for extracting the Gabor transform coefficients [4]-[10]. Bastiaan [5] derived an analytic solution for the 1-D case based on the expansion onto another set of discrete functions that are biorthogonal to the Gaussian elementary functions. This method was extended to the 2-D case by Porat and Zeevi [7]. Daugman [4] proposed a three-layer neural network for extracting the Gabor coefficients. The learning of the neurons is accomplished using a least mean squares (LMS) type algorithm [11]. Teuner and Hosticka [8] presented an algorithm that computes the Gabor transform coefficients using the complex LMS algorithm. Recently, Wang *et al.* [9] proposed a method to calculate the Gabor transform coefficients based on the biorthogonal functions. They used the FFT algorithm for the computation of the Gabor transform coefficients. Yeo [10] proposed a method to calculate the coefficients by multiplying a constant complex matrix and the inverse of a sparse real matrix.

Generally, these methods are based on either finding an analytical solution [5]-[7], [9], [10] or solving a set of normal equations using the LMS algorithm [4], [8]. The analytical solution requires a significant number of computations, and further, the solution may never exist. On the other hand, the main shortcoming of the LMS-based approaches is that the choice of the step size results in a tradeoff between accuracy and speed of convergence [11]. The primary objective of this correspondence is to find a solution to these problems by introducing an adaptive learning for the Gabor transform computation. The proposed algorithm uses the recursive least squares (RLS) learning algorithm instead of the LMS, which converges to an optimal solution in only few iterations. After convergence is achieved, Gabor transform coefficients can be extracted at the weights of the adaptive system. This RLS-based learning algorithm offers better accuracy and faster convergence when compared with the LMS-based algorithm. In addition, it does not have the accuracy-speed trade-off problems of the LMS method and provides better numerical stability compared with the analytical solutions. Simulation results are presented that demonstrate the applications of this method for image dimensionality reduction areas.

II. TWO-DIMENSIONAL GABOR TRANSFORMATION USING RLS LEARNING RULE

The goal of the 2-D Gabor transform is to represent a digital image $f(x, y)$, where x and y represent spatial coordinates, either exactly or in some optimal sense (e.g., minimizing the mean squared error between the reconstructed image and the original image) by projecting it onto a set of 2-D Gabor elementary functions. For a finite extent image $f(x, y)$, $x = 0, 1 \ldots, X - 1$; $y = 0, 1, \ldots, Y - 1$ partitioned

into $K \times L$ nonoverlapping lattices of size $M \times N$ where it is assumed that $X = KM$ and $Y = LN$, the approximated or reconstructed image $\hat{f}(x, y)$ can be written as

$$\hat{f}(x, y) = \sum_{m=0}^{K-1} \sum_{n=0}^{L-1} \sum_{r=0}^{M-1} \sum_{s=0}^{N-1} a_{mnrs} G_{mnrs}(x, y). \qquad (1)$$

In this representation, there are $KM \times LN$ coefficients a_{mnrs}'s that have a symmetry property, i.e., the real part has an even symmetry, and the imaginary part has an odd symmetry. Thus, only half of the coefficients will be sufficient for the image representation. Each Gabor elementary function G_{mnrs}'s in (1) constitutes a sine and cosine wave modulated by a Gaussian window, i.e.

$$G_{mnrs}(x, y) = w(h_1(x) - mM, h_2(y) - nN)$$
$$\cdot \exp\left(2\pi j \left[\frac{h_1(r)h_1(x)}{M} + \frac{h_2(s)h_2(y)}{N}\right]\right) \quad (2)$$

where $h_1(z) := z - \frac{M-1}{2}$, and $h_2(z) := z - \frac{N-1}{2}$ are introduced to center the Gaussian window at the center of the lattices; parameters M and N define the spatial distance between the centers of the elementary functions, which are usually called the Gabor logons [1], and $w(x, y)$ is the Gaussian window function of the form

$$w(x, y) = (\sqrt{2}\alpha)^{\frac{1}{2}} (\sqrt{2}\beta)^{\frac{1}{2}} \cdot \exp[-\pi(\alpha^2 x^2 + \beta^2 y^2)]. \quad (3)$$

The parameters α and β in $w(x, y)$ define the scaling of the Gaussian in the spatial domain along x and y coordinates, respectively. The values of these parameters are typically chosen to be the same, in which case, we get

$$w(x, y) = (\sqrt{2}\alpha) \cdot \exp[-\pi\alpha^2(x^2 + y^2)]. \quad (4)$$

From (4) and (2), it can clearly be seen that the Gabor elementary functions are parameterized for an invariant Gaussian window, which is positioned in fully overlapping Cartesian lattice location $\{x_m, y_m\} = \{mM, nN\}$, where (m, n) is the index for the lattice. The complex exponential that modulates these overlapping Gaussian are parameterized for the Cartesian lattice of 2-D spatial frequencies: $\{u_r, v_s\} = \{r/M, s/N\}$ for integer increments of (r, s) spanning their range inside each lattice $[-(M - 1)/2, (M - 1)/2], [-(N - 1)/2, (N - 1)/2]$, respectively.

Note that for fixed values of M and N, the choice of the Gaussian scale α determines the amount of effective overlap of the Gabor elementary functions across the neighboring lattice locations. It also determines the required support size (number of pixels) of each Gabor elementary function so that the truncation of the Gaussian tail produces negligible error. Therefore, if the truncation of the Gaussian is done outside each lattice the value of α should be chosen so that most of the energy of the Gaussian window lies inside the lattice. It must also be pointed out that since the image is partitioned into lattices and the Gabor elementary functions for each lattice are all centered, only translations that are equal to multiples of the lattice dimensions will result in a simple shifting relation of the transform coefficients.

To optimally represent a given pixel in the image by projecting it onto a chosen set of Gabor elementary functions, the mean squared error (MSE)

$$E = \frac{1}{XY} \sum_{x,y} [f(x, y) - \hat{f}(x, y)]^2. \quad (5)$$

is minimized with respect to all the coefficients. This gives the following normal equation [4]:

$$\sum_x \sum_y f(x, y) G^*_{mnrs}(x, y) = \sum_x \sum_y \hat{f}(x, y) G^*_{mnrs}(x, y) \quad (6)$$

which can be solved by LMS or RLS-type methods [11], taking into account the expression (1) for $\hat{f}(x, y)$.

To formulate the problem in the context of the RLS learning algorithm, let us rewrite (1) as

$$\hat{f}(x, y) = \sum_{m=0}^{K-1} \sum_{n=0}^{L-1} \hat{f}_{mn}(x, y) \quad (7a)$$

where $\hat{f}_{mn}(x, y)$ is the Gabor transform representation of lattice m, n i.e.

$$\hat{f}_{mn}(x, y) := \sum_{r=0}^{M-1} \sum_{s=0}^{N-1} a_{mnrs} G_{mnrs}(x, y). \quad (7b)$$

Now, assuming that the Gaussian windows are truncated outside each lattice, then the above equation becomes

$$\hat{f}_{mn}(x, y)$$
$$= \sum_{r=0}^{M-1} \sum_{s=0}^{N-1} a_{mnrs} G_{mnrs}(x, y)$$
$$\text{for } x \in [mM, mM + M - 1], \ y \in [nN, nN + N - 1]$$
$$= 0 \quad \text{otherwise.} \quad (8)$$

In this case, the optimal values of the Gabor transform coefficients a_{mnrs}'s for each lattice can be obtained independently by minimizing the "local" MSE

$$E_{mn} = \frac{1}{MN} \sum_{x,y} [f_{mn}(x, y) - \hat{f}_{mn}(x, y)]^2 \quad (9)$$

within each lattice m, n. This gives the following "local" complex normal equation

$$\sum_{x,y} f_{mn}(x, y) G^*_{mnrs}(x, y) = \sum_{x,y} \hat{f}_{mn}(x, y) G^*_{mnrs}(x, y). \quad (10)$$

An adaptive structure as shown in Fig. 1 can be constructed to solve this normal equation and extract the Gabor coefficients in each lattice at its weights. The input data set to this adaptive system consists of the Gabor elementary functions at each pixel within the lattice, i.e.

$$U_{mn}(x, y)$$
$$= [G^*_{mn00}(x, y) G^*_{mn01}(x, y) \cdots G^*_{mn,M-1,N-1}(x, y)]^H \quad (11a)$$

where superscript 'H' denotes complex conjugate transpose. The desired output is the corresponding pixel value of the original image, i.e., $f_{mn}(x, y)$. If the associated weight vector of the adaptive system is defined by

$$W_{mn}^H = [w_{mn00} \quad w_{mn01} \quad \cdots \quad \cdots \quad w_{mn,M-1,N-1}] \quad (11b)$$

then the output of the adaptive system for each pixel is

$$\hat{f}_{mn}(x, y) = \sum_{r=0}^{M-1} \sum_{s=0}^{N-1} w_{mnrs} G_{mnrs}(x, y)$$
$$= W_{mn}^H U_{mn}(x, y). \quad (12)$$

The error signal defined by $e_{mn}(x, y) := f_{mn}(x, y) - \hat{f}_{mn}(x, y)$ is used to drive the adaptation process.

Owing to the fact that the computations of the Gabor coefficients in each lattice are performed independently, we drop the lattice indices m, n for the sake of simplicity in notation. Moreover, in the RLS equations given below, the iteration index $k \in [1, MN]$ is obtained by mapping the 2-D spatial indices (x, y) inside each lattice in row-wise fashion, i.e., $k = Nx + y + 1$; $x \in [0, M - 1]$; $y \in [0, N - 1]$. The RLS algorithm [11], [12] starts with some chosen initial conditions and then uses the information contained in the new data samples to
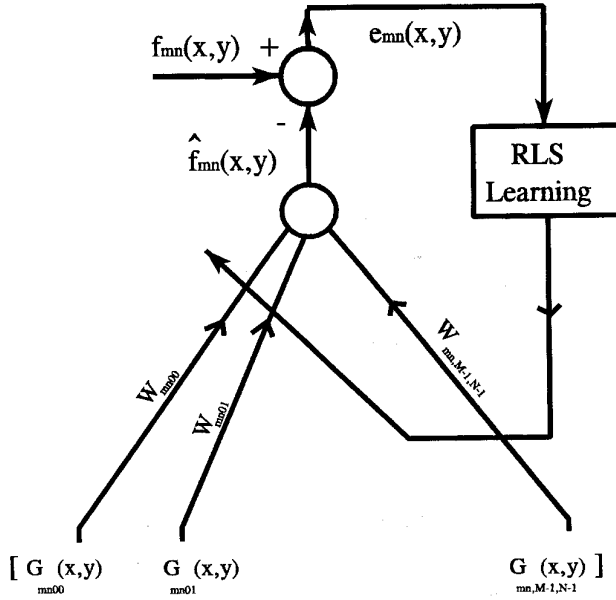
Fig. 1. Adaptive filter structure.

Fig. 2.   Original 'Lena' image 128 × 128.

update the weights. The algorithm starts by initializing $P(0) = \delta^{-1}I$, where δ is a very small positive constant $(\delta = 10^{-4})$, and $W(0) = 0$. Then, iterate

$$K(k) = \frac{\lambda^{-1}P(k-1)U(k)}{1 + \lambda^{-1}U^H(k)P(k-1)U(k)} \tag{13a}$$

$$e(k) = d(k) - \hat{W}^H(k-1)U(k) \tag{13b}$$

$$\hat{W}(k) = \hat{W}(k-1) + K(k)e*(k) \tag{13c}$$

$$P(k) = \lambda^{-1}P(k-1) - \lambda^{-1}K(k)U^H(k)P(k-1) \tag{13d}$$

where

$U(k)$ input data vector at iteration k
$d(k)$ desired output (i.e., $f_{mn}(x,y)$)
$\hat{W}(k)$ estimate of the weight vector at iteration k
λ forgetting factor $0 < \lambda \leq 1$ (in our case, $\lambda = 1$, i.e., infinite memory)
$K(k)$ Kalman gain vector
$P(k)$ inverse of data covariance matrix.

Upon the completion of the training, the weights converge to the 2-D Gabor coefficients a_{mnrs}'s, and thus, $\hat{f}_{mn}(x,y)$ will approach $f_{mn}(x,y)$. The training is performed by consecutive presentation of the training data set to the adaptive system and updating the weights according to the RLS equations. An epoch is completed when all the training data $k \in [1, MN]$ are presented to the adaptive system. This process is repeated for a number of epochs until the weights converge to within an acceptable solution. The training for all the other lattices can be done in parallel, even though within each lattice, the coefficient are computed sequentially.

III. SIMULATION RESULTS FOR IMAGE DATA REDUCTION

In this section, application of the proposed method was studied in an image data reduction example. The RLS algorithm of Section II was used to extract the Gabor transform coefficients of the "Lena" image in Fig. 2. This image is of size 128 × 128 with 256 gray levels. The entropy associated with this image was measured from the histogram to be 7.6. The image was partitioned into nonoverlapping lattices of size $M \times N$, and the RLS learning was applied to extract the Gabor transform coefficients within each lattice. Simulations were

done for several lattice sizes in order to study the effects on the convergence rate as well as on the entropy. Moreover, the effects of changing the scale of the Gaussian window α on the speed of convergence, on the existence of solution, and on the entropy were also studied.

For a lattice size of 16 × 16 and the Gaussian parameter $\alpha = .065$, 2-D Gabor transform coefficients were extracted using both the LMS and RLS algorithms. It was found that about 78% of the coefficients were concentrated in a small range [0, 3]. The histogram of the magnitude of the 2-D Gabor transform coefficients is shown in Fig. 3. As can be seen, this histogram is quite compact. The entropy of these coefficients was found to be 2.6. The Gabor transform coefficients were linearly quantized to 8 b and plotted in Fig. 4 as an image of (r, s), i.e., the frequency variables, centered at that spatial lattice position, with 256 gray levels. It was interesting to note that the RLS algorithm converged in only two epochs. The reconstructed image based on the entire extracted transform coefficients is shown in Fig. 5. The signal-to-noise ratio (SNR) for this image was found to be 35.7 dB. The same simulations were repeated using the LMS learning with a learning factor or step size of $\mu = 0.35$. The network took 165 epochs to converge to approximately the same MSE value as in the RLS method. The reconstructed image is shown in Fig. 6. The SNR for this image was measured to be 25.9 dB. Increasing the learning factor to $\mu = 0.4$ resulted in convergence after 150 epochs, i.e., faster convergence, but the results were less accurate comparing to the previous case.

For fixed lattice size, the effect of changing α, i.e., the Gaussian scale on the performance of the RLS and the LMS algorithms was studied next. For the LMS algorithm, decreasing α to 0.05 with the same learning factor $\mu = 0.35$ and lattice size 16 × 16 made the network converge in 120 epochs in contrast to 165 epochs when α was 0.065. This is due to the fact that as α decreases, the elementary functions become nearly orthogonal as they get closer to the FT basis functions. As a result, the network takes fewer number of epochs to converge. For the RLS learning, however, the network converged in only two epochs, irrespective of the choice of the value of α, i.e., the performance of this algorithm is not sensitive to the choice of the parameter α because RLS utilizes information contained in the input data. The entropy of the extrated coefficients was found to be 2.56.

The effect of varying α on the compactness of the coefficients was also studied. As α decreases, the coefficients become more compact, and the ability of the transform to capture the textural information from the images decreases as the spatial resolution deteriorates. Increasing α, however, improves the spatial resolution and, hence, the
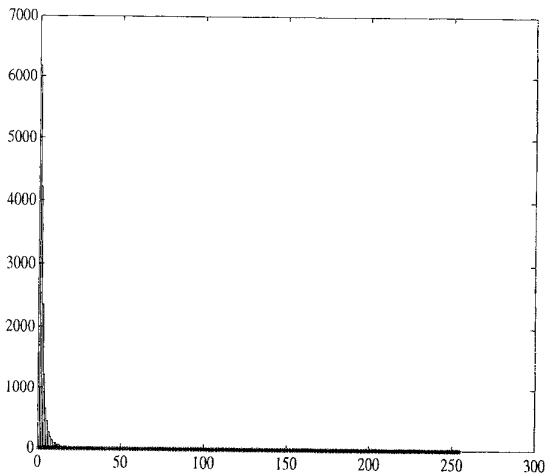
Fig. 3. Histogram of Gabor transform coefficients for 16×16 lattice.
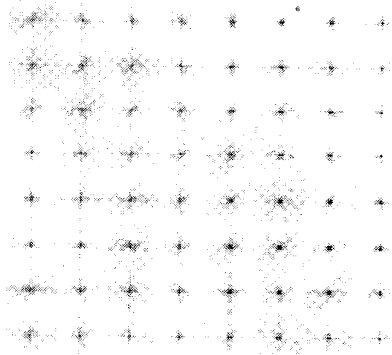

Fig. 4. Gabor coefficients plotted as an image with 256 gray scale.

Fig. 5. Reconstructed image using RLS for 16×16 lattice. (SNR = 35.7 dB).

Fig. 6. Reconstructed image using LMS for 16×16 lattice (SNR = 25.9 dB).

Fig. 7. Reconstructed image using RLS for 11×11 lattice.

ability of the transform to capture the textural information becomes better.

The effect of changing the lattice size was studied next. For the lattice cell dimension of 11×11, the original "Lena" image subsampled to 154×154 pixels was used. To assure that same amount of energy is contained in the truncated Gaussian window, the Gaussian scale of $\alpha = 0.095$ was used. The results showed that the LMS-based algorithm (same $\mu = 0.35$) converged to the Gabor coefficients in 120 epochs, as opposed to the RLS method, which still converged in only two epochs. The histogram of the magnitude of the 2-D Gabor transform coefficients was not as compact as in the 16×16 lattice size case. This is due to the fact that decreasing the lattice size will allow more efficient analysis of the image in the spatial domain as it captures more information. From the resultant histogram of the magnitude of the Gabor transform coefficients obtained for this case, the entropy was found to be 2.7. Decreasing the lattice size to 11×11 reduced the computational time compared with the 16×16 case. The reconstructed image based on these coefficients is shown in Fig. 7. Owing to the fact that the energy is contained in few coefficients, the image reconstruction can be done based on only the high energy coefficients. Fig. 8 shows the reconstructed image based on only 40% or actually 20% of the coefficients because of the symmetry property. The reconstructed image (SNR = 9.6 dB) still has an acceptable visual appearance.

IV. CONCLUSION

This correspondence introduces an RLS-based learning scheme for extraction of 2-D Gabor transform coefficients. In this approach, the image is divided into nonoverlapping lattices in which the 2-D Gabor transform coefficients are extracted. This method does not suffer from speed-accuracy tradeoff as with the LMS-type algorithms. Simulation results for image data reduction showed that the RLS-based algorithm converges to the Gabor transform coefficients in only two epochs, irrespective of the choices of parameters such as the lattice size and/or

Fig. 8. Reconstructed image based on 40% of the Gabor coefficients (SNR = 9.6 dB).

the Gaussian scale. On the contrary, the performance of the LMS-based scheme was shown to be very sensitive to the choices of these parameters.

REFERENCES

[1] D. Gabor, "Theory of communication," *J. Inst. Elec. Engr.*, pp. 429–457, 1946.

[2] J. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Amer.*, vol. 2, no. 7, pp. 1160–1169, 1985.

[3] ____, "Entropy reduction and decorrelation in visual coding by oriented neural receptive fields," *IEEE Trans. Biomed. Eng.*, vol. 36, pp. 107–114, Jan. 1989.

[4] ____, "Complete discrete 2-D Gabor transform by neural networks for image analysis and compression," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 7, pp. 1169–1179, July 1988.

[5] M. Bastiaans, "Gabor's expansion of a signal into Gaussian elementary signals," *Opt. Eng.*, vol. 20, no. 4, pp. 594–598, July 1981.

[6] J. Wexler and S. Raz, "Discrete Gabor expansion," *Signal Processing*, vol. 81, pp. 207–220, 1990.

[7] M. Porat and Y. Zeevi, "The generalized Gabor scheme of image representation in biological and machine vision," *IEEE Trans. Patt. Anal. Machine Intell.*, vol. PAMI-10, no. 4, pp. 452–468, July 1985.

[8] A. Teuner and B. J. Hosticks, "Adaptive Gabor transformation for image processing," *IEEE Trans. Image Processing*, vol. 2, no. 1, pp. 112–117, Jan. 1993.

[9] L. Wang, C. Chen and W. Lin, "An efficient algorithm to compute the complete set of discrete Gabor coefficients," *IEEE Trans. Image Processing*, vol. 3, no. 1, pp. 87–92, Jan. 1994.

[10] J. Yeo, "Complete Gabor transformation for signal representation," *IEEE Trans. Image Processing*, vol. 2, no. 2, pp. 152–159, Apr. 1993.

[11] S. Haykin, *Adaptive Filter Theory*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[12] A. Ibrahim, M. R. Azimi-Sadjadi, and S. Sheedvash, "A fast learning algorithm for Gabor transform extraction with application to image data reduction and pattern classification," in *Proc. 1994 IEEE Int. Conf. Neural Networks*, Orlando, FL, June–July 1994, pp. 4321–4326.

Separation of Image Parts Using 2-D Parallel Form Recursive Filters

Radhika Sivaramakrishna

Abstract—This correspondence deals with a new technique to separate objects or image parts in a composite image. A parallel form extension of a 2-D Steiglitz–McBride method is applied to the discrete cosine transform (DCT) of the image containing the objects that are to be separated. The obtained parallel form is the sum of several filters or systems, where the impulse response of each filter corresponds to the DCT of one object in the original image. Preliminary results on an image with two objects show that the algorithm works well, even in the case where one object occludes another as well as in the case of moderate noise.

I. INTRODUCTION

Parallel form realizations have been reported in the literature for 1-D signals [1], [2] and have the following advantages:-

1) Various components of the signal can be effectively separated from one another.
2) Coefficient sensitivities of the parallel and cascade forms are much lower than that for the direct form.
3) It is simpler to monitor stabilities for the parallel and cascade forms than for the direct form because they are of lower order.
4) The number of parameters to be estimated is identical to or less than that in the direct form, and thus, the computational requirements are approximately of the same order.
5) The techniques could be well adapted for a parallel machine.

In [3], Murthy and Prasad, have used a parallel form model to delineate P, QRS, and T wave components of the electrocardiogram signal by expressing the model for the discrete cosine transform (DCT) of the entire signal as a sum of models for the DCT of each individual component. The model is derived using the Steiglitz–McBride method [4]. They have shown that the DCT of a biphasic function in 1-D can be characterized by two poles and two zeros. A signal can be expressed as the sum of several biphasic functions with the DCT of each being expressed with a model of order (2,2). They have also shown that the location of peaks in the time signal is exactly related to the angle of the poles in the z domain. The same results can be carried over to 2-D. Hence, a similar technique can be adopted to separate 2-D biphasic functions. That is the focus of this paper, which proposes an extension to their method to separate objects in 2-D.

Murthy and Prasad claim that modeling the DCT of the signal, rather than the signal itself, reduces the model order considerably and quote that as a reason to use the DCT. However, we believe that, in 2-D at least, this reasoning does not seem to be appropriate and instead propound the following argument to model the DCT: In 2-D, it is not always possible to characterize objects in an image with 2-D rational models, but the DCT of an object or image part could be characterized either by a 2-D rational model or a sum of such models. This is because of the property of energy compaction of the DCT. The 2-D DCT of any arbitrary image has significant terms only in the first few rows or columns with the number of significant rows