ELSEVIER

Neurocomputing 12 (1996) 35–57

NEUROCOMPUTING

A multiresolution neural network approach to invariant image recognition

Stefanos D. Kollias *

Computer Science Division, Electrical Engineering Department, National Technical University of Athens, Zografou 15773, Greece

Received 4 August 1994; accepted 21 March 1995

Abstract

Triple-correlation-based representations of images have recently been combined with neural network architectures to derive invariant, with respect to translation, rotation and dilation, robust classification of images. Multiresolution image analysis is used in this paper to reduce the size of these representations in an optimal way, based on autoassociative linear networks. Hierarchical neural networks are then proposed as an efficient architecture for classification or retrieval of multiresolution invariant image representations. An effective procedure for designing and training such networks is also described and simulation results are presented which illustrate the capabilities of the proposed approach.

Keywords: Invariant; Multiresolution image analysis; Triple correlations; Autoassociative; Hierarchical neural networks

1. Introduction

Image recognition is an essential part of high-level systems used in a variety of fields, including robotics and industrial automation, military reconnaissance, remote sensing, optical character recognition and content-based image retrieval from large databases. Most existing recognition systems are based on the extraction of

* Email: stefanos@theseas.ntua.gr

SSDI 0925-2312(96)00041-0

appropriate features or small-sized representations of the images, thus reducing the redundancy, as well as the dimension of the data to be further classified and interpreted by the system [5]. Generation of a 'good' feature or small-sized image representation data set is a crucial aspect of the whole recognition procedure, requiring that as much as possible from the useful information of the original images be included in the derived feature data set. This requirement can ensure that small distortions in the shape of the objects shown in the images would not affect the subsequent feature-based classification process. Moreover, a flexible system should recognize an object regardless of its size, orientation or position in an image. This invariance property refers, therefore, to the requirement for extraction of features which are invariant with respect to transformations of the input image, such as translation, scale and rotation [19,11].

Multilayer perceptrons have been widely examined in the neural network field as a tool for image classification, based on appropriate feature extraction from the images. Apart from deterministic features, many statistical features, such as moments and linear prediction coefficients, have been used in the classification, for example, of textured images [11,22]. A crucial aspect concerning the performance of multilayer network classifiers is generalization, i.e. the ability of the network to classify correctly input data which were not included in its training set. Results from various applications have shown that good generalization is a result of appropriate network design; a rather small network size can make the network learn incomplete solutions, while an unnecessarily large size may lead the network learn only the specific training samples and noise. A small number of interconnection weights (i.e. free parameters during training) should be generally used and any a-priori knowledge about the problem should be included in the network structure. It should be mentioned that some very good results have been obtained when structured networks, including weight sharing and receptive fields are applied directly to image pixel values [15,6].

Third-order neural networks have been proposed as a classification method for invariant recognition of binary images [20,13]. However, the ability of such networks to provide solutions to complex real-life problems has been an open problem, mainly due to the excessive number of interconnections that are required in cases of large input image sizes. It has been recently shown that third-order neural networks are a specific category of triple-correlation-based neural network architectures [3]. Third order, or triple, correlations are higher-order signal statistics [17], which in case of deterministic signals have an one-to-one correspondence with the original signal (except of a shift ambiguity). Moreover, triple correlations of Gaussian or linear and symmetrically distributed noise are zero in the mean and tend to zero w.p.1 as the size of the available data record tends to infinity. In [3] appropriate clustering of the 4-D triple-correlation domain computed in the case of 2-D images has been proposed as an effective technique that provides invariant image representations to which neural networks can be applied for optimal classification purposes. However, in the case of real images these invariant representations are of large sizes. Consequently, the required number of the classifying network interconnection weights, especially between the units of the first hidden

layer and the network inputs, can be very large, resulting in prohibiting learning times, as well as in poor network generalization.

In this paper we propose a multiresolution analysis procedure to reduce the size of appropriate forms of the above-mentioned invariant representations in an optimal way. We also present an hierarchical image recognition approach, which achieves a major reduction in the number of interconnection weights of the classifying network, as well as in the required learning times. Section 2 gives a brief derivation of triple-correlation-based invariant image representations. Section 3 investigates multiresolution image representations and proposes a new technique for obtaining low resolution images in an optimal way, using autoassociative linear neural networks. Section 4 describes hierarchical neural network architectures which provide efficient solutions to the invariant image recognition problem. Section 5 presents simulation studies which illustrate the ability of the proposed approach to correctly classify images irrespectively of their position, orientation and resolution level.

2. Triple-correlation-based invariant image representations

Let $x(\mathbf{t})$ be a real two-dimensional signal with support $S = [0 \ldots N-1] \times [0 \ldots N-1]$. Its triple correlation is defined as,

$$x_3(\tau_1, \tau_2) = \frac{1}{N^2} \sum_S x(\mathbf{t}) x(\mathbf{t}+\tau_1) x(\mathbf{t}+\tau_2), \tag{1}$$

where τ_1, τ_2 are defined in $S' = [-(N-1), \ldots, (N-1)] \times [-(N-1), \ldots, (N-1)]$.

In general, there is one to one correspondence between the signal and triple correlation domain, implying that we can move indistinguishably from the signal domain to the triple correlation domain without loss of information, or, in other words, that we can distinguish two signals by comparing their triple correlations. Moreover, when the signal plane shifts, the triple correlation is unaffected and when the signal plane rotates and/or is scaled, the same happens in the triple correlation domain for both lag indices τ_1, τ_2. Triple correlation is also insensitive to additive Gaussian or any other linear and symmetrically distributed noise [17]. Various transformations of images have been recently proposed, based on triple correlation processing, which provide image representations that are invariant with respect to scale, rotation and translation of the original images [24,3].

The triple correlation of a 2-D signal $x(\mathbf{t})$ is a function of two 2-D vector indices, τ_1, τ_2, each of them spanning the subset S' of R^2. Let us cluster the resulting 4-D triple-correlation domain; by definition, $x_3(\tau_1, \tau_2)$ is the accumulation of all triple products formed by the values of $x(\mathbf{t})$ that lie on the corners of those triangles that are shifts of a prototype triangle defined by arbitrary vectors τ_1, τ_2. Let us define, next, classes $C(\tau_1, \tau_2)$ of triple correlation lags whose indices form, on the R^2 plane, triangles similar to the triangle defined by vectors τ_1, τ_2. This definition of $C(\tau_1, \tau_2)$ is such that rotation by an angle ϕ and/or scale by a

factor ρ (in log form) does not cause inter-class interference, while resulting in an internal circular shift of the content of each class. Based on this, it can be shown [3] that the amplitude and phase of the 2-D Fourier transformation of the triple correlations of each class, with respect to the space variables ρ and ϕ, provide a representation F_x which has a unique correspondence with the class of original images that are mutually related with scale-rotation-translation transformation. The dimensionality of this representation is, however, high; for this reason, it has been suggested to abolish 'uniqueness' of representation in favor of computational simplicity. This can be achieved, if another representation, say F_x^1, is used that contains only the amplitude of the Fourier transform of each triple-correlation class. A further reduction can be achieved, providing another representation, say F_x^2, if only the zero-th frequency Fourier coefficient is retained from each class.

It should be added that indices (τ_1, τ_2) can be replaced by the angles θ_1, θ_2 included between the plane vectors (τ_1, τ_2) and $(\tau_1, \tau_2 - \tau_1)$; these angles can be then quantized to, say, L_1 and L_2, levels respectively, leading to a considerable reduction of the four dimensional space spanned by (τ_1, τ_2), without any loss of information, to the 2-D space $[0, \pi/2] \times [0, \pi/2]$.

3. Triple-correlation-based neural network classifiers

All derived invariant representations consist of 2-D images of various sizes. The latter one, F_x^2, is a 2-D $(L_1 \times L_2)$ image, where L_1 and L_2 are the numbers of quantization levels for angles θ_1 and θ_2 respectively. In the former one, F_x^1, a 2-D $(R \times M)$ image corresponds to each class defined by (θ_1, θ_2), resulting in an invariant representation of size $(L_1 \times R, L_2 \times M)$, where R and M are the numbers of quantization levels for parameters ρ and ϕ respectively.

For small sizes of L_1, L_2, R and M, it is possible to apply multilayer fully-connected neural networks to perform the classification task. A multilayer network can, for example, accept any of the above-derived image representations directly in its input layer and be trained by some supervised learning algorithm, such as learning vector quantization, or a backpropagation variant [10,12], to classify these representations in different categories. It may also be shown [3] that third-order neural networks are a specific category of triple-correlation-based neural networks, in the sense that the input to the former networks is equivalent to representation F_x^2 of each triple-correlation class.

For larger, however, sizes L_1, L_2, R and M, the number of network inputs, which equals the number of pixels in the invariant representations, increases very rapidly; as a consequence, the number of free-parameters during network training, i.e. the number of interconnection weights, especially between the input and the first hidden layer, will become very large, imposing problems on the efficiency, as well as on the generalization ability of the network. Thus, the ability of the network to successfully perform the invariant recognition procedure requires a drastic reduction of the number of its weights. In the following we propose a technique for achieving such a reduction, by using algorithms that attempt to

define the correct network size, together with an hierarchical, multiresolution network architecture.

Since the optimal size of feedforward neural networks is generally an unknown quantity in most neural network applications, various techniques have been proposed for approximating it. Pruning algorithms start by considering a rather large network and delete, during training, nodes or weights that do not contribute to the minimization task. Weight decay, which penalizes the complexity of the network by letting each weight decay towards zero at a rate that is proportional to its magnitude, is such a simple technique; weight elimination [27], or weight sharing and receptive fields architectures [15] are similar methods, which are also amenable to parallel implementations [14]. Despite the fact that such methods can produce some reduction of the necessary network interconnection weights, they cannot provide a solution to the problem, especially for large input image sizes.

Constructive algorithms [21], on the other hand, first build an approximate model of low size and then add nodes to the network while learning more details, approaching, therefore, the optimum network size from below. Algorithms like node or network splitting [28,9], as well as procedures such as the ones used by cascade-correlation [7] can be included in this framework. Nevertheless, simple use of constructive methods cannot provide an effective solution to the problem, mainly due to the large size of the input network layer, which generally equals the size of the invariant representation. It is, therefore, essential to combine the above algorithms with a reduction of the input layer size.

A reduction of the network input layer size could be achieved, if local averaging of the invariant image representations was used as a preprocessing step. These representations could, for example, be separated in non-overlapping blocks of 8×8, or 16×16, pixels and each block be then replaced by the mean value of all pixels belonging to it; significant reduction of the number of input layer nodes would be achieved in this way, but at the cost of losing 'detail' information about the input images that may be significant for the classification task. In the next section we present a multiresolution decomposition of the invariant image representations, which can optimally be designed in specific applications so that the lost 'detail' information is as low as possible; following that, we propose a constructive procedure for designing a feedforward network to classify the derived invariant multiresolution image representations.

4. Invariant multiresolution image representations

4.1. The multiresolution decomposition

Representation of signals at many resolution levels has gained much popularity especially with the introduction of the discrete wavelet transform, implemented in a straightforward manner by filter banks using quadrature mirror filters (QMFs) [16,25]. In image processing the above are equivalent to subband processing [29]. Image decomposition is performed with an appropriate filter bank of decomposi-

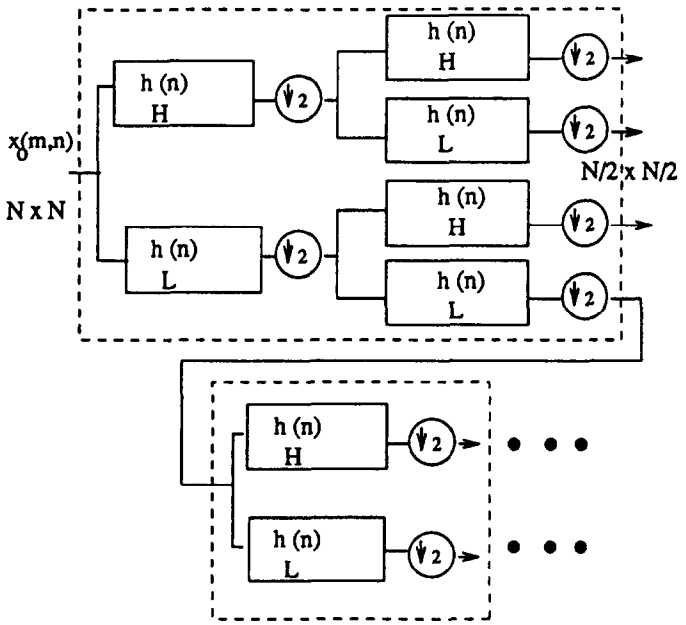

Fig. 1. Multiresolution (subband) image decomposition.

tion (or decimating) QMF filters. An appropriate bank of reconstruction (interpolating) QMF filters guarantee perfect reconstruction of the original image from its subband components. Multiresolution decompositions result in approximation images of low resolution that contain coarser information of the image content and in a set of detail images which contain more information as resolution is gradually increasing. A serious problem of multiresolution representations is that they can only provide scale-invariant features and not shift and rotation invariant ones. In the following this problem is aleviated by applying multiresolution decompositions to the derived triple-correlation-based invariant image representations, decomposing them in pyramids of images of gradually decreasing resolution.

Let x_0 denote an $N \times N$ invariant image representation. Using appropriate finite impulse response (FIR) perfect reconstruction filters $h_L(n)$ and $h_H(n)$, where $h_L(n)$ generally is a low-pass and $h_H(n)$ a high-pass filter, we can split the image into four lower resolution $N/2 \times N/2$ images [2], as shown in Fig. 1. Applying, for example, the low-pass filter $H_L(n)$ in the horizontal and then in the vertical direction of the original image (let us consider the separable case, for simplicity, at this point), we get the *approximation* image at the lower resolution level $j = -1$, denoted as x_{-1}^{LL}, where

$$x_{-1}^{LL}(m, n) = \sum_{k=1}^{N} \sum_{l=1}^{N} h_L(2m-k) h_L(2n-l) x_0(k, l) \tag{2}$$

By applying all other possible combinations of the above FIR filters shown in Fig. 2, we get three lower resolution *detail* images, denoted as x_{-1}^{LH}, x_{-1}^{HL}, x_{-1}^{HH}. As is

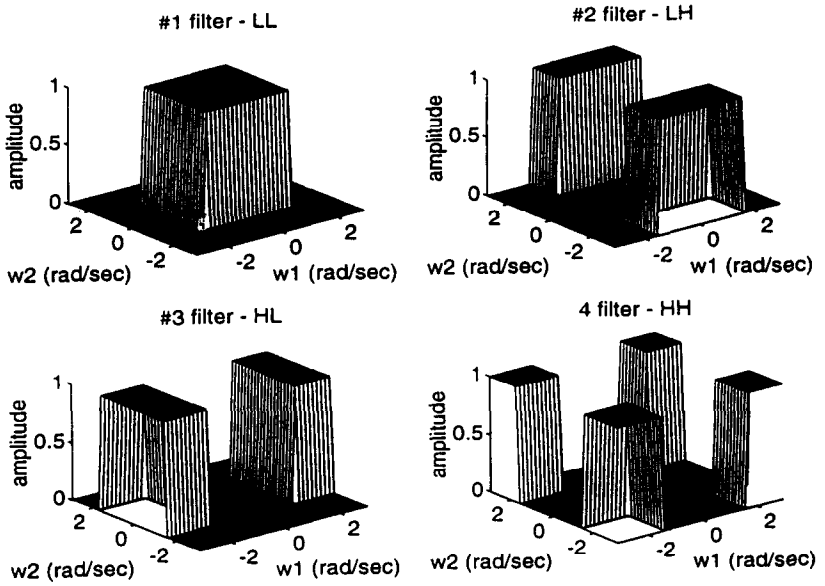

Fig. 2. Conventional decimation subband filters.

described in the next section, it is possible to use non-separable analysis (and synthesis) filters to perform the multiresolution decomposition. In this case, Eq. (2) takes the form

$$x_{-1}^{LL}(m, n) = \sum_{k=1}^{N} \sum_{l=1}^{N} h_{LL}(2m - k, 2n - l) x_0(k, l) \tag{3}$$

Perfect reconstruction of the original image $x_0(k, l)$ can be achieved through synthesis of all four subband components, i.e. the approximation and the three detail images. It is, however, possible to obtain an approximate reconstruction $x_0(k, l)$ of the original image, by using only the approximation image $x_{-1}^{LL}(m, n)$ and synthesis filter $f_{LL}(m, n)$ as follows

$$x_0(m, n) = \sum_{k=1}^{N/2} \sum_{l=1}^{N/2} f_{LL}(m - 2k, n - 2l) x_{-1}^{LL}(k, l) \tag{4}$$

If the image decomposition procedure, described by Eq. (3), is successively applied to the *approximation* images we have a *multiresolution approximation* of the original image, providing images of continuously decreasing size. Optimal design of the analysis and synthesis h and f filters in specific applications is examined next.

4.2. Optimal design of the multiresolution decomposition

The design of perfect reconstruction filter banks is based on the assumption that all the subband signals are available to the interpolation bank with infinite

precision. This is not, however, true, when only a part of subband components, and particularly only one of them, is used for reconstruction; in this case perfect reconstruction filters lose their optimality. Design techniques for analysis and synthesis filters that perform optimal reconstruction of an original image from a low-resolution representation of it have been recently proposed in [23]. Based on the minimization of the mean squared error between the original signal and the low-resolution representation of it, the 2-D filters are optimally adjusted to the statistics of the input images, so that most of the signal's energy is concentrated in the low resolution subband component. The procedure that is followed to achieve this goal is briefly described next.

Let us separate the input image into blocks of $P \times P$ pixels and then vectorize each block by placing its columns, one after the other, in an M-dimensional vector, say $\mathbf{x}(m, n)$, where $M = P^2$ and m,n denote the position of the first element of the block within the image. Let $\mathbf{x}(m, n)$ be considered as a zero-mean process with an $M \times M$ autocorrelation matrix which generally is space-varying over the whole image

$$\mathbf{R}_{xx}(m, n; u_1, u_2) = E\{\mathbf{x}(m+u_1, n+u_2)\mathbf{x}^T(m, n)\} \tag{5}$$

Let us then form $\bar{\mathbf{R}}_{xx}(u_1, u_2)$ as the spatially-averaged autocorrelation matrix

$$\bar{\mathbf{R}}_{xx}(u_1, u_2) = \lim_{N\to\infty} \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} \mathbf{R}_{xx}(m, n; u_1, u_2) \tag{6}$$

The 2-D Fourier-transform of $\bar{\mathbf{R}}_{xx}(u_1, u_2)$ will be the spatially-averaged spectral density matrix $\bar{\mathbf{S}}_{xx}(\omega_1, \omega_2)$. The design problem examined in this subsection searches for optimal non-separable $(Q \times M)$ and $(M \times Q)$ matrix filters $\mathbf{H}(n_1, n_2)$ and $\mathbf{F}(n_1, n_2)$. Filter $\mathbf{H}(n_1, n_2)$, when applied to each vectorized block of the image $\mathbf{x}(m, n)$, produces a low resolution vectorized image block, say $\mathbf{y}(m,n)$,

$$\mathbf{y}(m, n) = \sum_k \sum_l \mathbf{H}(k, l)\mathbf{x}(m-k, n-l) \tag{7}$$

with $Q = L^2$ elements, where the block size L of the low-resolution image in most practical situations approximately equals half of the original image block size P. The synthesis filters reconstruct the original image blocks, also in vectorized form, as follows

$$\hat{\mathbf{x}}(m, n) = \sum_k \sum_l \mathbf{F}(k, l)\mathbf{y}(m-k, n-l) \tag{8}$$

Filters $\mathbf{H}$ and $\mathbf{F}$ are formed in terms of the analysis and synthesis filters h and f so as to fit the vectorized $\mathbf{x}$ and $\mathbf{y}$ image blocks. The design of the optimal filters is based on minimization of the following criterion

$$J = \lim_{N\to\infty} \frac{1}{N^2} \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} E\{[\hat{\mathbf{x}}(m, n) - \mathbf{x}(m, n)]^T [\hat{\mathbf{x}}(m, n) - \mathbf{x}(m, n)]\} \tag{9}$$

In [23] it is shown that the 2-D Fourier transforms of the optimal analysis and synthesis filter matrices, i.e. $\mathbf{H}(\omega_1, \omega_2)$ and $\mathbf{F}(\omega_1, \omega_2)$, can be computed as follows

$$\mathbf{H}(\omega_1, \omega_2) = \begin{bmatrix} \mathbf{v}_1(\omega_1, \omega_2)^{T*} \\ \vdots \\ \mathbf{v}_Q(\omega_1, \omega_2)^{T*} \end{bmatrix}, \quad \mathbf{F}(\omega_1, \omega_2) = \mathbf{H}(\omega_1, \omega_2)^{T*} \tag{10}$$

where $\mathbf{v}_i(\omega_1, \omega_2)$ is the eigenvector corresponding to the ith largest eigenvalue of $\bar{\mathbf{S}}_{xx}(\omega_1, \omega_2)$ and $\mathbf{v}^{T*}$ denotes the complex transpose of $\mathbf{v}$.

For each frequency (ω_1, ω_2), the above procedure computes these filters in the frequency domain by performing an eigenvector/eigenvalue analysis of the spectral matrix [24,23], requiring a very high computational load. In the following we present a computationally efficient technique for designing the optimal filters in the spatial, and not the frequency, domain, based on autoassociative linear neural networks.

4.3 Neural-network-based optimal multiresolution filter design

Let us concentrate next on the problem of generating four subband components from each image, as shown in Fig. 1, only one of which is retained, as the low resolution representation. Using the notation adopted in the former two subsections, let the M-dimensional vector $\mathbf{x}(m, n)$ denote the vectorized $P \times P$ blocks of the input image $x_0(m, n)$, with $M = P^2$, the Q-dimensional vector $\mathbf{y}(m, n)$ denote the corresponding $(L \times L)$ blocks of the low-resolution representation $x_{-1}(m, n)$ also in vectorized form with $Q = L^2$ and finally the M-dimensional vector $\hat{\mathbf{x}}(m, n)$ represent the reconstructed vectorized image blocks.

The above vector notations have been adopted in this section, so that it be possible to denote the whole convolutional analysis and synthesis operations described by Eqs. (3) and (4) as multiplications of the above defined vectors by appropriate matrices, say H and F respectively, as was mentioned in the previous subsection. In particular Eqs. (7) and (8) can be written as

$$\mathbf{y}(m, n) = H\mathbf{x}(m, n) \tag{11}$$

$$\hat{\mathbf{x}}(m, n) = F\mathbf{y}(m, n) \tag{12}$$

Straightforward but tedious calculating, using Eqs. (3) and (4) provides analytical expressions of the $(Q \times M)$ and $(M \times Q)$ H and F matrices in terms of the, say $(J \times J)$, optimal filters h and f respectively. First, it can easily be verified that the dimension Q of the low-resolution vector $\mathbf{y}(m, n)$ can be expressed in terms of the input vector dimension M and the length J of filter h. In particular, if P and J add to an even value, L will be equal to $(P+J)/2$, otherwise it will equal $(P+J+1)/2$. Moreover, the dimension of reconstructed vector $\hat{\mathbf{x}}$ will be greater than the input signal dimension M due to the effects of the synthesis filter length J. In this case, it is the first M elements of $\hat{\mathbf{x}}$ which are retained for comparison

with the original vector. Following the above, if, for example, $M = 8$ and $J = 4$, then matrix H has the following structure

$$H = \begin{bmatrix} H_3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ H_1 & H_2 & H_3 & 0 & 0 & 0 & 0 & 0 \\ 0 & H_0 & H_1 & H_2 & H_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & H_0 & H_1 & H_2 & H_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & H_0 & H_1 & H_2 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & H_0 \end{bmatrix} \tag{13}$$

where H_0 is a submatrix of the form

$$H_0 = \begin{bmatrix} h(3,0) & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ h(1,0) & h(2,0) & h(3,0) & 0 & 0 & 0 & 0 & 0 \\ 0 & h(0,0) & h(1,0) & h(2,0) & h(3,0) & 0 & 0 & 0 \\ 0 & 0 & 0 & h(0,0) & h(1,0) & h(2,0) & h(3,0) & 0 \\ 0 & 0 & 0 & 0 & 0 & h(0,0) & h(1,0) & h(2,0) \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & h(0,0) \end{bmatrix} \tag{14}$$

where $h(0, 0)$, $h(1, 0)$, $h(2, 0)$, $h(3, 0)$ form the first column of filter $h(k, l)$. Matrices H_1, H_2 and H_3 are formed in exactly the same way using the corresponding columns of filter h, while extension to other values of M and J can easily be obtained. Matrix F is similarly formed in terms of corresponding matrices F_0, F_1, F_2 and F_3.

Based on Eqs. (11) and (12), we propose next to use a feedforward neural network to compute the optimal $J \times J$ analysis and synthesis filters, h and f respectively, through minimization of the mean squared difference between the original and reconstructed images. The network contains one hidden layer and linear hidden and output units. In particular the network accepts at its input the M-dimensional input image vector $\mathbf{x}$, uses Q hidden units and is trained to produce a reconstructed vector, at its M output units, that is equal to the input vector. As a consequence, the network operates in autoassociative form and during training is provided with the same input and desired output image blocks, in which the particular image, or a sequence of images, has been separated into; a backpropagation variant (see e.g. [12]) with a linear activation function can be the training algorithm. It is desired that the interconnection weights between the hidden units and the network inputs form a matrix W_{IH} equal to matrix H defined above in terms of the optimal filter h, while the interconnection weights between the output and hidden units form a matrix W_{HO} equal to the corresponding matrix F, so that the network implements the operations described in Eqs. (11) and (12). It should be added that formulations using similar autoassociative linear networks have been proposed for principal component analysis of data [18,30]. In the following, we impose appropriate constraints in the proposed network architecture, so that it is able to solve the filterbank design problem.

Based on the fact [23,1] that the optimal synthesis filter is related to the analysis one through Eq. (14) in the frequency domain, or equivalently in the spatial domain

$$f(m, n) = h(-m, -n) \tag{15}$$

the following constraint on the network structure is easily verified

$$W_{HO} = W_{IH}^T \tag{16}$$

Moreover, in order to force matrices W_{IH} and W_{HO} obtain the required forms (as, for example, the ones given in Eqs. (13)–(14) for the analysis matrix filter H), the weights corresponding to zero entries in the matrices are fixed to zero during training. Furthermore, when a specific weight of matrix W_{IH} (similarly for W_{HO}) is updated, its value is copied to all other weights that correspond to the same sample value of the optimal analysis filter $h(m, n)$, as determined by Eqs. (13)–(14); this procedure is the same as the one used for training time-delay networks (see e.g. [26]), where the need for copying the updated weight values to groups of weights with identical values also arises.

5. Multiresolution invariant neural network classifiers

Hierarchical neural network architectures are proposed next as an efficient scheme for classifying the resulting optimal multiresolution invariant representations. A feedforward multilayer network is used first to classify an *approximation* image of quite low resolution, trained by some backpropagation variant, including

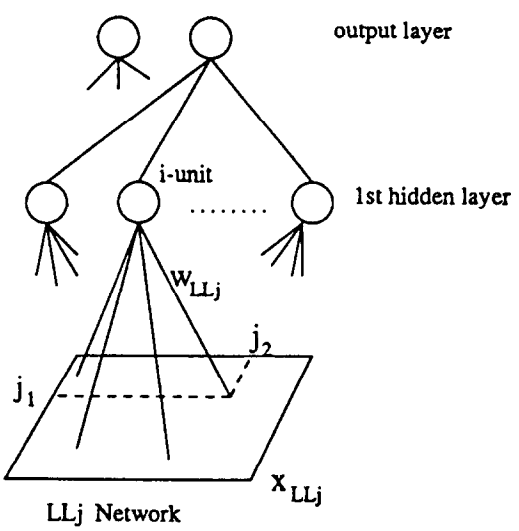

Fig. 3. Network training at level j.

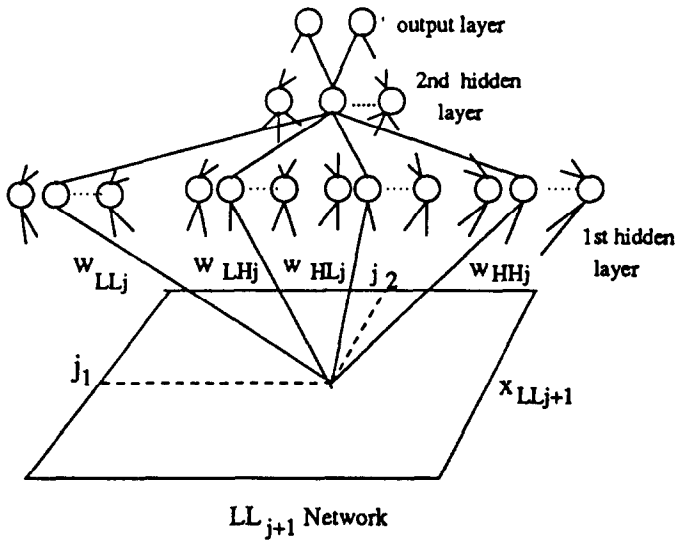

Fig. 4. Transfer of weights between levels j and $j+1$.

some pruning technique, such as weight decay. The hierarchical network is then recursively constructed to handle the image at higher resolution levels. More specifically, the proposed procedure starts by training a network at, say, resolution level j with $j \leq -1$, (network LL_j) to classify *approximation* images x_j^{LL} at that resolution level (see Fig. 3). After training, the network performance is tested, using a validation set of *approximation* images at the same resolution level j. If the performance is not acceptable, training is repeated at the $j+1$ resolution level.

In this approach, it would be desired that the network at level $j+1$ (network LL_{j+1}) a priori includes as much as possible from the 'knowledge' of the problem acquired by the former network at level j. Some early results in this topic [8] suggested to use the computed weights of the low resolution network as initial conditions for the weights of the high resolution one. In this paper, we use the property that the contents of the derived optimal *approximation* and *detail* images at level j are uncorrelated to each other [23] and that the information of the *approximation* image at level $j+1$ is equivalent to the information included at both the *approximation* and *detail* images at level j. As a consequence, we can train three more networks (LH_j, HL_j, HH_j), separately (or in parallel) from the former one, to classify the *detail* images at level j and let the network at level $j+1$ contain in its first hidden layer a number of units equal to the union of the first hidden layer units of all four lower resolution networks, as shown in Fig. 4.

We then derive forms that permit transfer of the generally large number of (already computed) weights between the input and the first hidden layer of the low resolution networks in corresponding positions of the high resolution network LL_{j+1} and keep these weights fixed during training of the high resolution network. A small number of nodes can be added to the first hidden layer of the LL_{j+1}

network, while computation of the resulting new interconnection weights, as well as of the generally less complex upper hidden layers is performed then by training the corresponding parts of the high resolution network LL_{j+1}. The above-mentioned addition of hidden nodes can be performed sequentially during training, similarly to the cascade correlation methodology [7]. To implement the weight transfer, we impose the constraint that the inputs to the units of the first hidden layer of network LL_{j+1} be identical to the corresponding inputs of the units of networks at level j. In the case of network LL_j, for example, with $j = -1$, the input to each unit of the first hidden layer is

$$\sum_{j_1=1}^{L} \sum_{j_2=1}^{L} w_{-1}^{LL}(j_1, j_2) x_{-1}^{LL}(j_1, j_2) \tag{17}$$

where $w_{-1}^{LL}(j_1, j_2)$ is the weight connecting each hidden unit to the (j_1, j_2) pixel of the $L \times L$ image block at level -1. Then in the first hidden layer of the LL_0 network classifying each $N \times N$ image block, the input to the corresponding unit will be analogously

$$\sum_{k_1=1}^{N} \sum_{k_2=1}^{N} w_0^{LL}(k_1, k_2) x_0^{LL}(k_1, k_2) \tag{18}$$

If the computed values in the above equations are required to be equal to each other, then it can be easily shown using Eqs. (3), (17) and (18) that

$$w_0^{LL}(k_1, k_2) = \sum_{j_1=1}^{K} \sum_{j_2=1}^{K} h_{LL}(2j_1 - k_1, 2j_2 - k_2) w_{-1}^{LL}(j_1, j_2) \tag{19}$$

A similar form can be derived relating the weights in each network LH_j, HL_j, HH_j and the corresponding weights in network LL_{j+1}, as shown in Fig. 4. The above forms permit computation of the generally large number of weights between network's input and first hidden layer be efficiently performed at lower resolution.

It should, however, be mentioned that training and use of all LL, LH, HL, HH networks at level j is not always meaningful; this is due to the fact that in most cases only some of the four subband images contain significant portion of the content of the original image. To overcome this problem, we propose to use only one set of weights in Eq. (19) corresponding to the low resolution image which contains the most significant part of the original image among the four subband low-resolution images; this image should be created following the procedure presented in the previous section.

6. Simulation results

The performance of the proposed triple-correlation-based multiresolution neural network classifiers was examined next, using inspection of solder joints in printed circuit board manufacturing as an application in which conventional

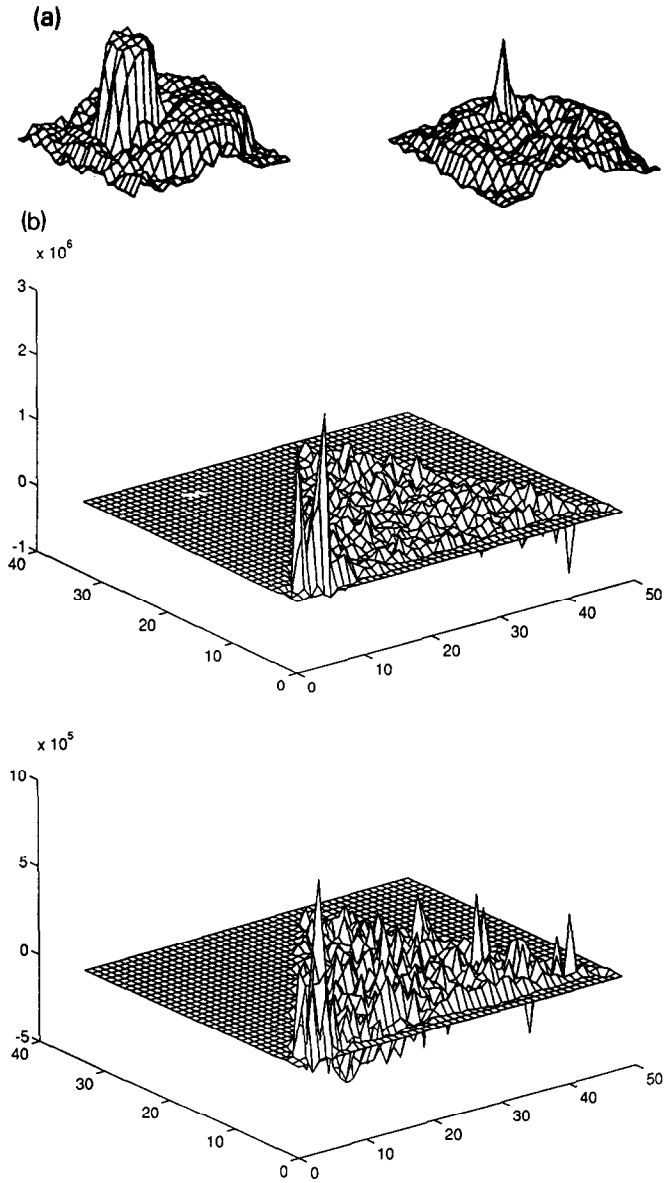

Fig. 5. (a) Characteristic images of two categories of solder joints; (b) Corresponding invariant representations.

pattern recognition techniques have not shown sufficient reliability. 2-D gray-scale images, showing either the height or the intensity as functions of the position across solder joints, were obtained by an optical laser scanner and used as signals to be classified in two categories; namely, good or poor solder joints, the latter containing insufficient amount of solder. 100 input images, 50 of each category,

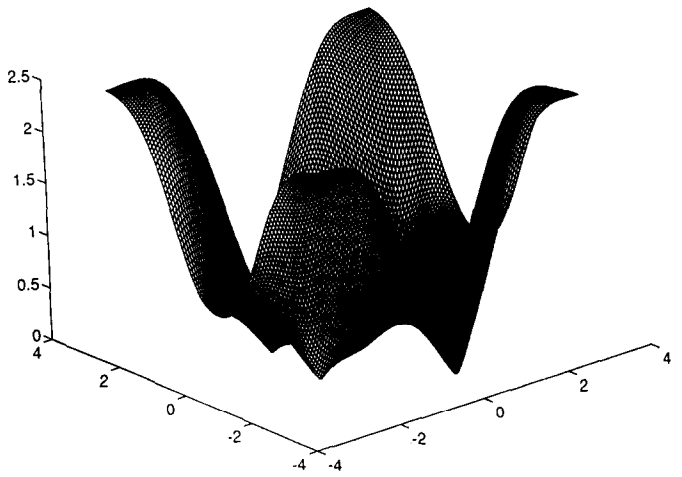

Fig. 6. Optimal analysis filter at level 0.

were transformed in this domain and used as training data, while another set of 100 images, 50 of each category, was used to test the generalization ability of the networks. An example of each category is shown in Fig. 5(a), while the corresponding F_x^2 invariant representations are shown in Fig. 5(b), the values of (θ_1, θ_2) being quantized to a discrete grid of 34×49 pixels.

Training a fully connected network at this resolution level, using a pruning mechanism like weight decay, did not provide good generalization results; only 86% of the test data were correctly classified. We then used multiresolution analysis to reduce the input image size. As has been discussed in Section 4, it is crucial to include as more information as possible into the *approximation* representation at each resolution level. We, therefore, examined first the performance of the linear autoassociative networks described in Section 4.3 for obtaining optimal analysis and synthesis filters. We used five images from each category, separated in blocks of 8×8 pixels, as training data at level 0. After training, we obtained the optimal analysis filter shown in Fig. 6. We applied this filter to all images of both categories. The low resolution representations (level -1) containing 21×29 pixels, which correspond to the images of Fig. 5(b), are shown in Fig. 7.

It should be mentioned that the optimal filter is of high frequency type. However, it is quite different from the *HH* filter that is used in conventional multiresolution analysis and is included in Fig. 2. For comparison purposes, we used the derived optimal synthesis filter to reconstruct the images at level 0. We then did the same, using the high pass (*HH*) conventional synthesis filter applied to the subband low resolution image obtained through the *HH* conventional analysis filter; the reconstructed images are shown in Fig. 8 and 9, verifying the optimality of the single-band representations provided by the proposed linear autoassociative networks. We continued the decimation procedure one step more, obtaining low resolution representations of 15×19 pixels at level -2; the derived

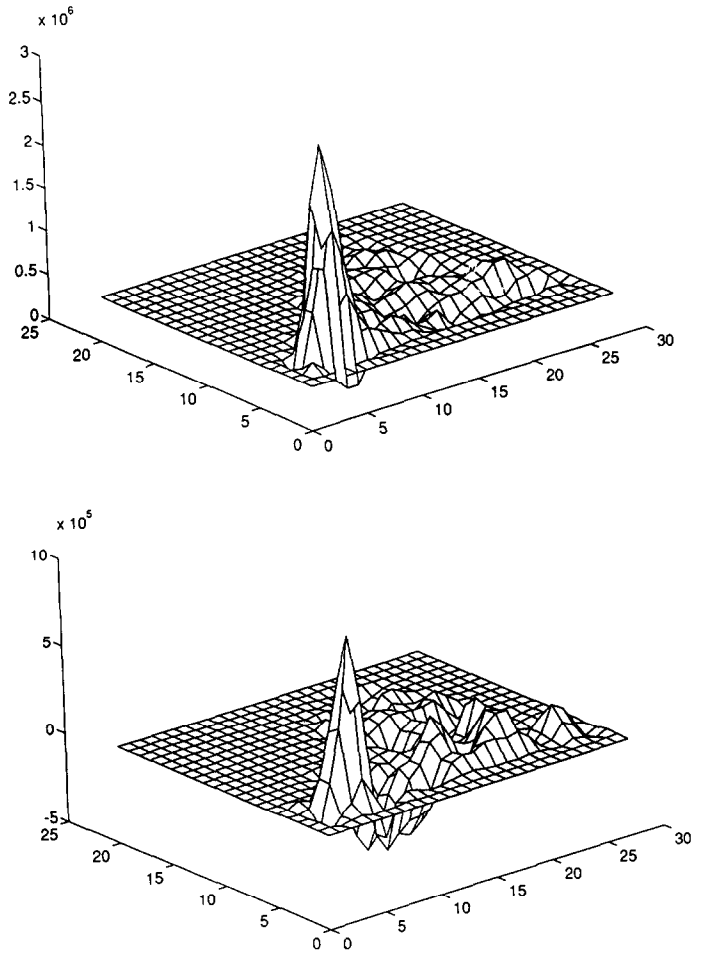

Fig. 7. Optimally derived low resolution representations at level (−1).

optimal analysis filter, shown in Fig. 10, is quite similar to that obtained at the previous level, verifying the high frequency nature of the multiresolution invariant triple-correlation-based representations.

We then examined the performance of the proposed hierarchical network classifiers using the optimally derived low resolution representations. We started by training a two hidden layer network at level −2, using 8 and 4 hidden units respectively; the generalization ability of it was found to be 84%. We then used the procedure described in section 5 to transfer weights between the input and first hidden layer of this network to corresponding weights of the network at level −1. Using a node splitting procedure [28] we constructed a network with 2 more units in the first hidden layer and two more hidden layers with 4 and 2 units respec-

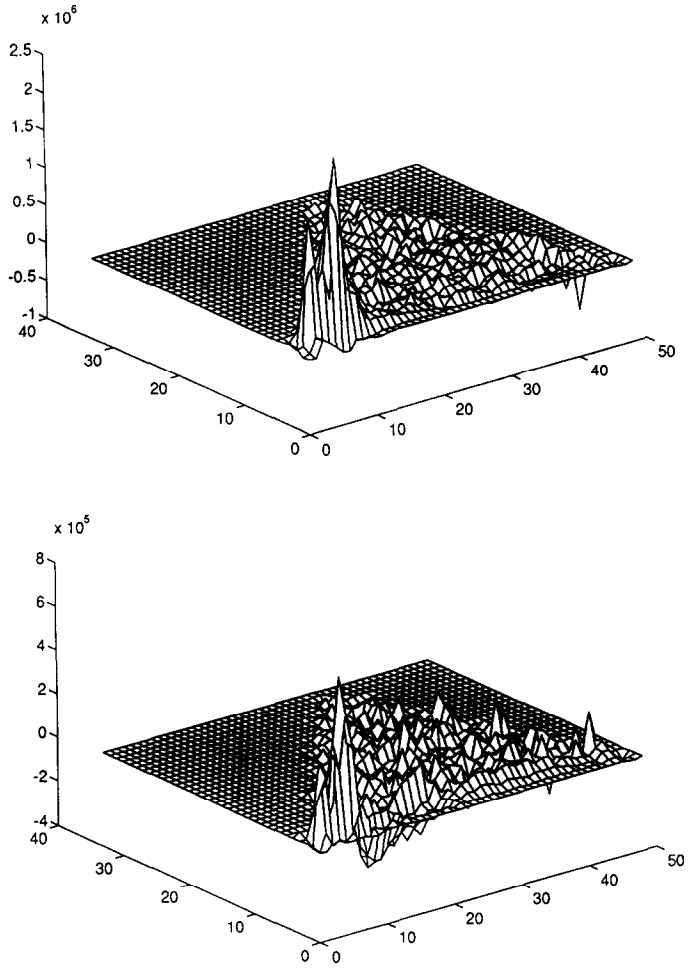

Fig. 8. Image reconstruction at level 0 using optimal results of linear autoassociative network.

tively; generalization of this network was very good, providing 92% correct classifications in the test data set. Adopting this procedure for transferring weights between networks at -1 and 0 levels, we managed to obtain a classification rate of 94% at level 0, which was higher than the one obtained by direct training of a network at this level.

In the last experiment we considered the problem of printed character recognition using image data obtained by a laser scanner as a means to further investigate the nature of the computed optimal filters, in relation to the invariant representations used. Fig. 11 depicts some of the images that were used, the average size of which was 20×20 pixels.

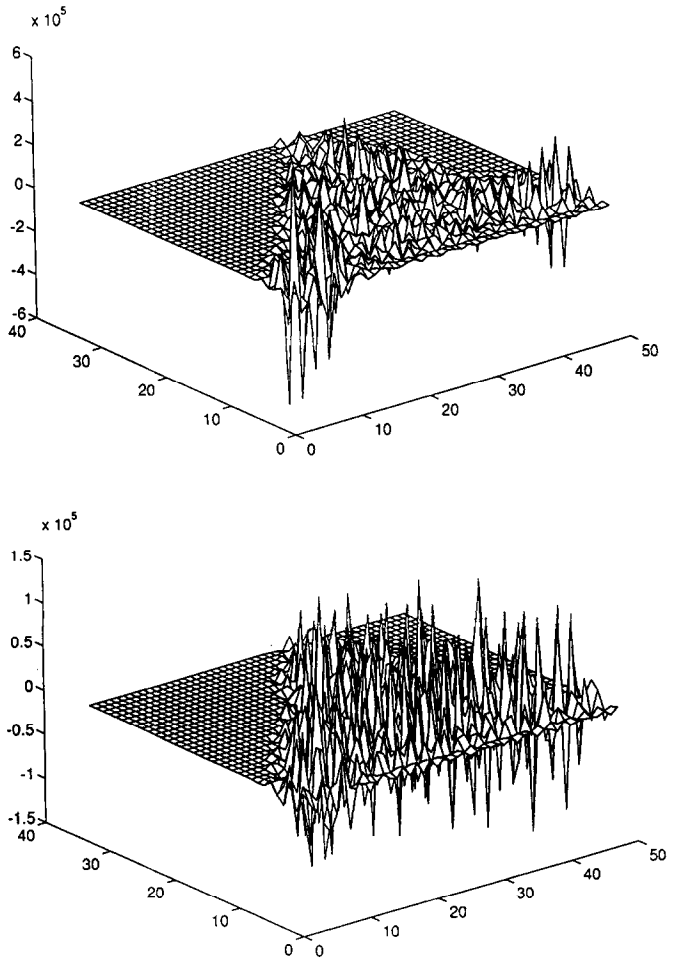

Fig. 9. Image reconstruction at level 0 using conventional HH analysis and synthesis filters.

The F_x^1 criterion, which uses the amplitude information of the Fourier transform, was used next to obtain invariant triple-correlation-based representations, using 16 quantization levels for each θ_1 and θ_2 angle and 8 levels for quantization of each inter-class parameter ρ and ϕ.

Fig. 12 shows a characteristic single class (Fourier transformed in the ρ, ϕ domain) of the representations of the horizontal capital letters 'N', 'T', 'U', 'A'. These representations are quite different from each other, while on the contrary, the representations shown in Fig. 13, which belong to three scaled and rotated versions of letter 'N' are almost identical, demonstrating invariance as well as robustness to small shape distortions. Tables 1 and 2 use the Euclidean distance as a measure of the difference between representations of the above letters and between rotated and scaled versions of them respectively.

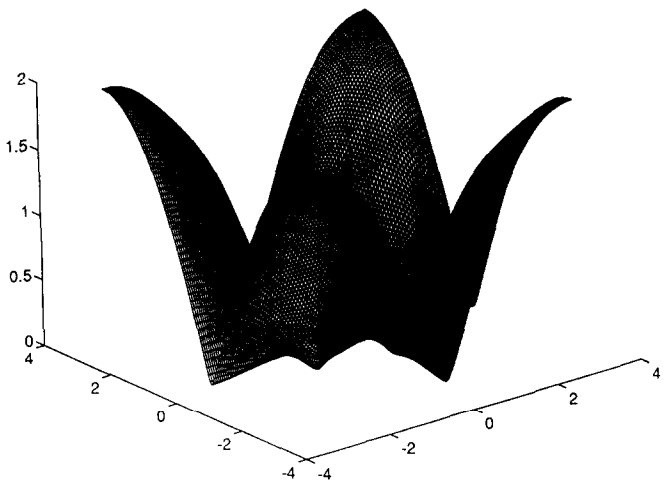

Fig. 10. Optimal analysis filter at level (−1).

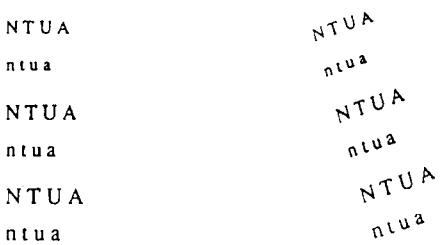

Fig. 11. Printed character sample data.

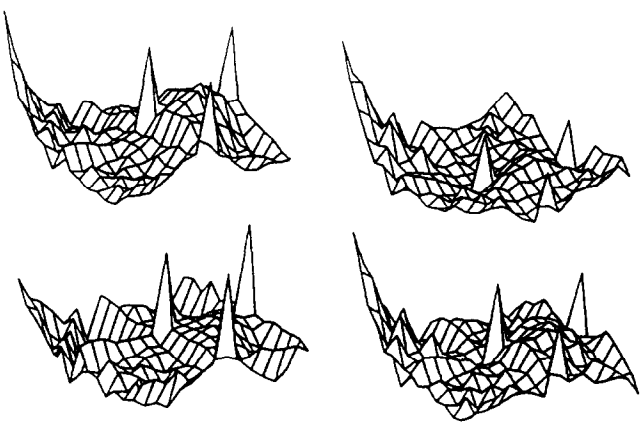

Fig. 12. A single class of the invariant representations of different characters.

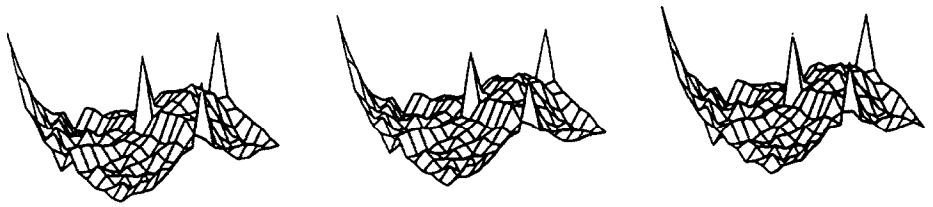

Fig. 13. A single class of the invariant representations of scaled and rotated versions of character '*N*'.

The derived 2-D invariant character representations in total, i.e. including all computed (i.e., 16×16) classes, consist of 128×128 pixels. Fig. 14 shows the representation of horizontal character 'N'; only the first quarter part of the representation, consisting of 64×64 pixels, is shown, since the rest parts are computed through third order correlation symmetries. Optimal analysis filters

Table 1
Euclidean distances of representations of different characters
Horizontal, pointsize 10

	N	T	U	A	n	t	u	a
N	0.00	0.96	1.70	0.44	1.31	1.63	1.02	0,85
T	0.96	0.00	2.65	0.63	2.78	0.89	2,19	1,79
U	1.70	2.65	0.00	1.82	0.88	4.58	1,17	1,47
A	0.44	0.63	1.82	0.00	1.49	1.59	1,04	0,77
n	1.31	2.78	0.88	1.49	0.00	4.40	0,24	0,55
t	1.63	0.89	4.58	1.59	4.40	0.00	3,55	3,09
u	1.02	2.18	1.17	1.04	0.24	3.55	0,00	0,34
a	0.85	1.79	1.47	0.77	0.55	3.09	0,34	0,00

Table 2
Euclidean distances of representations of scaled and rotated versions of characters
Rotated, pointsize 9

	N	T	U	A	n	t	u	a
N	0.06	0,97	2,49	0,49	1,55	1,88	1,68	0,97
T	1,35	0,07	3,69	0,62	2,98	1,34	3,20	1,99
U	1,94	2,44	0,18	1,97	1,45	4,57	1,36	1,46
A	0,57	0,55	2,48	0,08	1,47	1,77	1,68	0,84
n	1,30	2,54	1,23	1,64	0,47	4,47	0,37	0,60
t	1,90	1,21	5,92	1,52	4,61	0,16	4,87	3,34
u	1,00	1,94	1,62	1,15	0,33	3,59	0,29	0,72
a	0,91	1,66	2,11	0,84	0,64	3,35	0,76	0,13

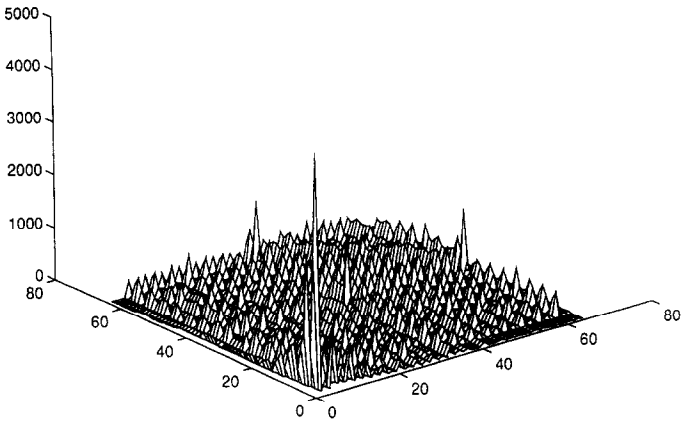

Fig. 14. The invariant representation of horizontal character '*N*', including all classes.

were then computed, using the proposed linear autoassociative neural networks and training data from all character representations in the form of Fig. 14. To indicate the nature of such filters, the filter computed using the image in Fig. 14 is shown in Fig. 15. This filter, apart from high pass areas, also includes a strong low-pass component; this is due to the large portion of 'smooth' small valued information that the image of Fig. 14, as well as other images of this form, contain.

7. Conclusions

Appropriate 2-D representations based on third-order correlations of images have been used for invariant with respect to translation, rotation and scale

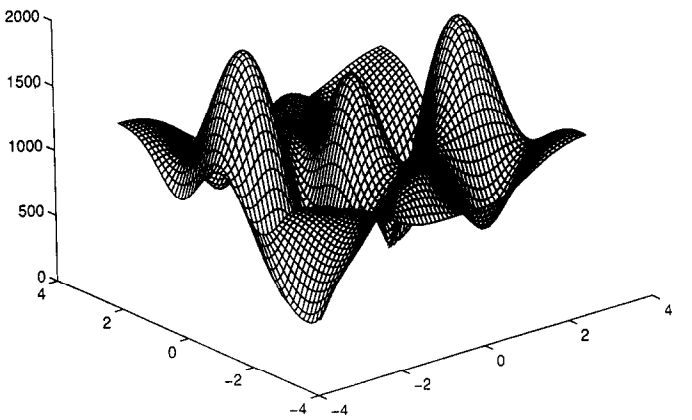

Fig. 15. Optimal analysis filter based on the image shown in Fig. 14.

classification by neural network architectures. An efficient scheme has been proposed for this purpose, introducing hierarchical neural networks, which are created in a constructive way. Linear autoassosiative networks have been proposed for optimally selecting filterbanks which are used in the above constructive approach. Very promising results have been presented, testing the performance of the method in image classification problems; other studies in efficient retrieval of images, or video, from large image databases, using the proposed multiresolution network approach are currently investigated.

Acknowledgement

I wish to thank the anonymous reviewers for their constructive comments which helped in giving the paper its final form.

References

[1] P. Baldi and K. Hornik, Neural networks and principal component analysis: Learning from examples without local minima, *Neural Networks* 2 (1989) 53–58.
[2] M. Antonini, M. Barlaud, P. Mathiew and I. Daubechies, Image coding using the wavelet transform, *IEEE Trans. Image Processing* 1 (1992) 205–220.
[3] A. Delopoulos, A. Tirakis and S. Kollias, Invariant image classification using triple-correlation-based neural networks, *IEEE Trans. Neural Networks* (May 1994).
[4] S. Kung, K. Diamantaras and J. Taur, Adaptive principal component extraction and applications, *IEEE Trans. Signal Processing* 42 (1994) 1202–1217.
[5] R. Duda and P. Hart, *Pattern Classification and Scene Analysis* (Wiley, New York, 1973).
[6] Esprit Project 2092 Handbook, ANNIE: Neural Networks for Industrial Pattern Recognition Problems, 1991.
[7] S. Fahlman and C. Lebiere, The cascade-correlation learning architecture, in *Advances in Neural Information Processing Systems 2* (Morgan Kaufman 1990) 524–532.
[8] C. Hand, M. Evans and S. Ellacott, A neural network feature detector using a multiresolution pyramid, in *Neural Networks for Vision, Speech and Natural Processing* (Chapman and Hall, 1992) 65–82.
[9] Y. Hirose, K. Yamashita and S. Hiilya, A backpropagation algorithm which varies the number of hidden units, *Neural Networks* 4 (1991) 61–66.
[10] D. Hush and B. Horne, Progress in supervised neural networks, *IEEE Signal Processing Mag.* 10 (1993) 8–39.
[11] A. Khotanzad and J. Lu, Classification of invariant image representations using a neural network, *IEEE Trans. ASSP* 38 (1990) 1028–1038.
[12] S. Kollias and D. Anastassiou, An adaptive least aquares algorithm for efficient training of artificial neural networks, *IEEE Trans. CAS* 36 (1989) 1092–1101.
[13] S. Kollias, A. Stafylopatis and A. Tirakis, Performance of higher order neural networks in invariant recognition, *Neural Networks: Advances and Applications* (North-Holland, 1991) 79–108.
[14] S. Kollias and A. Stafylopatis, Parallel implementations of backpropagation based on network topology, in *Parallel Algorithms for DSP, Computer Vision and Neural Networks* (Wiley, 1993) 233–258.
[15] Y. LeCun, Generalization and network design strategies, *Connectionism in Perspective* (North-Holland, Amsterdam, 1989) 143–155.

[16] S. Mallat, A theory for multiresolution signal decomposition: the wavelets representation, *IEEE Trans. PAMI* 11 (1989) 674–693.
[17] J.M. Mendel, Tutorial in higher-order statistics (spectra) in signal processing and system theory, *Proc. IEEE* 79 (1991) 278–305.
[18] E. Oja, A simplified neuron model as a principal component analyzer, *J. Math. Biol.* 15 (1982) 267–273.
[19] Y.H. Pao, *Adaptive Pattern Recognition and Neural Networks* (Addison-Welsey, 1989).
[20] M.B. Reid, L. Spirkovska, and E. Ochoa, Simultaneous position, scale, and rotation invariant pattern classification using third-order neural networks, *Int. J. Neural Networks* 1, (3) (July 1989).
[21] F. Smieja, Neural network constructive algorithms: Trading generalization for learning efficiency, *Circuits, Systems and Signal Processing* 12 (1993) 331–374.
[22] L. Sukissian, S. Kollias and Y. Boutalis, Adaptive classification of textured images using linear prediction and neural networks, *Signal Processing* 36 (1994) 209–232.
[23] A. Tirakis, A. Delopoulos and S. Kollias, 2-D filter bank design for optimal reconstruction using limited subband information, *IEEE Trans. Image Processing* accepted for publication.
[24] M. Tsatsanis and G. Giannakis, Object and texture classification using higher order statistics, *IEEE Trans. Pattern Analysis and Machine Intelligence* 14 (1992) 733–750.
[25] P.P. Vaidyanathan, *Multirate Systems and Filter Banks* (Prentice-Hall, Englewood Cliffs, NJ, 1993).
[26] A. Waibel et al., Phoneme recognition using time delay neural networks, *IEEE Trans. ASSP* 37 (1989) 328–339.
[27] A. Wigend, D. Rumelhart and B. Huberman, Generalization by weight elimination with applications to forecasting, in *Advances in Neural Information Processing Systems 3* (Morgan Kaufman, 1991) 875–882.
[28] M. Wynne-Jones, Node splitting: A constructive algorithm for feedforward neural networks, *Neural Computing Applications* 1 (1993) 17–22.
[29] J. Woods and S. O'Neil, Subband coding of images, *IEEE Trans. ASSP* 34 (Oct. 1986) 1278–1288.
[30] L. Xu, Least mean square error reconstruction principle for self-organization, *Neural Networks* 6 (1993). 627–648.

Stefanos D. Kollias was born in Athens, Greece in 1956. He received the Diploma degree in Electrical Engineering from the National Technical University of Athens (NTUA) in 1979, the MSc degree in Communication Engineering from the University of Manchester (UMIST), England in 1980 and the PhD degree in Signal Processing from the Computer Science Division of NTUA in 1984. In 1982 he received a ComSoc Scholarship from the IEEE Communications Society. From 1986 to 1992 he was Lecturer and Assistant Professor in the Department of Electrical and Computer Engineering, NTUA. From 1987 to 1988 he was a Visiting Research Scientist in the Department of Electrical Engineering and the Center for Telecommunications Research, Columbia University, New York. Since 1992 he has been an Associate Professor in the Computer Science Division of NTUA.

His research interests include image processing and analysis, artificial neural networks, image and video coding, multimedia systems and medical imaging. He has published 60 papers, 25 in international journals and 35 in proceedings of international conferences. He is reviewer of more than ten European and IEEE publications. He and his team have been participating in fifteen European and National projects.