

HYPERSPECTRAL ENDMEMBER EXTRACTION AND SIGNATURE CLASSIFICATION WITH MORPHOLOGICAL NETWORKS

Mark S. Schmalz Gerhard X. Ritter

Department of Computer and Information Science and Engineering
University of Florida, Gainesville FL 32611-6120
mssz@cise.ufl.edu ritter@cise.ufl.edu

ABSTRACT

Accurate multispectral or hyperspectral signature classification is key to the nonimaging detection and recognition of space objects. In imaging applications, signature classification accuracy depends on accurate spectral endmember determination. Previous approaches to endmember computation and signature classification were based on classical neural networks (CNNs) expressed in terms of the algebra $(\mathbf{R}, +, \times)$. Unfortunately, class separation in CNNs tends to be suboptimal, the number of signatures that can be accurately classified often depends linearly on the number of NN inputs, and computational cost is high. This can lead to poor endmember distinction, as well as potentially significant classification errors in the presence of noise or densely interleaved signatures.

In contrast to traditional CNNs, autoassociative morphological memories (AMMs) are defined on the $(\mathbf{R}, +, \vee)$ lattice algebra. Unlimited storage and perfect recall of noiseless real valued patterns has been proven for AMMs. Although AMMs suffer from sensitivity to specific noise models such as erosive and dilative noise, it is possible to specify input error limits such that individual patterns can be correctly classified with pattern-specific error tolerances. In nonimaging applications, this supports the variable specification of spectral signature noise or error in terms of individual spectral bands as well as for each spectral signature. It has been shown that AMMs can be based on dendritic computation. These techniques yield improved accuracy and class segmentation/separation ability in the presence of highly interleaved signature data. In imaging applications, the prior definition of a set of endmembers corresponds to material spectra lying on vertices of the minimum convex region covering the image data. These vertices can be characterized as morphologically independent patterns that can be extracted with AMMs.

In this paper, we overview theory and examples of AMM based classification of complex patterns at very high input dimensionality, in the presence of high levels of noise and input error. We show how the superior performance of AMMs can support spectral signature classification in nonimaging detection of space objects. We also summarize recent results in spectral endmember determination based on morphological computation. We show that detected endmembers can be extracted that closely match known endmember spectra. Additional discussion pertains to computational complexity of AMMs, which are superior to classical NN based techniques.

Keywords: Pattern recognition, Space object detection, Morphological neural networks

1. INTRODUCTION

The detection and classification of space objects using spectral data is an emerging area of research that depends upon precise spectrometry, as well as accurate but comprehensive classifiers. Advances in passive remote sensing have produced imaging devices with increasing spectral coverage and resolution. High spectral resolution produced by current hyperspectral devices facilitates identification of fundamental materials that comprise remotely sensed objects, and thus supports discrimination based on such derived measures. The spectra of such fundamental materials (e.g., aluminum, silicon in solar panels, Mylar, etc.) are called *endmembers*.

In this paper, we discuss recent developments in an emerging technology called *morphological neural networks* (MNNs), which are neural nets that utilize a nonlinear kernel whose operations are derived from the theory of mathematical morphology [1]. Here, MNNs are evaluated for their applicability to two problems in spectral classification of space objects. Firstly, we consider non-imaging applications, where a spectrometer is affixed to a telescope that does not detect a space object as an image, but as a collection of scalar intensities at different wavelengths. Secondly, we expand on the non-imaging case to overview the analysis of hyperspectral imagery from which endmembers are extracted using morphological associative memories.

In both imaging and non-imaging applications, hyperspectral image segmentation and analysis takes the form of a pattern recognition problem. In principle, the segmentation problem reduces to the challenge of matching the

hyperspectral data to predetermined spectra stored in a library. In practice, template-based approaches applied directly to spectral wavelengths are more often found in non-imaging applications.

In imaging applications, endmembers cannot be determined *a priori*, and must be selected from the image by identifying the pixel spectra that are most likely to represent the fundamental materials. For example, if one seeks to detect an orbiting satellite, then one would look for metals, other inorganic materials, or plastics that are expected to be part of the satellite. This ability to automatically determine endmembers is called the *autonomous endmember detection problem*. Unfortunately, the spatial resolution of a sensor makes it often unlikely that any pixel is composed of a single endmember. Thus, the determination of endmembers becomes a search for image pixels with the least contamination from other endmembers. These are occasionally referred to as *pure pixels*. The pure pixels exhibit maximal reflectance in certain spectral bands and correspond to vertices of a high-dimensional simplex. The simplex whose vertices are the endmembers encloses most, if not all, of the pixel spectra.

Our discussion of imaging applications assumes that the linear mixing model, which is based on the fact that points of a simplex can be represented as a linear sum of the vertices that determine the simplex. We assume that this is a reasonable model, and overview a method for autonomous endmember detection that is based on the theory of morphological associative memories, which is directly related to the theory of morphological neural networks.

This paper is organized as follows. In Section 2, we overview the theory of morphological neural nets and morphological associative memories. This section provides the necessary background for morphological autoassociative memories and established facts that are pertinent to endmember detection. Section 3 discusses implementational issues for non-imaging and imaging applications of spectral detection and classification using morphological memories. We also overview the theory of linear spectral unmixing with endmembers, with an example of endmember extraction using morphological memories. It is shown that morphological memories are useful and accurate for non-imaging applications, and that spectral endmember detection is feasible using morphological techniques. Conclusions and suggestions for future work are given in Section 4.

2. THEORETICAL SUMMARY

2.1. Morphological Neural Networks

Attempts to mathematically model the structure and function of biological neurons such as those illustrated in Figure 1a have, until recently, centered on the linear inner product, which sums m partial products. In particular, artificial neural networks are customarily specified in terms of network connectivity, node characteristics, and training or learning rules. The underlying algebra contains the real numbers \mathbf{R} with the operations of addition and multiplication as well as their governing laws, and forms a *ring* $(\mathbf{R}, +, \times)$. The classical NN model has the following governing equation

$$\mathbf{a}_j(t+1) = f\left(\sum_{i=1}^N \mathbf{a}_i(t) \cdot \mathbf{w}_{ij} - \theta_j\right), \quad (1)$$

where $\mathbf{a}_j(t)$ denotes the value of the j -th neuron at time t , N represents the total number of neurons, \mathbf{w}_{ij} denotes the synaptic connectivity value between the i -th and j -th neurons, θ_j denotes a threshold, and f represents the next-state function that usually introduces a nonlinearity into the network. All current network models cannot be precisely described in this manner, but can be viewed as variations of Equation (1), which describes a *Hopfield network*. This neural model is illustrated schematically in Figure 1b.

It is well known that content-addressable memories based on the Hopfield net have two key deficiencies. Firstly, the number of patterns M that can be stored and accurately recalled is *severely* limited. If M is too large, then the net may converge to a spurious pattern that differs from all stored exemplars, which produces a “no-match” output in classifier applications [2]. Hopfield remedied this defect by generating patterns *randomly* and keeping the number of classes $N_c < 0.15N$. For example, a Hopfield net for a simple ($N_c = 10$ class) problem would require more than 70 nodes and over 5,000 connection weights. Secondly, exemplar patterns in the Hopfield net are *unstable* if they share many bits in common with other exemplars. By unstable, we mean that after the exemplar is applied to the net’s input, the net converges to a different exemplar. Thirdly, the multiplication operations inherent in the inner product formulation incur high computational cost, especially when classification is iterative.

In response to this situation, UF has developed under DoD sponsorship an emerging pattern recognition technology called *Morphological Neural Networks* (MNNs), which is revolutionizing neural network based information classification and decision systems by (a) increasing system information capacity, speed, and reliability, as well as (b) drastically increasing classification accuracy in the presence of noise, sensor error, or partial

information. Such goals are achieved due to the enhanced information storage capacity, fast and accurate convergence behavior, short training times, and much lower computational cost exhibited by MNNs [3,4].

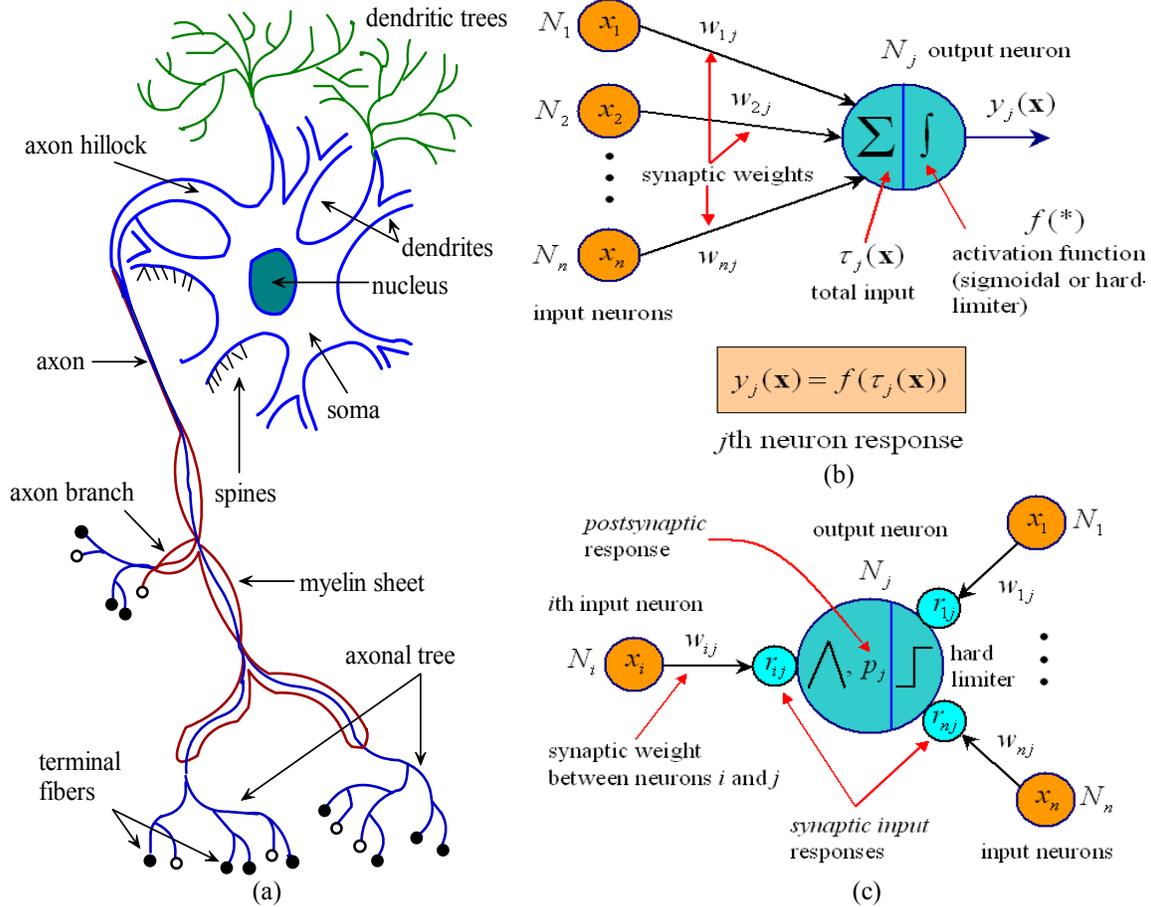


Figure 1. Neural Models: (a) biological neuron, (b) artificial neuron computed in terms of a linear inner-product, and (c) artificial morphological neuron.

Morphological neural nets are a natural outgrowth of the theory of image algebra [5]. A subalgebra of image algebra expresses currently popular NN models. Since the first attempts at formulating useful MNNs, only a few papers involving morphological NNs have appeared. Davidson employed MNNs for template identification and target classification problems [6]. C.P. Suarez-Araujo applied MNNs to compute homothetic auditory and visual invariances [7]. However, the preceding two researchers devised dedicated, special-purpose MNNs only. In contrast, Ritter and Sussner present a more general approach to MNNs that provides foundations for future research concerning the behavior, capabilities, and applications of MNNs [3,4]. Ritter and Sussner proved that MNNs are useful as associative memories and perceptrons, where MNNs have the computing capabilities of their classical counterparts. Additionally, it was shown that MNNs are computationally complete and efficient.

As previously noted, computation represented by the left member of Equation (1) is based on the algebraic structure $(\mathbf{R}, +, \times)$. In contrast, the basic computation occurring in a morphological neuron is based on the *semi-ring* $(\mathbf{R}_{-\infty, \vee, +})$ or $(\mathbf{R}_{\infty, \wedge, +})$. Here, the *extended real numbers* $\mathbf{R}_{-\infty} = \mathbf{R} \cup \{-\infty\}$ and $\mathbf{R}_{\infty} = \mathbf{R} \cup \{\infty\}$ have the following basic arithmetic and logic operations. The symbol $+$ denotes the usual addition operation with the stipulation that $a + (-\infty) = (-\infty) + a = -\infty \forall a \in \mathbf{R}_{-\infty}$. Similarly, $a + \infty = \infty + a = \infty \forall a \in \mathbf{R}_{\infty}$. The symbols \vee and \wedge respectively denote the binary operations of maximum and minimum, with the additional stipulation that $a \vee (-\infty) = (-\infty) \vee a = a \forall a \in \mathbf{R}_{-\infty}$ and $a \wedge \infty = \infty \wedge a = a \forall a \in \mathbf{R}_{\infty}$.

Note that the symbol $-\infty$ acts like a *zero* element in $(\mathbf{R}_{-\infty, \vee, +})$ if one views \vee as *addition* and $+$ as *multiplication*. Similar comments hold for ∞ in $(\mathbf{R}_{\infty, \wedge, +})$. Also, the role of the multiplicative identity in the

structure $(\mathbf{R}_{+, \times})$ is filled by the number 1 (i.e., $1 \cdot a = a \cdot 1 = a \quad \forall a \in \mathbf{R}$). In the structures $(\mathbf{R}_{-\infty, \vee, +})$ and $(\mathbf{R}_{\infty, \wedge, +'})$, this role is filled by 0, since $0 + a = a + 0 = a \quad \forall a \in \mathbf{R}$.

Using $(\mathbf{R}_{-\infty, \vee, +})$, the morphological neural net is governed by the following equation:

$$\mathbf{a}_j(t+1) = f \left(\mathbf{p}_j \cdot \bigvee_{i=1}^N \mathbf{r}_{ij} (\mathbf{a}_i(t) + \mathbf{w}_{ij}) \right), \quad (2)$$

where $\mathbf{r}_{ij} = \pm 1$ denotes the excitatory (+) or inhibitory (-) influence of the i -th neuron on the j -th neuron, and $\mathbf{p}_j = \pm 1$ denotes the post-synaptic response of the j -th neuron to the total received input. Via the dual structure $(\mathbf{R}_{\infty, \wedge, +'})$, the preceding equation becomes

$$\mathbf{a}_j(t+1) = f \left(\mathbf{p}_j \cdot \bigwedge_{i=1}^N \mathbf{r}_{ij} (\mathbf{a}_i(t) + \mathbf{w}_{ij}) \right). \quad (3)$$

Equations (2) and (3) represent the basic morphological operations of *dilation* and *erosion* – hence the term *morphological neural network*. The first of these two equations is instantiated in the *morphological neuron* model depicted schematically in Figure 1c.

In contrast to the Hopfield memory, in the absence of noise a morphological autoassociative memory will *always* provide perfect recall as long as any one of the exemplar patterns is not a subset of another exemplar pattern. We have found that the morphological autoassociative memory provides perfect recall for any arbitrary number of patterns [4]. In contrast, the Hopfield net can neither recall a large number of noiseless exemplar patterns nor perform well when patterns are noise or partially occluded.

The autoassociative morphological memories investigated in our previous and current DoD-funded research in morphological neural nets are extremely robust to certain types of noise and occlusions. For example, erosive noise due to missing elements of an input vector scarcely affects a morphological memory W that is compensated for erosive noise. Similarly, the dual of W is robust in the presence of dilative noise. Additionally, due to the replacement of multiplication in a classical NN with addition in the morphological NN, computation is extremely fast, which is further reinforced by convergence in one pass of the network.

Recent advances in morphological neural net technology include the development of morphological associative memories that compute with dendrites, as overviewed in the following subsection.

2.2. Morphological Associative Memories and Dendritic Computation

Examination of the biological neuron depicted in Figure 1a reveals two processes – *dendrites* and *axons*. The axon is the principal fiber that forms toward its ends a multitude of branches called the *axonal tree*. The tips of these branches, called *nerve terminals* or *synaptic knobs*, make contact with the dendritic structure of other neurons. These sites of contact are called *synaptic sites*. The synaptic sites of dendrites are the places where synapses occur. Dendrites have many branches that create large and complicated trees: the number of synapses on a *single* neuron of the human cerebral cortex typically ranges between 500 and 200,000. It is also well known that there exist two types of synapses: *excitatory synapses*, which play a role in exciting the postsynaptic cell to fire impulses; and *inhibitory synapses*, which attempt to prevent the neuron from firing impulses in response to excitatory synapses. The postsynaptic membranes of the dendrites will thus either accept or inhibit the received input from other neurons.

It is worthwhile to note that dendrites comprise the largest component in both the surface and the volume of the human brain. Part of this is due to the fact that dendrites span all cortical layers in all regions of the cerebral cortex [8-10]. Thus, when attempting to model artificial brain networks one cannot ignore dendrites, which make up more than 50 percent of the neuron's outer structure. This is especially true in light of the fact that some researchers have proposed that dendrites, and not the neurons, are the elementary computing devices of the brain, capable of implementing such logical functions as *and*, *or*, and *not* [8-11].

In order to take advantage of recent advances in neurobiology and the biophysics of neural computation, and to nudge the field of artificial neural networks (ANNs) back to its original roots, Ritter proposed a model of single-neuron computation that takes into account the computations performed by dendrites [12]. Extrapolating on this model, we constructed a single-layer, feedforward neural net based on dendritic computing within the lattice domain [13]. In this model, a set of n input neurons N_1, \dots, N_n provides information through its axonal arborization to the dendritic trees of a set of m neurons M_1, \dots, M_m . Explicitly, the state value of a neuron N_i ($1 \leq i \leq n$) propagates

through its axonal tree all the way to the terminal branches that make contact with the neuron M_j ($1 \leq j \leq m$). The weight of an axonal branch of neuron N_i terminating on the k -th dendrite of M_j is denoted by w_{ijk}^l , where the superscript $l \in \{0,1\}$ distinguishes between *excitatory* ($l = 1$) and *inhibitory* ($l = 0$) input to the dendrite. The k -th dendrite of M_j will respond to the total input received from the neurons N_1, \dots, N_n , and will either accept or inhibit the received input. The computation of the k -th dendrite of M_j is given by

$$\tau_k^j(\mathbf{x}) = p_{jk} \bigwedge_{i \in I(k)} \bigwedge_{l \in L(i)} (-1)^{1-l} (x_i + w_{ijk}^l), \quad (4)$$

where $\mathbf{x} = (x_1, \dots, x_n)$ denotes the input value of the neurons N_1, \dots, N_n with x_i representing the value of N_i , $I(k) \subseteq \{1, \dots, n\}$ corresponds to the set of all input neurons with terminal fibers that synapse on the k^{th} dendrite of M_j , $L(i) \subseteq \{0,1\}$ corresponds to the set of terminal fibers of N_i that synapse on the k^{th} dendrite of M_j , and $p_{jk} \in \{-1,1\}$ denotes the excitatory ($p_{jk} = 1$) or inhibitory ($p_{jk} = -1$) response of the k^{th} dendrite of M_j to the received input.

It follows from the formulation $L(i) \subseteq \{0,1\}$ that the i^{th} neuron N_i can have at most two synapses on a given dendrite k . Also, if the value $l = 1$, then the input $(x_i + w_{ijk}^1)$ is excitatory, and inhibitory of $l = 0$, since in this case we have $-(x_i + w_{ijk}^0)$.

The value $\tau_k^j(\mathbf{x})$ is passed to the cell body of the neuron, and the state of M_j is a function of the input received from all its dendrites. The total value received by M_j is given by

$$\tau^j(\mathbf{x}) = p_j \bigwedge_{k=1}^{K_j} \tau_k^j(\mathbf{x}), \quad (5)$$

where K_j denotes the total number of dendrites of M_j , and $p_j = \pm 1$ denotes the response of the cell body to the received dendritic input. As before, $p_j = 1$ denotes the input is accepted, while $p_j = -1$ denotes that the cell rejects the received input. The *next* state of M_j is determined by an activation function f , namely, $y_j = f[\tau^j(\mathbf{x})]$. The total computation of M_j is, therefore, given by

$$y_j(\mathbf{x}) = f \left\{ p_j \bigwedge_{k=1}^{K_j} \left[p_{jk} \bigwedge_{i \in I(k)} \bigwedge_{l \in L(i)} (-1)^{1-l} (x_i + w_{ijk}^l) \right] \right\}. \quad (6)$$

Figure 2a provides a graphic representation of this model.

Note that, at first glance, Equations (4) and (5) appear to involve only addition and minimization. However, due to the duality $a \vee b = -(-a \wedge -b)$, and the fact that $(-1)^{1-l} = -1$ for $l = 0$, as well as $p_j, p_{jk} \in \{-1, 1\}$, it follows that the maximum operation is implicitly included in these equations.

A modification of the lattice-based perceptron with dendritic structure leads to a different autoassociative memory than the lattice matrix memories discussed previously. For this new autoassociative memory, a set of sensor (input) neurons N_1, \dots, N_n that receives input \mathbf{x} from the space \mathbf{R}^n with N_i receiving input x_i , the i^{th} coordinate of \mathbf{x} is defined. If, as before, $X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\} \in \mathbf{R}^n$ represents the set of exemplar patterns, then the input neurons will propagate their input values x_i to a set of k hidden neurons H_1, \dots, H_k , where each H_j has exactly one dendrite. Every input neuron N_i has exactly two axonal fibers terminating on the dendrite of H_j , and the weights of these fibers are given by

$$w_{ij}^l = \begin{cases} -(x_i^j - \alpha) & \text{if } l = 1 \\ -(x_i^j + \alpha) & \text{if } l = 0 \end{cases}, \quad (7)$$

where $i = 1, \dots, n$ and $j = 1, \dots, k$. The parameter $\alpha > 0$ is a user-defined error limit that must satisfy the inequality $\alpha < \frac{1}{2}d_{\min}$, where

$$2\alpha < d_{\min} = \min \{d(\mathbf{x}^\xi, \mathbf{x}^\gamma) : \xi < \gamma \text{ with } \xi, \gamma \in \{1, \dots, k\}\} \quad (8)$$

and $d(\mathbf{x}^\xi, \mathbf{x}^\gamma)$ denotes the Chebyshev (checkerboard) distance between the patterns \mathbf{x}^ξ and \mathbf{x}^γ , defined by

$$d(\mathbf{x}^\xi, \mathbf{x}^\gamma) = \max \left\{ |x_i^\xi - x_i^\gamma| : i = 1, \dots, n \right\}. \quad (9)$$

For a given input $\mathbf{x} \in \mathbf{R}^n$, the dendrite of the hidden unit H_j computes

$$\tau^j(\mathbf{x}) = \bigwedge_{i=1}^n \bigwedge_{l=0}^1 (-1)^{l-1} (x_i + w_{ij}^l). \quad (10)$$

The state of the neuron H_j is determined by the hard-limiter activation function

$$f(z) = \begin{cases} 0 & \text{if } z \geq 0 \\ -\infty & \text{if } z < 0 \end{cases}. \quad (11)$$

The output of H_j is given by $f[\tau^j(\mathbf{x})]$ and is passed along its axonal fibers to n output neurons M_1, \dots, M_n . The activation function defined by Equation (11) is a hard-limiter in the algebra $\mathcal{H} = (\mathbf{R}_{-\infty}, \vee, +)$ since the zero of \mathcal{H} is $-\infty$ (for the operation \vee) and the unit of \mathcal{H} corresponds to 0. This mirrors the hard-limiter in the algebra $(\mathbf{R}, +, \times)$ defined by $f(z) = 0$ if $z < 0$ and $f(z) = 1$ if $z \geq 0$, since in this algebra the zero is 0 and the unit is 1.

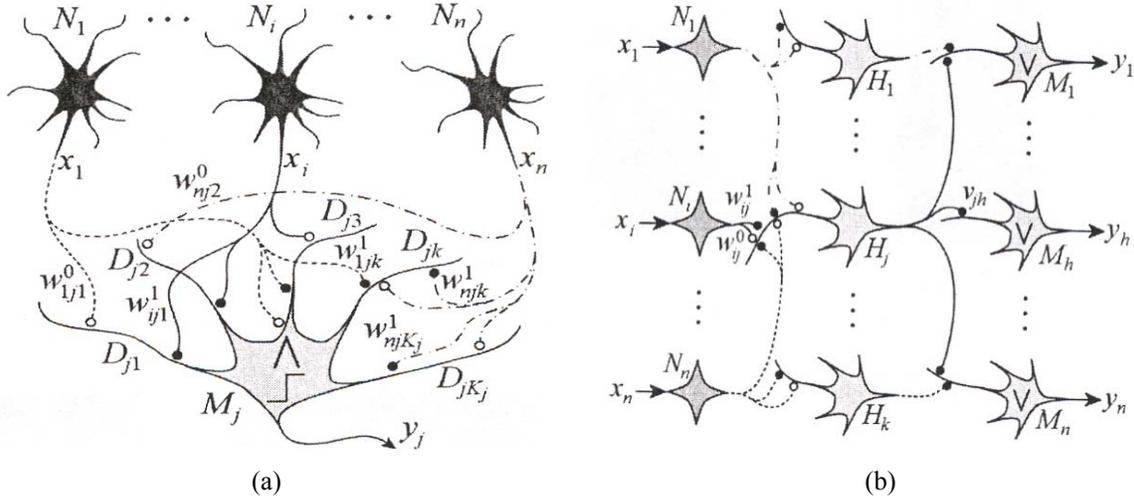


Figure 2. Dendritic models: (a) Morphological perceptron with dendritic structure. Terminations of excitatory and inhibitory fibers are marked with \bullet and \circ , respectively. D_{jk} denotes dendrite k of M_j and K_j its number of dendrites. Neuron N_i can synapse D_{jk} with excitatory or inhibitory fibers (e.g., weights w_{1jk}^1 and w_{nj2}^0 respectively denote excitatory and inhibitory fibers from N_1 to D_{jk} and from N_n to D_{j2}); (b) Topology of a dendritic model of an autoassociative memory, where the network is fully connected: all axonal branches from input neurons synapse via two fibers on all hidden neurons, which in turn connect to all output nodes via excitatory fibers. For clarity, only a few connections are illustrated.

Similar to the hidden layer neurons, each output neuron M_h has one dendrite, where $h = 1, \dots, n$. However, each hidden neuron H_j has exactly one excitatory axonal fiber and no inhibitory fibers terminating on the dendrite of M_h . Figure 2b illustrates this dendritic network model, where the excitatory fiber of H_j terminating on M_h has synaptic weight $v_{jh} = x_h^j$, and the computation performed by M_h is given by

$$\tau^h(\mathbf{s}) = \bigvee_{j=1}^k (s_j + v_{jh}), \quad (12)$$

with s_j denoting the output of H_j . Namely, $s_j = f[\tau^j(\mathbf{x})]$, where f was defined in Equation (11). The activation function g for each output neuron M_h is the linear identity function $g(x) = z$.

Each neuron H_j will have the output value $s_j = 0$ if and only if \mathbf{x} is an element of the hypercube

$$B^j = \left\{ \mathbf{x} \in \mathbf{R}^n : x_i^j - \alpha \leq x_i \leq x_i^j + \alpha \right\} \quad (13)$$

and $s_j = -\infty$ whenever $\mathbf{x} \in \mathbf{R}^n \setminus B^j$. Thus, the output of this network will be

$$\mathbf{y} = (y_1, \dots, y_n)' = (x_1^j, \dots, x_n^j)' = \mathbf{x}^j \quad (14)$$

if and only if $\mathbf{x} \in B^j$. That is, whenever \mathbf{x} is a corrupted version of \mathbf{x}^j with each coordinate of \mathbf{x} not exceeding the allowable error α , then \mathbf{x} will be associated with \mathbf{x}^j . If the error magnitude exceeds α , then the network rejects the input, which is indicated by the output vector

$$\mathbf{y} = (-\infty, \dots, -\infty)' \quad (15)$$

It is readily seen that each non-erroneous pattern \mathbf{x}^ξ will be associated with \mathbf{x}^ξ .

A further improvement on this model makes the error parameter α dependent on each $\xi = 1, \dots, k$ and each coordinate $i = 1, \dots, n$. This allows greater freedom in network design, to meet the constraints of a specific problem. This approach is especially important in spectral signature classification. For example, consider a library of n exemplar signatures, against which a test signature \mathbf{x} is to be compared and classified. Some of the n signatures will have been acquired by different sensors, or by the same sensor but under different environmental conditions. As a result, the response of each spectral band can vary across the n acquired signatures. Thus, it is important to vary the error parameter α with respect to the spectral band index modeled by ξ , and the exemplar index modeled by i .

In order to implement this highly precise and flexible error specification technique, the network weights can be bounded as follows:

$$w_{ij}^l = \begin{cases} -(x_i^j - \alpha_i^1(j)) & \text{if } l = 1 \\ -(x_i^j + \alpha_i^0(j)) & \text{if } l = 0 \end{cases} \quad (16)$$

As shown in Figure 3, the hyperboxes are not centered at \mathbf{x}^ξ , thereby allowing the aforementioned error specification. For example, each point in the box denoted by

$$B^j = \left\{ \mathbf{x} \in \mathbf{R}^n : x_i^j - \alpha_i^1(j) \leq x_i \leq x_i^j + \alpha_i^0(j) \right\} \quad (17)$$

is associated with the pattern \mathbf{y}^j . This figure also illustrates the superiority of this model over the lattice correlation memory through the elimination of problems associated with the set of fixed points of W_{XX} and M_{XX} .

A simple modification of the dendritic autoassociative memory leads to a heteroassociative memory that exploits the dendritic computing concept. The modification is simple because we only change the weights of the axonal fibers of the hidden neurons and at the output layer. In particular, suppose that the associated pattern sets are given by $X = \{\mathbf{x}^1, \dots, \mathbf{x}^k\} \subset \mathbf{R}^n$ and $Y = \{\mathbf{y}^1, \dots, \mathbf{y}^k\} \subset \mathbf{R}^m$. In this case, the topology of the network is the same as that of the autoassociative memory shown in Figure 2b, except that the output layer now consists of m neurons, namely, M_1, \dots, M_m . As before, each output neuron M_h , where $h = 1, \dots, m$, has one dendrite, and each hidden neuron H_j has exactly one excitatory axonal fiber and no inhibitory fibers terminating on the dendrite of M_h . However, the excitatory fiber of H_j , terminating on M_h , now has synaptic weight $v_{jh} = y_h^j$. The computation performed by M_h is given by Equation (12) and the activation function g for each output neuron M_h is the linear identity function $g(z) = z$.

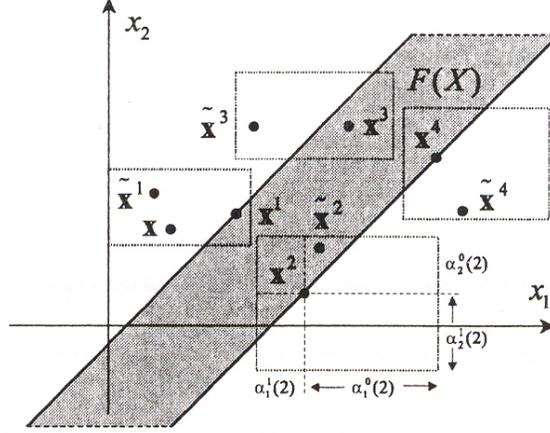


Figure 3. Hyperboxes with variable error levels $\alpha_i^l(j)$, where each pattern in the box containing \mathbf{x}^j will be associated with \mathbf{x}^j .

Hence, again, each neuron H_j will have the output value $s_j = 0$ if and only if \mathbf{x} is an element of the hypercube B^j defined by Equation (13), and $s_j = -\infty$ whenever $\mathbf{x} \in \mathbf{R}^n \setminus B^j$. Thus, the output of this network will be

$$\mathbf{y} = (y_1, \dots, y_m)' = (y_1^j, \dots, y_m^j)' = \mathbf{y}^j \quad (18)$$

if and only if $\mathbf{x} \in B^j$. That is, whenever \mathbf{x} is a corrupted version of \mathbf{x}^j with each coordinate of \mathbf{x} not exceeding the allowable error level α , then \mathbf{x} will be associated with \mathbf{y}^j . If the amount of error (e.g., input noise) exceeds the level α , then the network rejects the input by yielding the output vector $(-\infty, \dots, -\infty)'$. It is readily seen that each uncorrupted pattern \mathbf{x}^ξ will be associated with \mathbf{y}^ξ .

Example 1. As an example of this associative memory, consider the associated image pairs $P = \{\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3\}$ and $Q = \{\mathbf{q}^1, \mathbf{q}^2, \mathbf{q}^3\}$ shown in Figure 4a. Each \mathbf{p}^ξ is a 50x50-pixel 256-level grayscale image, whereas each \mathbf{q}^ξ is a 30x50-pixel 256-level grayscale image, with $\xi = 1, 2, 3$. Using the standard row-scan technique, each pattern image \mathbf{p}^ξ and \mathbf{q}^ξ was converted into an associated pair of pattern vectors $\mathbf{x}^\xi = (x_1^\xi, \dots, x_{2500}^\xi)$ and $\mathbf{y}^\xi = (y_1^\xi, \dots, y_{1500}^\xi)$. Thus, for this particular example, we have $X = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3\} \subset \mathbf{R}^{2500}$ and $Y = \{\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3\} \subset \mathbf{R}^{1500}$.

Example 2. The images in Figure 4 were distorted by randomly corrupting 95 percent of the coordinates of each \mathbf{x}^ξ with a noise level α that satisfies the inequality in Equation (8). Here, we set $\alpha = 2d_{min}/5$. Numerically, we have $d(\mathbf{x}^1, \mathbf{x}^2) = 198$, $d(\mathbf{x}^1, \mathbf{x}^3) = 211$, and $d(\mathbf{x}^2, \mathbf{x}^3) = 188$, where the range of pixel values (feature vectors in pattern space) is $[0, 255]$. Hence, $d_{min}(\mathbf{x}^2, \mathbf{x}^3) = 188$ and $\alpha = 75.2$. The top row of Figure 4b shows the corrupted images, while the bottom row shows the perfect recall association achieved by the network. In this experiment, recall succeeds because the amount of distortion is controlled not to exceed the allowable level α . When the error or distortion exceeds α , the memory described thus far fails to recognize the input patterns and rejects them by outputting a vector $(-\infty, \dots, -\infty)'$.

This model can be refined to achieve greater error tolerance, by employing the concept of dimension- and pattern-specific noise intervals, as discussed previously (e.g., in Equation 16). That is, we set individual error parameters for each $\xi \in \{0, \dots, k\}$, which satisfy the inequality

$$\alpha_\xi < \frac{1}{2} \min \{d(\mathbf{x}^\xi, \mathbf{x}^\gamma) : \gamma \in K(\xi)\}, \quad (19)$$

where $K(\xi) = \{1, \dots, k\} \setminus \{\xi\}$. This model has weights of axonal fibers of N_i terminating on the dendrite of H_j that are given by

$$w_{ij}^l = \begin{cases} -(x_i^j - \alpha_j) & \text{if } l = 1 \\ -(x_i^j + \alpha_j) & \text{if } l = 0 \end{cases}, \quad (20)$$

where $i = 1, \dots, n$ and $j = 1, \dots, k$, instead of being set as shown in Equation (16). For a given input $\mathbf{x} \in \mathbf{R}^n$, computation at the hidden and output layers proceeds as before, according to Equations (10) through (12).

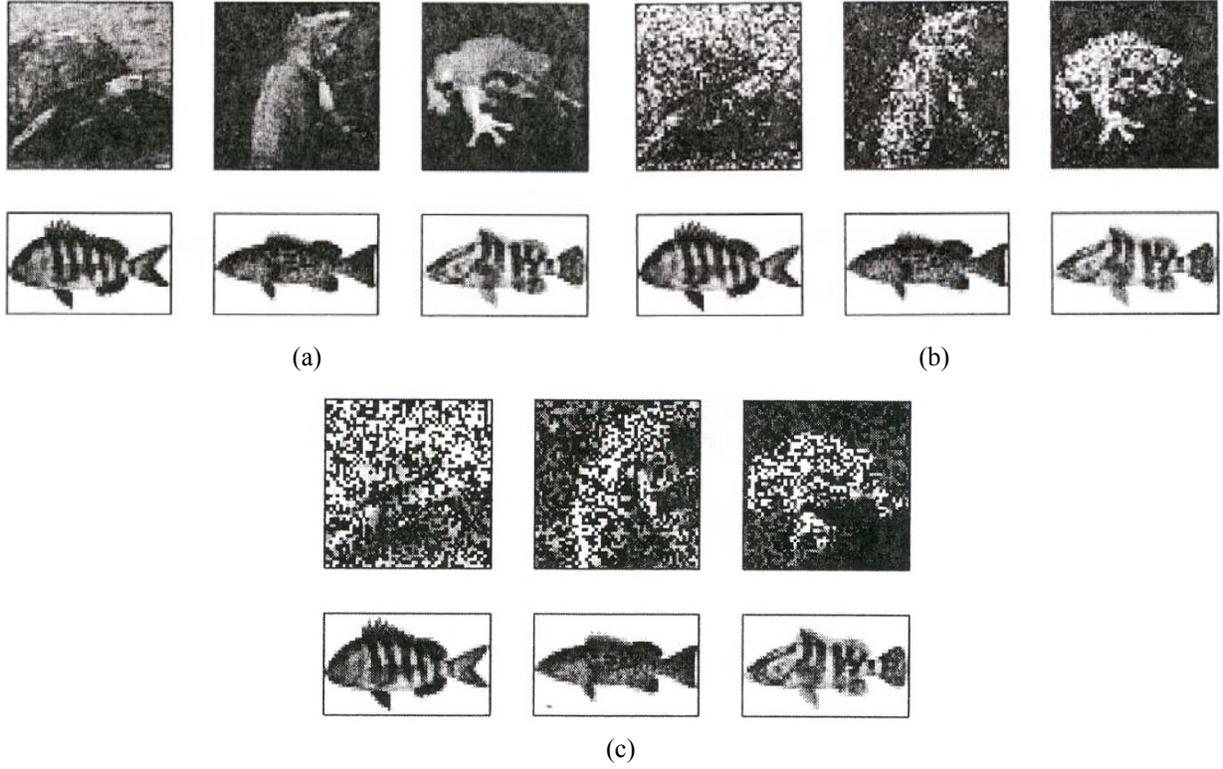


Figure 4. Examples of MNN-based classification of complex grayscale patterns: (a) Uncorrupted source images, where the top row of images illustrates input patterns $\mathbf{p}^1, \mathbf{p}^2, \mathbf{p}^3$, which were converted to prototype patterns of the set $X = \{\mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3\}$, while the bottom row shows the images from which associated patterns from the set $Y = \{\mathbf{y}^1, \mathbf{y}^2, \mathbf{y}^3\}$ were generated; (b) Noise-corrupted versions of the input patterns in a) yield perfect recall, per Example 2; (c) Extremely noise-corrupted version of input patterns in a), with perfect recall achieved via pattern-specific error specification, per Example 3.

In the revised model, each output neuron H_j has a value $s_j = 0$ if and only if \mathbf{x} is an element of the hypercube

$$B^j = \left\{ \mathbf{x} \in \mathbf{R}^n : x_i^j - \alpha_j \leq x_i \leq x_i^j + \alpha_j \right\} \quad (21)$$

and $s_j = -\infty$ whenever $\mathbf{x} \in \mathbf{R}^n \setminus B^j$. In contrast to Equation (13), the hypercubes constructed at the hidden neurons have different sizes. These hypercubes are at least as large as the one in Equation (13), since the parameters α_ξ can be chosen greater than α and still satisfy the inequality in Equation (19).

Example 3. The patterns shown in Figure 4c were obtained by distorting 100 percent of the vector components of each \mathbf{x}^ξ within a noise level α_ξ , which was chosen to satisfy the inequality in Equation (19). Specifically, $\alpha_1 = 0.49 \min[d(\mathbf{x}^1, \mathbf{x}^2), d(\mathbf{x}^1, \mathbf{x}^3)] \sim 97$, $\alpha_2 = 0.49 \min[d(\mathbf{x}^1, \mathbf{x}^2), d(\mathbf{x}^2, \mathbf{x}^3)] \sim 92$, and $\alpha_3 = 0.49 \min[d(\mathbf{x}^1, \mathbf{x}^3), d(\mathbf{x}^2, \mathbf{x}^3)] \sim 92$, where the values were obtained by truncation to the nearest integer. Figure 4c shows that the refined model achieves perfect recall, in the presence of severe noise-induced distortion.

3. SIGNATURE CLASSIFICATION AND ENDMEMBER DETECTION

Although the examples shown in Figure 4 may initially appear to have pedagogic value only, they are directly related to the problem of spectral signature classification that is the focus of this paper. In particular, scanning each input image in Figure 4 (top row of each group of six images) yields an input pattern vector with 2500 dimensions, which exceeds the number of bands in current hyperspectral detectors. Each dimension is characterized by a

number that has 256 graylevels, similar to the single-channel output of a hyperspectral detector having 256 possible values. Hyperspectral signature classification typically involves inputting a signature vector to a classifier that outputs an index which points into a table of signature names, each containing at most several bytes of data. However, the examples shown in Figure 4 address the much more difficult problem of mapping a 2500-dimensional input to a 1500-dimensional output, which the morphological network technology achieves with perfect recall in the presence of high noise levels. Furthermore, this morphological mapping is performed with relatively small computational effort, since convergence occurs in one pass through the network.

Example 4. Figure 5a illustrates how a morphological dendritic classifier could be applied to classify two test spectra S_1 and S_2 , in terms of a reference spectrum S_R . Here, the error intervals particular to each spectrum are shown in red. Observe that S_1 is similar to S_R , because the values of S_1 occur within the error bounds placed on S_R . Thus, an MNN-based classifier would identify S_1 as equal to S_R . In contrast, the values of S_2 do not all occur within the associated error limit placed on S_R , so S_2 is classified as not equal to S_R .

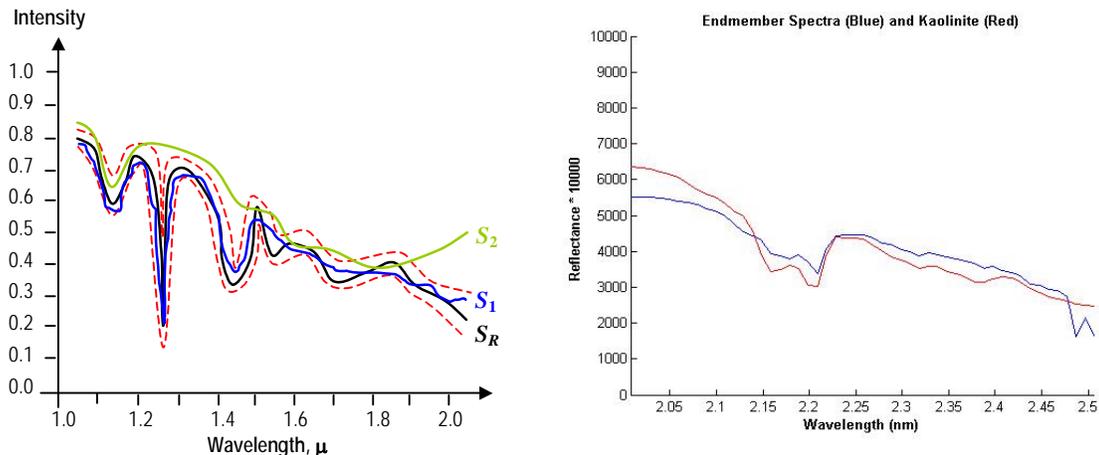


Figure 5. (a) Example of morphological dendritic classifier applied to the determination of whether a test signature S_1 or S_2 is similar to a reference signature S_R , within error bounds placed on S_R ; (b) Example of morphologically based endmember detection in an imaging application: reconstruction (blue) of a reference spectrum (red) of kaolinite from AVIRIS hyperspectral image dataset taken at the Cuprite, NV test site.

In contrast, the imaging application of hyperspectral endmember detection is based on the theory of linear spectral unmixing. In particular, a typical pixel of a multi- or hyper-spectral image generally represents a spatial region consisting of several features. For example, in airborne remote sensing of terrestrial natural resources, the instantaneous field of view (IFOV) of a pixel is usually several square meters, which can subtend features such as a roof, grass or soil, rocks, etc. It is, therefore, useful to know the percentage of different object features that are represented in an image pixel. The most widely used spectral mixing model is the linear mixing model, which assumes that the observed reflectance spectrum of a given pixel is a linear combination of a small number of unique constituent deterministic spectral signatures known as *endmembers*. This model has been used by numerous researchers beginning with Adams et al. and their analysis of Martian terrain using four endmembers [15].

The mathematical equation and its constraints, which define the linear mixing model are given by:

$$\mathbf{x} = \sum_{k=1}^m a_k \mathbf{x}^k + \mathbf{n} = \mathbf{X} \cdot \mathbf{a} + \mathbf{n} \quad (22)$$

$$\sum_{k=1}^m a_k = 1 \quad \text{and} \quad a_k \geq 0 \text{ for } k = 1, \dots, m, \quad (23)$$

where \mathbf{x} denotes the measured spectrum of an image pixel, $\mathbf{x}^1, \dots, \mathbf{x}^m$ are the endmember spectra assumed to be affinely independent, a_1, \dots, a_k denote the corresponding fractional abundances (fraction of endmember spectra present in \mathbf{x}), and \mathbf{n} is an additive noise vector.

In practice, endmembers may be obtained from spectral libraries for certain specific materials, or autonomously from the image by a variety of techniques [16-19]. Autonomous endmember detection has received wide coverage in the literature since signatures of various objects that may be present in an image are unknown *a priori*. Boardman

[16] uses the framework of the geometry of convex sets to identify the $m+1$ endmembers as the vertices of the smallest simplex that bounds the measured spectral data. A major problem is that the vertices need not have a physical relationship to the actual image data.

Winter's N-FINDR is based on inflating a simplex within the dataset to determine the largest simplex "inscribed" within the data. It is not completely clear how pixels outside the inscribed dataset are handled, and the algorithm code is not yet available publicly. Additionally, the algorithm is computationally intensive, as individual pixels need to be examined and the simplex volume recalculated for each image pixel.

In contrast, autonomous endmember detection using morphological memories is extremely fast and carries little computational overhead [14,20]. The method is derived from examining the morphological autoassociative memory that stores the hyperspectral image cube. For example, a variety of iterative solutions have been proposed for endmember extraction, including recent developments by Myers using theory derived from morphological neural networks [20]. Initially, all hyperspectral signatures in an image dataset are located in the n -dimensional pattern space defined by an n -band hyperspectral signature. A convex region is constructed around the signatures. Myers' algorithm adapts the boundaries of the convex region to yield a simplex – a Euclidean geometric element having the minimum number of boundary points for a given value of n . The vertices of the simplex are the extracted spectral endmembers. Although a different type of morphological memory from that discussed in Section 2.2 was employed in Myers' algorithm, the governing principles are similar. Due to space limitations, the extensive theoretical development is not reproduced here, but can be referenced at [20].

In practice, Myers applied his technique for morphologically-based endmember detection to the segmentation of AVIRIS hyperspectral imagery in a publicly-available Cuprite, NV dataset. Spectral endmembers were extracted with high accuracy, as shown in Figure 5b. This accuracy has since been improved by recent enhancements to be discussed in a forthcoming publication. Additionally, Myers' approach requires much less computational work to extract endmembers, albeit with accuracy better than Winter's N-FINDR algorithm.

After the endmembers have been determined, the next step is to obtain a good estimate $\tilde{\mathbf{a}}$ of the abundances $\mathbf{a} = (a_1, \dots, a_m)'$. This step is called *the unmixing problem* and corresponds to a matrix inversion problem. More precisely, if $\tilde{\mathbf{a}}$ is an estimate of \mathbf{a} , then $\tilde{\mathbf{x}} = \mathbf{X} \cdot \tilde{\mathbf{a}}$ is an estimate of \mathbf{x} . Thus, the problem is to find an estimate $\tilde{\mathbf{a}}$ that minimizes the error

$$\varepsilon = \|\mathbf{x} - \tilde{\mathbf{x}}\|^2 = \|\mathbf{x} - \mathbf{X} \cdot \tilde{\mathbf{a}}\|^2 . \quad (24)$$

If the columns of \mathbf{X} are linearly independent, then there exists a unique vector

$$\tilde{\mathbf{a}} = (\mathbf{X}'\mathbf{X})^{-1} \mathbf{X}'\mathbf{x} \quad (25)$$

that minimizes the least-squares error ε .

4. CONCLUSIONS

The accurate and efficient classification of spectral signatures is key to detection of space objects in nonimaging applications. Morphological associative memories, derived from the theory of lattice algebra, can yield pattern classifiers that accurately categorize or match test signatures to a potentially large number of spectral exemplars stored in a reference library. This paper shows how a morphological memory is derived, and how error bounds are specified to facilitate spectral signature classification in realistic nonimaging applications. A neural network model was developed that supports accurate classification using concepts derived from dendritic computation in biological and artificial neural nets. Advantages of the morphological approach include theoretical maximum information storage capacity, fast convergence, accurate classification, and reduced computational cost.

Several examples of complex pattern classification in the presence of noise support the capability of morphological memories to classify hyperspectral signatures. Severely noise-corrupted patterns were classified correctly, even though these patterns were difficult to discern visually. We also showed how a morphological classifier could be configured to provide band- and signature-specific error levels for multiple signature classification tasks in the presence of significant levels of input noise. The problem of endmember extraction in hyperspectral imaging applications was overviewed, and preliminary results were presented vis-à-vis Myers' algorithm for iterative endmember extraction using morphological memories. It was demonstrated that Myers' technique accurately and efficiently approximates known endmember signatures.

Future work will involve improving the accuracy of endmember extraction in imaging applications, and providing for partial matching in nonimaging signature classification.

5. REFERENCES

1. Serra, J. *Image Analysis and Mathematical Morphology*, Orlando, FL: Academic Press, 1983.
2. Lippman, R. P. "An introduction to computing with neural networks", IEEE Transactions on Acoustics, Speech, and Signal Processing, pp. 4-22, April 1987.
3. Ritter, G.X., P. Sussner, and J.L. Diaz de Leon, "Morphological associative memories", IEEE Transactions on Neural Networks, Vol. 9, 281-293, 1998.
4. Ritter, G.X., J.L. Diaz de Leon, and P. Sussner, "Morphological bi-directional associative memories", Neural Networks, Vol. 12, 851-867, 1999.
5. Ritter, G.X. and J.N. Wilson, *Handbook of Computer Vision Algorithms in Image Algebra*, Second Edition, Boca Raton, FL: CRC Press, 2001.
6. Davidson, J.L. and F. Hummer, "Morphology neural networks: An introduction with applications", Circuits, Systems and Signal Processing, Vol. 12, 177-210, 1993.
7. Suarez-Araujo, C. "Novel neural network models for computing homothetic invariances: An image algebra notation", Journal of Mathematical Imaging and Vision, Vol. 7, 69-83, 1997.
8. Eccles, J.C., *The Understanding of the Brain*, New York: McGraw-Hill, 1977.
9. Koch, C. and I. Segev (Eds.), *Methods in Neuronal Modeling: From Synapses to Networks*, Boston: MIT Press, 1989.
10. Segev, I., "Dendritic Processing", in Arbib, M.A. (Ed.) *The Handbook of Brain Theory and Neural Networks*, pp. 282-289, Boston: MIT Press, 1998.
11. Holmes, W.R. and W. Rall, "Electronic Models of Neuron Dendrites and Single Neuron Computation", in McKenna, T., J. Davis, and S.F. Zornetzer (Eds.), *Single Neuron Computation*, pp. 7-25, San Diego: Academic Press, 1992.
12. Ritter, G.X. and G. Urcid, "Lattice algebra approach to single neuron computation", IEEE Transactions on Neural Networks, Vol. 14, 282-295, 2003.
13. Ritter, G.X., L. Iancu, and M.S. Schmalz, "A new auto-associative memory based on lattice algebra", in Sanfeliu, A., J.F. Trinidad, and J.A. Ochoa (Eds.), *Progress in Pattern Recognition, Image Analysis, and Applications*, in LNCS, Vol. 3287, 148-155, Berlin-Heidelberg: Springer-Verlag, 2004.
14. Grana, M., B. Raducanu, P. Sussner, and G.X. Ritter, "On endmember detection in hyperspectral images with morphological associative memories", Lecture Notes in Computer Science, Vol. 2757, 526-535, Berlin-Heidelberg: Springer, 2002.
15. Adams, J.B., M.O. Smith, and P.E. Johnson, "Spectral mixture modeling: A new analysis of rock and soil types at the Viking Lander 1 site", Journal of Geophysical Research, Vol. 91, 8098-8112, 1986.
16. Boardman, J.W., "Automated spectral unmixing of AVIRIS data using convex geometry concepts", AVIRIS Workshop Proceedings, JPL Publication 93-26, Jet Propulsion Laboratory, Pasadena, CA, pp. 11-14, 1993.
17. Roberts, D.A., M. Gardner, R. Church, S. Ustin, G. Sheer, and R.O. Green, "Mapping chaparral in the Santa Monica mountains using multiple endmember spectral mixture models", Remote Sensing of the Environment, Vol. 65, 267-279, 1998.
18. Winter, M.E., "Fast autonomous spectral endmember determination in hyperspectral data", Proceedings of the 13th International Conference on Applied Remote Sensing, Vol. 2, 337-344, Vancouver, BC, Canada, 1999.
19. Winter, M.E., "N-FINDR: An algorithm for fast autonomous spectral endmember determination in hyperspectral analysis", Proceedings SPIE, Vol. 3753, 266-275, Denver, CO, 1999.
20. Myers, D.S., "Hyperspectral endmember detection using morphological autoassociative memories", M.S. Thesis, University of Florida Department of Computer and Information Science and Engineering, 2005.