Survey of Neural Network Technology for Automatic Target Recognition

MICHAEL W. ROTH, MEMBER, IEEE

Abstract—Automatic target recognition (ATR) is a challenging problem that involves extraction of critical information about a target vehicle from a sequence of complex images. Previous ATR systems have been only partially successful and have produced high false-alarm rates. This paper reviews ATR and presents some of the highlights of neural network technology developments that have the potential for making a significant impact on ATR. In particular, neural network technology developments in the areas of collective computation, learning algorithms, expert systems, and neurocomputer hardware could provide crucial tools for developing improved algorithms and computational hardware for ATR.

I. Introduction

Automatic target recognition (ATR) is a very specific field of study within the general scope of image processing and image understanding. From a sequence of images (usually within the visual or infrared spectral bands), it is desired to recognize a target such as a ground-based tank. Sometimes there is additional image data such as a radar or laser range image which can be combined with the visual image to give three-dimensional contours. Automatic recognition of the target can be a very difficult task, particularly if there is structured background and a low signal-to-noise ratio. There are partial solutions, but work is still continuing to derive complete solutions to this difficult problem.

The United States Defense Advanced Research Projects Agency (DARPA) has conducted a study of neural networks and has selected ATR as one of the four application areas for evaluating neural network technology [1]. (The other three application areas are speech recognition, sonar signal processing, and seismic signal recognition.) A considerable body of work has been published recently with respect to neural network techniques and tools which have the potential for direct application to ATR. The purpose of this review paper is to give the reader an introduction to the overall problem of ATR and to enumerate current efforts in neural networks that might help in the solution of the problem.

The term "automatic target recognition" originated with the Low Altitude Navigation and Targeting Infra-Red for Night (LANTIRN) program in the early 1980's. One objective of the LANTIRN program was to develop a capability for an airborne forward-looking infrared (FLIR) system which would automatically detect and locate targets in an area on the ground known to contain targets. The LANTIRN ATR had to distinguish tanks from trucks, jeeps, and other less important targets. This multiclass discrimination requirement was and still is one of the most difficult requirements for ATR. Prior to the LANTIRN program, little had been done in the area which became known as ATR.

A number of ATR systems, based on previous methods, have been developed and tested on a very limited data set, and good classification performance has been reported (for reviews of ATR see [2] and [3]). However, in practice these efforts have been only partially successful and have produced high false-alarm rates. Some of the key reasons for this are the nonrepeatability of the target signature, competing clutter objects having the same shape as the actual targets, experience with a very limited data base, obscuration of targets, and limited use of *a priori* information. These are key technical challenges for the ATR problem.

ATR is a problem which involves extraction of critical information from complex and uncertain data for which the traditional approaches of signal processing, pattern recognition, and rule-based artificial intelligence (AI) techniques have been unable to provide adequate solutions. Target recognition of fixed signatures in stationary backgrounds is a straightforward task for which numerous effective techniques have been developed. If the target signatures and the backgrounds are variable in either a limited or known manner, more complex techniques such as those using rule-based AI (i.e., expert systems) methods can be effective. However, rule-based AI systems exhibit brittle rather than robust behavior (i.e., there is great sensitivity to the specific assumptions and environments). When the target signatures or the backgrounds vary in an unlimited or unknown manner, the traditional approaches have not been able to furnish appropriate solutions. A new approach is therefore required.

Neural network technology provides a number of tools which could form the basis for a potentially fruitful approach to the ATR problem. ATR needs methods to represent targets and backgrounds that are both sufficiently descriptive yet robust to signature and environmental variations. Neural networks offer potentially powerful collective-computation techniques for designing special-

Manuscript received June 13, 1989; revised November 21, 1989.
The author is with the Applied Physics Laboratory, Johns Hopkins University, Laurel, MD 20707.
IEEE Log Number 8933629.

1045-9227/90/0300-0028$01.00 © 1990 IEEE

purpose hardware which can implement fast optimization for a number of potential computational vision and multisensor fusion methods. ATR needs the capability for adaptation to additional targets and environments. The existence of powerful learning algorithms is one of the main strengths of the neural network approach. ATR needs to construct a compact set of maximally discriminating target features. There are a number of neural network inspired techniques which can be used for the selection or development of such a feature set. Finally, effective ATR performance could be enhanced if *a priori* knowledge about target signatures and background is used as much as possible. Whereas previous techniques for integrating diverse forms of knowledge were limited (e.g., for combining diverse feature detectors with no obvious metric) neural network technology provides expert-system capabilities for automatic integration. These ATR needs and neural network methods are discussed in considerably more detail in the following sections.

This paper presents a discussion of some of the trends relevant to application of neural network technology for ATR. No comprehensive review of the state of the art will be attempted. Rather, the emphasis will be selectively on certain current trends, as well as new ideas, that in the author's opinion, show promise for the future. However, the reader should note that neural network technology is in a state of flux with several alternative theoretical models and approaches. One approach that is different from that discussed in this paper is the work of Grossberg and his coworkers. Because a review of this approach has recently been published [4], the reader is referred to it for further discussion.

The next section presents a brief review of previous ATR systems developments and Section III highlights some key ATR issues and needs. Section IV describes how developments in computational vision combined with neural network technology could contribute to the derivation of effective target and background representations. Section V discusses neural network learning algorithms for ATR. Section VI describes a number of neural network techniques which could form useful tools for effective feature extraction. Section VII shows how neural network tools can contribute to a variety of higher vision tasks and describes a number of neural network based expert systems. Section VIII discusses developments in neural network hardware technology that have the potential for significant impact on the ATR problem. Finally, Section IX contains some comments and conclusions.

II. Previous ATR System Efforts

This section presents a brief review of previous ATR system developments. For more detailed discussions see [2] and [3]. The challenges for the early ATR LANTIRN effort were formidable, falling into three major categories: processing power limitations, sensor imaging deficiencies, and performance limitations of recognition algorithms. Digital processing hardware was not fast enough to keep up with the FLIR frame rate. The characteristics of first generation common-module FLIR sensors precluded determination of absolute target temperature. The result was image variability that compounded the problem of detecting, classifying, and discriminating targets from background clutter. The algorithm error rate and the ability to separate the various classes of targets (e.g., tanks from trucks, civilian vehicles, etc.) were less than satisfactory.

A. ATR Sensor Developments and Multisensor Fusion

Sensor problems have been substantially reduced by developments in such areas as focal-plane-array detector technology. However, as FLIR sensor technology improved, a number of other limitations were realized which affected ATR algorithm development. Some of these limitations were the high number of false alarms due to background clutter, vegetation, and animals. Other FLIR limitations included range uncertainty and occlusions from terrain and vegetation. For FLIR-based recognition, problems included target signature variability, ambiguity, and aspect-angle dependence. These limitations motivated examination of other potential sensors such as millimeter wave (MMW) radar, synthetic aperture radar (SAR), laser radar, and visible electrooptical (EO) sensors. Table I lists the various advantages and critical issues for sensor options.

The effect of topography and vegetation on sensor performance is to occlude the target and generate clutter. Clutter may be defined as natural or man-made objects whose image or signature may be mistaken for a target. Vegetation that may entirely or partially obscure a target from a FLIR may be penetrated by a SAR. Shrubs or animals that would result in false alarms with a FLIR would generally not result in a false alarm by a SAR. Variations in topography result in significant degradation in the ability to estimate target range with directional sensors such as FLIR and visible EO sensors. Structures and rocks are clutter for SAR but not generally for FLIR sensors.

The effects of climate, season, weather, and time of day on sensor performance and effectiveness are important. Algorithms must generally take into account differences in ground cover and associated clutter for different seasonal and climatic conditions. Weather conditions have, in general, greater effect on FLIR and visible sensors than on radio frequency (RF) sensors. The effects of clouds and rain on FLIR and lasers depend on how the sensor is used (e.g., below the clouds). Generally, RF sensor performance is the least affected by climatic, seasonal, and weather conditions.

The relative advantages and disadvantages for the exclusive use of either FLIR or SAR for detection are essentially marginal. Each has distinct advantages and the area of intersection of comparable performance is relatively large (i.e., day/night capability and effectiveness for a wide range of targets). SAR has a wider all-weather capability, can penetrate vegetation to some extent, and detect targets at ambient temperatures. FLIR's are less

TABLE I
PERFORMANCE TRADEOFFS FOR ATR SENSOR OPTIONS (ADAPTED FROM [3])

Sensor	Advantages	Critical Issues
FLIR	High target to background contrast Day & night operation Penetrates fog, haze, & dust	False alarms from background clutter, vegetation, & animals Range uncertainty Occlusions from terrain & vegetation Target signature variability Aspect angle dependence
MMW Radar	All weather Day & night operation	False alarms from background clutter, rocks, isolated buildings, & metal structures Terrain occlusions Target signature varies with aspect angle
SAR	All weather Day & night operation Large target to background contrast	False alarms from background clutter, rocks, isolated buildings, and metal structures Terrain occlusions Target signature varies with aspect angle
Laser Radar	Penetrates fog, haze, & dust Potentially high level of discrimination for range map signatures Vibration signature show promise Doppler laser radar for moving targets	Target signature varies with aspect angle Complex technology Power requirements Requires long dwell time on target Very precise tracking & stabilization required
Passive Electro-Optic	Light weight Inexpensive High Resolution Reliable	Relatively low target to background contrast No night or all-weather capability

susceptible to certain classes of clutter and are effective in detecting heat-generating targets.

Target recognition generally requires more information and greater resolution than that required for detection. FLIR and SAR signatures are variable, ambiguous, and highly dependent on aspect angle (although SAR less so than FLIR). A high-resolution differential range image of a target, as derived from a scanning-laser radar, appears to have potential for significantly increasing target recognition performance. Of all target signature classes, the range map, which is relatively insensitive to target state and time of day, is most easily modeled and synthesized. However, such high-resolution differential-range sensors also have narrow fields of view and therefore must be accurately pointed at potential targets using data derived from the detection computation and the launch platform navigation system.

A variety of problems and deficiencies with the various sensors available for ATR suggest that two or more sensors may significantly improve overall performance. The utilization of multiple sensors to acquire data for target detection and recognition is a major consideration in significantly improving ATR performance.

Multiple sensors could be utilized concurrently for detection and recognition. The data would be "fused" in this processing scheme to exploit the spectral and geometric differences and arrive at a more reliable decision. However, there are many issues for multisensor fusion such as the method for correlating nonsimultaneous data from multiple, independent sensors and the determination of correct classification of targets when there are conflicting reports.

An alternative approach is based on the observation that the sensors tend to fall into two categories. The first category consists of wide field-of-view medium resolution sensors (such as FLIR and microwave SAR) that are suitable for detecting the presence of targets but are susceptible to false detections. The second category contains narrow field-of-view relatively high resolution sensors that are more suitable for confirming detections and discriminating between different target classes. Thus, instead of "fusion," a complementary system may be configured whereby one sensor or set of sensors is used for detection and another for recognition.

B. Previous ATR Algorithm Developments

Although processing power and sensors have improved considerably, progress in the algorithm area has not kept pace with processors and sensors. There are a number of reasons for this. However, one reason is the previous deficiencies in processors and sensors. The limitations of previous sensor systems were described in the preceding paragraphs and the noise and error characteristics were frequently major problems for algorithms. In addition, the complexity of the algorithm was always limited by the processing technology at the time that the hardware design had to be frozen. Work of a very basic nature is needed in the algorithm area to bring it to the state of maturity commensurate with that of current sensors and processors. Many algorithm approaches have been proposed. The major categories of algorithms, functions, and methods previously applied to ATR are outlined in Table II. These approaches included classical statistical pattern recognition, syntactical pattern recognition, and rule-based AI approaches.

Statistical pattern recognition techniques typically involve the reduction of the image into a binary silhouette from which primitive shape features can be extracted. After segmenting the image into objects, a set of features is computed for each object. These shape features form the basis for object recognition. The reliability of these features is essential for target classification. Most of the features previously used by researchers were geometric, topological, or spectral. Various shape, gray-scale, and projection features were commonly used. Semantic features, such as those which are geographical, temporally contextual, and environmental, have been used to only a very limited extent. In real imagery, the global shape is frequently too severely perturbed to reliably generate these features. Often, it is lost in the silhouette formation process, which is highly vulnerable to noise and variations in target and scene conditions. In addition, statistical pattern recognition techniques do not provide a means for recovering from occlusion. Although these factors degraded statistical techniques, they are significant problems which would affect a number of algorithmic approaches.

In addition to the specific restrictions imposed by the classification techniques (e.g., linear and quadratic classifiers), the desirable properties of the features are invari-

TABLE II
PREVIOUS ALGORITHMIC FUNCTIONS AND METHODS FOR ATR (ADAPTED FROM [3])

Function/Method	Description	Advantages	Critical Issues
Statistical Pattern Recognition	Statistical procedures to compute discriminant function parameters	Satisfactory for patterns with well behaved distributions	Requires large representative training set; not robust
Discriminant Function	Mathematical function to determine class membership	Relatively easy to implement; wide applicability	Difficult to apply for exploitation of "knowledge"
Segmentation	Determination of segments of data that may contain target signatures	Systematic method for isolation of target-like signatures	Methods for consistent segmentation of wide range of data not solved
Model-Based Vision	Methods for detection/recognition based on physical properties of sensor, scene, target	Reduces dependence on large training sets	Practical modeling of spectrum of operational reality far from solved
Feature Extraction	Computation of numerical values of target signature from segmented data	Relatively easy to implement; applicable to well defined target types	Features are very dependent on aspect angle, target state, etc.
Detection	Detection of evidence of presence of target signature in data	Necessary first step in recognition process	Large number of false alarms required to achieve high probability of detection
Expert Systems			
Knowledge Base (KB)	Data base or "knowledge" about target, order of battle, scene, etc.	Method for including "knowledge" that is lacking in pattern recognition schemes	Problems of acquisition and representation of knowledge not solved yet
Production Rules	Linguistic rules for applying data in KB to target detection & recognition	Can be used to incorporate data & knowledge about target, background, etc.	Problem of formulating rules for non-linguistic domains such as target recognition not solved
Inference Engine	Computer program used to design procedures to draw inferences from data	Method for developing chains of inference rules	Applicability and effectiveness for targeting not yet established
Syntactic Pattern Recognition	Pattern recognition based on linguistic rules describing target structure	Rules can be formulated without recourse to processing large training sets	Very difficult to apply to "noisy" variable data

ance with respect to geometry (rotation, scale, and translation), computational efficiency, and extractability under adverse conditions. However, it should be noted that some features which are invariant with respect to some variations exhibit sensitivity with respect to other variations. For example, moments which are invariant with respect to size, position, and orientation are not contrast invariant and the contrast change of an image introduces a nonlinear scaling effect. Resolution is affected by scale, so that the invariant moments are no longer strictly invariant under rotation and scale changes.

The primary goal of feature selection is to obtain features which maximize the similarity of objects in the same class while maximizing the dissimilarity of objects in different classes. Feature selection in the context of the ATR problem has been performed by histogram examination, Bhattacharyya distance measure, Kolmogorov–Smirnov test, F-statistic, physical reasoning, linear regression techniques, etc. In feature selection the assumption of Gaussian distribution of the data has generally been made even though it may not have a multivariable Gaussian distribution. Features and classifiers have been optimized with respect to aspect angle. Classification has been mostly done using a K-nearest neighbor algorithm, projections, linear and quadratic classifiers, structural classifiers, tree-based classifiers, or clustering techniques. In a tree-classifier design it was required to partition the samples at each node into classes and select the subset of features which was most effective in separating the classes.

The limitations of the statistical approach to ATR systems led to the hope that better performance might be achieved by using rule-based AI or expert-system/knowledge-based techniques. These techniques offer suboptimal ways of handling context, rather than optimal ways of handling the local structure. The ATR problem was proposed to be suited for building an AI knowledge-based system for a specific operational environment and geographical area. Attempts have been made to use context (temporal, global, local, and ancillary information, such as map data, sensor data, seasonal, and intelligence information), semantics, and problem domain knowledge in an AI framework to improve the performance of ATR systems.

The AI knowledge-based approach basically has three parts. The first part consists of the development of low-level image analysis techniques and pattern recognition methods, as in the classical pattern recognition approach. The second part includes techniques for symbolic representation, strategies to be used for the integration of knowledge, search and control methods, and the implementation of a knowledge base so that appropriate knowledge is available at the right place in the search and decision making process. The third part combines the first two parts so that the system as a whole can be implemented. The reasoning in the expert-system/knowledge-based approach is essentially common sense rules such as: if the object detected is on a road, in motion and in a column, then it is a target. The antecedent condition (i.e., automatically finding roads, detecting motion, determining if objects are in a column) remains a difficult processing task.

While the payoff for successful exploitation of context could be significant, theoretical and practical implementation problems are far from solved. These problems include extracting contextual information from the sensor data, acquiring and exploiting external ancillary contextual data (such as terrain and map data), and utilizing inference methods for ATR under constraints of time and data processing power. Intrinsic deficiencies of applying rule-based AI or expert-system/knowlege-based systems to ATR include the following: 1) the difficulty in extracting the necessary data from the sensor to support the inference procedures; 2) the difficulty in acquiring, representing, and exploiting prior knowledge (such as terrain data); and 3) the present lack of an overall knowledge representation structure applicable to ATR.

III. ATR ISSUES AND NEEDS

There are several critical issues that are important for solving ATR algorithm problems such as the high false-alarm rate experienced by previous ATR systems. One issue is the complex and unpredictable characteristics of three-dimensional targets and scene clutter. One of the many things that makes ATR so hard is that the same target can vary wildly in appearance depending on lighting, aspect angle, atmospheric effects, and a host of other variables. Thus, the first priority in a system is to represent these possible variations in a form that is simple enough and compact enough for the system to handle.

Another issue is occlusion and obscuration. When multiple targets are present in the image, they may occlude or be near each other. Thus, separation of individual targets may be difficult. In addition, targets may be obscured by or be partially hidden in smoke, dust, and shadows.

Finally, it is important to use critical *a priori* knowledge. In many situations, a simple model of black and white intensity definition is not sufficient to extract the target if part of the target is brighter and part is darker than the background. This is a common occurrence in FLIR images. In this case context and shape information about the targets is important. In addition, it often happens that target boundaries are poorly defined and buried in the background. In such cases, the use of textural, structural, and contextual scene information may be useful. Size of the target is an important parameter in the ATR system design. As the range increases, the target occupies a reduced number of pixels in the image. If the number of pixels on the target becomes very small, the target may dissolve into the background. Thus, range information may be of crucial importance. Frame-to-frame analysis is also important because it may reduce the amount of computation. It may also increase classification accuracy by requiring repeated consistency of the classification decision.

Solving these issues will be extremely difficult. Therefore, it is necessary to address a number of fundamental needs before reasonable performance can be expected. Neural networks can provide a number of critical tools towards solution. Fig. 1 presents a diagram of some of the major ATR issues, needs, and neural network tools. Description of fundamental ATR needs and applicable neural network technology are as follows.

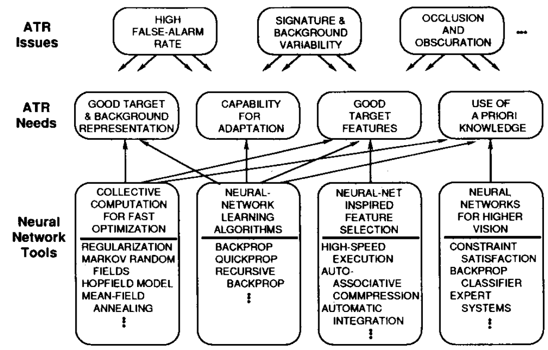

Fig. 1. Critical issues and needs for the ATR problem. Neural network technology can provide a variety of tools to meet the basic needs of the ATR problem. See text for discussion.

A. Good Representations of Target Signatures and Backgrounds

It is desirable to use image representations which are both sufficiently descriptive yet robust to target and environment variations. Several potential representation techniques have been proposed (e.g., regularization and Markov random fields for geometric-shape estimation) however such techniques involve optimization and subsequently potentially prohibitive computational requirements. The collective-computation capabilities of neural networks offer powerful techniques for designing special-purpose computational hardware which can implement fast optimization for these methods. This topic is discussed in more detail in Section IV.

B. Adaptation to Target or Environment Changes

Current target recognition systems are unable to modify their behavior based on the dynamic environmental changes which occur around them. In order to perform robustly in unconstrained environments, an ATR system must be able to adapt its representation of this dynamic environment while maintaining acceptable performance levels. Neural network learning algorithms offer some promising solutions to many of these problems faced in ATR. Learning allows a system to adapt its representation of the changing environment and to improve the system's recognition performance over time. The existence of powerful learning algorithms is one of the main strengths of the neural network approach. Although there exist many different learning algorithms, the number of practical successes for neural network learning algorithms has been impressive. In addition, there exist direct theoretical relationships between other successful learning algorithms (e.g., Kalman [5] and adaptive [6] filters) and neural network learning algorithms. This topic is discussed in Section V.

C. Good Features for Target Representation

A good feature must be sufficiently discriminating to be useful. However, in order to keep the classification problem tractable, the total number of features must be limited. Neural network technology offers techniques for selecting, developing, clustering, and compressing features into a useful set. This topic is discussed in more detail in Section VI.

D. Use of A Priori Knowledge About Target Signatures and Backgrounds

Probably the greatest difficulty with the use of diverse forms of *a priori* knowledge is integration. Whereas previous rule-based techniques for integrating diverse forms of knowledge were limited, neural network technology provides automatic knowledge acquisition and integration techniques for developing expert systems. This topic is discussed in more detail in Section VII.

IV. Early Vision and Collective Computation

ATR requires representations of targets and backgrounds which are sufficiently descriptive yet robust to signature and environmental variations. This is also a central problem in computational vision and is regarded as a significant part of early vision processing. A major theme in computational vision research is that a geometric-shape description could form the basis for such a representation

[7]. However, the extraction of such three-dimensional information from images or set of images is an ill-posed problem. Several techniques have been developed by researchers in computational vision to address such problems.

Regularization theory is one approach towards solving ill-posed problems [8]. A problem is ill-posed when either the solution does not exist, is not unique, or does not depend continuously on the initial data. Several problems in early vision are ill-posed such as stereo matching, determining structure from motion, computing optical flow, detecting edges, calculating shape from shading, and reconstructing surfaces. The main idea of regularization theory is to restrict the class of admissible solutions by introducing suitable *a priori* constraints on the possible solutions. Standard regularization theory imposes the constraints on a problem by a variational principle with a cost function.

The use of Markov random fields is another approach which can be briefly described as follows. More detailed information about the Markov-random-field approach can be found in [9] and [10]. As an example, consider the problem of approximating a surface f given sparse and noisy depth data g on a two-dimensional discrete grid. First define the prior probability of the class of surfaces of interest and make the Markov assumption that the probability of a certain depth at any given site on the grid depends only upon neighboring sites. Because of the Clifford–Hammersley theorem [11], the prior probability $P(f)$ is guaranteed to have the Gibbs form

$$P(f) = \frac{1}{Z_1} e^{-U(f)/T} \tag{1}$$

where Z_1 is a normalization constant, T is a constant called the temperature, and $U(f)$ is an energy function that can be computed as the sum of local contributions from each neighborhood. As a simple example, when the surfaces are expected to be smooth, the prior probability can be given in terms of

$$U(f) = \sum_i \sum_{j \in C_i} (f_i - f_j)^2 \tag{2}$$

where f_i is the value of f at site i and C_i is the set of all sites in the neighborhood of i.

If a model of the noise process is available, then one can write the conditional probability $P(g \mid f)$ of the sparse observation g for any given surface f. A simple example of such a model is

$$P(g \mid f) = \frac{1}{Z_2} \exp\left[-\frac{1}{T}\sum_i \alpha\gamma_i (g_i - f_i)^2\right] \tag{3}$$

where α is a constant, Z_2 is another normalization constant, and $\gamma_i = 1$ only where data are available. More complicated cases can be handled in a similar manner.

Bayes theorem then allows one to write the posterior distribution

$$P(f \mid g) = \frac{P(g \mid f)P(f)}{P(g)} = \frac{1}{Z_3} e^{-U(f \mid g)/T} \tag{4}$$

where in this simple example

$$U(f \mid g) = \sum_i \alpha\gamma_i (f_i - g_i)^2 + \sum_{j \in C_i} (f_i - f_j)^2 \tag{5}$$

Z_3 is the corresponding normalization constant and $P(g)$ is the prior probability for g. The maximum *a posteriori* (MAP) estimate of f, which is the surface corresponding to the maximum of $P(f \mid g)$, corresponds to finding the global minimum of $U(f \mid g)$. Note that the temperature parameter T, which controls the noise in the model, drops out of the minimization process and so does not need to be estimated. One of the main attractions of Markov random fields is that the prior probability distribution can be made to embed more sophisticated assumptions about the world. For example, Geman and Geman [9] introduced the idea of adding other processes, called line processes, representing explicitly the presence or absence of discontinuities that break the smoothness assumption (2). In addition, rule-based constraints corresponding to regular grammar languages can also be expressed in this fashion ([12] and [13]).

Markov random fields can also be used for integrating different image information. In real imagery, the segmented image is frequently too severely perturbed to generate reliable characteristics. Often, it is lost in the segmentation process which is vulnerable to noise and variations in aspect and environmental conditions. In order to cope with the uncertainties in natural scenes, highly redundant information is retained. The redundancy of the image data can be achieved by transforming the input image into a number of independent representations as shown in Fig. 2. While redundant representations are recognized as being important to successful computational vision, the algorithmic problem of combining several sometimes conflicting image representations to recognize an object is a technical challenge. Considerable interest has recently focussed on the use of MAP techniques and Markov random fields for dealing with this challenging problem ([14] and [15]). Because the integration problem for multisensor fusion in ATR is analogous to the computational vision problem of integrating different image information, integration techniques developed for computational vision could have significant impact on multisensor fusion techniques for ATR.

A central and common characteristic of such early vision approaches is the formulation of a cost or energy function which, when minimized, provides the desired solution. Because the function to be minimized is very complex, with large dimensionality and multiple local minima, sophisticated and computationally intensive minimization techniques, such as simulated annealing [16] were previously required. Simulated annealing operates as follows. An optimization problem involves a cost function $F(X) = F(x_1, x_2, \cdots, x_N)$ to be minimized and a set of candidate solutions generated by trial moves. Gradient descent methods accept only those moves which reduce $F(X)$ and therefore cannot escape from local minima. Simulated annealing accepts trial moves that decrease

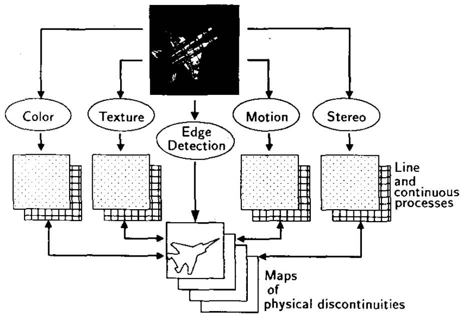

Fig. 2. Example of a current approach (the M.I.T. vision machine) to computational vision which consists of computing and integrating multiple representations (adapted from [14]). This problem is analogous to multisensor fusion for ATR and the same techniques of Markov random fields can be employed. Neural network collective computation can supply the rapid optimization tools required.

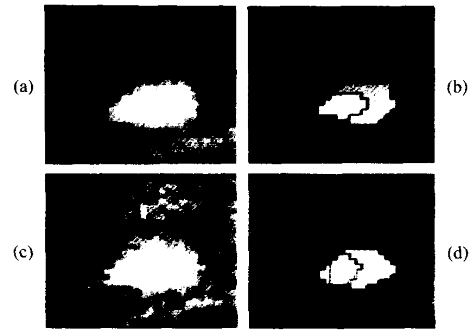

Fig. 3. Example of Markov-random-field approach for restoration and segmentation of IR data: (a) the original IR image including one vehicle with hot engine, (b) the restored and segmented image, (c) the original image corrupted by adding white noise, (d) restoration and segmentation of the corrupted image (adapted from [17]). The reason for the middle boundary is that the segmentation process has separated the hot engine from the rest of the vehicle and is an appropriate result. Although simulated annealing was used for optimization, neural network technology can achieve much more rapid optimization.

$F(X)$ but also allows some moves which increase $F(X)$. If a trial move increases $F(X)$ by $\Delta F(X)$, it is accepted with Boltzmann probability $\exp[-\Delta F(X)/T_a]$ where T_a is a control parameter referred to as the annealing temperature (which is different from the Markov temperature T in (1), (3), and (4)). A system evolving under these rules eventually reaches thermal equilibrium at any given annealing temperature. At high annealing temperatures the probability of accepting moves that increase $F(X)$ is greater and equilibrium is reached more quickly. At low annealing temperatures equilibration takes longer but the system is more heavily weighted towards low-cost states. The strategy is to do a coarse search of the space at high annealing temperature and then reduce T_a to focus on the low-cost states. The method is sensitive to the specific schedule for reducing T_a (known as the annealing schedule) which must be chosen with care.

Fig. 3 shows an example of using the Markov-random-field approach and simulated annealing for the restoration and segmentation of IR data [17]. Fig. 3(a) is the original IR picture containing one vehicle with the engine running. The intensity data represents thermal radiation contaminated by a number of blurring and noise effects. The restored and segmented picture, Fig. 3(b) was calculated using a simple degradation model. Fig. 3(c) was additionally corrupted by adding white noise with the resulting calculated image in Fig. 3(d).

The collective computation capabilities of neural network technology provides powerful new techniques for optimization which have the potential for solving such complex minimization problems much more rapidly. In 1982, Hopfield introduced a neural network model in which the processing element was a threshold logic unit, the weights were selected as the sum of the outer products of desired pattern values, all the units were connected to each other, and the units were updated in a random, asynchronous, and recursive manner [18]. The motivation for this model was to show that a network of simple nonlinear recursive units could achieve significant computational capabilities. Because of the symmetry of the weight matrix, Hopfield showed that this model could be interpreted in terms of an energy function such that stable states of the model corresponded to minima of the energy function.

In 1984, Hopfield introduced a significant extension [19] of the earlier Hopfield model [18] in order to overcome the computationally intensive procedure of asynchronous updating necessary in the original model for the system to converge to global rather than local minima. Unlike the earlier model, this Hopfield model was synchronous, continuous, and deterministic with a sigmoidal response function. Hopfield showed that there is also a corresponding energy function such that in the high-gain limit, the system is equivalent to the original Hopfield model. The sigmoidal nonlinearity is an important feature of the model because a sharp nonlinearity causes convergence to local rather than global minima. The later Hopfield model [19] inspired many efforts at analog VLSI implementations and computational experiments because of the advantages of the deterministic formulation over the stochastic one. Researchers have shown that analog VLSI implementations of this Hopfield model [19] have network time constants and convergence times of the order of a few microseconds [20]. This and other hardware developments are discussed further in Section VIII.

One of the first applications of neural network technology was for the solution of complex optimization problems. Because many different kinds of problems can be formulated as optimization problems, the potential for neural network technology to quickly solve such problems is a very important development. The first demonstration of this capability was by Hopfield and Tank [21] for the traveling salesman problem. The traveling salesman problem is an NP-complete problem which asks what is the shortest tour of several cities so that each city is visited only once. A problem is NP complete if the solution requires a number of computational steps that grow faster than any finite power of some appropriate measure of the problem. Hopfield and Tank showed that the later Hopfield [19] model can be used to compute good solutions to this problem within a few network time constants.

A number of researchers have used this optimization capability of the Hopfield model for specific early-vision optimization problems. Such work include implementations of shape from shading [22], optical flow [23]–[25], image restoration [26], and image segmentation ([27] and [28]). However, the most studied application is stereopsis. For example, the Marr–Poggio stereo algorithm [29] can be directly implemented in a Hopfield-like neural network ([30]–[32]). In addition, more complicated binocular-stereo ([30], [33], and [34]) as well as motion-stereo [35] algorithms can also be implemented in the framework of Hopfield-like neural networks.

The functional minimization characteristics of the later Hopfield model [19] have been enhanced by observations of its relationship with simulated annealing. Bilbro *et al.* [36] (see also [37] and [38]) have shown that if an approximation used in physics models, known as the mean-field approximation, is introduced, then the relaxation process of simulated annealing at a fixed annealing temperature is equivalent to the later Hopfield model [19]. Bilbro *et al.* extend this result further by showing that if the annealing temperature is gradually reduced according to the annealing schedule, that the resulting method is orders of magnitude faster than simulated annealing with comparable solution performance. This technique, which the authors refer to as mean-field annealing, has been successfully applied to MAP restoration of noisy images and segmentation ([39] and [40]). Consequently, if the good solutions found by the Hopfield model alone are not adequately close to the absolute minimum, then the mean-field-annealing technique effectively provides an enhancement to the Hopfield model which can find solutions comparable to simulated annealing. Thus, a hardware implementation of the later Hopfield model [19], which may be combined with an annealing schedule may provide the speed potential for solving ill-posed vision problems in real time.

V. Learning and Adaptation for ATR

Previous ATR systems lacked the ability to adapt to changing target and environmental conditions. In order to be effective in dynamic scenarios, a robust ATR system should be able to automatically acquire necessary information from the environment. The performance of previous ATR systems began to quickly degrade when subjected to the problems of variable lighting conditions, image noise, and object occlusion.

Neural network learning could facilitate two main advances for ATR: automatic knowledge acquisition and continuous system refinement. The use of learning in system construction would save the user from spending the enormous amount of time necessary to derive rule-based data bases for targets and environments. System refinement could then be incorporated to make any necessary changes to improve the performance of the recognition system. These two modifications alone could serve to significantly advance the present abilities of ATR systems.

The back-propagation learning algorithm for layered, feedforward networks [41] represents a major advance in effective neural network learning. Back propagation is a learning algorithm which is designed to solve the problem of choosing weight values for hidden units in a layered feedforward network. Hidden units are units in intermediate layers which are neither input nor output units. Hidden units are introduced in order to give the network enhanced internal processing capabilities it could not have if only input and output units were present. The limitations of networks with only feedforward input and output units, such as the single-layer perception [42], were well documented by the work of Minsky and Papert [43]. The introduction of one hidden layer gives the network the capability to represent an arbitrary Boolean function and two layers allows the network to represent an arbitrary decision space. In addition, hidden unit layers can be introduced to automatically represent geometrical invariances such as translation invariance (for additional references, see [44]).

Although the introduction of hidden units gives a feedforward network the potential for an arbitrary mapping, before the introduction of the back-propagation algorithm, no known technique existed for determining the weights of a deterministic, feedforward network with hidden layers. This was the essence of the credit-assignment problem: how to determine the weights of intermediate hidden units so that desired responses occurred as a result of specified inputs. Several algorithms were known for single-layer feedforward networks without hidden units, such as the perceptron learning algorithm [42] and the Widrow–Hoff algorithm [45].

The back-propagation algorithm has demonstrated several advantages in addition to having the potential for determining networks with arbitrary mapping properties. For example, the backpropagation algorithm has been shown to be directly related to the extended Kalman filter [5]. In addition, the back-propagation algorithm can also be used to learn distributed representations [46]. A distributed representation is one whereby a knowledge unit is distributed over several units as opposed to being represented by a single unit. Using back-propagation, weights can automatically evolve to show relationships which were not explicitly input. Whereas neural networks can learn topological mappings using lateral inhibition and modified Hebbian-type learning [47], the back-propagation algorithm can also be used to discover topological mappings [48]. Indeed, it has been proposed that back-propagation can be modified to minimize the log-likelihood function rather than the error and thereby learn probability distributions [49]. Although the gradient-descent nature of the back-propagation algorithm leads to slow convergence, a number of modifications to the back-propagation algorithm have been proposed in order to speed up convergence (e.g., quick-prop [50]). Finally, the number of training examples required for a given level of performance can be estimated. Baum [51] has derived bounds on the number of training examples required for a given error rate for a feedforward network assuming that training and testing patterns are chosen from the same arbitrary distribution. The rule is that the number of examples needed is approximately the number of weights divided by the error rate.

Pineda [52] has derived a recursive version of the back-propagation algorithm by first noting that feedforward processing elements are a special case of the Hopfield [19] dynamical equations and then deriving a learning algorithm for a network with arbitrary feedforward and feedback connections for which back-propagation is the special case limited to feedforward networks only. The learning equations can also be interpreted as another network. Although the early version of this algorithm cannot serve as an associative memory because all the initial states are in a single basin of attraction, the problem can be circumvented by deriving a related algorithm whereby the external units are constrained during learning to be the desired patterns [53].

One example of the utility of neural network learning algorithms is for the important problem of estimating the parameters of the Markov-random-field models described in Section IV. The values of these parameters determine both the distribution over the configuration space to which the system converges and the speed of convergence. Thus, rigorous methods for estimating these parameters are essential for the practical success of the method and for meaningful results. One way to estimate parameter values is from the set of examples in which data and desired solution are given. Neural network learning algorithms have the potential for providing powerful techniques for learning the parameters from examples. This technique of neural network learning for the parameters of the energy function has already been demonstrated for the case of discovering the Marr–Poggio stereo algorithm (which can be formulated as a Markov-random-field model [10]) parameters using Pineda's algorithm and examples of random-dot stereograms [54]. Other examples of the utility of neural network learning algorithms are discussed in the following two sections.

VI. Feature Extraction

Selection of appropriate target features is one of the most important tasks for ATR algorithm development. Because it is impractical to match a given input image or image representation with all the image templates of all possible targets and their variations, it is necessary to find a compact set of features which can represent the critical aspects of a target. It is also important that the feature set be sufficiently complete so that targets can be appropriately discriminated from nontargets. In this way, the selection of the feature set is linked to the classification task. Clearly, features which are invariant with respect to target and environmental variations (e.g., translation, rotation, scale, context, etc.) are of more interest than noninvariant features. Neural network technology can contribute to the feature-selection task in three major ways.

The first way that neural network technology can contribute to the features selection process is by providing hardware for massively parallel implementations of traditional feature detection algorithms. Present and future neurocomputers have applications beyond neural network models and algorithms because the hardware architecture consists of massive parallelism of simple processing units. Because any finite-state computing machine can be computed with a set of Boolean functions and because neural networks can be configured to compute an arbitrary Boolean function, any finite-state computing machine can be configured with neural networks. Whereas state machines are usually complemented with finite memory, neural networks use the values of connection weights as its form of memory. However, some algorithms can be more efficiently implemented on conventional rather than neural network hardware. Likewise, many well-known image processing and feature selection algorithms can be efficiently implemented using neural network algorithms and hardware. For example, it has been shown that neural networks can efficiently compute convolution integrals, Gaussian-filtered images, edge-enhanced images, and the Hough transform [55]. Whereas special-purpose parallel hardware could always, in principle, be constructed for a specific problem, the hardware would, in general, be limited to only that particular problem. One of the things that a neurocomputer offers is the possibility of not only solving one problem in parallel, but a broader variety of problems.

Neural networks can implement optimum feature receivers for extraction of weak features from high-clutter environments [56]. In general, detection devices must set high thresholds in order to achieve a reasonable false-alarm rate. This is especially true for environments, such as radar at low elevation angles, where the clutter distributions have long tails. The problem with setting a high threshold to cut down on false alarms is that detection of small and medium-size features can be missed. A previously impractical idea for dealing with this problem was to implement a large bank of matched filters to cover the feature variations. Because neural network hardware is precisely designed to implement massively parallel computations, it offers new opportunities to implement such previously impractical ideas. In particular, a modified Hopfield [19] model can implement an optimum post-detection target feature receiver. In this application, the desired states represent single features and spurious states would correspond to multiple features. Since it is desirable to detect multiple features, the spurious states for this application are not a drawback but rather a desirable characteristic. Simulations have been conducted and show that over 12-dB signal-to-noise gain can be achieved.

Another example is provided by Daugman [57] who showed how a Hopfield-like network can be used to compress an image into a nonorthogonal set of Gabor filters. Minimizing the mean square error between the original and reproduced images can be implemented by a parallel relaxation network that is related to the Hopfield [19] model. In addition to being able to achieve compression factors of 3 to 8 with excellent reproduction, the Gabor-filter set can also be used for edge detection of textured images. Two-dimensional Gabor filters are of special interest because they have been recently shown to provide an excellent fit to the visual receptive fields found in mammalian brains [58].

However, it was previously believed that neural networks could implement only limited kinds of feature detectors. Although feedforward neural networks can perform a number of computationally significant operations, there also is a significant number of computational operations that cannot be performed by a feedforward network with a finite number of layers, as pointed out by Minsky and Papert [43]. An example of such an operation is determining whether a given figure is connected. However, such a restriction does not apply to recursive or feedback networks such as the Hopfield model. In particular, Roth [44] showed that a three-layer network with two of the layers recursive can compute the connectedness of a figure. As shown in Fig. 4, the first layer is a lateral inhibition (or winner-take-all) network which merely picks out a single point in the figure (noise can be added to insure that only a single point is selected). The result is directly transferred by feedforward connections to a Hopfield network whose weights are the outer product of the binary input pattern restricted to nearest neighbor weights only. This second layer acts like a cellular automaton in that if a single point of the input figure is activated, then all points which are connected to that point will also be activated. Finally, the result of the Hopfield network layer is fed forward to a single threshold logic unit whose weights are the input pattern but with a threshold sufficiently high so that a perfect match is required. It should be noted that the weights of the second and third layers are determined by only the current input pattern. In this manner, a network with a combination of feedforward and recursive layers can compute the connectedness of a figure and overcome the computational limitations of feedforward networks alone. A related network for determining figure from ground has been described by Kienker *et al.* [59].

The second major way in which neural network technology can contribute to the feature selection task is through automatic discovery of clustered features by using neural network learning algorithms. For example, Kohonen feature maps [47] are neural networks which consist of a set of interconnected adaptive units that have the ability to change their responses in such a way that they will adapt to represent the characteristics of the input signal. This model can be used for vector quantization of images [60] and gives comparable performance to the Linde-Buzo-Gray [61] vector-quantization algorithm.

Cottrell *et al.* [62] have introduced a potentially powerful technique for feature discovery and image compression. The technique involves an autoassociative multi-

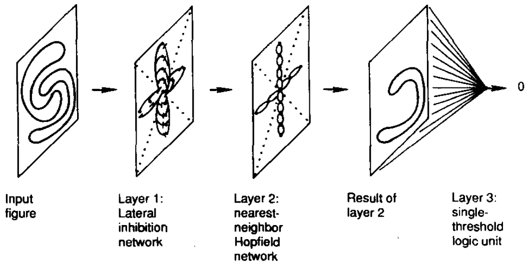

Fig. 4. A layered neural network architecture for computing the connectedness of a given figure using both feedforward and feedback connections (adapted from [44]). This example shows that recursive neural networks can be used for more complex ATR feature extraction than that provided by feedforward networks alone.

layer feedforward neural network that is trained with the back-propagation algorithm and reduces the dimensionality of continuous-valued inputs. The network has the same number of input and output nodes and one or more layers of hidden nodes. This net is trained to reproduce the input at the output nodes through a small layer of hidden nodes. Output of hidden nodes after training can be used as reduced dimensional inputs for further processing. This technique has been applied to aircraft IR data (after Fourier polar transform to take out translations and rotations) and found that the middle layer did appear to separate the data into clustered classes [63].

This autoassociative technique is closely related to a standard statistical technique called principal component analysis [64]. In particular, the linear version of the autoassociative compression technique corresponds to a projection onto the subspace spanned by the first principal eigenvectors of a covariance matrix associated with the training patterns. The error function has, up to an equivalence, a unique local and global minimum. All other critical points of the error function are saddle points. This gives a precise meaning to generalization in that the distortion on a new pattern is exactly given by its distance to the subspace generated by the first p eigenvectors of the covariance matrix where p is the number of hidden units. However, the neural network compression differs in that near equal variance of the components results [65]. In addition, the nonlinearity of the neural network approach can solve encoding problems where the principal-component-analysis solution is degenerate.

The third major way in which neural networks can contribute to feature selection is in its capabilities to automatically integrate a diverse set of features. *A priori* knowledge may suggest that a specific set of diverse features (i.e., size, Fourier-polar coefficients, presence of hot spots, etc.) is important for the classification problem. However, there is a significant problem of combining such diverse features into an effective classifier, especially when there is no obvious metric available. The back-propagation algorithm, combined with an appropriate training set, can be used as an effective way to automatically integrate such diverse features into a classification system. Analysis of the resulting neural network weights can then determine whether a particular feature has effectively participated in the classification task, or whether it can be removed without loss of overall performance. See [66] for an example of this approach. The metric that back propagation learns is the one that minimizes the mean square error. Back propagation is also capable of learning metrics with respect to other criteria (e.g., log-likelihood [49]) as well.

VII. Higher Vision and Expert Systems

Higher vision processing in ATR systems must perform the final object recognition using information from early vision processing and feature extraction. Neural networks can contribute to the higher vision task in two major ways: 1) extensions of the Hopfield model for knowledge integration which requires optimization and 2) neural network learning for development of classifiers and expert systems.

There are a number of higher vision tasks which can benefit from the enhanced optimization capabilities of neural network technology. For example, when shape data is represented as discrete critical points (endpoints of common line segments for a hierarchy of successive convex subshape reductions of the original shape), the Hopfield model can be used for pattern recognition of two-dimensional shapes as well as various views of three-dimensional shapes [67]. Another application is to the problem of forming stable perceptual groups from primitive image tokens [68]. Additional examples are constraint satisfaction of structure relationships for collated image features [69], selecting a region label with location and adjacency constraints [70], and hypothesis optimization for finding the simplest and most likely description of a segmented object's structure [71].

The back-propagation algorithm represents a significant development in methods for representing and computing the parameters of discriminant functions for classification. This provides a formal technique for determining decision surfaces between multiclass training samples with irregular and even multimodal distributions. Although in many cases the error rate of a back-propagation classifier is close to that of a Bayesian classifier, back-propagation classifiers are also robust and can outperform conventional techniques by a wide margin when inputs have disjoint probability distributions [72].

As an example, Glover [73] decribes a remarkable application of back-propagation to computational vision for industrial automated inspection. The image is transformed into 32 elements of the ring-wedge Fourier-polar transform using optical computing techniques. These feature vectors are used to train examples of items with and without defects. Results with back-propagation are superior to those using the Fisher linear discriminant.

An example of the application of a back-propagation classifier to the ATR problem was described by Troxel *et al* [74]. The technique was applied to laser radar data of tanks and trucks using the Fourier-polar transform for position, scale, and rotation invariance. The input representation is selected to be shape values around the peak of a correlation with Fourier-polar templates. Excellent classification accuracy is reported.

Neural network technology can also provide tools for developing expert systems. Neural networks can implement propositions and constraints with the capability to backtrack for explanations [75]. They also have features which support both robust reasoning and knowledge acquisition. Inferencing techniques can be developed to allow the expert system to reach conclusions when only a fraction of the input values are known [76]. Also techniques for confidence estimation, question generation, and conclusion explanation can be developed.

However, the major advantage to the neural network approach to developing expert systems is the significant reduction in knowledge-engineering development time over rule-based approaches when a suitable training data base is available. Neural network learning algorithms, such as the back-propagation algorithm, can be applied to the automatic development of expert-system rules, and thereby reduce the knowledge acquistion bottleneck for the rule-based approach. Several neural-network-based expert systems (e.g., [77] for medical diagnosis, and [78] for solar-flare forecasting) have been directly compared to rule-based expert systems developed for the same problem. In each case, the neural network-based system provided comparable or superior performance but at an enormous reduction in system development and execution time.

It should be noted that even with the automated-learning capabilities of neural networks, a considerable body of knowledge engineering is still required for practical and high-performance neural-network-based expert systems. For example, Denker *et al.* [79] describe their design of a neural network system for recognition of handwritten digits on postal mail. Considerable preprocessing (scaling, deskewing, skeletonization, and feature extraction) is required before classification. The high-performance preprocessor plus a large training base gave the back-propagation-trained network better performance over alternative classical-method classifiers.

VIII. Neurocomputer Hardware

Neural network models and algorithms are computationally intensive on general-purpose computers. However, because of the computational simplicity of the basic processing element, neural networks can be implemented on special-purpose massively parallel hardware which can vastly outperform implementations on even the most powerful serial computers. Consequently, a number of groups are developing such special-purpose neurocomputer hardware for implementing neural network applications. The recent appearance of these neurocomputers has been an essential ingredient for the development of practical applications of neural network technology. The ATR problem also requires that computational hardware be configurable into a reasonably compact volume. There are a number of neural-network-hardware research efforts which have the promise of providing both high performance and compact design.

The first generation of neurocomputers were based on pipelined implementations of digital VLSI technology with some low-level parallelism (for a review, see [80]). It should be noted that this first generation of neurocomputers consist of simulators of neural network models and algorithms and does not exploit the powerful computational potential of direct implementations using device physics.

A number of groups have recognized the computational potential of direct implementations of neural networks using the device physics of analog VLSI. The main advantage of going to analog VLSI is that an enormous processing advantage is gained by using analog circuitry to perform the neural network computations. For example, one analog VLSI chip implementation of the Hopfield [19] model has 256 processing elements and 130 000 fixed resistive weights and can converge in less than 1.4 μs [20]. An excellent and detailed introduction to this approach can be found in [81] (see also [82]).

Analog VLSI chips have been developed for application to early vision processing. Mead and Mahowald [83] describe an analog VLSI implementation of a 48 by 48 array simulating image processing in the retina. The chip has logarithmic photoreceptors, a horizontal network of resistive elements, and an implementation of lateral inhibition. Similar chips for computing optical flow [84] and tracking the center of intensity [85] have also been developed. In addition, chips have been designed and tested for implementing the line processes [86] proposed by Geman and Geman [9] and also for implementing stereo algorithms [87].

Although the analog VLSI approach presents a number of technical challenges, a number of techniques are being introduced to deal with them. A major challenge is dealing with the results of variations in chip lithography, dopant density, and other process parameters. Although large output current response differences have been observed on account of these effects, the variations can be reduced with careful circuit layout. In the orientation-detector-array chip described by Allen *et al.* [88], the remaining noise was a fixed pattern which could be eliminated by subtracting an image of the offsets. In addition, Mead [89] has developed an adaptive circuit based on a floating-gate technique which can overcome this offset problem. For other chip designs, especially for ones with more feedback connections, circuit parasitics and delays can result in undesirable spontaneous oscillations and instability. Standley and Wyatt [90] (see also [91]) propose design rules which can guarantee stability in spite of a lack of precision in component parameters.

Another major challenge is overcoming the limitation in the number of connections that can be achieved because of the two-dimensional nature of the chips. One proposed solution is to go to wafer-scale technology [92] which may be viable if the network weights can be reprogrammed (especially around imperfections) by some technique such as laser modification of resistive weights (see also [93]).

For many applications it is desirable to have the capability for variable weights on-board the chip. Early designs for variable weights took up so much room that only a small number of variable weights could be put onto a chip [94]. With a later approach using pairs of capacitors and access and charge transistors, over 1000 variable weights with 10 bits of dynamic range can be achieved [95]. Using floating-gate techniques, a hybrid analog-digital chip with 10 240 variable weights has been fabricated [96].

Tsividis [97] (see also [98]) proposes that analog MOS technology previously used for analog-to-digital converters, pulse-code modulators, and filters could have utility for VLSI neural network chips. Analog MOS technology has developed self-correcting, self-compensating, and self-calibrating techniques to eliminate the errors traditionally associated with analog circuits. These techniques could be especially important for sensors with integrated on-board processing. For example, a design using periodic switched-capacitor networks has no unwanted continuous-time oscillations [99]. Another example suggests that the variability of transistor conductance with bias can also implement variable weights [100]. The bias is held using capacitors for temporary storage. A control voltage generator can be time-shared among several weights through switches and addressing schemes similar to semiconductor memories. This technique would use much less chip real estate than alternatives

Optical implementations of neural networks have the potential for achieving very high connectivity because beams of light can pass through one another without interaction (for reviews, see [101] and [102]). There may be a significant interplay between optical computer technology and neural network algorithms because of their complementary strengths and limitations. Future applications of neural networks will require massive parallelism which is a strength of optical computers. However, optical computers have limited dynamic range which may not be a problem for neural networks. Finally, there are possibilities for all-optical computation loops which would avoid the interface bottleneck between electronic and optical components. Optical neurocomputers are another good example of the computational potential for neural networks models and algorithms to exploit device physics.

The first optical neurocomputer designs employed optical matrix-vector multipliers using light-emitting diodes, photodiodes, light masks, and electronic feedback ([103] and [104]). However, this matrix-vector-multiplier approach has evolved considerably to the point where designs have now been proposed with optoelectronic integration on VLSI chips. Photodiodes integrated into VLSI chips could provide optically controlled weights and the light intensity could be varied by masks or acoustooptical crystals [105]. Using a photoconductive array, over 10 000 weights can be reprogrammed over three decades of dynamic range by incident light patterns [106]. Another chip design proposes using magnetooptical spatial light modulators [107]. A review of the various techniques for implementing compact designs using the cross-strip matrix-vector multiplier is presented in [108].

Another optical approach to implementing neural networks is the use of holographic devices such as volume holograms. Combined with optical feedback, systems could have the capability to retrieve associatively potentially millions of image patterns in a very short time [101]. The use of photorefractive crystals represents a promising approach to volume holograms because of the ease of dynamic holographic modification of interconnections [109]. The density of interconnections which may be implemented in these crystals is of the order of 10^8 to 10^{10} weights per cm^3. The Hopfield model can be precisely implemented in an all-optical design using computer-generated holograms [110] and a design for an all-optical implementation of back-propagation has been presented [111]. Holograms used in conjunction with optoelectronic resonator cavities have also been considered [112]. A design using page-orientated holographic memory has the potential for 10^9 interconnections [113]. Finally, experimental results for an all-optical associative memory have been reported [114].

Although the volume-hologram approach has the potential for large number of interconnections, achieving physically compact design is a technical challenge. Stoll and Lee [115] describe a system used to store and recall high resolution imagery using a coherent ring laser, volume holograms, and bipolar optics. This system is also capable of feature extraction and optimization. Using microoptics, the system can be reduced to approximately the size of an audiocassette.

IX. Comments and Conclusions

Previous ATR approaches were ultimately constrained by the limits of computer technology at the time and by the compact volume available for onboard ATR processing. This limited the kind of preprocessing, feature extraction, and pattern recognition that could be accomplished. Thus potentially more effective methods which were too computationally intensive could not be implemented. For example, the available processing power limited expert-system approaches by limiting the number of rules that could be handled despite the complexity of the ATR problem.

Although the technology for on-board processing power has improved significantly in recent years, greater computational resources may be required. Researchers in computer vision have realized that the computational resources required may be beyond the level of supercomputers. Consequently, they have focused on the development and use of massively parallel machines such as the Connection Machine. However, there is a great degree of difficulty in programming such massively parallel machines. Furthermore, it is unclear whether such designs could be made sufficiently compact for ATR application.

What neural network technology brings to the ATR problem is two new ingredients that open up the range of possibilities for solutions: 1) the potential for compact massively parallel computational hardware and 2) the potential for improved programmability by using powerful models and learning algorithms. Because the basic processing elements of neural networks are relatively simple, basic device physics can be used to implement them in hardware. Analog VLSI is one neurocomputer approach that uses its analog nature to achieve high-speed local computation. If only local connectivity is required (e.g., Markov random fields) then the hardware can be implemented at chip scale. For more global connectivity, research is in progress for wafer-scale implementation. Optical neurocomputers also have the potential for massive parallelism because intersecting beams of light do not interact. Through the use of techniques such as holographic memories and microoptics, compact designs are also possible.

The connection weights of a neurocomputer can be efficiently programmed by using the powerful computational characteristics of neural network models and learning algorithms. In addition, these models and algorithms can be used directly as important tools for the ATR problem. In particular, the collective computational characteristics of neural network models like the Hopfield model [19] and related optimization techniques like mean-field annealing [36] offer the potential for achieving very fast optimization. Such methods can be used to implement realizations of optical flow, stereopsis, and image restoration and segmentation which would otherwise be very computationally intensive on serial computers or even on conventional massively parallel machines. These computational-vision techniques have the potential for producing improved representations of ATR target signatures and backgrounds.

Finally, powerful learning algorithms are one of the main strengths of the neural network approach. If it is known that a particular neural network can represent a desired function and if an appropriate data base exists of input and output examples, research is showing that neural network learning can be used to calculate the connection weights so that the network can approximate the function. Such learning can be used for automatic ATR knowledge acquisition and system refinement. It has already been shown in some cases that this can lead to comparable performance but at a significant reduction in the development and execution time as compared to rule-based expert systems. Ultimately, neural network learning could be used for addressing the ATR needs for adaptation to target and environment changes, selection of good target features, and integration of *a priori* knowledge about target signatures and backgrounds.

Acknowledgment

The author wishes to thank B. G. Boone and R. L. Kulp for their comments and discussions on this paper. The author would also like to thank R. C. Mann for helpful discussions.

REFERENCES

[1] *DARPA Neural Network Study*. Fairfax, VA: AFCEA International Press, 1988.

[2] B. Bhanu, "Automatic target recognition: State of the art survey," *IEEE Trans. Aerospace Electron. Syst.*, vol. AES-22, no. 4, pp. 364-379, July 1986.

[3] M. Gyer *et al.*, "Automatic target recognition panel report," SAIC, San Diego, CA, rep. ET-TAR-002, Dec. 1988.

[4] S. Grossberg, "Nonlinear neural networks: Principles, mechanisms, and architectures," *Neural Networks*, vol. 1, pp. 16-61, 1988.

[5] D. W. Ruck, S. K. Rogers, P. S. Maybeck, and M. Kabrisky, "Back propagation: A degenerate Kalman filter?," Wright-Patterson AFB, OH, Air Force Institute of Technology preprint, 1989.

[6] A. Lapedes and R. Farber, "Nonlinear signal processing using neural networks: Prediction and system modeling," Los Alamos Nat. Lab., Los Alamos, NM, preprint LA-UR-87-2662, July 1987.

[7] D. Marr, *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. San Francisco, CA: W. H. Freeman, 1982.

[8] T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature*, vol. 317, pp. 314-319, Sept. 1985.

[9] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Patt. Anal. Mach. Intel.*, vol. PAMI-6, no. 6, pp. 721-741, Nov. 1984.

[10] J. Marroquin, S. Mitter, and T. Poggio, "Probabilistic solution of ill-posed problems in computational vision," *J. Amer. Stat. Ass.*, vol. 82, no. 397, pp. 76-89, Mar. 1987.

[11] J. Besag, "Spatial interaction and the statistical analysis of lattice systems (with discussion)," *J. Royal Stat. Soc.*, ser. B, vol. 36, pp. 192-326, 1974.

[12] B. Roysam and M. I. Miller, "Mapping deterministic rules to stochastic representations via Gibbs' distributions on massively parallel analog networks: Application to global optimization," in *Proc. 1988 Connectionist Models Summer School*, D. Touretzky, G. Hinton, and T. Sejnowski, Eds. San Mateo, CA: Morgan Kaufmann, 1989, pp. 229-238.

[13] M. I. Miller, B. Roysam, and K. R. Smith, "Mapping rule-based and stochastic constraints to connection architectures: Implication for hierarchical image processing," *Proc. SPIE Int. Soc. Opt. Eng.*, 1988, pp. 1078-1085.

[14] T. Poggio *et al.*, "The M.I.T. vision machine," in *Proc. Image Understanding Workshop 1988*. San Mateo, CA: Morgan Kaufmann, Apr. 1988, pp. 177-198.

[15] G. L. Bilbro, H. Hiriyannaiah, and W. E. Snyder, "Fusion of range and reflectance image data using Markov random fields," in *Proc. 1988 IEEE Int. Symp. Intelligent Control*. Piscataway, NJ: IEEE, 1988.

[16] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671-680, 1983.

[17] D. Geman and S. Geman, "Bayesian image analysis," in *Disordered Systems and Biological Organization*, E. Bienenstock, F. Fogelman, and G. Weisbuch, Eds. Heidelberg, W. Germany: Springer-Verlag, 1986, pp. 301-319.

[18] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proc. Nat. Acad. Sci.*, vol. 79, pp. 2554-2558, 1982.

[19] J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two-state neurons," *Proc. Nat. Acad. Sci.*, vol. 81, pp. 3088-3092, 1984.

[20] H. P. Graf *et al.*, "VLSI implementation of a neural network memory with several hundreds of neurons," in *Neural Networks for Computing, Snowbird, Utah, 1986, AIP Conf. Proc. 151*, J. S. Denker, Ed. New York: American Institute of Physics, 1986, pp. 182-187.

[21] J. J. Hopfield and D. W. Tank, "'Neural' computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141-152, 1985.

[22] C. Koch, J. Marroquin, and A. Yuille, "Analog neuronal networks in eary vision," *Proc. Nat. Acad. Sci.*, vol. 83, pp. 4263-4267, June 1986.

[23] C. Koch, "Computing motion in the presence of discontinuities—Algorithm and analog networks," in *Neural Computers*, R. Eckmiller and C. von der Malsburg, Eds. New York: Springer-Verlag, 1988, pp. 101-110.

[24] Y. Zhou, "Artificial neural network algorithms for some computer vision problem," Ph.D. dissertation, Univ. of Southern California, Los Angeles, CA, 1988.

[25] H. Bülthoff, J. Little, and T. Poggio, "A parallel algorithm for real-time computation of optical flow," *Nature*, vol. 337, pp. 549-553, Feb. 9, 1989.

[26] Y. T. Zhou, R. Chellappa, A. Vaid, and B. K. Jenkins, "Image restoration using a neural network," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 36, no. 7, pp. 1141-1151, July 7, 1988.

[27] C. Cortes and J. A. Hertz, "A network system for image segmentation," in *IJCNN Int. Joint Conf. Neural Networks*, vol. I. Piscataway, NJ: IEEE, 1989, pp. 121-125.

[28] M. J. Daily, "Color image segmentation using Markov random fields," in *Proc. Image Understanding Workshop 1989*. San Mateo, CA: Morgan Kaufmann, 1989, pp. 552-562.

[29] D. Marr and T. Poggio, "Cooperative computations of stereo disparity," *Science*, vol. 194, pp. 283-287, 1976.

[30] E. Goles and G. Y. Vichniac, "Lyapunov functions for parallel neural networks," in *Neural Networks for Computing, AIP Conf. Proc. 151*, J. S. Denker, Ed. New York: American Institute of Physics, 1986, pp. 165-181.

[31] A. F. Gmitro and G. R. Gindi, "Optical neurocomputer for implementation of the Marr-Poggio stereo algorithm," in *IEEE First Int. Conf. Neural Networks*, vol. III, M. Caudill and C. Butler, Ed. Picataway, NJ: IEEE, 1987, pp. 599-606.

[32] G. Z. Sun, H. H. Chen, and Y. C. Lee, "Learning stereopsis with neural networks," in *IEEE First Int. Conf. Neural Networks*, vol. IV, M. Caudill and C. Butler, Ed. Piscataway, NJ: IEEE, 1987, pp. 345-355.

[33] Y. T. Zhou and R. Chellappa, "Stereo matching using a neural network," in *1988 Int. Conf. Acoust., Speech, Signal Processing*. Piscataway, NJ: IEEE, 1988, pp. 940-943.

[34] S. P. Clifford and N. M. Nasrabadi, "Integration of stereo vision and optical flow using Markov random fields," in *IEEE Int. Conf. Neural Networks*, vol. I. Piscataway, NJ: IEEE, 1988, pp. 577-584.

[35] Y. T. Zhou and R. Chellappa, "Neural network algorithms for motion stereo," in *IJCNN Int. Joint Conf. Neural Networks*, vol. II. Piscataway, NJ: IEEE, 1989, pp. 251-258.

[36] G. Bilbro, R. Mann, T. K. Miller, W. E. Snyder, D. E. Van den Bout, and M. White, "Optimization by mean field annealing," in *Advances in Neural Information Processing Systems 1*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 91-98.

[37] G. C. Fox and W. Furmanski, "Load balancing by a neural network," California Institute of Technology, Pasadena, CA, tech. rep. C^3P363, Sept. 1986.

[38] J. R. Anderson, "A mean field computational model for PDP," in *Proc. 1988 Connectionist Models Summer School*, D. Touretzky, G. Hinton, and T. Sejnowski, Eds. San Mateo, CA: Morgan Kaufmann, 1989, pp. 217-223.

[39] G. L. Bilbro and W. E. Snyder, "Range image restoration using mean field annealing," in *Advances in Neural Information Processing Systems 1*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 594-601.

[40] H. Hiriyannaiah, G. L. Bilbro, W. E. Snyder, and R. Mann, "Restoration of piecewise constant images via mean field annealing," *J. Opt. Soc. Amer.*, to be published.

[41] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing, Explorations in the Microstructure of Cognition*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: M.I.T. Press, 1986, pp. 318-362.

[42] F. Rosenblatt, *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*. Washington, DC: Spartan, 1961.

[43] M. Minsky, and S. Papert, *Perceptrons*. Cambridge, MA: M.I.T. Press, 1969.

[44] M. W. Roth, "Neural-network technology and its applications," *Heuristics*, vol. 2, no. 1, pp. 46-62, 1989 (reprinted version); also originally appeared in *The Johns Hopkins APL Tech. Dig.*, vol. 9, no. 3, pp. 242-253, 1988.

[45] B. Widrow and M. E. Hoff "Adaptive switching circuits," in *IRE Western Electron. Show and Conv.*, pt. 4, 1960, pp. 96-104.

[46] G. E. Hinton, "Learning distributed representations of concepts," in *Eighth Annu. Conf. Cognitive Sci. Soc.* Hillsdale, NJ: Lawrence Erlbaum, 1986, pp. 1-12.

[47] T. Kohonen, *Self-Organization and Associative Memory*. Berlin: Springer-Verlag, 1984.

[48] E. Saund, "Abstraction and representation of continuous variables in connectionist networks," in *Proc. Fifth Nat. Conf. Artificial Intell.* San Mateo, CA, Morgan Kaufmann, 1986, pp. 638-644.

[49] E. B. Baum and F. Wilczek, "Supervised learning of probability distributions by neural networks," in *Neural Information Process-*

ing Systems, D. Z. Anderson, Ed. American Institute of Physics, 1988, pp. 52-61.

[50] S. E. Fahlman, "Faster learning variations on back propagation: An empirical study," in *Proc. 1988 Connectionist Models Summer School*, D. Touretzky, G. Hinton, and T. Sejnowski, Eds. San Mateo, CA: Morgan Kaufmann, 1989, pp. 38-51.

[51] E. B. Baum, "What size net gives valid generalization?," *Neural Computation*, vol. 1, no. 1, pp. 151-160, 1989.

[52] F. J. Pineda, "Generalization of back propagation to recurrent neural networks," *Phys. Rev. Lett.*, vol. 59, pp. 2229-2232, 1987; also "Generalization of back propagation to recurrent and high-order networks," in *Neural Information Processing Systems*, D. Z. Anderson, Ed. New York: American Institute of Physics, 1988, pp. 602-611.

[53] F. J. Pineda, "Dynamics and architecture in neural computation," *J. Complexity*, vol. 4, pp. 216-245, 1988.

[54] N. Qian and T. J. Sejnowski, "Learning to solve random-dot stereograms of dense and transparent surfaces with recurrent back propagation," in *Proc. 1988 Connectionist Models Summer School*, D. Touretzky, G. Hinton, and T. Sejnowski, Eds. San Mateo, CA: Morgan Kaufmann, 1989, pp. 435-443.

[55] J. M. Oyster, F. Vicuna, and W. Broadwell, "Associative network applications to low-level machine vision," *Appl. Opt.*, vol. 26, no. 10, pp. 1919-1926, 1987.

[56] M. W. Roth, "Neural networks for extraction of weak targets in high clutter environments," *IEEE Trans. Syst., Man., Cybern.*, vol. 19, no. 5, pp. 1210-1217, 1989.

[57] J. G. Daugman, "Relaxation neural network for nonorthogonal image transforms," in *IEEE Int. Conf. Neural Networks*, vol. I. Piscataway, NJ: IEEE, 1988, pp. 547-560.

[58] J. Jones and L. Palmer, "An evaluation of the two-dimensional Gabor filter model of simple receptive fields in cat striate cortex," *J. Neurophysiol.*, vol. 58, pp. 1233-1258, 1987.

[59] P. K. Kienker, T. J. Sejnowski, G. E. Hinton, and L. E. Schumacher, "Separating figure from ground with a parallel network," *Perception*, vol. 15, pp. 197-216, 1986.

[60] N. M. Nasrabadi and Y. Feng, "Vector quantization of images based upon the Kohonen self-organizing feature maps," in *IEEE Int. Conf. Neural Networks*, 1988, pp. I-101-I-108.

[61] Y. Linde, A. Buzo, amd R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.*, vol. COM-28, pp. 84-95, Jan. 1980.

[62] G. W. Cottrell, P. Munro, and D. Zipser, "Image compression by back propagation: An example of extensional programming," in *Models of Cognition: A Review of Cognitive Science*, vol. 1, N. E. Sharkey, Ed. Norwood, NJ: Ablex, pp. 208-240, 1989.

[63] R. M. Kuczewski, M. H. Myers, and W. J. Crawford, "Exploration of backward error propagation as a self-organizational structure," in *IEEE First Int. Conf. Neural Networks*, vol. II, M. Caudill and C. Butler, Eds. Piscataway, NJ: IEEE, 1987, pp. 89-95.

[64] P. Baldi, and K. Hornik, "Neural networks and principal component analysis: Learning from examples without local minima," *Neural Networks*, vol. 2, pp. 53-58, 1989.

[65] G. W. Cottrell and P. Munro, P., "Principal components analysis of images via back propagation," in *Proc. SPIE Int. Soc. Opt. Eng.*, pp. 1070-1077, 1988.

[66] R. L. Kulp and A. S. Gilbert, "Combination of dissimilar features in neural networks for ship recognition," Johns Hopkins Univ. Appl. Phys. Lab. rep. F1A(1)-89U-022, Oct. 31, 1989.

[67] L. B. Moorehead and R. A. Jones, "A neural network for shape recognition," in *IEEE Region 5 Conf. 1988*. Piscataway, NJ: IEEE, 1988, pp. 187-191.

[68] D. T. Lawton, T. S. Levitt, and P. Gelband, "Knowledge based vision for terrestrial robots," in *Proc. Image Understanding Workshop, 1988*. San Mateo, CA: Morgan Kaufmann, 1988, pp. 103-110.

[69] R. Mohan and R. Nevatia, "Preceptual grouping for the detection and description of structures in aerial images," in *Proc. Image Understanding Workshop 1988*. San Mateo, CA: Morgan Kaufmann, 1988, pp. 512-526.

[70] T. A. Jamison and R. J. Schalkoff, "Image labelling: A neural network approach," *Image and Vision Computing*, vol. 6, no. 4, pp. 203-213, Nov. 1988.

[71] A. Pentland, "Part segmentation for object recognition," *Neural Computation*, vol. 1, no. 1, pp. 82-91, 1989.

[72] W. Y. Huang and R. P. Lippmann, "Comparisons between neural net and conventional classifiers," in *IEEE First Int. Conf. Neural Networks*, vol. IV, M. Caudill and C. Butler, Eds. Piscataway, NJ: IEEE, 1987, pp. 485-492.

[73] D. E. Glover, "An optical Fourier/electronic neurocomputer automated inspection system," in *IEEE Int. Conf. Neural Networks*, vol. I. Piscataway, NJ: IEEE, 1988, pp. 569-576.

[74] S. E. Troxel, S. K. Rogers, and M. Kabrisky, "The use of neural networks in PSRI target recognition," in *IEEE Int. Conf. Neural Networks*, vol. I. 1988, pp. 593-600.

[75] S. D. Post, "A bipartite connectionist model to represent N-ary Boolean and linear constraints," in *IEEE First Int. Conf. Neural Networks*, M. Caudill, and C. Butler, Eds. Piscataway, NJ: IEEE, 1987, pp. III-87-III-94.

[76] S. I. Gallant, "Connectionist expert systems," *Commun. Ass. Comput. Mach.*, vol. 31, no. 2, pp. 152-169, 1988.

[77] D. G. Bounds, P. J. Lloyd, B. Mathew, and G. Waddell, "A multilayer perceptron network for the diagnosis of low back pain," in *IEEE Int. Conf. Neural Networks*, vol. II, 1988, pp. 481-489.

[78] G. Bradshaw, R. Fozzard, and L. Ceci, "A connectionist expert system that actually works," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 248-255.

[79] J. S. Denker *et al.*, "Neural network recognizer for handwritten zip code digits," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 323-331.

[80] R. Hecht-Nielsen, "Neurocomputing: Picking the human brain," *IEEE Spectrum*, vol. 25, no. 3, pp. 36-41, 1988.

[81] C. Mead, *Analog VLSI and Neural Systems*. New York: Addison-Wesley, 1989.

[82] H. P. Graf and L. D. Jackel, "Analog electronic neural network circuits," *IEEE Circuits, Devices Mag.*, pp. 44-49, 55, July 1989.

[83] C. A. Mead and M. A. Mahowald, "A silicon model of early visual processing," *Neural Networks*, vol. 1, pp. 91-97, 1988.

[84] C. Koch, J. Luo, C. Mead, and J. Hutchinson, "Computing motion using resistive networks," *Proc. SPIE Int. Soc. Opt. Eng.*, pp. 108-113, 1988; also J. Hutchinson, C. Koch, J. Luo, and C. Mead, "Computing motion using analog and binary resistive networks," *Computer*, pp. 52-63, Mar. 1988.

[85] S. P. DeWeerth and C. A. Mead, "A Two-dimensional visual tracking array," in *Advance Research in VLSI*, J. Allen and F. T. Leighton, Eds. Cambridge, MA: M.I.T. Press, 1988, pp. 259-275.

[86] J. Harris, C. Koch, J. Luo, and J. Wyatt, "Resistive fuses: Analog hardware for detecting discontinuities in early vision," in *Analog VLSI Implementation of Neural Systems*, C. Mead and M. Ismail, Eds. Boston, MA: Kluwer Academic, 1989, pp. 27-55.

[87] M. A. Mahowald and T. Delbrück, "Cooperative stereo matching using static and dynamic image features," in *Analog VLSI Implementation of Neural Systems*, C. Mead and M. Ismail, Eds. Boston, MA: Kluwer Academic, 1989, pp. 213-238.

[88] R. Allen, C. Mead, F. Faggin, and G. Gribble, "Orientation-selective VLSI retina," *Proc. SPIE Int. Soc. Opt. Eng.*, pp. 1040-1046, 1988.

[89] C. Mead, "Adaptive retina," in *Analog VLSI Implementation of Neural Systems*, C. Mead and M. Ismail, Eds. Boston, MA: Kluwer Academic, 1989, pp. 239-246.

[90] D. L. Standley and J. L. Wyatt, "Stability criterion for lateral inhibition and related networks that is robust in the presence of integrated circuit parasitics," *IEEE Trans. Circuits, Syst.*, vol. 36, no. 5, pp. 675-681, 1989.

[91] M. J. S. Smith and C. L. Portmann, "Practical design and analysis of a simple 'neural' optimization circuit," *IEEE Trans. Circuits Syst.*, vol. 36, no. 1, pp. 42-50, Jan. 1989.

[92] J. I. Raffel, "The application of wafer-scale technology to neuromorphic systems," in *Advanced Research in VLSI*, J. Allen and F. T. Leighton, Eds. Cambridge, MA: M.I.T. Press, 1988, pp. 121-129.

[93] J. I. Raffel, J. R. Mann, R. Berger, A. M. Soares, and S. Gilbert, "A generic architecture for wafer-scale neuromorphic systems," *Lincoln Lab. J.*, vol. 2, no. 1, pp. 63-75, 1989.

[94] J. Alspector, R. B. Allen, V. Hu, and S. Satyanarayana, "Stochastic learning networks and their electronic implementation," in *Neural Information Processing Systems, Denver, Co., 1987*, D. A. Anderson, Ed. American Institute of Physics, 1988, pp. 9-21.

[95] D. B. Schwartz, R. E. Howard, and W. E. Hubbard, "Adaptive neural networks using MOS charge storage," in *Advances in Neural Information Processing Systems I*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1989, pp. 761-768.

[96] M. Holler, S. Tam, H. Castro, and R. Benson, "An electrically trainable artificial neural network (ETANN) with 10240 'floating gate' synapses," in *IJCNN Int. Joint Conf. Neural Networks*, vol. II. Piscataway, NJ: IEEE, 1989, pp. 191-196.
[97] Y. P. Tsividis, "Analog MOS integrated circuits—certain new ideas, trends, and obstacles," *IEEE J. Solid-State Circuits*, vol. SC-22, no. 3, pp. 317-321, June 1987.
[98] S. Bibyk and M. Ismail, "Issues in analog VLSI and MOS techniques for neural computing," in *Analog VLSI Implementation of Neural Systems*, C. Mead and M. Ismail, Eds. Boston, MA: Kluwer Academic, 1989, pp. 103-133.
[99] Y. P. Tsividis and D. Anastassiou, "Switched-capacitor neural networks," *Electron. Lett.*, vol. 23, no. 18, pp. 958-959, Aug. 27, 1987.
[100] Y. P. Tsividis and S. Satyanarayana, "Analogue circuits for variable-synapse electronic neural networks," *Electron. Lett.*, vol. 23, no. 24, pp. 1313-1314, Nov. 19, 1987.
[101] Y. S. Abu-Mostafa and D. Psaltis, "Optical neural computers," *Sci. Amer.*, vol. 256, no. 3, pp. 88-95, 1987.
[102] T. Williams, "Optics and neural nets—trying to model the human brain," *Computer Design*, pp. 47-62, Mar. 1, 1987.
[103] D. Psaltis, and N. Farhat, "Optical information processing based on an associative-memory model of neural nets with thresholding and feedback," *Opt. Lett.*, vol. 10, pp. 98-100, 1985.
[104] N. H. Farhat, D. Psaltis, A. Prata, and E. Paek, "Optical implementation of the Hopfield model," *Appl. Opt.*, vol. 24, pp. 1469-1475, 1985.
[105] G. D. Boyd, "Optically excited synapse for neural networks," *Appl. Opt.*, vol. 26, pp. 2712-2719, 1987.
[106] C. D. Kornfeld, R. C. Frye, C. C. Wong, and E. A. Rietman, "An optically programmed neural network," in *IEEE Int. Conf. Neural Networks*, vol. II, 1988, pp. 357-364.
[107] N. H. Farhat, "Optoelectronic analogs of self-programming neural nets—architecture and methodologies for implementing fast stochastic learning by simulated annealing," *Appl. Opt.*, vol. 26, pp. 5093-5103, 1987.
[108] R. A. Athale and C. W. Stirk, "Compact architectures for adaptive neural nets," *Opt. Eng.*, vol. 28, no. 4, pp. 447-455, Apr. 1989.
[109] D. Psaltis, K. Wagner, and D. Brady, "Learning in optical neural computers," in *IEEE First Int. Conf. on Neural Networks*, M. Caudill and C. Butler, Eds. Piscataway, NJ: IEEE, vol. III, 1987, pp. 549-555; also D. Psaltis, D. Brady, and K. Wagner, "Adaptive optical networks using photorefractive crystals," *Appl. Opt.*, vol. 27, no. 9, pp. 1752-1759, 1988.
[110] H. J. White, N. B. Aldridge, and I. Lindsay, "Digital and analog holographic associative memories," *Opt. Eng.*, vol. 27, no. 1, pp. 30-37, 1988.
[111] K. Wagner, and D. Psaltis, "Multilayer optical learning networks," *Appl. Opt.*, vol. 26, pp. 5061-5076, 1987.
[112] Y. Owechko, "Optoelectronic resonator neural networks," *Appl. Opt.*, vol. 26, pp. 5104-5111, 1987.
[113] H. J. Caulfield, "Parallel N^4 optical interconnections," in *IEEE First Int. Conf. Neural Networks*, vol. III, M. Caudill and C. Butler, Eds. Piscataway, NJ: IEEE, 1987, pp. 595-597.
[114] K. Hsu and D. Psaltis, "Invariance and discrimination properties of the optical associative loop," in *IEEE Int. Conf. Neural Networks*, 1988, pp. 395-402.
[115] H. M. Stoll and L. S. Lee, "A continuous-time optical neural network," in *IEEE Int. Conf. Neural Networks*, vol. II, 1988, pp. 373-384.

*

Michael W. Roth (M'87) received the B.A. degree in physics and mathematics from MacMurray College in 1971 and the M.S. and Ph.D. degrees in physics from the University of Illinois in 1972 and 1975, respectively.

He is a Principal Staff Physicist at the Johns Hopkins University Applied Physics Laboratory (JHU/APL). He has served as a Project Leader for a number of different efforts relating to sensors, signal and image processing, guidance systems, and neural networks. He has also been a member of several JHU/APL and government committees and panels. He joined JHU/APL in 1977 after serving as a Research Associate at Fermilab. He has written published papers on a variety of topics, has received a number of academic and professional awards, and is an Associate Editor of IEEE TRANSACTIONS ON NEURAL NETWORKS.