A SELF-ORGANIZING NEURAL SYSTEM FOR LEARNING TO RECOGNIZE TEXTURED SCENES

Stephen Grossberg[1] and James R. Williamson[2]

Department of Cognitive and Neural Systems
and Center for Adaptive Systems
Boston University

Vision Research, 39 (1999) 1385-1406.

All correspondence should be addressed to:

Professor Stephen Grossberg
Department of Cognitive and Neural Systems
Boston University
677 Beacon Street
Boston, MA 02215
Phone: 617-353-7858
Fax: 617-353-7755
E-mail: steve@cns.bu.edu

Keywords: pattern recognition, boundary segmentation, surface representation, filling-in, texture classification, neural network, adaptive resonance theory

[1]Supported in part by the Defense Research Projects Agency and the Office of Naval Research (ONR N00014-95-1-0409) and the Office of Naval Research (ONR N00014-95-1-0657).

[2]Supported in part by the Defense Research Projects Agency and the Office of Naval Research (ONR N00014-95-1-0409).

Abstract

A self-organizing ARTEX model is developed to categorize and classify textured image regions. ARTEX specializes the FACADE model of how the visual cortex sees, and the ART model of how temporal and prefrontal cortices interact with the hippocampal system to learn visual recognition categories and their names. FACADE processing generates a vector of boundary and surface properties, notably texture and brightness properties, by utilizing multi-scale filtering, competition, and diffusive filling-in. Its context-sensitive local measures of textured scenes can be used to recognize scenic properties that gradually change across space, as well as abrupt texture boundaries. ART incrementally learns recognition categories that classify FACADE output vectors, class names of these categories, and their probabilities. Top-down expectations within ART encode learned prototypes that pay attention to expected visual features. When novel visual information creates a poor match with the best existing category prototype, a memory search selects a new category with which classify the novel data. ARTEX is compared with psychophysical data, and is benchmarked on classification of natural textures and synthetic aperture radar images. It outperforms state-of-the-art systems that use rule-based, backpropagation, and K-nearest neighbor classifiers.

1 Introduction

1.1 Background and Benchmarks

The brain's unparalleled ability to perceive and recognize a rapidly changing world has inspired an increasing number of models aimed at exploiting these properties for purposes of automatic target recognition. On the perceptual side, the brain can cope with variable illumination levels and noisy scenic data that combine information about edges, textures, shading, and depth that are overlaid in all parts of a scene. This type of general-purpose processing enables the brain to deal with a wide range of imagery, both familiar and unfamiliar. On the recognition side, the brain can autonomously discover and learn recognition categories and predictive classifications that shape themselves to the statistics of a changing environment in real time. The present article develops a new self-organizing neural architecture that combines perceptual and recognition models that exhibit these desirable properties.

These models have individually been derived to explain and predict data about how the brain generates perceptual representations in the striate and prestriate visual cortices (e.g., Arrington, 1994; Baloch & Grossberg, 1997; Francis & Grossberg, 1996; Gove, Grossberg, & Mingolla, 1995; Grossberg, 1994, 1997; Grossberg, Mingolla, & Ross, 1997; Pessoa, Mingolla, & Neumann, 1995) and uses these representations to learn attentive recognition categories and predictions through interactions between inferotemporal, prefrontal, and hippocampal cortices (e.g., Bradski & Grossberg, 1995; Carpenter & Grossberg, 1993; Grossberg, 1995; Grossberg & Merrill, 1996). The perceptual theory in question is called FACADE theory. It consists of subsystems called the Boundary Contour System (BCS) and the Feature Contour System (FCS) that generate 3-D boundary and surface representations that model the cortical interblob and blob processing streams, respectively. The adaptive categorization and predictive theory is called Adaptive Resonance Theory, or ART. ART models are capable of stably self-organizing their recognition codes using either unsupervised or supervised incremental learning in any combination through time (Carpenter & Grossberg, 1991; Carpenter *et al.*, 1992).

The present work develops the ARTEX model to classify scenes that include complex textures, both natural and artificial. The ARTEX architecture was built up from specialized versions of FACADE and ART models that have been designed to achieve high competence in classifying textured scenes without also incorporating mechanisms that are not essential for understanding this competence. Just as the properties of the FACADE and ART models are "emergent" properties that are due to interactions of their various parts, the properties of the ARTEX architecture are also emergent properties due to interactions within and between its FACADE and ART modules. These new emergent properties are not merely "the sum of the parts" of the modules of which they are derived, and need to be analysed on their own terms.

In order to understand the emergent properties that are achieved by joining a FACADE

vision preprocessor to an ART adaptive classifier, ARTEX is benchmarked against state-of-the-art alternative models of texture classification. Our most striking results are derived through benchmark studies that classify natural textures from the Brodatz (1966) texture album, which is often used as a standardized test of texture classification models. ARTEX benchmarks emulated the conditions under which others benchmarked their algorithms on Brodatz textures. A single trial of on-line incremental category learning by ARTEX can outperform another leading model's off-line batch learning using a complex rule-based system (Greenspan, 1996; Greenspan *et al.*, 1994). ARTEX also outperforms K-nearest neighbor models in both accuracy and data compression, and multilayer perceptrons (back propagation) in both accuracy and processing time.

The classification errors that ARTEX does produce are compared with human perception of texture similarities (Rao & Lohse, 1993, 1996). A correlation exists between the psychophysically measured similarity between two textures and the probability that ARTEX will confuse them.

ARTEX is also used to classify regions in real-world scenes that have been processed by synthetic aperture radar (SAR). SAR imagery has recently become popular in many satellite image processing applications because the SAR sensor can penetrate variable weather conditions (Novak *et al.*, 1990; Waxman *et al.*, 1995). The SAR images present a challenge for texture classifiers because they contain pixel intensities that vary over five orders of magnitude and are corrupted by high levels of multiplicative noise, yielding incomplete and discontinuous boundary and surface representations. Results below on natural texture and SAR images illustrate how pattern recognition models that are based on biological principles and mechanisms can outperform models that have been derived from more traditional engineering concepts.

1.2 Psychophysical Data and Model Properties

At least two different approaches exist to texture classification. In one approach, the focus is on separating regions with different textures by finding the boundaries between them (Bergen & Adelson, 1988; Fogel & Sagi, 1989; Gurnsey & Browse, 1989; Malik & Perona, 1990; Rubenstein & Sagi, 1990; Bergen & Landy, 1991). Another approach attempts to classify the textures within small regions of a scene (Caelli, 1985, 1988; Bovik, Clark, & Geisler, 1990; Jain & Farrokhnia, 1991; Greenspan *et al.*, 1994). Such an approach discovers texture boundaries by classifying the textures within each region differently. It can also classify local regions whose textural properties vary gradually across space, and thus are not separated by a distinct boundary.

Gurnsey and Laundry (1992) have provided psychophysical data in support of the latter type of processing by showing that human texture recognition is only slightly impaired when the boundaries between different textures in a texture mozaic are blurred. ARTEX does the latter type of classification. It derives a 17-dimensional feature vector from multiple-scale boundary features of the BCS and a surface brightness feature

of the FCS. This feature vector utilizes filters of four different scales, as suggested by psychophysical experiments (Harvey & Gervais, 1978; Richards, 1979; Wilson & Bergen, 1979). The spatial filters are evaluated at four different orientations, thereby leading to a 16-dimensional (4×4) feature vector. The 17^{th} dimension is a surface brightness feature. The ARTEX model uses these feature vectors to generate a context-sensitive classification of local texture properties. These BCS and FCS operations are designed to be as simple and fast as possible without incurring a loss of accuracy in classifying texture data.

A large psychophysical literature supports the FACADE hypothesis that the human brain forms distinct boundary and surface representations before they are bound together by object recognition categories. Experimental results that support the role of boundary representations include the following: (1) Object superiority effects occur using outline stimuli with little surface detail (Davidoff & Donnelly, 1990; Homa, Haver, & Schwartz, 1976). (2) The number of errors in tachistoscopic recognition and the speed of identification are often comparable using appropriately and inappropriately colored objects (Mial, Smith, Doherty, & Smith, 1979; Ostergaard & Davidoff, 1985). (3) There is no difference in recognition speed using black-and-white photographs or line drawings that are carefully derived from them (Biederman & Ju, 1988).

Several types of data also implicate a separate surface brightness and color process. These include the following: (4) Colored surfaces may be bound to an incorrect form during illusory conjunctions (McLean, Broadbent, & Broadbent, 1983; Stefurak & Boynton, 1986; Treisman & Schmidt, 1982). (5) Color can facilitate object naming if the objects to be named are structurally similar or degraded (Christ, 1975; Price & Humphreys, 1989). (6) Colors are coded categorically prior to the processing stage at which they are named (Davidoff, 1991; Rosch, 1975). Two of the most recent studies in support of the boundary-surface distinction were carried out by Elder and Zucker (1998) and Rogers-Ramachandran and Ramachandran (1998).

FACADE theory proposes that 3-D boundary and surface features that are formed in the prestriate visual cortex are categorized in the inferotemporal cortex (Grossberg, 1994, 1997). Both boundary and surface properties are proposed to be combined during the categorization process within bottom-up and top-down adaptive pathways that are modeled by an ART system. Two consequences of this conception are that unambiguous boundaries can generate category recognition by themselves, and that boundaries can prime 3-D object representations even if they need to be supplemented by 3-D surface information in order to achieve unambiguous recognition. Cavanagh (1997) has reported data consistent with this latter prediction.

In the ARTEX implementation of this concept, the feature vectors that are formed from the 17-dimensional boundary and surface features of the FACADE preprocessor are input to an ART classifier, which categorizes the textures using a biologically-motivated learning algorithm. Humans learn to discriminate textures by looking at them and becoming sensitive to their statistical properties in small regions. This is how our model is trained. Intuitively speaking, model training is like having an observer look at a number

of locations and trying to learn to categorize them based on their local properties. The ART classifier we used, called Gaussian ARTMAP, or GAM, incrementally constructs internal categories that have Gaussian receptive fields in the input space, and that map to output class predictions (Williamson, 1996, 1997). Cells with Gaussian receptive fields are ubiquitous in the brain, and have been used to model data about how the inferotemporal cortex learns to categorize visual input patterns (Logothetis *et al.*, 1994). Such models are not, however, typically able to self-organize their own recognition categories and to autonomously search for new ones with which to classify novel input patterns. ART models overcome this weakness by showing how complementary attentional and orienting systems are designed with which to balance between the processing of familiar and expected events, on the one hand, and unfamiliar and unexpected events on the other (Carpenter & Grossberg, 1991; Grossberg, 1980; Grossberg & Merrill, 1996). All learned categorization goes on within the attentional system. The orienting subsystem is activated in response to events that are too novel for the attentional system to successfully categorize them. Interactions between the attentional and orienting subsystems then lead to a memory search which discovers a more appropriate population of cells with which to categorize the novel information. These interactions are designed to explain how the brain continues to learn quickly about huge amounts of new information throughout life, without being forced to just as quickly forget useful information that it has previously learned.

After each input is presented (i.e., each location is "observed"), GAM automatically activates cells whose receptive fields adapt to represent the input by amounts proportional to their level of *match* with the input. However, if the input is too novel for any existing receptive field to match the input well enough, then a memory search is triggered which leads to the selection of a previously uncommitted cell population with which a new category can be learned. During unsupervised learning, the correct names of the regions that are being classified are not supplied, and the level of match that is required for a category to learn is constant. The parameter that determines this degree of match is called the "vigilance" parameter because it computationally realizes the intuitive process of being more or less vigilant in respose to information of variable importance (Carpenter & Grossberg, 1991). Low vigilance allows the network to learn general categories in which many input exemplars may share the same category prototype. High vigilance enables the network to learn more specific categories, even categories in which only a single exemplar may be represented. Thus the choice of vigilance can trade between prototype and exemplar learning, even within a single ART system. Experimental evidence consistent with vigilance control has been reported in monkeys when they attempt to perform classifications during easy vs. difficult discriminations (Spitzer, Desimone, & Moran, 1988).

Learning typically starts with a low vigilance value, which leads to the formation of the most general categories that are consistent with the input data. Because ART models are self-organizing, such learning can proceed on its own in an unsupervised mode. Starting with a low vigilance value conserves memory resources, but it can also create the tendency, also found in children, to overgeneralize until further learning leads to category

refinement (Chapman, *et al.*, 1986; Clark, 1973; Smith *et al.*, 1985; Smith & Kemler, 1978; Ward, 1983). For example, it might happen that, after learning a category that classifies variations on the letter "E", the letter "F" will also activate that category, based on the visual similarity between the two types of letters. The difference between the letters "E" and "F" is determined by cultural factors, not by visual similarity. Supervised learning is often essential to prevent errors based on input similarity which do not correspond to cultural understandings, or other environmentally dependent factors. ART models can operate in both unsupervised and supervised learning modes, and can switch between the two seamlessly during the course of learning.

During supervised learning, the vigilance parameter, or required match level, is raised if an incorrect prediction is made (e.g., if there is negative reinforcement) by just enough to trigger a memory search for a new category. This type of vigilance control sacrifices category generality only when more specific categories are needed to match the statistical properties of a given environment. Categories of variable generality are hereby automatically learned based upon the success or failure of previously learned categories in predicting the correct classification. A block diagram of the ARTEX architecture is shown in Figure 1.

2 Multiple-scale Oriented Filter

The ARTEX multiple-scale oriented filter further develops the BCS filter that was introduced to explain texture data in Grossberg and Mingolla (1985). Variants of this BCS filter have since become standard in many texture segmentation algorithms (Malik & Perona, 1989; Sutter, Beck, & Graham, 1989; Bovik *et al.*, 1990; Bergen, 1991; Bergen & Landy, 1991; Jain & Farrokhnia, 1991; Graham, Beck, & Sutter, 1992; Greenspan *et al.*, 1994).

Figure 2 diagrams the ARTEX version of BCS processing (Stages 1–5) for a single spatial scale. As in Richards (1979), we used 4 spatial frequency channels. Each channel computed 4 orientational contrast features. These filter equations and parameters are described in Appendix I. A functional description is given here. Stage 1 of the BCS filter uses an on-center off-surround network whose cells obey membrane equations, or shunting laws (Grossberg, 1980, 1983) to discount the illuminant, compute contrast ratios of the image, and normalize image intensities. Stage 2 accomplishes multiple-scale oriented filtering using odd-symmetric Gabor filters at the 4 orientations and spatial scales. Stage 3 computes a local measure of absolute orientational contrast by full-wave rectifying the filter activities from Stage 2. These operations are neurally interpreted as follows: Stage 1 operations occur in the retina and LGN, Stage 2 operations at cortical simple cells, and Stage 3 operations at cortical complex cells (Grossberg & Mingolla, 1985). Stage 4 simplifies the BCS operations of boundary grouping by computing a smooth, reliable measure of orientational contrast that spatially pools responses within the same orientation. Stage 5 performs an optional orientational invariance operation which

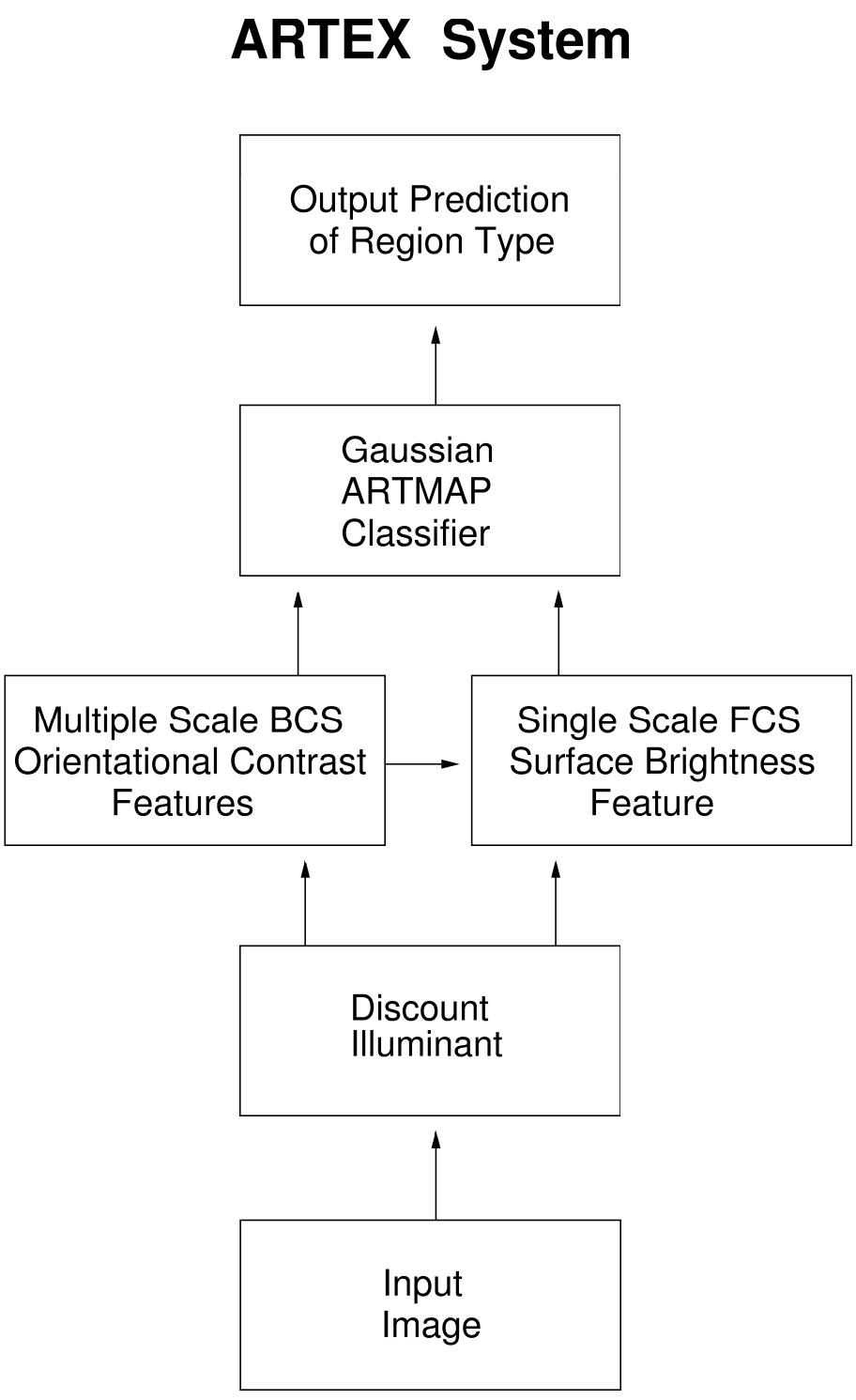


Figure 1: Block diagram of ARTEX image classification subsystems.

shifts orientational responses at each scale into a canonical ordering. This computation shifts, with wrap around, the smoothed orientational responses from Stage 4 so that the orientation with maximal amplitude is in the first orientation plane. The usefulness of this operation is task-dependent, as shown by our simulations below.

Graham *et al.* (1992) also simplified Stage 4 of the BCS by pooling responses from Stage 3. They then used a hand-crafted sigmoidal discrimination measure to convert Stage 4 output into a probabilistic output function that could be compared with subjects' ratings of texture discriminability. In the present benchmark studies, the BCS filter outputs forms part of the input vector to a GAM classifier which autonomously learns the probabilistic recognition categories with which texture discriminations are made. We note in Section 3 how the Graham *et al.* (1992) study has been extended to explain a larger data base about texture discrimination using additional FACADE theory mechanisms.

3 Filled-in Surface Brightness

The FACADE model suggests how the BCS and FCS interact to generate filled-in 3-D surface representations within the FCS. These surface representations are derived from scenic data after the illuminant has been discounted, as in Stage 1 of Figure 2. In general, these surface representations combine information about brightness, color, depth, and form. Our simulations below demonstrate the utility of using a filled-in surface brightness feature to help learn recognition categories for texture discrimination.

The simplest surface feature is one that is based on first-order differences in illumination intensity. An improved surface feature discounts the illuminant to compute a measure of local contrast. Such a feature, however, can still be corrupted by various sorts of specular noise in an image. In the brain, such noise can be due to the blind spot, retinal veins, and the retinal layers through which light must pass to activate photodetectors. In artificial sensors, too, such noise can derive from sensor characteristics. Discounting the illuminant is also insensitive to contextual groupings of image features. A filled-in surface brightness feature overcomes these deficiencies by smoothing local contrast values when they belong to the same region, while maintaining contrast differences when they belong to different regions. Filling-in hereby smoothes over image noise in a form-sensitive way, and generates a representation that reflects properties of a region's form by being contained within the region boundaries. It also tends to maximize the separability, in brightness space, of different region types by minimizing within-region variance while maximizing between-region variance. This sort of preattentive and automatic separation simplifies the task of an attentive pattern classifier such as GAM.

In Grossberg *et al.* (1995), a multiple-scale FACADE network was developed to process noisy SAR images for use by human operators. There the goal was to generate reconstructions of SAR images that were pleasing to the eyes of expert photointerpreters. The BCS in this simulation used a grouping network with a feedback process that

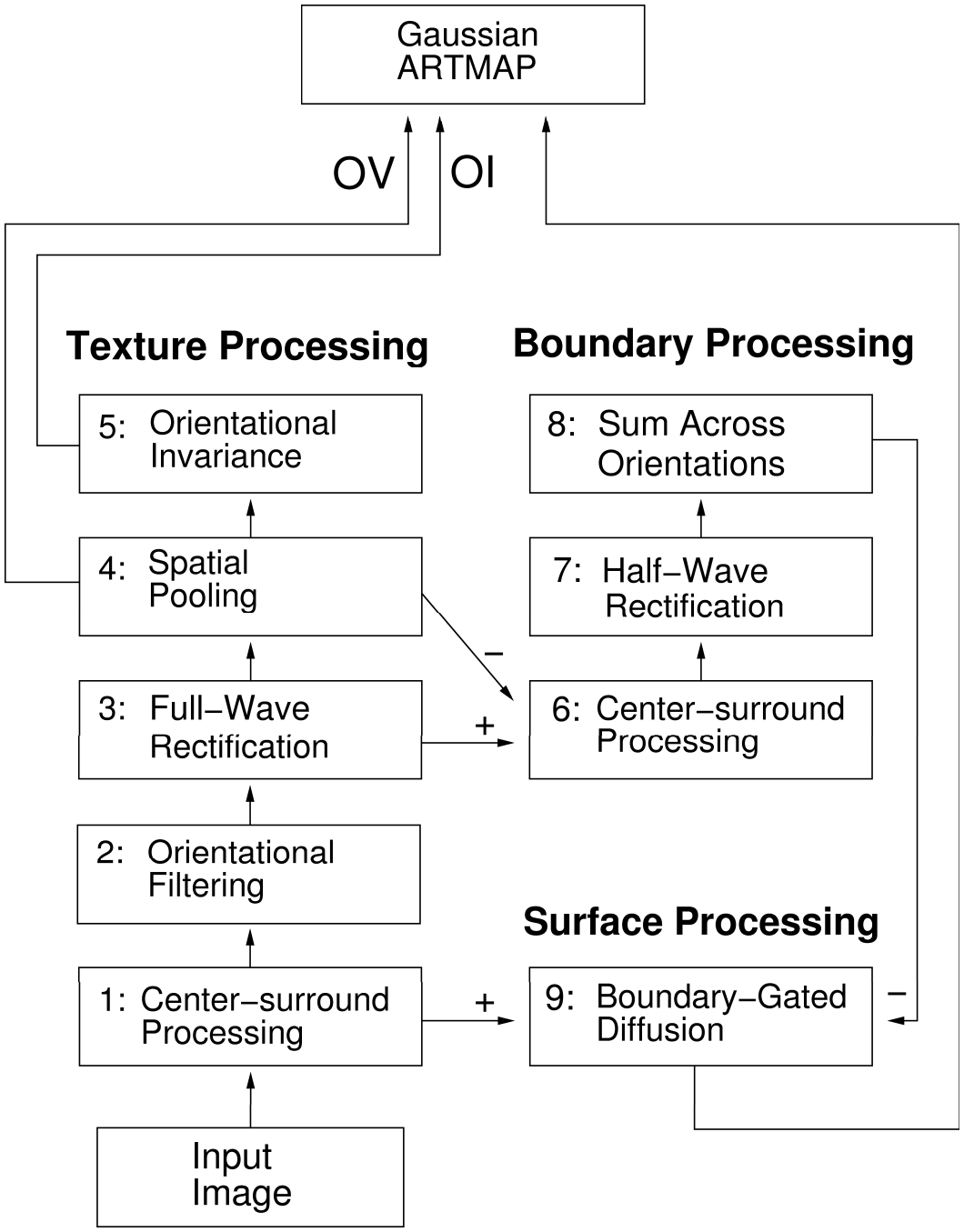


Figure 2: Boundary and surface preprocessing stages. OV = orientationally variant representation. OI = orientationally invariant representation. Either OV or OI, but not both, are active in any given problem.

can complete and sharpen boundary representations. These boundary groupings created sharply delineated image regions and filled-in surfaces. Although such a feedback grouping network has the remarkable property of converging within 1 to 3 feedback iterations, it still has the disadvantage, at least in software simulations, of slowing down processing time.

Here we replace the full BCS filter and grouping network by a multiple-scale BCS filter and a single scale of one-pass feedforward boundary processing to control filling-in of the brightness feature. Computer simulations summarized below demonstrate that this simplification does not impair classification benchmarks on Brodatz textures and on SAR textured scenes. The simplified boundary segmentation is, moreover, computationally 75 times faster than the feedback network. The slower feedback benchmarks are not reported here. Accurate texture classification thus does not seem to depend upon photorealism of the corresponding percept. Stages 6–9 of Figure 2 show how the BCS filter output is used to derive the one-pass boundary segmentation. Appendix II contains the equations and parameters of this simplified brightness filling-in process.

These FACADE preprocessing results can be placed into a larger framework to better understand their relevance for understanding human texture discrimination. Three issues need to be considered: (1) the use of a simplified Stage 4 spatial pooling operation instead of long-range grouping by a feedback network; (2) the role of surface representations; and (3) the need for 3-D boundary and surface representations. When are long-range groupings, such as illusory contours, not needed to improve texture discriminability? This is more true when the images contain dense enough textures to obviate the need for grouping over long distances. Not all of the data considered even by Graham *et al.* (1992) were of this type, however, since their displays contained regularly placed features that could group together in orientations colinear, perpendicular, or oblique to their defining edges. Cruthirds *et al.* (1993) showed that a multiple-scale BCS filter, supplemented by the long-range groupings of a feedback network, could simulate the pairwise ordering of human ratings of texture discriminability better than the Graham *et al.* (1992) variant of the BCS filter on its own.

Grossberg and Pessoa (1997) have simulated a variant of FACADE theory in which both 2-D and 3-D boundary and surface operations were needed to simulate psychophysical data about the discrimination of textured regions composed of regular arrays of equiluminant colored regions on backgrounds of variable luminance, as in the experiments of Beck (1994) and Pessoa, Beck, & Mingolla (1996). This latter simulation study was restricted, however, to textures composed of colored squares on achromatic backgrounds, rather than the stochastic factors that arise in Brodatz and SAR textures. The Grossberg and Pessoa (1997) study also does not analyze how recognition categories for discriminating textures are learned. Taken together, however, these several studies provide converging evidence that FACADE mechanisms can explain challenging properties of data concerning human texture segregation.

4 ART Heuristics

The 16-dimensional feature vector produced by Stages 1–5 (representing orientational contrast at 4 orientations and 4 spatial scales) and the single filled-in brightness feature produced by Stages 6–9 yield a 17-dimensional boundary-surface feature vector. GAM must learn a mapping from the input space populated by these feature vectors to a discrete output space of associated region class labels. As noted above, GAM shares a number of key properties with other ARTMAP architectures (Carpenter, Grossberg, and Reynolds, 1991; Carpenter *et al.*, 1992). GAM learns mappings incrementally, without any prior knowledge of the problem domain, by self-organizing an efficient set of recognition categories that shape themselves to the statistics of the input environment, as well as a map from recognition categories to class labels, which are supplied during supervised learning. Because GAM learns its mappings incrementally, a previously trained GAM network may be retrained with new input/output contingencies, including new class labels, without any need to retrain the network on the previous data. Finally, although GAM is trained only with individual class labels, it also learns to accurately estimate the probabilities of its class label predictions, as we show in our simulations below.

In a typical ART network (Carpenter & Grossberg, 1987, 1991), an input vector activates feature selective cells within the attentional system that store the vector in short-term memory. This short-term memory pattern then activates bottom-up pathways whose signals are filtered by learned adaptive weights, or long-term memory traces. The filtered signals are added up at target category nodes which compete via recurrent lateral inhibition to determine which category activities will be stored in short-term memory and thereby represent the input vector. The degree of activation of a category provides an estimate of the likelihood that an input belongs to the category. Activating a category is like "making a hypothesis".

As they are being activated, the selected categories read-out learned top-down expectations, or prototypes, which are matched against the input vector at the feature detectors. This matching process plays the role of "testing the hypothesis". The vigilance parameter defines the criterion for a good enough match. As noted above, low vigilance leads to the learning of general categories, whereas high vigilance leads to the learning of specialized categories, even a single exemplar, in the limit of very high vigilance. By varying vigilance, an ART system can hereby learn both abstract prototypes and concrete exemplars.

If the chosen category's match function exceeds the vigilance parameter, then the bottom-up and top-down exchange of feedback signals locks the system into a resonant state. The resonant state signifies that the hypothesis matches the data well enough to be accepted by the system. ART proposes that these resonant states focus attention upon relevant feature combinations, and that only resonant states enter conscious awareness (Grossberg, 1980). Resonance triggers learning in both the bottom-up adaptive weights that are used to activate the selected recognition category, and in the top-down weights that represent its prototype. This learning incorporates the new information supplied by

the input vector into the long-term memory of the attentional system.

If the category's match function does not exceed vigilance, this designates that the hypothesis is too novel to be incorporated into the prototype of the active category. A bout of memory search, or hypothesis testing, is then triggered through activation of the orienting system. Memory search either discovers a category that can better represent the data or, if no such learned category already exists, automatically chooses uncommitted cells with which to learn a new category. ART hereby incrementally discovers new categories whose degree of generalization varies inversely with the size of the vigilance parameter. Neurobiological data about recognition learning in inferotemporal cortex that are consistent with these hypotheses are reviewed by Carpenter and Grossberg (1993) and Grossberg and Merrill (1996).

All of the above properties proceed autonomously in ART networks as they undergo unsupervised learning. ARTMAP extends these ART designs to include both supervised and unsupervised learning (Carpenter, Grossberg, & Reynolds, 1991; Carpenter *et al.*, 1992). In ARTMAP, the chosen ART categories learn to make predictions which take the form of mappings to the names of output classes. In such an ARTMAP system, many different recognition categories can all learn to map into the same output name, much as many different visual fonts of a given letter of the alphabet can be grouped into several different visual recognition categories, based upon visual similarity, before these visual categories are mapped into the same auditory category that is used to name that letter.

ARTMAP systems propose how to correct a prediction, as in the case where the letter "E" is disconfirmed by environmental feedback that the correct letter is "F", using only local operations in environments that may be filled with unexpected events. ARTMAP does this using a *minimax learning principle*, which conjointly maximizes predictive generalization while it minimizes predictive error. ARTMAP does this by trying to form the largest categories that are consistent with environmental feedback. A *match tracking* process realizes this principle by increasing the vigilance value after each disconfirmation until it exceeds the chosen category's match function. This vigilance increase is the minimal one that can trigger new hypothesis testing on that learning trial. Match tracking hereby gives up the minimum amount of generalization that is required to correct the error. In summary, an ARTMAP system organizes its categorization of experience based both on the similarity of the input feature vectors and upon feedback from the environmental response, whether culturally or otherwise determined, to the names or other behaviors that its categories predict.

5 Gaussian ARTMAP

Gaussian ART (Williamson, 1996, 1997) provides a means for an ART system to learn the statistics of an input environment. Each of its categories defines a Gaussian distribution in the input space, with a mean and variance in each input dimension, as well as an

overall *a priori* probability. The Gaussian ART bottom-up activation function evaluates the probability that the input belongs to a category, given its Gaussian distribution and *a priori* probability. The match function evaluates how well the input fits the category's distribution, which is normalized to a unit height. This match is a measure of the distance, in units of standard deviation, between the input vector and the category's mean. Vigilance specifies the maximum allowable size of this distance.

Gaussian ART also uses distributed learning, in which multiple categories can all cooperate to classify an input event. Gaussian ART hereby avoids the problems incurred by "grandmother cell" models of recognition. Each such category is assigned credit based on its proportion of the net activation, which is determined by all categories whose match functions satisfy the vigilance criterion. Each category then learns by an amount that is determined by its credit. When Gaussian ART is extended to Gaussian ARTMAP to enable it to benefit from both supervised and unsupervised learning, each category's credit is determined by its proportion of the net activation of its *ensemble*, which consists of all categories that map to the same output prediction. The normalized strength of each ensemble's prediction is a probability estimate for that prediction. The equations and parameters for Gaussian ARTMAP are found in Appendix III.

6 Some Alternative Texture Classifiers

6.1 Comparison of Feature Extraction Methods

In order to evaluate the promise of any vision system, particularly one that attempts to explain such a complex competence as textured scene classification, one needs to evaluate that it really "works". This is particularly the case when the key behavioral properties emerge due to interactions across the entire system. There is thus no substitute for running such a system on benchmarks on which competing systems have also been evaluated. Our benchmark comparisons, presented in Section 7, evaluate ARTEX under conditions that are as similar as possible to those under which these competing systems have been evaluated.

ARTEX performance is first compared to that of a system that was used to classify natural textures in Greenspan *et al.* (1994) and Greenspan (1996). We call their model the Hybrid System because it is a hybrid architecture that used a log-Gabor Gaussian pyramid for feature extraction followed by one of three alternative classifiers. Although the Hybrid System was not developed to explain biological data, it has the virtue of having been developed to the point that it could be successfully tested on benchmark data bases that use textures or textured scenes as their inputs. Most other biologically derived models have not yet reached this level of development.

The Hybrid System's log-Gabor pyramid uses three levels, or spatial scales, and four orientations at each scale. Each level, after the first one, of the Gaussian pyramid is

obtained by blurring the previous lower level (i.e., smaller spatial scale) with a Gaussian kernel (with standard deviation $\sigma = 1$) and then decimating the image (i.e., removing 3 out of 4 pixels in each 2x2 pixel block). Due to decimation, the Gaussian at each successive level effectively has twice the σ of the Gaussian used in the previous level. The final outputs of all three pyramid levels of the Hybrid System have the same net amount of blurring, produced by three successive blur/decimate steps. This amount of blurring is equivalent to convolving with a single Gaussian kernel with $\sigma = \sqrt{21} = \sqrt{1^2 + 2^2 + 4^2}$, which produces an 8x8 pixel resolution. That is, each patch of 8×8 pixels in the input image yields a single pixel in an output image for each oriented contrast feature. In Greenspan (1996), classification results at 16×16, 32×32, and 64×64 resolution were also reported.

Without further preprocessing, ARTEX produces feature images at single pixel resolution. To make a fair comparison with the results reported by Greenspan *et al.* (1994) and Greenspan (1996), ARTEX feature images need to be reduced, via blurring and decimation, to the same resolution used there. For example, to change the ARTEX features to 8×8 resolution, the smaller-scale ARTEX features require additional blurring prior to decimation so that their net amount of blurring is equivalent to convolving with a single Gaussian kernel with $\sigma = \sqrt{21}$.

The net amount of blurring is a crucial consideration for the two types of tasks on which the systems are compared. The first task is classification of a library of texture images. Because this task does not include transitions between different textures, performance monotonically improves as blurring is increased, since blurring reduces variance and thus improves the signal-to-noise ratio. The second task is classification of a texture mosaic. Here, texture transitions need to be accurately resolved, so performance degrades with over-blurring. We demonstrate both of these phenomena below.

6.2 Comparison of Classification Methods

In the Hybrid System's first classification scheme, the extracted features are clustered independently in each feature dimension using the K-means procedure. Mappings from these clusters to class labels are then formed using a batch learning, rule-based algorithm called ITRULE (Goodman, *et al.*, 1992). The clusters in this scheme are formed to discretize the input, so that ITRULE can form explicit rules mapping them to the output classes. ITRULE forms a large number of rules. The exact number is never stated in Greenspan (1996). On the large problems, however, a maximum of 10,000 is allowed, and as many as 430 rules per class are reported for discriminating only two textures. Another drawback of this approach is that unsupervised discretization via K-means clustering throws away potentially important information because the clusters may span discrimination boundaries in the input space. Finally, GAM enjoys a major practical advantage in that it uses a simple incremental learning procedure as opposed to the complex and computationally expensive batch learning procedure used by ITRULE.

The two alternative classifiers used in Greenspan (1996) are standard incremental learning schemes: the K-nearest neighbor (K-NN) classifier and the multilayer perceptron (MLP), backpropagation algorithm. These two approaches have complementary advantages and flaws. K-NN learns quickly (one training epoch) but achieves no data compression. MLP, on the other hand, achieves better data compression but learns very slowly (500 slow-learning training epochs in Greenspan, 1996). An additional drawback of MLP is that it uses a form of mismatch learning that may suffer from "catastrophic forgetting" if trained on new data with different contingencies from previous data. As demonstrated by our results below, GAM combines the good properties of the above three classifiers: like ITRULE, GAM predicts the posterior probabilities of the output classes; like K-NN, GAM learns local mappings quickly; like MLP, GAM achieves significant data compression. Although GAM use a more local representation than MLP, and thus could, in principle, require more memory, GAM compensates for this by constructively forming a representation of appropriate size for whatever problem it is trained on.

7 Texture Classification Results

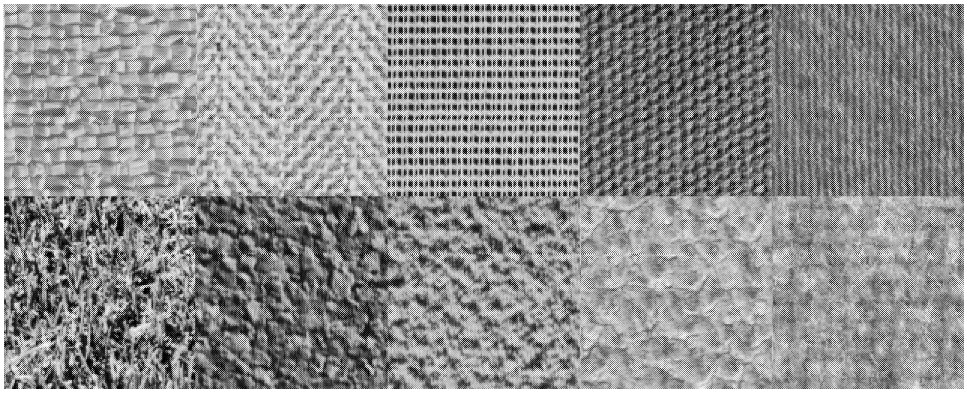
7.1 10-Texture Library

ARTEX was first compared to the Hybrid System on the library of ten textures shown in Figure 3A, whose top row contains structured textures and whose bottom row contains unstructured textures. Each texture image consists of 128 $\times$ 128 pixels. Three other images of each texture are not shown. In Greenspan (1996), classification results of the Hybrid System using ITRULE, K-NN, and MLP classifiers were published for this database. The classifiers were trained on data at three different levels of spatial resolution, with a different number of training samples per class at each resolution: 300 samples at 8 $\times$ 8 resolution, 125 samples at 16 $\times$ 16 resolution, and 40 samples at 32 $\times$ 32 resolution. ARTEX was trained on the same data set under the same conditions. Like the Hybrid System, ARTEX used an orientationally variant, or OV, representation on this problem since generalization to novel orientations of the same texture during testing was not required. ARTEX was evaluated with five random orderings of the data, and the results were averaged.

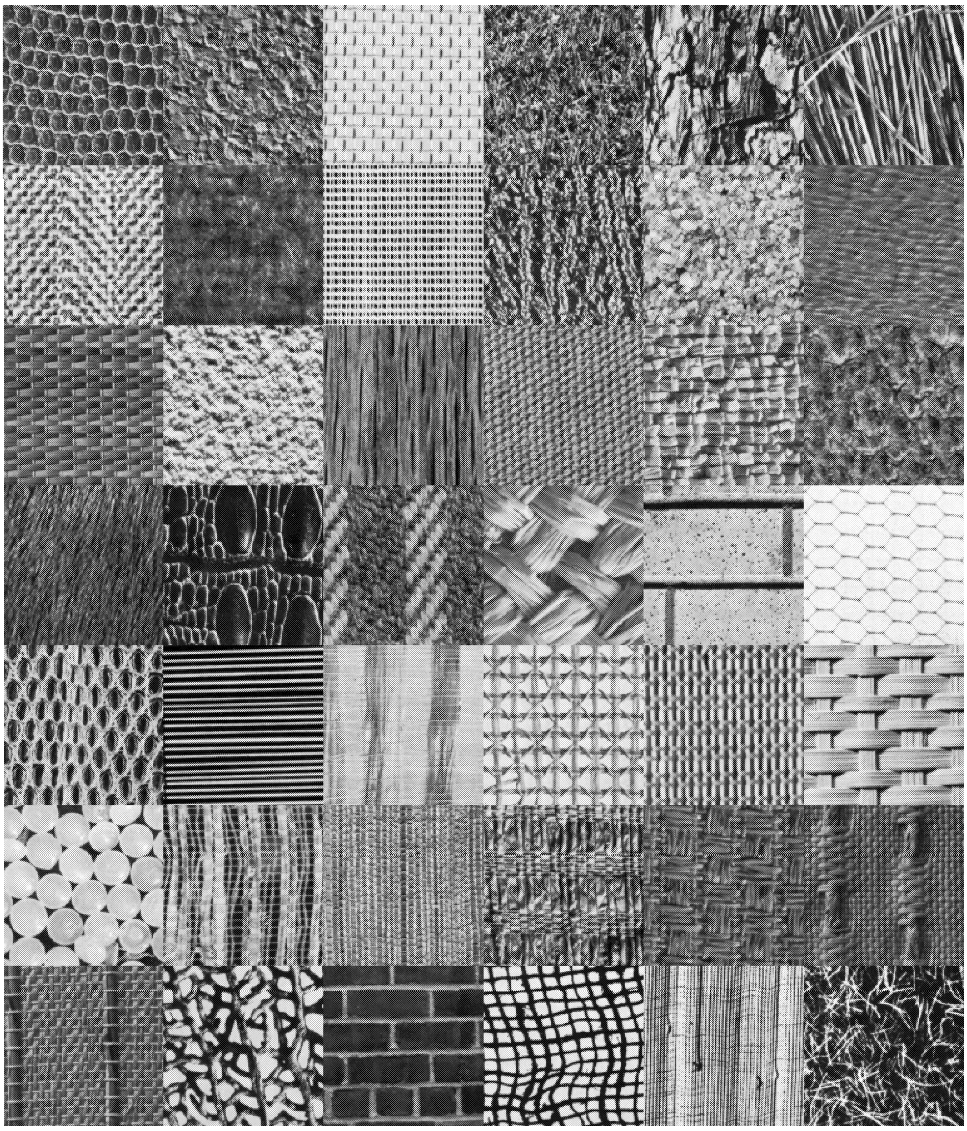
Table 1 shows comparative results for the Hybrid System and ARTEX at the three spatial resolutions. Table 1 lists the classification rate, number of epochs, and number of categories (or hidden units, stored exemplars, etc.) for each system configuration. The number of epochs indicates how many training trials were needed. The number of categories indicate how well the model compresses the data. In the case of K-NN, there is no compression, so each input or exemplar forms a different category. The number of weights indicate the memory resources, or computational complexity, that is needed to achieve this degree of compression. The goal is to minimize the number of epochs, categories, and weights. 60 hidden units are listed for MLP because the average MLP

Figure 3: (Next page). **A)** 10-texture database of textures corresponding to Figure 2 of Greenspan et al. (1994). Top row consists of structured textures, and bottom row of unstructured textures. Textures from Brodatz album are labeled with plate number. Top row (left to right): raffia (D84), herringbone weave (D17), french canvas (D21), cotton canvas (D77), jeans. Bottom row (left to right): grass (D9), pressed cork (D4), handmade paper (D57), pigskin (D92), and wool (D19). **B)** 42-texture database from Brodatz album. ROW 1: reptile skin (D3), cork (D4), wire (D6), grass (D9), bark (D12), straw (D15). ROW 2: herringbone (D17), wool (D19), french canvas (D21), calf (D24), sand (D29), water (D38). ROW 3: straw matting (D55), handmade paper (D57), wood (D68), cotton canvas (D77), raffia looped (D84), pigskin (D92). ROW 4: fur (D93), crocodile skin (D10), homespun wool (D11), raffia weave (D18), ceramic brick (D26), netting (D34). ROW 5: lizard skin (D36), straw screening (D49), raffia woven (D50), oriental cloth (D52), oriental cloth (D53), oriental rattan (D65). ROW 6: plastic pellets (D66), oriental grass fiber (D76), oriental cloth (D78), oriental cloth (D80), oriental cloth (D82), woven matting (D83). ROW 7: straw matting (D85), sea fan (D87), brick (D95), burlap (D103), cheesecloth (D105), grassy fiber (D110).

A)



B)



10-Texture Problem

Configuration	Class. Rate	# Samples/Class	# Epochs	# Categories	# Weights
8 × 8 Resolution:					
Hybrid System, ITRULE	94.3%	300	Batch	—	—
Hybrid System, MLP	94.5%	300	500	60	1,500
Hybrid System, K-NN	87.0%	300	1	3,000	48,000
ARTEX, all features	95.8%	300	1	26.6	958
ARTEX, all features	96.3%	300	5	34.0	1,224
ARTEX, no large-scale features	97.1%	300	5	41.0	1,148
ARTEX, no brightness feature	95.6%	300	5	38.4	1,306
ARTEX, no large-scale or brightness features	95.7%	300	5	47.2	1,227
16 × 16 Resolution:					
Hybrid System, ITRULE	95.0%	125	Batch	—	—
Hybrid System, MLP	96.0%	125	500	60	1,500
Hybrid System, K-NN	93.0%	125	1	1,250	20,000
ARTEX, all features	97.2%	125	1	17.4	626
32 × 32 Resolution:					
Hybrid System, ITRULE	97.8%	40	Batch	—	—
Hybrid System, MLP	100.0%	40	500	60	1,500
Hybrid System, K-NN	99.0%	40	1	400	6,400
ARTEX, all features	100.0%	40	1	10.6	382

Table 1: Recognition statistics on 10-texture library at three pixel resolutions: 8 × 8, 16 × 16, and 32 × 32. The number of weights is determined by multiplying the number of categories times the number of weights per category, or WPC. WPC is calculated based on the dimension of the input space, M, and the number of output classes, K. $M = 15$ for the Hybrid System, $M = 17$ for ARTEX, and $K = 10$ because there are 10 textures. For MLP, $WPC = M + K = 25$. For K-NN, $WPC = M + 1 = 16$. For ARTEX with all features, $WPC = 2M + 2 = 36$. For ARTEX with no large-scale features ($M = 13$), $WPC = 28$. For ARTEX with no brightness feature ($M = 16$) , $WPC = 34$. For ARTEX with no large-scale or brightness features ($M = 12$), $WPC = 26$. For example, the 48,000 weights for K-NN are computed as follows. The Hybrid System uses 15 features per input sample. With K-NN, these 15 features plus the correct class label must be stored for each training sample. Therefore, the number of weights that must be stored is 16 × (number of training samples). Since there are 300 samples/class and 10 classes, there are 3,000 training samples. In all $16 \times 3,000 = 48,000$ weights.

results were reported for 30, 60, and 90 hidden units.

ARTEX was tested with several configurations, with different subsets of its features removed. With its full 17-dimensional feature set, ARTEX achieved 95.8% correct after only one incremental training epoch, and 96.3% after five epochs. By comparison, the Hybrid System with K-NN achieved only 87.0% correct after one training epoch, at the cost of 3,000 stored exemplars compared to 23 internal categories for ARTEX. With much longer training times (i.e., 500 training epochs using MLP, or the computationally expensive batch-learning procedures using K-means and ITRULE), the Hybrid System did not match the performance of ARTEX with only one incremental learning epoch, and exhibited 49% more errors than ARTEX with 5 training epochs.

Three alternative ARTEX configurations were also tested to elucidate why ARTEX achieved better results than the Hybrid System. ARTEX uses four spatial scales versus only three for the Hybrid System. Therefore, perhaps its largest spatial scale conferred an advantage to ARTEX. This possibility was tested by removing the largest scale, resulting in a slight performance increment (97.1%). Another unique feature used by ARTEX is its filled-in surface brightness feature, which seems to be more effective than the multi-scale Gaussian blurring used by the Hybrid System. Removing the brightness feature resulted in a performance decrement (95.6%). This difference quantifies how much surface as opposed to boundary properties influence recognition accuracy on these data. Finally, both the large-scale and the brightness features were removed. This resulted in a similar performance decrement (95.7%).

The modest role played by the surface brightness feature in classifying these data is consistent with cognitive evidence summarized above suggesting that boundary inputs that go directly to the human cognitive recognition system are often sufficient to accurately recognize many objects. Surface brightness and color properties become more important insofar as the boundary information, by itself, is ambiguous. Given that boundaries are predicted to be perceptually invisible within the BCS itself (viz., the interblob cortical processing stream), these results are consistent with the possibility of being able to quickly begin to recognize certain objects using their invisible boundaries even before these objects become visible through their surface properties.

The ARTEX advantage, even with five ARTEX features removed, is probably due to some remaining differences between the systems: (1) the nature of band-pass filtering prior to orientational filtering, (2) the bandwidth characteristics of the orientational filters, (3) spatial pooling at the third spatial scale, and/or (4) the classification scheme. The first difference is in the Stage 1 band-pass filtering operation prior to the orientational Gabor filtering. The Hybrid System uses a Laplacian pyramid in which both the center and surround Gaussians that make up the band-pass filter double in size with each scale. In ARTEX, on the other hand, only the surround Gaussian grows with each successive spatial scale. It preserves on-center resolution while varying the scale of image normalization and noise suppression. Thus, the Hybrid System is much more restrictive in the range of spatial frequencies that are passed through to its orientational filtering stage. The second

difference is that the oriented filters used by the two models have different bandwidth characteristics: the ARTEX Gabor filters are defined with higher-frequency sinewaves (50% higher frequency; see Appendix I for parameters). The third difference is that Stage 4 of ARTEX performs spatial pooling following orientational filtering at each spatial scale. The Hybrid System does not do this in its largest spatial frequency channel at 8×8 resolution. Therefore, this discrepancy might help explain why ARTEX outperforms the Hybrid System at 8×8 resolution, but not at lower resolutions. The fourth difference is the classification stage. The advantages of the self-organizing Gaussian ARTMAP classifier over those used by the Hybrid System are described above.

7.2 Larger Texture Libraries

In Greenspan (1996), recognition statistics of the Hybrid System on a 30-texture library were presented. This library consists of 19 textures from the Brodatz album, and 11 additional textures of comparable complexity. We were unable to obtain this database, and so we chose to evaluate ARTEX on a library of similar textures obtained solely from the Brodatz album, which contains the 19 textures used in Greenspan (1996) as a subset. Figure 3B shows this library of 42 Brodatz textures. The plate numbers from the Brodatz album are listed in the caption. The 19 textures evaluated in Greenspan (1996) comprise the first three rows of Figure 3, as well as the first texture of the fourth row.

ARTEX was trained on this database at the same three resolutions as above, as well as at a 64×64 pixel resolution, which uses 12 samples per class. It is useful to compare performance at different resolutions. However, the training set sizes used in Greenspan (1996) are not consistent across resolutions. Using 12 samples per class at 64×64 resolution corresponds to using 768 samples per class at 8×8 resolution, rather than the 300 samples per class that were actually used, in terms of the image extent from which the samples are actually derived. Therefore, in order to obtain a meaningful measure of the performance increment resulting from 64×64 pixel resolution versus 8×8 resolution, we also trained ARTEX using 768 samples per class, as well as 300 samples per class, at 8×8 resolution.

ARTEX was evaluated on different-sized subsets of the library shown in Figure 3. ARTEX was evaluated on row 1 (6 textures), on rows 1 and 2 (12 textures), on rows 1–3 (18 textures), etc., up to all 42 textures. ARTEX was evaluated with five random orderings of the data, and the results were averaged. For the 300 samples/class case, the results are shown after 5 training epochs, and for the 768 samples/class case, the results are shown after 2 training epochs. Thus, GAM was trained on about 1,500 net samples/class in both cases.

Figure 4 plots the results at 8×8 resolution for all the texture set sizes, from 6 up to 42 textures. Figure 4 (top) plots the classification rates, and Figure 4 (bottom) plots the average number of categories that were learned in the ensembles that predicted each texture class. Note that the classification rate degrades gracefully as the number of

30-Texture Problem

Configuration	Class. Rate	# Samples/Class	# Epochs	# Categories	# Weights
8 × 8 Resolution:					
Hybrid System, ITRULE	80.0%	300	Batch	—	—
Hybrid System, MLP	89.6%	300	500	60	2,700
Hybrid System, K-NN	82.0%	300	1	9,000	144,000
ARTEX	92.5%	300	5	208.0	7,488
ARTEX	94.3%	768	2	357.6	12,874
16 × 16 Resolution:					
Hybrid System, ITRULE	84.0%	125	Batch	—	—
Hybrid System, MLP	93.4%	125	500	60	2,700
Hybrid System, K-NN	88.0%	125	1	3,750	60,000
ARTEX	95.5%	125	1	68.0	2,448
32 × 32 Resolution:					
Hybrid System, ITRULE	94.4%	40	Batch	—	—
Hybrid System, MLP	98.2%	40	500	60	2,700
Hybrid System, K-NN	96.6%	40	1	1,200	19,200
ARTEX	98.9%	40	1	38.4	1,382
64 × 64 Resolution:					
Hybrid System, ITRULE	97.5%	12	Batch	—	—
Hybrid System, MLP	97.3%	12	500	60	2,700
Hybrid System, K-NN	95.0%	12	1	360	5,760
ARTEX	100.0%	12	1	33.0	1,188

Table 2: Recognition statistics on 30-texture library, at four pixel resolutions: 8 × 8, 16 × 16, 32 × 32, and 64 × 64. Here, $K = 30$ because there are 30 textures. For MLP, $WPC = 45$. For K-NN, $WPC = 16$. For ARTEX, $WPC = 36$.

classes is increased, while the average number of categories per class gradually increases. Thus, ARTEX scales well as the number of textures increases. ARTEX achieves higher classification rates, and creates more categories, for the 768 samples/class case than it does for the 300 samples/class case. Table 2 lists the results of the Hybrid System on the 30-texture library reported in Greenspan (1996), along with the results of ARTEX on the 30 textures in the first five rows of Figure 3, at four spatial resolutions. As Table 2 shows, ARTEX obtains higher classification rates than all three variations of the Hybrid System at all the resolutions. At lower resolutions, as the classification problem becomes easier, ARTEX creates smaller representations. These representations range from about 13,000 weights at 8 × 8 resolution down to about 1,000 weights at 64 × 64 resolution.

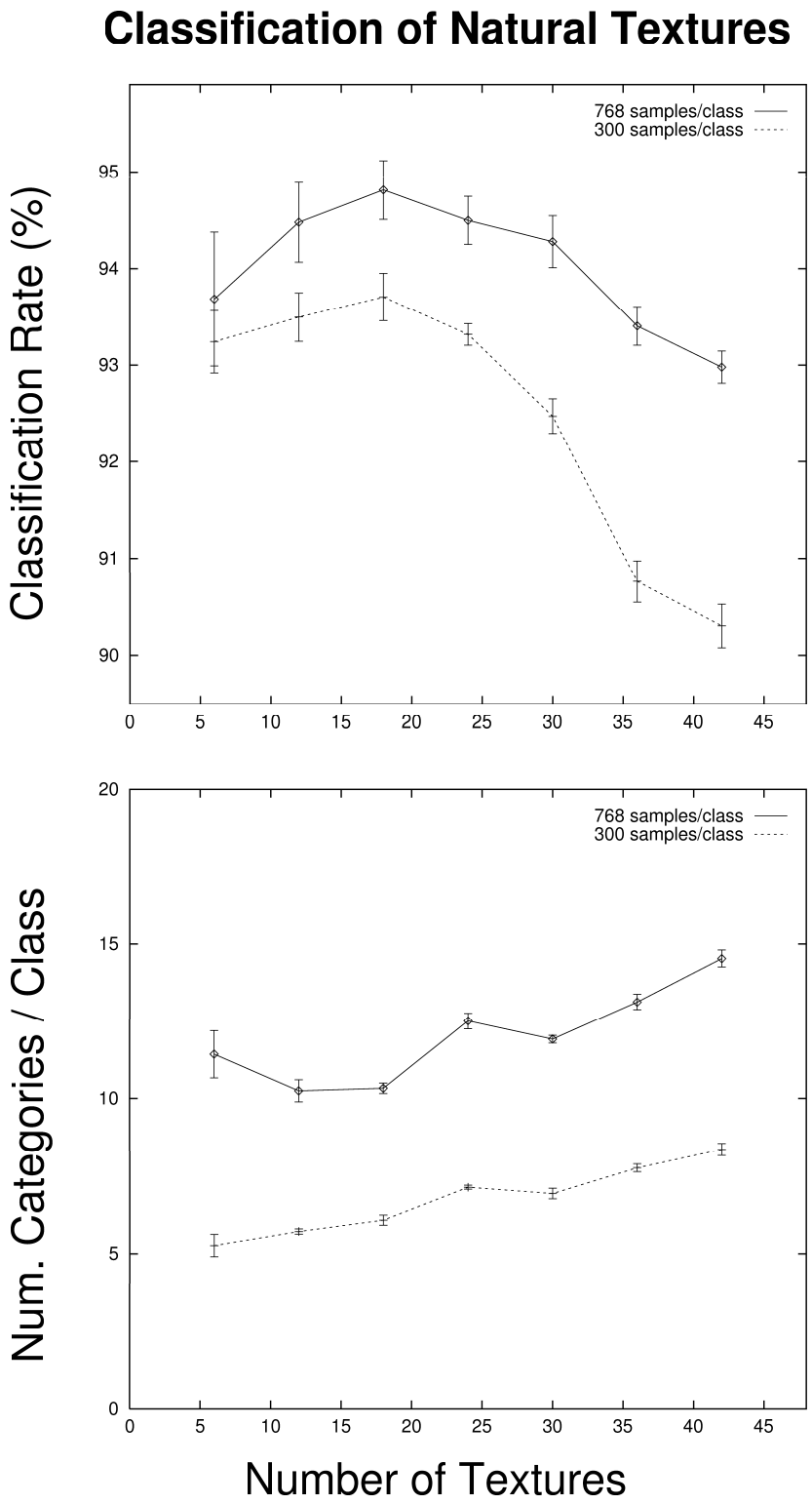


Figure 4: ARTEX performance on various subsets of the texture library in Figure 3B.

7.3 Texture Mosaic

ARTEX was also trained and tested on a texture mosaic problem reported in Greenspan *et al.* (1994) in order to evaluate classification accuracy at texture boundaries. Such an analysis indicates the extent to which a system that classifies textured scenes based on local texture properties, as suggested by the human psychophysical data summarized above, can also identify texture boundaries. The test mosaic is a 256x256 pixel image (Figure 5, TOP) which consists of five textures (grass, raffia, wood, herringbone, and wool). As in Greenspan *et al.* (1994), ARTEX was trained on these textures as well as on an additional sixth texture (sand). ARTEX was trained at four spatial resolutions, and its resulting class predictions for the texture mosaic are shown in Figure 5. From black to white, the class predictions correspond (in order) to sand, grass, raffia, wood, herringbone, and wool. Unlike the texture library problems above, performance here degrades (from 95.7% correct down to 79.5% correct) at lower resolutions because of a loss of accuracy at texture boundaries. The texture predictions of the Hybrid System on this problem (at 8×8 resolution), shown visually in Figure 5 of Greenspan *et al.*, (1994), appear to be less accurate than those obtained by ARTEX.

7.4 Comparison to Psychophysical Results

ARTEX is able to classify a large number of textures, and to localize the transitions between textures, with high accuracy. But is the performance of ARTEX consistent with what we know about human texture perception? To investigate this question we compared the errors that ARTEX produces with measures of the perceived similarities between pairs of textures (Rao & Lohse, 1993, 1996). Rao and Lohse derived these measures from subjects' hierarchical clustering of 56 Brodatz textures based on their similarity, via multidimensional scaling (MDS). 3-D coordinates for the 56 textures were obtained, which preserved 88% of the variance contained in the clustering statistics. These MDS measurements were also independently validated by comparison with subjects' ratings of the textures on 12 dimensions such as "high contrast", "repetitive", and "granular".

Our data set (which was used in the previous benchmarks) contains 21 of the 56 textures used by Rao and Lohse. We trained ARTEX on these 21 textures using the same procedures as described above. ARTEX obtained 93.9% correct on the test set after training with 768 samples/class for 2 learning epochs, and 87.9% correct after training with 300 samples/class for 5 learning epochs. For each pair of the 21 textures (210 pairs), we tallied the number of times ARTEX mistook one of the two textures for the other. Despite the difference in absolute number of errors, both training regimes yielded the same negative correlation (correlation coefficient $= -0.3$) between the number of pairwise confusions and the MDS distance between the textures. Therefore, the more similar two textures appear to people, the more likely ARTEX is to confuse them. This correlation may not be higher because of the difference between the sets of textures that are used in the simulations and the experiments, and the fact that texture similarity and texture

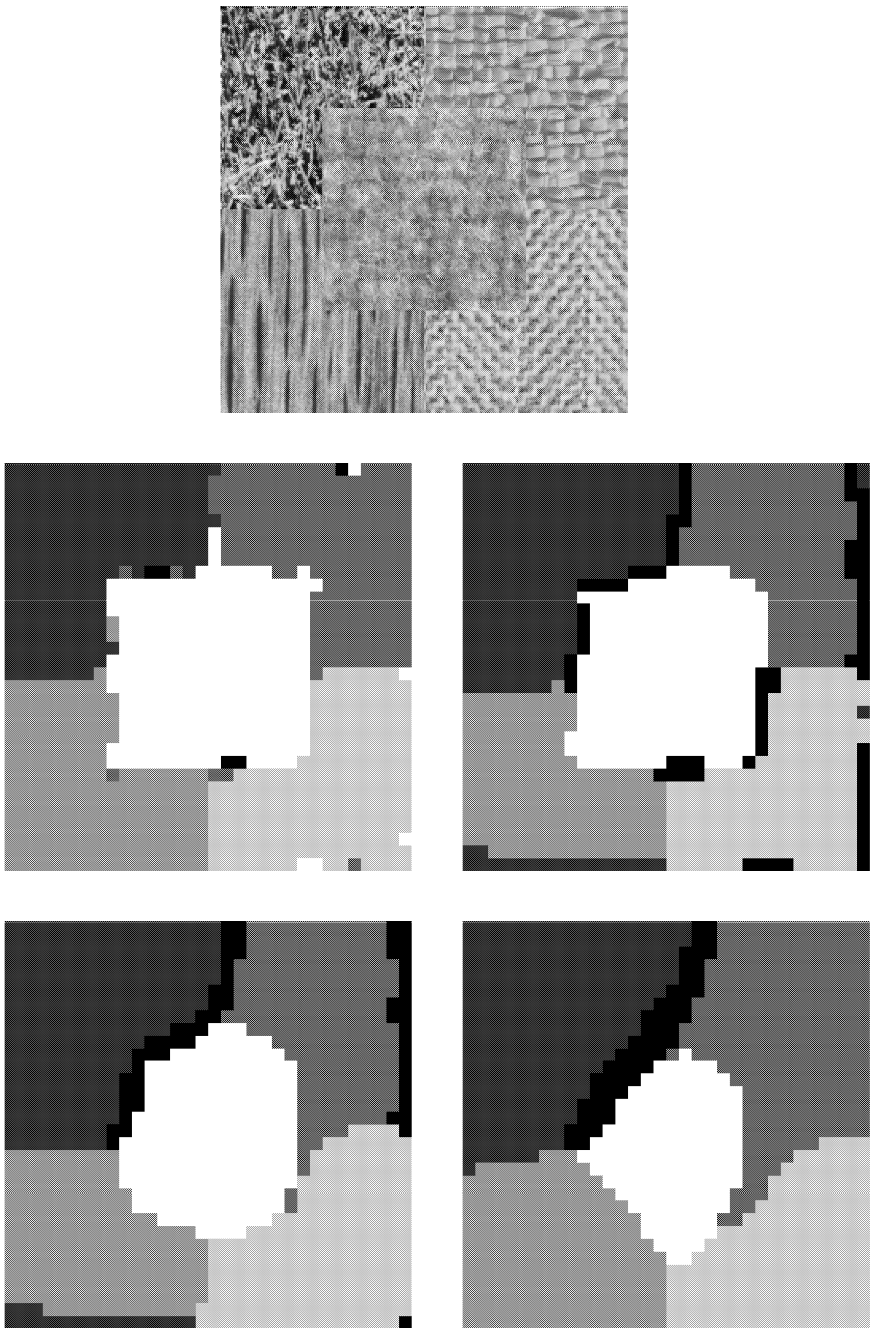


Figure 5: **Top:** Texture mosaic consisting of grass (D9), raffia (D84), wood (D68), herringbone (D17), and wool (D19, inset). **Rows 2 and 3:)** Classification results, at four levels of blurring, following training on the five textures in the mosaic, as well as on a sixth texture (sand, D29).

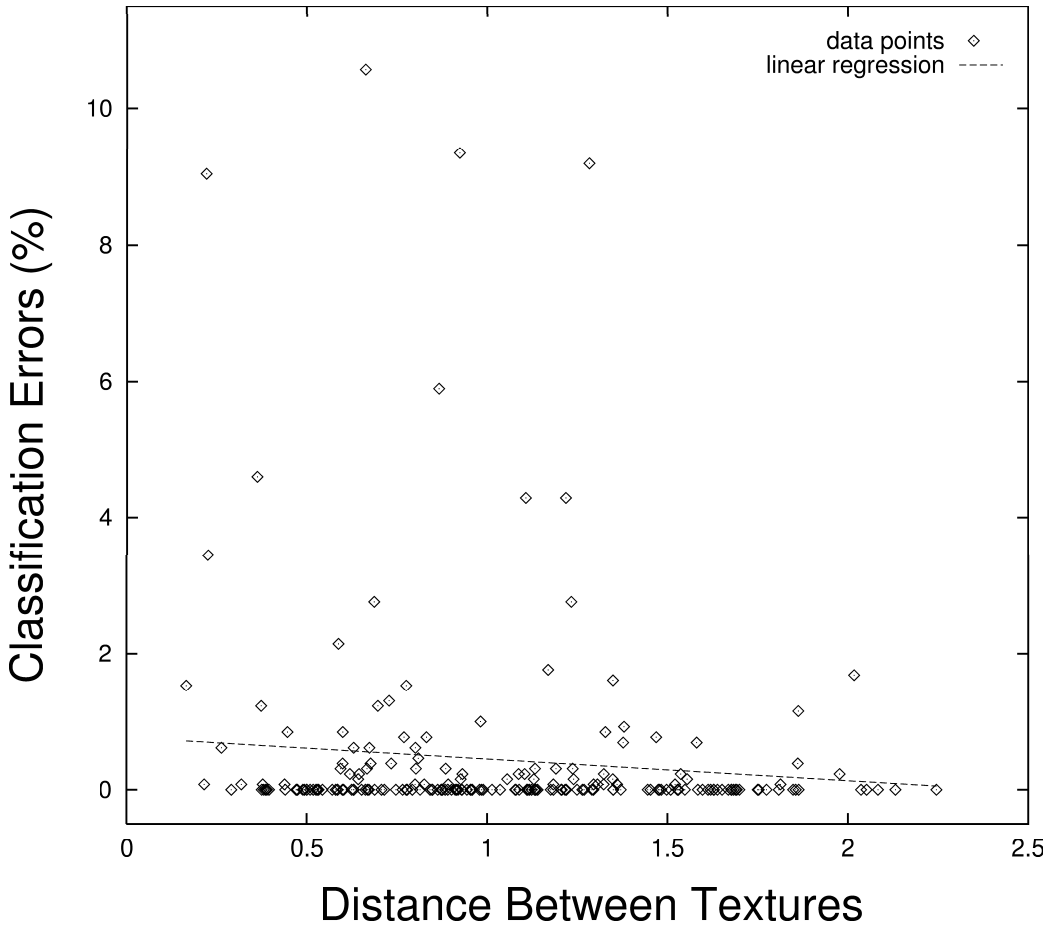


Figure 6: The percentage of all errors due to confusion between pairs of textures is plotted as a function of the distance in MDS coordinates between the textures (see Rao & Lohse, 1993, 1996). The data set consists of the following 21 textures: D3, D9, D10, D11, D15, D18, D26, D29, D34, D50, D52, D55, D57, D73, D78, D80, D82, D83, D86, D87, D93, D110 (see Figure 3).

confusability are not identical measures of performance. Figure 6 plots the confusion errors between each pair of textures as a function of their distance in MDS coordinates after training ARTEX with 768 samples/class for 2 epochs.

8 Classifying SAR Image Regions

ARTEX was also evaluated on classification of textured regions in real-world synthetic aperture radar (SAR) images at single-pixel resolution. We are grateful to Allen Waxman of MIT Lincoln Laboratory for making these SAR images available. SAR textures can vary gradually and stochastically across space, and exhibit a great deal of speckle and drop-out of image pixels. This is the type of problem that our brains need to solve when they are confronted by the noisy images created by retinal photoreceptors. Our simulations illustrate how the types of processes that have evolved to cope with biologically occurring

noise and pixel drop-out work just as well with man-made sensors. Indeed, we propose that human observers who become expert in interpreting SAR images use similar biological mechanisms to the ones that we report herein.

The SAR images were obtained using a 35-GHz synthetic aperture radar with 1 foot by 1 foot resolution and a slant range of 7 km (Novak *et al.*, 1990). We have not found any classification benchmarks on SAR imagery of sufficiently high resolution to provide meaningful comparisons to our results. The images were taken of upstate New York scenery, and contain four region types—grass, trees, roads, and radar shadows—that we trained the system to classify. We selected nine 512x512 SAR images that contain large amounts of these four regions, and hand-labeled them with the help of optical photographs of the scenes. The labels, from dark gray to white, correspond to radar shadows, roads, grass, and trees, respectively. For computational tractability, the images were reduced via grey-level consolidation from their original size of 512x512 pixels to 200x200 pixels. Following the feature extraction steps, the outer 10 pixels from each image were disregarded in order to avoid border effects. Therefore, only the interior 180x180 pixel area of the images will be shown.

Figure 7 (top left) shows the output of Stage 1 of ARTEX (see Figure 2) at the third spatial scale, for one of these images. It converts five orders of magnitude of power in the radar return into a normalized image that preserves the (Weber-law modulated) ratio contrast of the original. Substantial multiplicative noise remains, however. Figure 7 (top middle) shows the Stage 8 BCS boundaries. They are far less precise than those achievable using a CC Loop (see Grossberg, Mingolla, & Williamson, 1995). Figure 7 (top right) shows the Stage 9 filled-in brightness feature that is organized by these boundaries. Note that the surface brightness representation smooths out the noise in a form-sensitive way. Figure 7 (middle left) shows the hand-labeled class labels of the four region types for this image.

This SAR classification problem requires accurate classification of the region interiors as well as many region transitions. Unlike the texture mosaic problem described above, this problem involves training on the same types of images that are used for testing. Like texture mosaics, SAR images contain many region transitions. In addition, the hand-labeled region classes are rather crude, and, at single-pixel resolution, there is no spatial averaging to reduce the variance of features within regions. Therefore, this problem requires learning an extremely noisy mapping from the set of input features to the region labels.

Before evaluating ARTEX, we first analyze the discriminability of the image regions based on the surface brightness feature (Stage 9 in Figure 2) in order to clarify the utility of using surface brightness as compared with using the outcome of center-surround processing (Stage 1 in Figure 2). Figure 8 (top) shows the brightness distributions of the four region types following only Stage 1 center-surround processing. As these histograms show, a great deal of overlap exists between the region types. Figure 8 (bottom) shows the distributions of the Stage 9 filled-in brightness outputs. This figure quantifies how

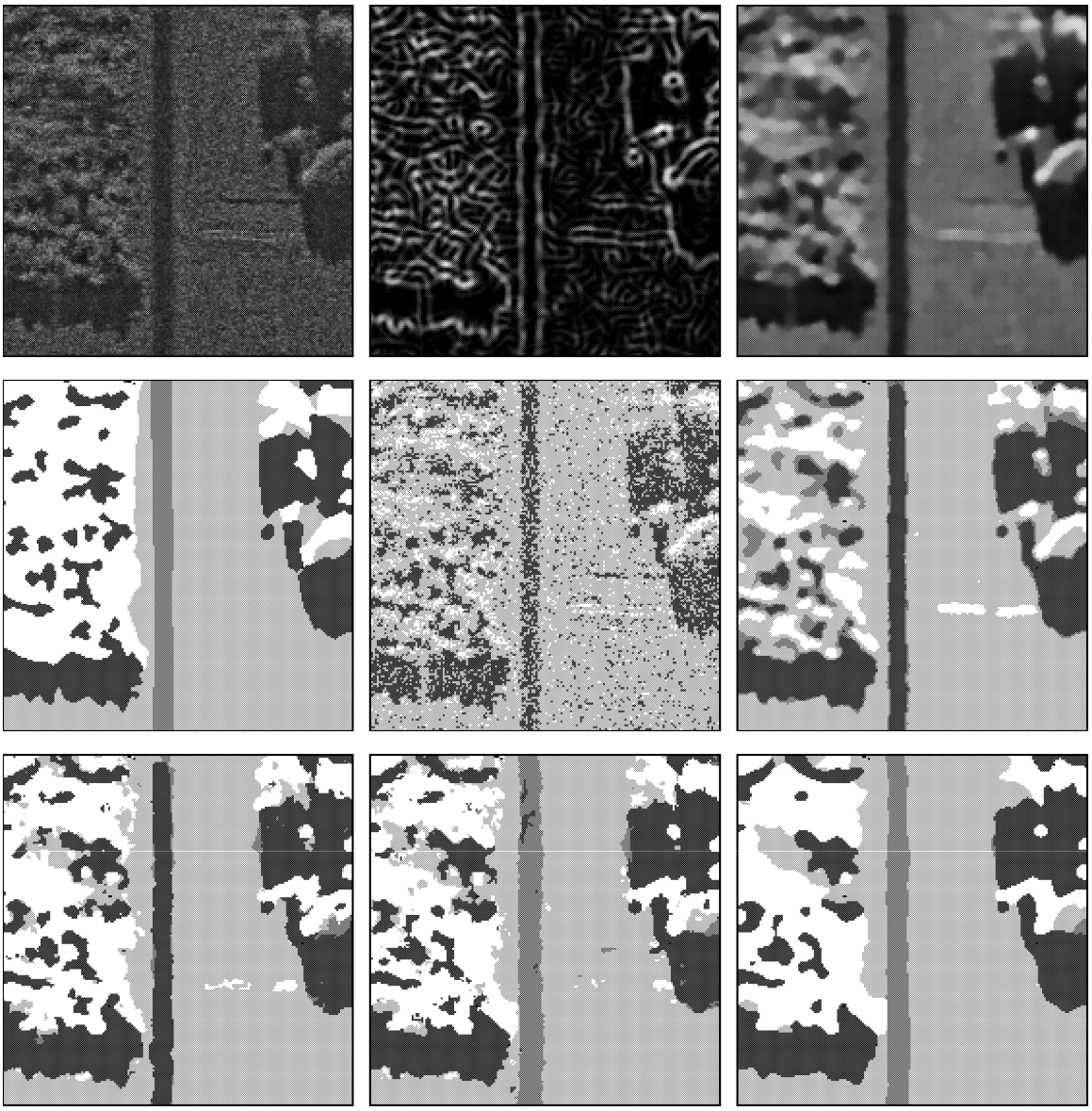


Figure 7: Results are shown on a 180x180 pixel SAR image, which is one of nine images in data set. Top row: Stage 1 output (left); Stage 8 BCS boundaries (middle); Stage 9 FCS filled-in output (right). Middle row: Hand-labeled classes for SAR regions. From dark to lightest, regions are radar shadows, roads, grass, and trees (left). Classification using the Stage 1 center-surround feature and a Gaussian classifier yields 57.8% correct (middle). Classification using the Stage 9 surface brightness feature and a Gaussian classifier yields 71.4% correct (right). Bottom Row: GAM classification using all 17 features. 81.7% correct using OV representation (left). 82.9% correct using OI representation (middle). 85.9% correct using OI representation, with filled-in probability estimates.

SAR Classification

Configuration	Total	Shadow	Road	Grass	Tree
Stage 1 Feature	57.8	58.7	0.0	87.5	21.7
Stage 9 Feature	71.4	71.6	21.0	93.6	50.5
ARTEX OV (no voting)	80.8	76.6	62.5	88.0	79.5
ARTEX OV (voting)	81.7	78.0	62.4	88.9	80.3
ARTEX OI (no voting)	81.9	76.5	68.9	88.6	78.7
ARTEX OI (voting)	82.9	78.4	69.6	89.4	79.4
ARTEX FP (no voting)	85.0	79.5	72.6	91.3	82.8
ARTEX FP (voting)	85.9	80.1	72.2	91.6	82.6

Table 3: Classification results on SAR images for different configurations. Left column shows net classification rate, with remaining columns showing breakdown in the four individual region types. The first two rows show results (using a Gaussian classifier) based on a single brightness feature, the Stage 1 center-surround feature (1st row) and the Stage 9 filled-in feature (2nd row). The remaining rows show the classification results of different ARTEX configurations, with and without voting. ARTEX OV is ARTEX with an orientation variant representation, ARTEX OI is ARTEX with an orientation invariant representation, and ARTEX FP uses the ARTEX OI region probability estimates by filling them in within the BCS boundaries.

brightness helps to separate input features in a natural scene. This result is made intuitively clear by comparing the Stage 1 image in Figure 7 (top left) with the Stage 9 image in Figure 7 (top right). The latter image is much clearer looking and more pleasing to the eye, even though the boundaries that organize it are rather coarse.

The usefulness of the surface brightness processing is further elucidated by comparing classification rates based on only the Stage 1 and Stage 9 features. These unimodally distributed data were classified using a Gaussian classifier, in which the distribution for each region type was represented by a single Gaussian distribution. The result for Stage 1 is shown in the middle image of Figure 7 and in the middle, right image of Figure 7. Quantitative performance measures are listed in Table 3. These results quantify the usefulness of FACADE preprocessing, particularly in overcoming frequent misclassifications due to multiplicative noise, and also provide a baseline for evaluating the effectiveness of the complete image classification system, which also uses a multiscale oriented filter, and GAM rather than a Gaussian classifier.

GAM was trained and tested on the nine images using a leave-one-out method at the level of images (i.e., test each of the 9 images after training on the other 8 images) to ensure independence between testing and training image data. All image pixels were used for training and testing in one study. This result was compared to results obtained by training with as little as 0.01% of the training set. A total of about 260,000 training

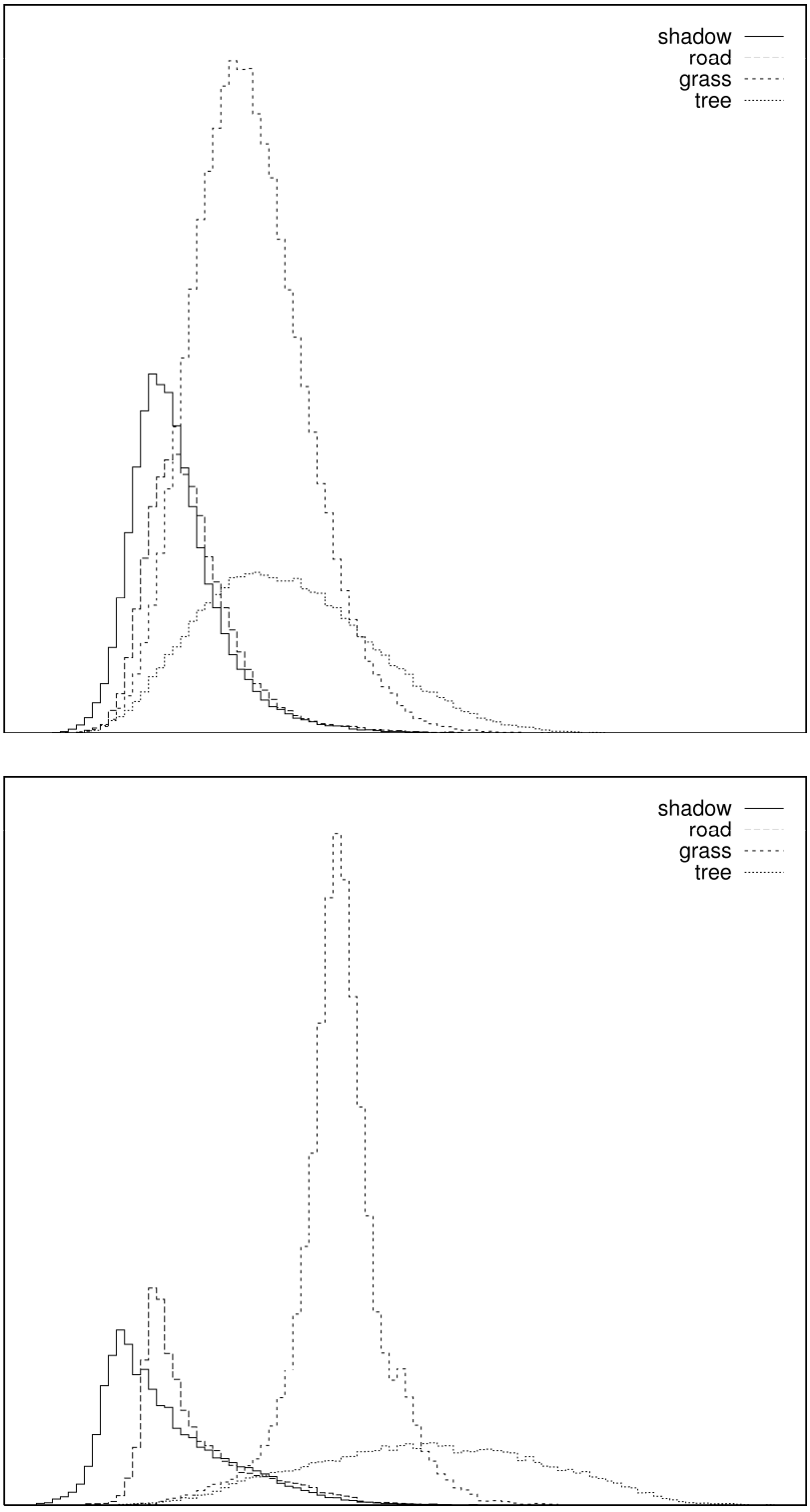


Figure 8: Brightness distributions of four region types: shadows, roads, grass, and trees. Top: Stage 1 center-surround output. Bottom: Stage 9 filled-in output. BCS/FCS processing effectively separates regions.

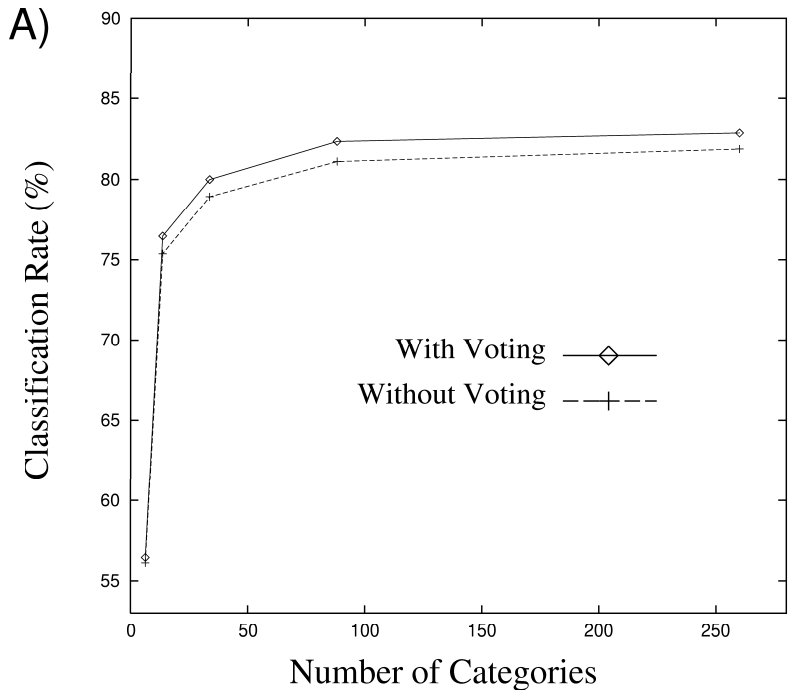
samples were used on a single training epoch. Five GAM networks were trained on independent orderings of the data. The classification rates attained on these five runs were averaged. In addition, *voting* was done among these five systems. Voting involves averaging the probability estimates among GAM networks trained on different random orderings of the training data, and choosing the class prediction with the highest average probability estimate.

Orientationally Variant (OV) Representation. First, results obtained without orientational invariance of the BCS filter are reported. GAM self-organized on average 285.3 categories. The classification result (with voting) is displayed in Figure 7 (bottom left). The non-voting and voting results are quantified in Table 3. With voting, the net classification rate is slightly improved, from 80.8% to 81.7% correct. The improvement in classification rate with voting must be weighed against the cost of using n voters. Here $n=5$, which entails 5 times more categories and training epochs. The main problem with the orientationally variant representation can be seen in how the roads are classified. For example, the thin vertical road is misclassified in the central image, presumably because the system was not trained on any thin vertical roads.

Orientationally Invariant (OI) Representation. With the orientational invariance step of the BCS filter included, GAM self-organized 260.0 categories. This represents 65 categories per output class, and a compression of 1000:1 from training samples to categories. The classification result (with voting) is displayed in Figure 7 (bottom middle). The non-voting and voting results are also listed in Table 3. With voting, the net classification rate is slightly improved, from 81.9% to 82.9% correct. Note that the classification errors on the thin vertical road are corrected since any orientation during training can generalize to any other orientation during testing.

Speed of Training. Good results are also obtained after training with much fewer samples. This is demonstrated in Figure 9A, which plots performance, with and without voting, after training with randomly selected subsets of the training set. From left to right, the plotted points correspond to training with 0.01%, 0.1%, 1%, 10%, and 100% of the training set. For each of these points, the number of self-organized categories (abscissa) and the classification rate (ordinate) are shown. Note that with 0.1%–10% of the training set, GAM obtains good performance (75–82% correct) using very few (13–88) categories.

Diffusing Probability Estimates. The probability estimates obtained with the OI representation and voting make good confidence measures because they predict reasonably well the probability that a prediction is correct. This suggests that each probability estimate should be weighted equally in any further operations that combine the estimates across



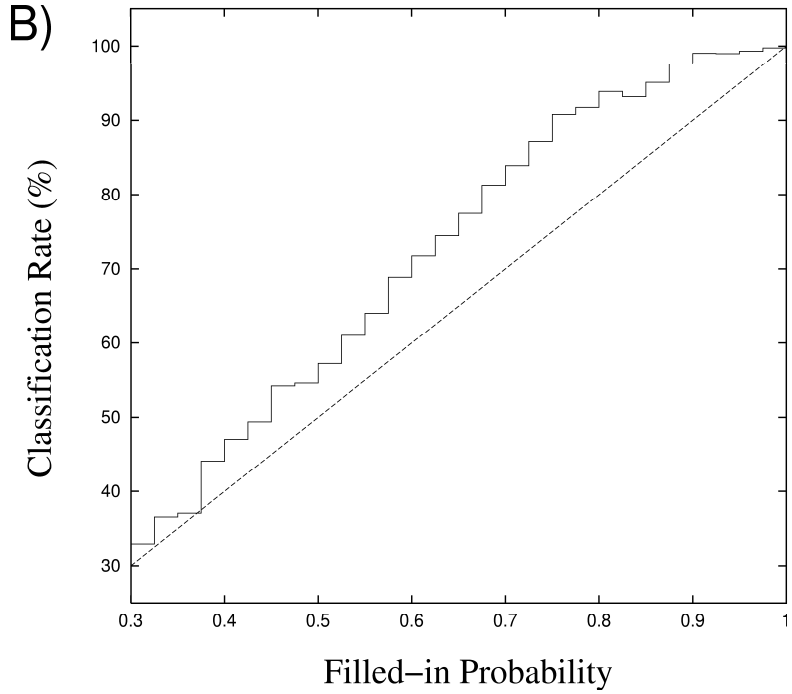


Figure 9: **A)** The number of self-organized categories (abscissa) and the classification rate (ordinate) are plotted for GAM, with and without voting, trained on different sized subsets of the training data. From left to right, the points correspond to training with 0.01%, 0.1%, 1%, 10%, and 100% of the training set. **B)** The accuracy of confidence measures is shown by plotting classification accuracy as a function of filled-in probability estimates in SAR images. The system's confidence measures are reasonably close to the ideal confidence measures represented by the dashed diagonal line.

space. One such operation is spatial averaging, which has the disadvantage of mixing probability estimates between different regions.

A better way to combine estimates is to take advantage of the information contained in the BCS boundaries (Figure 7, top middle), in order to maximize spatial averaging within regions while minimizing it between regions. This can be done by diffusing the probabilities within the BCS boundaries, in the same way that brightness estimates are diffused in Stage 9, in order to obtain diffused probabilities. See Appendix IV for details. Figure 7 (bottom right) shows the decision regions following diffusion of probability estimates. With probability diffusion, classification performance on all nine SAR images was improved from 82.9% to 85.2% correct. These results are also listed in Table 3. Figure 9B shows the accuracy of the filled-in probability estimates as confidence measures, plotting classification accuracy as a function of the probability estimate of the chosen region. This plot approximates that of an ideal confidence measure (diagonal line).

Further improvement in accuracy could be achieved by using cognitive comparisons between contiguous region types; for example, the fact that shadows do not occur below roads or grass could be used to improve classification of these regions. Such excursions into cognitive mechanisms go beyond the scope of the present study.

9 Concluding Remarks

The ARTEX system demonstrates the utility of combining neural models for visual perception that are based on FACADE theory with neural models for adaptive categorization and prediction that are based on Adaptive Resonance Theory. ARTEX extracts multiple-scale oriented contrast features and a filled-in surface feature that provide an informative context-sensitive representation of the textural and brightness properties of an image. ARTEX then incrementally learns an internal categorization of these features along with a mapping from multiple categories to the labels of the output classes. The model also learns class probabilities, which may be filled-in to yield surface probability maps that improve classification accuracy. ARTEX outperforms other leading texture-based classifiers on a variety of texture classification benchmarks, and provides good classification at single-pixel resolution on noisy SAR images whose intensities vary over 5 orders of magnitude.

Given the success of ARTEX in the domain of spatially localized scene recognition, it is interesting to compare it with approaches in the more large-scale and challenging domain of shape and object recognition. Two popular competing approaches are recognition by components (RBC) (Biederman, 1987; Hummel & Biederman, 1992), and memory-based recognition (Edelman & Poggio, 1992; Edelman, 1996). RBC posits the formation of an intermediate representation prior to shape or object recognition. This intermediate representation consists of a structural description of an object, made up of volumetric primitives called geons, and their spatial relations. The primary (and in our view, correct) criticism

of RBC has been that recovering useful volumetric primitives is often impossible given the complexity of real-world objects and the noisiness of real-world images. In addition, some of the key psychophysical data that motivated the Geon concept concerned how humans better recognize line drawings with deleted segments when the drawings contain recoverable features that the brain can restore using amodal illusory contours, than when they do not (Biederman, 1987). Grossberg (1987, Section 20) provided a FACADE theory explanation of these data by suggesting how and why amodal illusory contours would form in the recoverable case, before this completed boundary representation inputs to an ART classifier for correct recognition. Geons played no role in this explanation.

The alternative memory-based approach posits a direct mapping from "low-level" features to "high-level" internal categories. Invariance to 3-D rotation is obtained by interpolating between multiple categories that represent different aspects, or 2-D projections, of 3-D objects. ARTEX is consistent with a memory-based approach. Therefore, it is useful to compare ARTEX to a specific memory-based model, such as that outlined by Edelman (1996). This model consists of a stage of image filtering followed by a mapping, via radial basis functions (RBFs), into a distributed representation of internal categories that represent stored views of objects. The RBF model uses analog-valued training signals and learns via a matrix inversion or gradient descent algorithm. It has no specified mechanism for learning how many basis functions to use. ART models, in contrast, construct and learn internal categories in a generally unsupervised manner, receiving only a limited, biologically plausible, type of supervised feedback; namely, if a supervised ART system makes an incorrect prediction, then its active representation is reset, and its vigilance is raised.

Three self-organizing view-based ART models have already been developed and benchmarked, the Aspect Network model of Seibert and Waxman (1992), the VIEWNET model of Bradski and Grossberg (1995), and the ART-EMAP model of Carpenter and Ross (1995). The present work on ARTEX shows how to develop a texture-sensitive front end for such models, and how to use a GAM classifier, which is also based on RBFs, to learn a distributed representation of individual 2-D views, before these representations are joined together, using a working memory for temporal evidence accumulation, into 3-D object recognition categories.

References

Arrington, K. (1994). The temporal dynamics of brightness filling-in. *Vision Research*, 34, 3371-3387.

Baloch, A. & Grossberg, S. (1997). A neural model of high-level motion processing: Line motion and formotion dynamics. *Boston University Technical Report*, CAS/CNS-TR-96-020. *Vision Research*, in press.

Beck, J. (1994). Interference in the perceived segregation of equal-luminance element-arrangement texture patterns. *Perception and Psychophysics*, 56, 424-430.

Biederman, I. (1987). Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94, 115-147.

Biederman, I. & Ju, G. (1988). Surface versus edge-based determinants of visual recognition. *Cognitive Psychology*, 20, 38-64.

Biederman, I. & Hummel, J.E. (1992). Dynamic binding in a neural network for shape recognition. *Psychological Review*, 99, 480-517.

Bergen, J.R. (1991). Theories of visual texture perception, in *Spatial Vision*, D.M. Regan Ed. New York: Macmillan, 1991, 114-134.

Bergen, J.R., & Landy, M.S. (1991). Computational modeling of visual texture segregation, in *Computational models of visual processing*, M. S. Landy and J. A. Movshon, Eds. Cambridge, Mass: MIT Press, 1991, 253-271.

Bergen, J.R., & Adelson, E.H. (1988). Early vision and texture perception. *Nature*, 333, 363-364.

Bovik, A.C., Clark, M., & Geisler, W.S. (1990). Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, 55-73.

Bradski, G. & Grossberg, S. (1995). Fast learning VIEWNET architectures for recognizing 3-D objects from multiple 2-D views. *Neural Networks*, 8, 1053-1080.

Brodatz, P. (1966). *Textures*. New York: Dover, 1966.

Caelli, T.M. (1985). Three processing characteristics of visual texture segmentation. *Spatial Vision*, 1, 19-30.

Caelli, T.M. (1988). An adaptive computational model for texture segmentation. *IEEE Transactions on Systems, Man and Cybernetics*, 18, 9-17.

Carpenter, G.A. & Grossberg, S. (1987). A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37, 54-115.

Carpenter, G.A. & Grossberg, S. (Eds.), *Pattern recognition by self-organizing neural networks.* Cambridge, MA: MIT Press, 1991.

Carpenter, G.A. & Grossberg, S. (1993). Normal and amnesic learning, recognition, and memory by a neural model of cortico-hippocampal interactions. *Trends in Neurosciences*, 16, 131-137.

Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J., & Rosen, D.B. (1992). Fuzzy ARTMAP: A neural network architecture for incremental learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*, 3, 698-713.

Carpenter, G.A., Grossberg, S., & Reynolds, J. (1991). ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network. *Neural Networks*, 4, 565-588.

Carpenter, G.A. & Ross, W.D. (1995). ART-EMAP: A neural network architecture for object recognition by evidence accumulation. *IEEE Transactions on Neural Networks*, 6, 805-818.

Cavanagh, Patrick (1997). Direct recognition. *Proceedings of the International Conference on Vision, Recognition, Action: Neural Models of Mind and Machine*, 3, Department of Cognitive and Neural Systems, Boston, MA.

Chapman, K.L., Leonard, L.B., & Mervis, C.B. (1986). The effect of feedback on young children's inappropriate word usage. *Journal of Child Language*, 13, 101-107.

Christ, R.E. (1975). Review and analysis of color coding research for visual displays. *Human Factors*, 17, 562-570.

Clark, E.V. (1973). What's in a word? On the child's acquisition of semantics in his first language. In T.E. Moore (Ed.), *Cognitive Development and the Acquisition of Language*, pp. 65-110. New York: Academic Press.

Cohen, M. & Grossberg, S. (1984). Neural dynamics of brightness perception: Features, boundaries, diffusion, and resonance. *Perception and Psychophysics*, 36, 428-456.

Cruthirds, D., Mingolla, E., & Grossberg, S. (1993). Emergent groupings and texture segregation. *Investigative Optthalmology & Visual Science Suppl.,* 2623.

Davidoff, J.B. (1991). *Cognition through color.* Cambridge, MA: MIT Press.

Davidoff, J.B. & Donnelly, N. (1990). Object superiority: A comparison of complete and part probes. *Acta Psychologica*, 73, 225-243.

Dempster, A.P., Larid, N.M., & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Satistical Society Series B*, 39, 1-38.

Duda, R.O. & Hart, P.E. (1973). *Pattern Classification and Scene Analysis.* New York: John Wiley.

Edelman, S. (1996). Receptive fields for vision: From hyperacuity to object recognition. In R.J. Watt, (Ed.). *Vision*, 1996.

Edelman, S. & Poggio, T. (1992). Bringing the grandmother back into the picture: A memory-based view of object recognition. *International Journal of Pattern Recognition and Arificial Intelligence*, 6, 37-61.

Elder, J.H & Zucker, S.W. (1998). Evidence for boundary-specific grouping. *Vision Research*, 38, 143-152.

Francis, G. & Grossberg, S. (1996). Cortical dynamics of form and motion integration: Persistence, apparent motion, and illusory contours. *Vision Research*, 36, 149-173.

Fogel, I. & Sagi, D. (1989). Gabor filters as texture discriminator. *Biological Cybernetics*, 61, 103-113.

Ghahramani, Z. & Jordan, M.I. (1994). Supervised learning from incomplete data via an EM approach. In Cowan, J.D., Tesauro, G., and Alspector, J. (Eds.). *Advances in Neural Information Processing Systems 6.* Morgan Kaufmann Publishers, San Francisco, CA, 1994.

Goodman, R.M., Higgins, C., Miller, J., & Smyth, P. (1992). Rule-based networks for classification and probability estimation. *Neural Computation*, 4, 781-804.

Gove, A., Grossberg, S. & Mingolla, E. (1995). Brightness perception, illusory contours, and corticogeniculate feedback. *Visual Neuroscience*, 12, 1027-1052.

Graham, N., Beck, J., & Sutter, A. (1992). Nonlinear Processes in Spatial-frequency Channel Models of Perceived Texture Segregation: Effects of Sign and Amount of Contrast. *Vision Research*, 32, 719-743.

Greenspan, H. (1996). Non-parametric texture learning. In *Early Visual Learning*, S. Nayar and T. Poggio (Eds.), Oxford University Press, 1996.

Greenspan, H., Goodman, R., Chellappa, R., & Anderson, C.H. (1994). Learning texture discrimination rules in a multiresolution system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16, 894-901.

Grossberg, S. (1980). How does a brain build a cognitive code? *Psychological Review*, 87, 1-51.

Grossberg, S. (1983). The quantized geometry of visual space: The coherent computation of depth, form, and lightness. *Behavioral and Brain Sciences*, 6, 625-657.

Grossberg, S. (1987). Cortical dynamics of three-dimensional form, color, and brightness perception, I: Monocular theory. *Perception and Psychophysics*, 41, 87-116.

Grossberg, S. (1994). 3-D vision and figure-ground separation by visual cortex. *Perception and Psychophysics*, 55, 48-120.

Grossberg, S. (1995). The attentive brain. *American Scientist*, 83, 438-449.

Grossberg, S. (1997). Cortical dynamics of 3-D figure-ground perception of 2-D pictures. *Psychological Review*, 104, 618-658.

Grossberg, S. & Merrill, J.W.L. (1996). The hippocampus and cerebellum in adaptively timed learning, recognition, and movement. *Journal of Cognitive Neuroscience*, 1996, 8, 257-277.

Grossberg, S. & Mingolla, E. (1985). Neural dynamics of perceptual grouping: Textures, boundaries, and emergent segmentations. *Perception and Psychophysics*, 38, 141-171.

Grossberg, S. Mingolla, E., & Ross, W. (1997). Visual brain and visual perception: How does the cortex do perceptual grouping? *Trends in Neurosciences*, 20, 106-111.

Grossberg, S., Mingolla, E., & Williamson, J. (1995). Synthetic aperture radar processing by a multiple scale neural system for boundary and surface representation. *Neural Networks*, 8, 1005-1028.

Grossberg, S. & Pessoa, L. (1997). Texture segregation, surface representation, and figure-ground separation. *Vision Research*, in press.

Grossberg, S. & Todorović, D. (1988). Neural dynamics of 1-D and 2-D brightness perception: A unified model of classical and recent phenomena. *Perception and Psychophysics*, 43, 241-277.

Gurnsey, R. & Browse, R. (1989). Micropattern properties and presentation conditions influencing visual texture discrimination. *Perception and Psychophysics*, 41, 239-252.

Gurnsey, R. & Laundry, D.S. (1992). Texture discrimination with and without abrupt texture gradients. *Canadian Journal of Psychology*, 46, 306-332.

Harvey, L.O. & Gervais, M.J. (1978). Visual texture perception and Fourier analysis. *Perceptual Psychophysics*, 4, 534-542.

Homa, D., Haver, B., & Schwartz, T. (1976). Perceptibility of shematic face stimuli: Evidence for a perceptual gestalt. *Memory and Cognition*, 4, 176-185.

Jain, A.K. & Farrokhnia, F. (1991). Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24, 1167-1185.

Logothetis, N.K., Pauls, J., & Bulthoff, H.H. (1994). View-dependent object recognition by monkeys. *Current biology* 4, 401.

Malik, J. & Perona, P. (1989). Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America*, 7, 923-932.

McLean, J.P., Broadbent, D.E., & Broadbent, M.H.P. (1983). Combining attributes in rapid serial visual presentation tasks. *Quarterly Journal of Experimental Psychology*, 35, 171-186.

Mial, R.P., Smith, P.C., Doherty, M.E., & Smith, D.W. (1974). The effect of memory color on form identification. *Perception and Psychophysics*, 16, 1-3.

Novak, L., Burl, M., Chaney, R., & Owirka, G. (1990). Optimal processing of polarimetric synthetic-aperture radar imagery. *The Lincoln Laboratory Journal*, 3, 273-290.

Ostergaard, A.L. & Davidoff, J.B. (1985). Some effects of color on naming and recognition of objects. *Journal of Neurophysiology*, 42, 833-849.

Pessoa, L., Beck, J., & Mingolla, E. (1996). Perceived texture segregation in chromatic element-arrangement patterns: high luminance interference. *Vision Research*, 35, 2201-2223.

Pessoa, L., Mingolla, E., & Neumann, H. (1995). A contrast- and luminance-driven multiscale network model of brightness perception. *Vision Research*, 35, 2201-2223.

Price, C.J. & Humphreys, G.W. (1989). The effects of surface detail on object categorization and naming. *Quarterly Journal of Experimental Psychology*, 41A, 797-827.

Rao, A.R. & Lohse, G. (1993). Towards a texture naming system: Identifying relevant dimensions of texture perception. IBM Research Technical Report, No. RC 19140.

Rao, A.R. & Lohse, G. (1996). Towards a texture naming system: Identifying relevant dimensions of texture. *Vision Research*, 36, 1649-1669.

Richards, W. (1979). Quantifying sensory channels: Generalizing colorimetry to orientation and texture, touch, and tones. *Sensory Processes*, 3, 207-229.

Rogers-Ramachandran, D.C. & Ramachandran, V.S. (1998). Psychophysical evidence for boundary and surface systems in human vision. *Vision Research*, 38, 71-77.

Rosch, E. (1975). The nature of mental codes for color categories. *Journal of Experimental Psychology: Human Perception and Performance*, 1, 303-322.

Rubenstein, B.S. & Sagi, D. (1989). Spatial variability as a limiting factor in texture discrimination tasks: Implications for performance asymmetries. *Journal of the Optical Society of America A.*, 7, 1632-1643.

Seibert, M. & Waxman, A.M. (1992). Adaptive 3-D object recognition from multiple views. *IEEE Transactions on Pattern Analysis and Matchine Intelligence*, 14, 107-124.

Smith, C., Carey, S., & Wiser, M. (1985). On differentiation: a case study of the development of the concept of size, weight, and density. *Cognition*, 21, 177-237.

Smith, L.B. & Kemler, D.G. (1978). Levels of experienced dimensionality in children and adults. *Cognitive Psychology*, 10, 502-532.

Spitzer, H., Desimone, R., & Moran, J. (1988). Increased attention enhances both behavioral and neuronal performance. *Science*, 240, 338-340.

Stefurak, D.L. & Boynton, R.M. (1986). Independence of memory for categorically different colors and shapes. *Perception and Psychophysics*, 39, 164-174.

Sutter, A., Beck, J., & Graham, N. (1989). Contrast and Spatial Variables in Texture Segregation: Testing a Simple Spatial-Frequency Channels Model, *Perceptual Psychology*, 46, 312-332.

Treisman, A. & Schmidt, H. (1982). Illusory conjunctions in the perception of objects. *Cognitive Psychology*, 14, 107-141.

Ward, T.B. (1983). Response tempo and separable-integral responding: Evidence for an integral-to-separable processing sequencing in visual perception. *Journal of Experimental Psychology: Human Perception and Performance*, 9, 103-112.

Waxman, A.M., Seibert, M.C., Gove, A., Fay, D.A., Bernardon, A.M., Lazott, C., Steele, W.R., & Cunningham, R.K. (1995). Neural processing of targets in visible, multispectral IR and SAR imagery *Neural Networks*, 8, 1029-1051.

Williamson, J.R. (1996). Gaussian ARTMAP: A neural network for fast incremental learning of noisy multidimensional maps. *Neural Networks*, 9, 881-897.

Williamson, J.R. (1997). A constructive, incremental-learning network for mixture modeling and classification. *Neural Computation*, 9, 1555-1581.

Wilson, H.R. & Bergen, J.R. (1979). A four mechanism model for threshold spatial vision. *Vision Research*, 19, 19-31.

Appendix I

BCS Filter

The BCS filter computes 16 oriented contrast features from 4 scales and 4 orientations.

Stage 1. A shunting on-center off-surround network compensates for variable illumination and computes ratio contrasts in the image (Grossberg, 1983). The output at pixel (i, j) and scale g is

$$a_{ij}^g = \frac{I_{ij} - (G^g * I)_{ij} - \alpha\beta}{\alpha + I_{ij} + (G^g * I)_{ij}}, \tag{1}$$

where I_{ij} is the input to pixel (i, j), and $G^g * I$ denotes the convolution of input matrix $\mathbf{I} = \{I_{ij}\}$ and the Gaussian kernel G^g. Kernel G^g is defined by

$$G_{ij}^g(p, q) = \frac{1}{2\pi\sigma_g^2} \exp[-((i-p)^2 + (j-q)^2)/2\sigma_g^2], \tag{2}$$

with $\sigma_g = 2^g$, for the spatial scales $g = \{0, 1, 2, 3\}$. Parameter $\beta = 0.5$. The value of α is determined by the distribution of pixel intensities in the input image. We used $\alpha = 255$ for natural texture images, which have an amplitude range of pixel amplitudes of 0–255, and $\alpha = 2,000$ for SAR images, which have a range of about 100–110,000.

Stage 2. The output a_{ij}^g of equation (1) is convolved with an odd-symmetric Gabor filter D_k^g defined at four equally spaced orientations k,

$$b_{ijk}^g = (D_k^g * a^g)_{ij}, \tag{3}$$

where the horizontal Gabor filter ($k = 0$) is defined by

$$D_{ij0}^g(p, q) = G_{ij}^g(p, q) \cdot \sin[0.75\pi(j-q)/\sigma_g]. \tag{4}$$

Stage 3. A local measure of orientational contrast is obtained by full-wave rectifying the orientational filter output from (3):

$$c_{ijk}^g = |b_{ijk}^g|. \tag{5}$$

Stage 4. Orientational contrast responses may exhibit high spatial variability. A smooth, reliable measure of orientational contrast spatially pools responses within each orientation:

$$d_{ijk}^g = (G^g * c^g)_{ijk}. \tag{6}$$

Stage 5. This optional *orientational invariance* stage shifts orientational responses at each scale into a canonical ordering. This shift maps the same texture pattern, which may be viewed from different angles, into a canonical pattern of orientational contrast signals.

With wraparound, the d_{ijk} responses are shifted so that the orientation with maximal amplitude is in the first orientation plane:

$$e^g_{ijk} = d^g_{ije} \text{ where } e = [k + \arg\max_f (d^g_{ijf})] \bmod 4. \tag{7}$$

The usefulness of the orientational invariance step in equation (7) is task-dependent, as shown by our simulations.

Appendix II

Boundary and Surface Processing

The third spatial scale ($g = 3$) of the BCS filter is input to additional processing stages which form a boundary segmentation that is used to fill-in a surface representation, as diagrammed in Figure 2. Because the superscript g is constant in the following equations, it is left out.

Stage 6. At each orientation, boundaries are contrast-enhanced using shunting competition:

$$f_{ijk} = \frac{c_{ijk} - d_{ijk}}{\zeta + c_{ijk} + d_{ijk}}, \tag{8}$$

where $\zeta = 0.1$.

Stages 7 and 8. The rectified f_{ijk} are summed across orientation to yield the boundary signals:

$$g_{ij} = \sum_{k=1}^{4} [f_{ijk}]^+, \tag{9}$$

where $[\ ]^+$ is the half-wave rectification operator.

Stage 9. The boundary signals g_{ij} block diffusive filling-in of the discounted signals a_{ij} from (1), and thereby yield a filled-in surface brightness feature h_{ij}. FCS filling-in obeys the diffusion equation,

$$\frac{d}{dt} h_{ij} = -\lambda h_{ij} + \sum_{p,q \in N_{ij}} (h_{pq} - h_{ij}) P_{pqij} + a_{ij}, \tag{10}$$

where diffusion is among the four nearest neighbors, $N_{ij} = \{(i, j-1),\ (i-1, j),\ (i+1, j),\ (i, j+1)\}$, and the boundary-gated permeabilities obey

$$P_{pqij} = \frac{\delta}{1 + \epsilon(g_{pq} + g_{ij})} \tag{11}$$

(Cohen & Grossberg, 1984; Grossberg & Todorović, 1988). At equilibrium, equation (10) yields

$$h_{ij} = \frac{a_{ij} + \sum_{p,q \in N_{ij}} h_{pq} P_{pqij}}{\lambda + \sum_{p,q \in N_{ij}} P_{pqij}}. \tag{12}$$

In our implementation, the equilibrium equation (12) is iterated 1000 times. Parameters are $\lambda = 0.05$, $\delta = 10.0$, $\epsilon = 100.0$.

Appendix III

Gaussian ARTMAP

GAM consists of an input layer, F_1, and an internal category layer, F_2, which receives input from F_1 via adaptive weights. Activations at F_1 and F_2 are denoted, respectively, by $\boldsymbol{x} = (x_1, ..., x_M)$ and $\boldsymbol{y} = (y_1, ..., y_N)$, where M is the dimension of the input space, and N is the number of committed F_2 category nodes. In ARTEX, $\boldsymbol{x}$ is the output vector of the FACADE filter. Each F_2 category models a local density of the input space with a separable Gaussian receptive field, and maps to an output class prediction. The j^{th} category's receptive field is parametrized by two M-dimensional vectors: its mean, $\boldsymbol{\mu}_j$, and standard deviation, $\boldsymbol{\sigma}_j$. A scalar, n_j, represents the amount of training data for which the node has received credit.

Category Match and Activation. The input to category j is determined by the *match* value

$$G_j = \exp\left(-\frac{1}{2}\sum_{i=1}^{M}\left(\frac{x_i - \mu_{ji}}{\sigma_{ji}}\right)^2\right). \tag{13}$$

Function G_j measures how close the input vector $\boldsymbol{x}$ is to the category's mean $\boldsymbol{\mu}_j$, relative to its standard deviation $\boldsymbol{\sigma}_j$. The net input signal to category j is

$$g_j = \frac{n_j}{\prod_{i=1}^{M} \sigma_{ji}} G_j, \tag{14}$$

where $(\prod_{i=1}^{M} \sigma_{ji})^{-1}$ normalizes the Gaussian and n_j is proportional to its *a priori* probability.

A category succeeds in temporarily storing activity in short-term memory only if its match value G_j is large enough to exceed the vigilance parameter ρ. The category is thus stored only if $G_j > \rho$. Otherwise, it is rapidly reset. In addition, the category's stored activity, y_j, is normalized by a shunting competition that occurs across all active categories. That is,

$$y_j = \frac{g_j}{\xi + \sum_{l \in T(\rho)} g_l} \quad \text{if} \quad j \in T(\rho); \qquad y_j = 0 \text{ otherwise}, \tag{15}$$

where $T(\rho)$ is the set of all categories, j, such that $G_j > \rho$. Parameter $\xi = 0.00001$. The activity y_j represents the posterior probability $P(j|\boldsymbol{x})$ for the category given the input vector.

Output Prediction. Equations (13)–(15) describe activation of category nodes in an unsupervised learning Gaussian ART module. The following equations describe GAM's supervised learning mechanism, which incorporates feedback from class predictions made by the F_2 category nodes. When a category, j, is first chosen, it learns a permanent mapping to the output class, C, associated with the current training sample. All categories that map to the same class prediction C belong to the same *ensemble*, $E(C)$. Each time an input is presented, the categories in each ensemble sum their activities to generate a probability estimate, z_C, of the class prediction C that they share:

$$z_C = \sum_{j \in E(C)} y_j. \tag{16}$$

The system prediction, $C_{\max}$, is obtained from the maximum probability estimate by a winner-take-all competition across classes:

$$C_{\max} = \arg\max_C (z_C), \tag{17}$$

which also determines the chosen ensemble. Once the class prediction $C_{\max}$ is chosen, feedback from chosen class $C_{\max}$ to the categories selects those categories that map to the class and suppresses those that do not (Carpenter & Grossberg, 1987). As a result, the category activities y_j are renormalized to values

$$y_j^* = \frac{g_j}{\sum_{l \in T(\rho) \cap E(C_{\max})} g_l} \quad \text{if} \quad j \in T(\rho) \cap E(C_{\max}); \quad y_j^* = 0 \quad \text{otherwise.} \tag{18}$$

Just as y_j represents $P(j|\boldsymbol{x})$, y_j^* represents $P(j|\boldsymbol{x}, C_{\max})$.

Match Tracking. If $C_{\max}$ is the correct prediction, then the network resonates and learns the current input. If $C_{\max}$ is incorrect, then match tracking is invoked. As originally conceived, match tracking involved raising ρ continuously from a baseline value $\overline{\rho}$, thereby causing categories j with $G_j \leq \rho$ to be reset until the correct prediction was selected (Carpenter, et al., 1991). Because GAM uses a distributed representation at F_2, each z_C may be determined by multiple categories, according to (16). Therefore, it is difficult to determine numerically how much ρ needs to be raised in order to select a different prediction. It is inefficient (on a conventional computer) to determine the exact amount to raise ρ by repeatedly resetting the active category with the lowest match value G_j, each time re-evaluating equations (15), (16), and (17), until a new prediction is finally selected.

Instead, a one-shot match tracking algorithm is used. This algorithm involves raising ρ to the average match value of the chosen ensemble:

$$\rho = \exp\left(-\frac{1}{2}\sum_{j=1}^{N} y_j^* \sum_{i=1}^{M}\left(\frac{x_i - \mu_{ji}}{\sigma_{ji}}\right)^2\right). \tag{19}$$

In addition, all categories in the chosen ensemble are reset: $g_j = 0$ for all $j \in E(C_{\max})$. Equations (14)–(17) are then re-evaluated. Based on the remaining non-reset categories, a new prediction $C_{\max}$ in (17), and its corresponding ensemble, are chosen. This search cycle automatically continues until the correct prediction is made, or until all committed categories are reset; namely, $G_j \leq \rho$ for all $j \in \{1, ..., N\}$, and an uncommitted category is chosen. Match tracking assures that the correct prediction comes from an ensemble with a better match to the training sample than all the reset ensembles. Upon presentation of the next training sample, ρ returns to the baseline value: $\rho = \overline{\rho}$.

Learning. GAM learns when it makes a correct output prediction, or a prediction that is not disconfirmed. The F_2 parameters $\boldsymbol{\mu}_j$ and $\boldsymbol{\sigma}_j$ are then updated to represent the sample statistics of the input using local learning rules. When category j learns, n_j is updated to represent the amount of training data for which the j^{th} node has been assigned credit:

$$n_j := n_j + y_j^*. \tag{20}$$

The learning rate for $\boldsymbol{\mu}_j$ and $\boldsymbol{\sigma}_j$ is modulated by n_j, so that the parameters represent the sample statistics of the input:

$$\mu_{ji} := (1 - y_j^* n_j^{-1})\mu_{ji} + y_j^* n_j^{-1} x_i, \tag{21}$$

$$\sigma_{ji}^2 := (1 - y_j^* n_j^{-1})\sigma_{ji}^2 + y_j^* n_j^{-1}(x_i - \mu_{ji})^2, \tag{22}$$

GAM is initialized with $N = 0$. When an uncommitted category is chosen, N is incremented, and the new category, indexed by $j = N$, is initialized with $y_j^* = 1$ and $n_j = 0$, and with a permanent mapping to the correct output class. Learning then proceeds via (20)–(22), with one modification: a constant, γ^2, is added to the right hand side of equation (22), yielding $\sigma_{ji} = \gamma$. Initializing categories with this nonzero standard deviation makes (13) and (14) well-defined. Varying γ has a marked effect on learning: as γ is raised, learning becomes slower, but fewer categories are created. Generally, γ is much larger than the final standard deviation to which a category converges. Intuitively, a large γ represents a low level of certainty for, and commitment to, the location in the input space coded by a new category. As γ is raised, the network settles into its input space representation in a slower and more graceful way. All datasets are preprocessed to have a unit standard deviation in each feature dimension, so that γ has the same meaning in all the dimensions. On the texture classification problems in Section 7, we used $\gamma = 1$. On the noisier SAR classification problems in Section 8, we used $\gamma = 4$.

Appendix IV

Filling-in Output Probabilities

At each image position (i, j), the following diffusion equation is iterated.

$$q_{ijC} = \frac{z_C(i,j) + \sum_{p,q \in N_{ij}} q_{pqC} B_{pqij}}{\lambda + \sum_{p,q \in N_{ij}} B_{pqij}}. \tag{23}$$

Here $z_C(i, j)$ is GAM's probability estimate for output class C, N_{ij} consists of the 4 nearest neighbors to (i, j), and the boundary-gated permeabilities obey

$$B_{pqij} = \frac{\delta}{1 + \epsilon(g_{pq} + g_{ij})}, \tag{24}$$

where g_{ij} is the strength of the boundary computed in (9). Equation (23) is iterated 250 times. Otherwise, the same diffusion parameters are used as in Appendix II. Finally, the diffused values q_{ijc} are normalized to produce the diffused probability estimates,

$$Q_{ijC} = \frac{q_{ijC}}{\sum_d q_{ijd}}. \tag{25}$$

The size of the probability estimate, $z_C(i, j)$, is determined by both the absolute input magnitude to its ensemble, and the input magnitude to its ensemble relative to the input magnitude to other ensembles. The shunting decay parameter ξ in equation (15), which produces partial normalization of activity in the GAM categories, causes the absolute input magnitude to affect the probability estimate. A positive ξ is useful because it prevents the network from producing a large probability estimate when the input magnitude to all the ensembles is very low.