1994-01

Recognition of 3-D Objects from Multiple 2-D Views by a Self-Organizing Neural Architecture

RECOGNITION OF 3-D OBJECTS FROM MULTIPLE 2-D VIEWS BY A SELF-ORGANIZING NEURAL ARCHITECTURE

Gary Bradski and Stephen Grossberg

January 1994

Technical Report CAS/CNS–94–004

To appear in:
V. Cherkassky, J.H. Friedman, and H. Wechsler (Eds.)
From statistics to neural networks:
Theory and pattern recognition
New York: Springer-Verlag, 1994

Boston University Center for Adaptive Systems and
Department of Cognitive and Neural Systems
111 Cummington Street
Boston, MA 02215

Recognition of 3-D Objects from Multiple 2-D Views by a Self-Organizing Neural Architecture

Gary Bradski and Stephen Grossberg

Center for Adaptive Systems and
Department of Cognitive and Neural Systems,
Boston University, 111 Cummington Street,
Boston, Massachusetts 02215 USA

Abstract

The recognition of 3-D objects from sequences of their 2-D views is modeled by a neural architecture, called VIEWNET, that uses View Information Encoded With NETworks. VIEWNET illustrates how several types of noise and variability in image data can be progressively removed while incomplete image features are restored and invariant features are discovered using an appropriately designed cascade of processing stages. VIEWNET first processes 2-D views of 3-D objects using the CORT-X 2 filter, which discounts the illuminant, regularizes and completes figural boundaries, and removes noise from the images. Boundary regularization and completion are achieved by the same mechanisms that suppress image noise. A log-polar transform is taken with respect to the centroid of the resulting figure and then re-centered to achieve 2-D scale and rotation invariance. The invariant images are coarse coded to further reduce noise, reduce foreshortening effects, and increase generalization. These compressed codes are input into a supervised learning system based on the fuzzy ARTMAP algorithm. Recognition categories of 2-D views are learned before evidence from sequences of 2-D view categories is accumulated to improve object recognition. Recognition is studied with noisy and clean images using slow and fast learning. VIEWNET is demonstrated on an MIT Lincoln Laboratory database of 2-D views of jet aircraft with and without additive noise. A recognition rate of 90% is achieved with one 2-D view category and of 98.5% correct with three 2-D view categories.

1 Introduction

This article describes a neural architecture that is capable of learning to recognize 3-D objects in a self-organizing manner. Much prior research on 3-D visual object recognition relies on appearance based approaches. Appearance based approaches use input imagery to construct 3-D object models. Koenderink and van Doorn (1979) created *Aspect Graphs* consisting of 2-D views of a 3-D object along the nodes of the graph, with legal view transitions indicated by the arcs among nodes. In their approach, 2-D views and view transitions are equally important for recognizing the object. Gigus and Malik (1988, 1990) and Plantinga and Dyer (1990) have attempted to automatically construct aspect graphs from objects in a CAD database using convex polyhedra. Other efforts for automatically generating perspec-

tive projection aspect graphs from a CAD database using curved objects and non-convex polyhedra have been pursued by Bowyer, Eggert, Stewman, and Stark (1989), Sripradisvarakul and Jain (1989), Ponce and Kriegman (1990), Rieger (1990), Stewman and Bowyer (1990), Chang and Huang (1992), and Eggert and Bowyer (1993). Hidden Markov models have also been applied to learning an aspect graph from a view sequence by Rimey and Brown (1991).

Seibert and Waxman (1992) have developed a neural network architecture that self-organizes aspect graph representations of 3-D objects from 2-D view sequences. Images of rotating jets were binarized and points of high curvature and the object centroid were found using a reaction-diffusion process. A log-polar transform around the object centroid was used to remove 2-D rotation and scale variations. The result was coarse coded (compressed to 5x5 pixels from 128x128) using Gaussian filters. The coarse codes (25 data points) were fed into an ART 2 (Carpenter and Grossberg 1987) network for clustering and categorization. These "categorical" 2-D views were then fed into a series of cross-correlation matrices, or view graphs, one for each possible 3-D object, so that views and view transitions could be learned by a 3-D object categorization layer. The 3-D categorization layer incorporated "evidence accumulation" nodes which integrate activations that they receive from learned connections to the correlation matrix. The node receiving maximal evidence in the 3-D layer is chosen as the network's recognition of the 3-D object being viewed.

In the Seibert and Waxman model, given N 2-D views and M objects, the architecture must have the potential to encode on order of $M \times N^2$ 2-D view transitions. As reported in Seibert and Waxman (1992), 75% of the 2-D jet images were ambiguous to some degree. That is, 75% of the 2-D view categories formed by ART 2 gave evidence for more than one type of jet. Even if several views are ambiguous, the transitions between them may unambiguously identify a particular 3-D object. Thus, view transitions are critically important in the Seibert and Waxman architecture, which may then incur the cost of needing up to $M \times N^2$ view transitions.

Bradski, Carpenter, and Grossberg (1991) described an alternative architecture that potentially overcomes the problem of proliferating view transitions. It learns to code 2-D views in recognition categories, as do Seibert and Waxman, but stores these categories in a working memory, called a STORE network, whose activity pattern implicitly represents the order in which the 2-D views occurred, as well as the views themselves. An ART module then learns to categorize the stored combination of 2-D views and (implicitly coded) view transitions into a 3-D object category. Such an algorithm needs no more than N+M nodes to code N 2-D views in working memory for M 3-D objects.

This paper further develops the perspective that, although multiple views may facilitate recognition, view transitions, as such, may not be needed to achieve high recognition accuracy from one or more views. A neural net-

work architecture, called **VIEWNET**, for **V**iew **I**nformation **E**ncoded **W**ith **NET**works is proposed that can categorize individual views with high accuracy, in accord with the human experience that many objects can be identified with a single view, except when they are observed from an unfamiliar perspective or from a perspective that reduces the objects apparent dimension. Single view recognition accuracy of up to 90% is achieved by this architecture on the Seibert and Waxman database.

As diagramed in Figure 1, the architecture consists of three parts: an image preprocessor, a supervised self-organizing recognition network, and a network to accumulate evidence over multiple views. It is assumed that the figure to be recognized is separated from its background. Neural networks for figure-ground separation that use computations consistent with those in the preprocessors are described in Grossberg (1993) and Grossberg and Wyse (1992). The image figure is then processed by a feedforward network, called the CORT-X 2 filter (Carpenter, Grossberg, and Mehanian, 1989; Grossberg and Wyse, 1991, 1992) that suppresses image noise while it completes and regularizes a boundary segmentation of the figure. The noise-suppressed boundary segmentation is made invariant under 2-D rotation, translation, and scale invariance by a centering, log-polar, centering operation (Schwartz, 1977). The resulting spectra are coarse coded to gain some insensitivity to 3-D deformation effects and to reduce memory requirements. This coarse-coded, invariant spectrum of a noise-suppressed boundary segmentation defines the input vectors to the self-organizing neural network classifier.

Fuzzy ARTMAP (Carpenter, Grossberg, Markuzon, Reynolds and Rosen, 1992) was used to categorize the output spectra. This architecture is capable of fast, stable learning of recognition categories in response to nonstationary multidimensional data, and of learning to generate many-to-one output predictions from recognition categories to output labels. Erroneous predictions trigger hypothesis testing, or memory search, in the input classifier. Memory search discovers and learns recognition categories that conjointly maximize code compression and minimize predictive error using a mechanism that is called *match tracking*. Fuzzy ARTMAP can hereby use supervised learning to rapidly fit the number, size, and shape of input categories to the statistical demands of the environment. Each category codes a range of target views.

Evidence accumulation stores several learned categories in working memory, and derives decisions from a voting procedure. On the test set, a single view category leads to up to 90% recognition accuracy, voting with two view categories achieves up to 94%, and voting with three view categories up to 98.5%.

2 Data

The image database used to test the architecture described below consists of multiple 2-D images images of three jets [1]. Video images were taken of

[1]Special thanks to Michael Seibert, Alan Waxman and MIT Lincoln Laboratory for their assistance and use of their data.

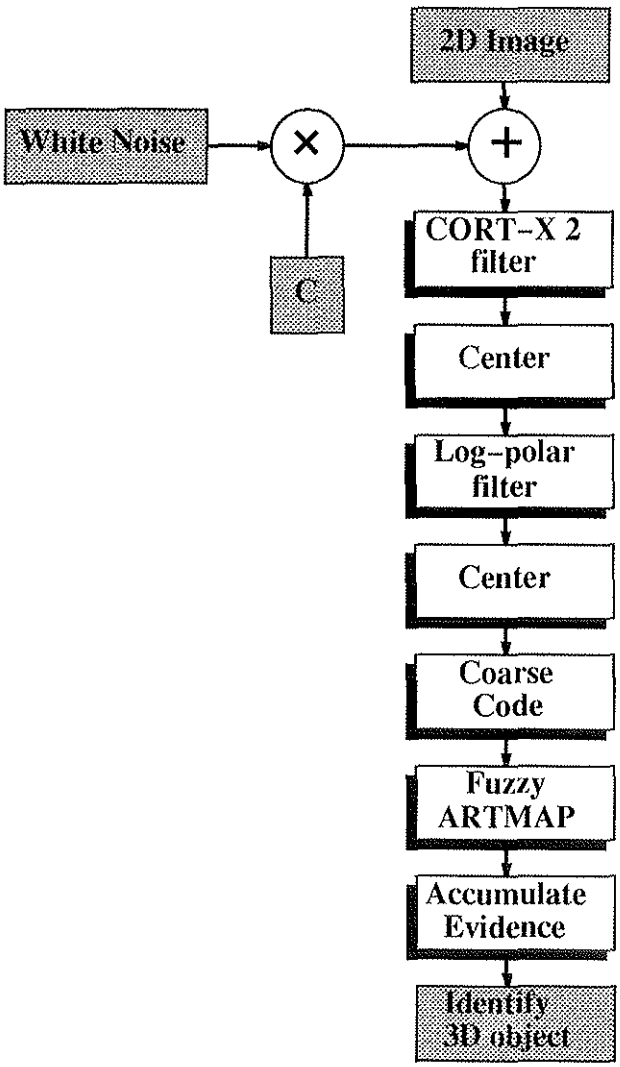


Figure 1: The image processing flow chart of the VIEWNET system from presenting a 2-D image in the image database till the read out of the predicted 3-D object.

3 jet models: an F-16, an F-18, and an HK-1. Each jet was painted black and suspended by string against a light background to facilitate figure-ground separation. The camera was mounted anywhere in an arc around the jets that started at 0.0 degrees above horizontal and went in increments of 4.5 degrees to a maximum of 72.0 degrees above horizontal. For each camera angle, the jets were spun and frames covering one full revolution (an average of 88 frames) were retained resulting in 1200 to 1400 images per object. The images themselves were 128x128 pixel gray scale. The images were then thresholded and binarized into a SUN raster format to form the "raw" database. Data were converted into a floating point format scaled between 0.0 and 1.0 and an additive noise process was introduced. The noise consisted of a 128×128 pixel images with each pixel taken from a uniform distribution between 0.0 and 1.0 scaled by a constant $C \geq 0.0$. These scaled, 128×128 noise images were then added to the 128×128 jet images prior to preprocessing. Thus, both noise-free and noisy 2-D views covering a half-sphere surrounding the 3-D object were collected, keeping their spatial relationships intact.

Even numbered rotation images from each camera angle were taken as the training set with the odd numbered images forming the test set. The system was trained using random walks over the half-sphere of training images. Testing was done using random walks over the half-sphere of test images so that the paths taken and views seen were never the same between the training and test sets.

3 CORT-X 2 filter

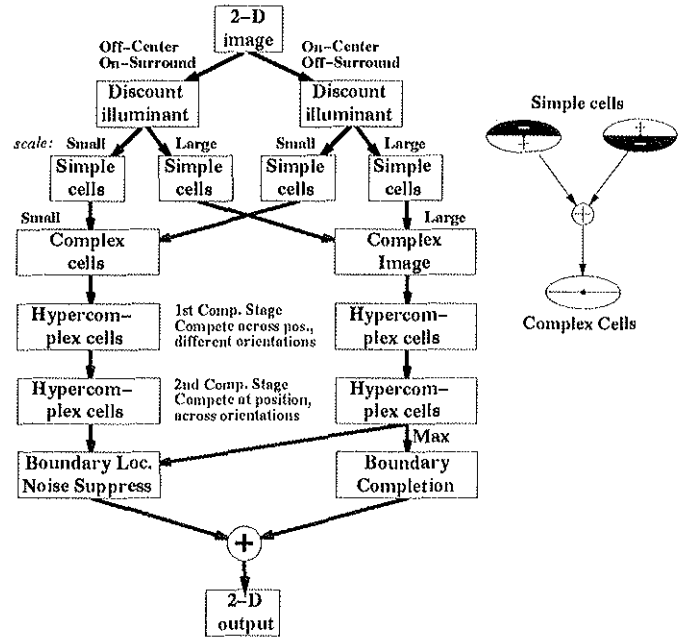
The CORT-X 2 filter (Grossberg and Wyse, 1991, 1992) discounts the illuminant and normalizes image contrasts, regularizes and completes figural boundaries, and suppresses image noise. See CORT-X 2 stages in Figure 2(a), filter kernels in Figure 2(b), and equations in the Appendix.

Step 1. Discount the illuminant. A shunting on-center/off-surround network ("ON-C") and an off-center/on-surround network ("OFF-C") operating on the image in parallel are used to discount variable illumination in the image. The ON-C network has a zero baseline activity and the OFF-C network has a positive baseline activity. The OFF-C filter performs an image inversion. Figure 3 shows a noise-free image as well as the ON-C and OFF-C outputs.

Along straight contrast boundaries in an image, both the ON-C and OFF-C networks enhance the contrast. The ON-C network has a stronger response to concave corners of activity in an image than the OFF-C network, while the converse is true at convex corners (Grossberg and Todorović 1988). These complementary responses are used to build better boundary segmentations, as illustrated below.

Step 2. Boundary Segmentation. Two sets of convolution kernels at two different scales are combined to produce a segmentation that takes advantage of another complementary processing property: larger scales achieve better noise reduction and smaller scales achieve better positional localiza-

(a) CORT–X 2 Flow Chart



(b) CORT–X 2 Filter Kernels

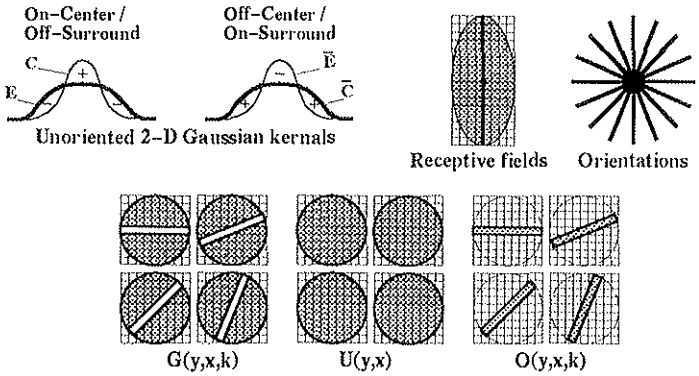


Figure 2: CORT-X 2 flow chart and filter kernels. The image is processed in parallel with small and large scale filters. Grey areas in the kernels are the active regions. All kernels are normalized to have an area equal to one.

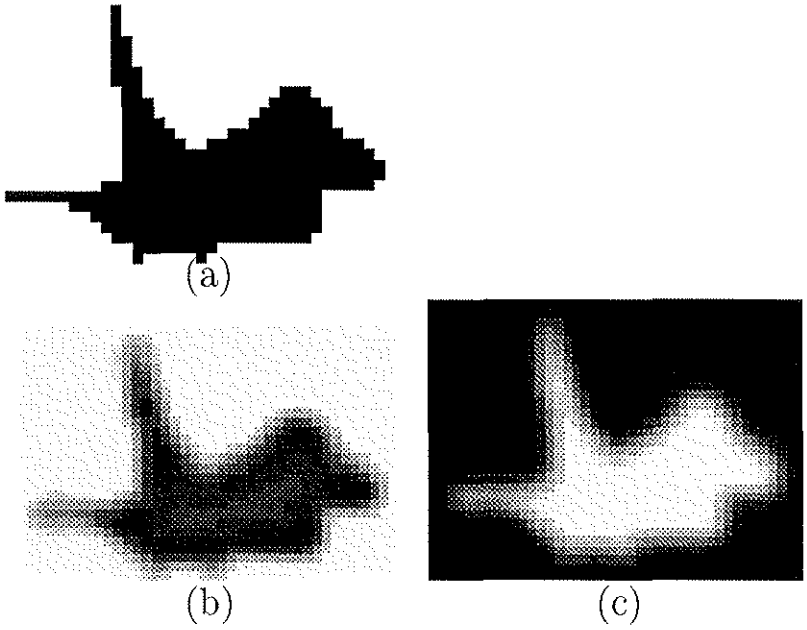


Figure 3: (a) The original F16 image. (b) CORT-X 2 ON-C output. (c) CORT-X 2 OFF-C output.

tion. The ON-OFF and large-small complementary properties are both exploited in the filter. The first stage, called the simple cell layer, consists of oriented contrast detectors that are sensitive to the orientation, amount, direction, and spatial scale of image contrast at a given image location. The orientation sensitivity results from an elliptically shaped kernel, or input field, one for each of eight orientations spaced 45° apart that operate in parallel at each position in the image. Sensitivity to direction-of-contrast results from a kernel in which one half is excitatory and the other half inhibitory. At each orientation, a pair of detectors sensitive to opposite directions-of-contrast processes the image. The net activity of each detector is rectified, giving rise to a half-wave rectified output signal. Figure 4(a) illustrates processing the ON-C and Figure 4(b) the OFF-C image with the small spatial scale (6x3 pixels) simple cell layer. The same thing is done using the larger spatial scale (10x5 pixels) simple cell layer. Lines in the figure indicate the magnitude of the simple cell response at each orientation at each position.

The complex cell layer combines outputs from the simple cells at each position. Complex cells sum up the half-wave rectified outputs of like-oriented simple cells of both directions-of-contrast at each position from the ON-C and

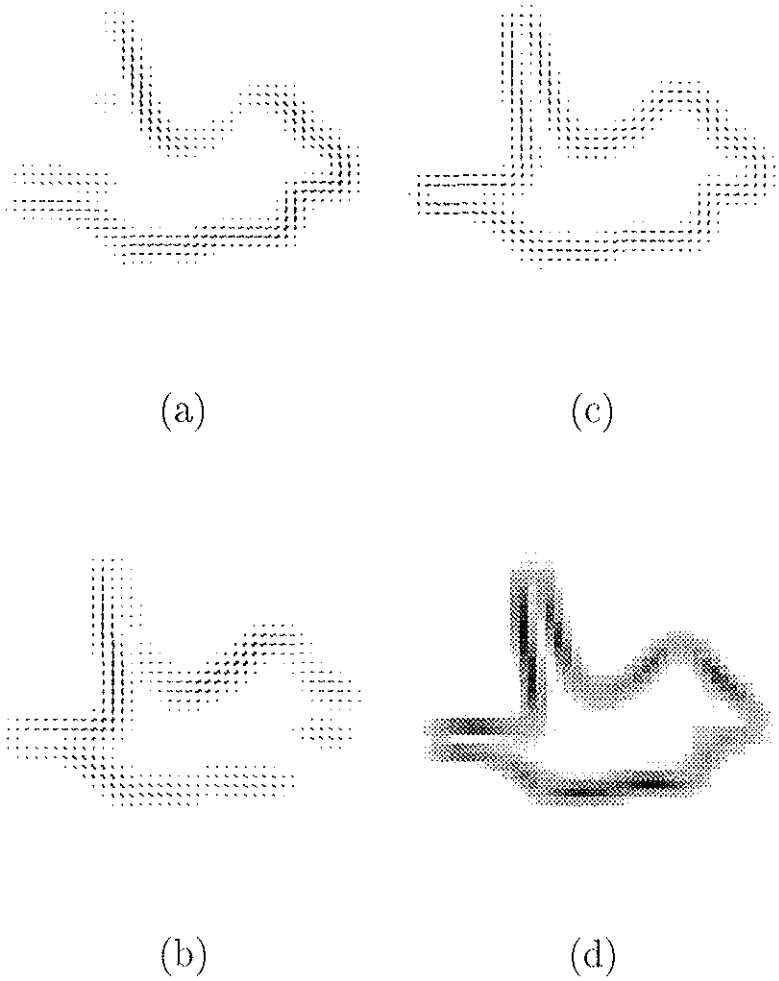


Figure 4: CORT-X 2 processing. (a) Output resulting from the ON-C network, using left sided elliptical filters (simple cell output). A "left sided" filter refers to filters that respond to a vertical left-to-right, high-to-low contrast transition area in the image when the filter is in vertical orientation. A "right sided" filter is the opposite. Lines in the figure are proportional to the magnitude of the response at each orientation at each position. (b) Output from the OFF-C network using left sided elliptical filters. (c) Hypercomplex cell output for the small scale. (d) The final CORT-X 2 output.

OFF-C networks to achieve sensitivity to the orientation, amount, and spatial scale of the contrast in the image, but not to its direction-of-contrast. The complementary deficiencies of the ON-C and OFF-C responses in Figure 4a and 4b are hereby overcome.

Complex cells excite hypercomplex cells in the next layer at their position and orientation while inhibiting hypercomplex cells at nearby locations that are not co-linear with the complex cell's orientation. This interaction is called the first competitive stage. Figure 4(c) shows the output of the hypercomplex cells for the small scale. The next layer, called the second competitive stage, chooses the hypercomplex cell whose orientation is maximally activated to represent the activity at each position.

The final stages of CORT-X 2 involve cooperative interactions between the large and small scale filters. Larger scale filters are better able to complete gaps in image boundaries and to suppress noise, but do a worse job of boundary localization than smaller scale filters. The final CORT-X 2 operations include cooperative interactions between both filter scales that enhance their desirable properties.

Boundary gaps become more likely as the noise in the image increases. To overcome this problem, cooperative interactions among the hypercomplex cells activate an inactive cell if enough cells that share the inactive cell's orientation are active on both sides of its oriented axis. Large and small scales are combined in such a way that the better localization properties of the smaller scale filters have an effect only within regions where the larger scales have located a boundary. Figure 4(d) shows the final CORT-X 2 output consisting of the sum of output of the cooperative and multiple scale interactions. Figure 5 shows the results of processing images with two levels of additive noise: C = 0.5 or 50% noise (a), and C= 1.0 or 100% noise (b).

4 Translation, rotation and scale invariance

The 2-D boundary segmentation is centered by dividing its 1^{st} moments by its 0^{th} moment to find the figure centroid, subtracting off the center of the image and then shifting the figure by this amount. A log-polar transform is then taken with respect to the center of the image. Each point (x, y) is represented as $re^{i\theta}$. Taking the logarithm yields coordinates of log radial magnitude and angle. As is well known (Schwartz, 1977), figural sizes and rotations are converted into figural shifts under log-polar transformation. Using these shift parameters to center the log-polar transformed image leads to a figural representation that is invariant under 2-D changes in position, size and rotation.

5 Coarse coding

Coarse coding reduces memory requirements while compensating for inaccuracies of figural alignment, 3-D viewpoint specific foreshortening, and self-occlusions. On the other hand, too much coarse coding can obscure critical input features and thereby harm recognition performance. The analysis below suggests how to balance these effects to maximize the benefits of coarse

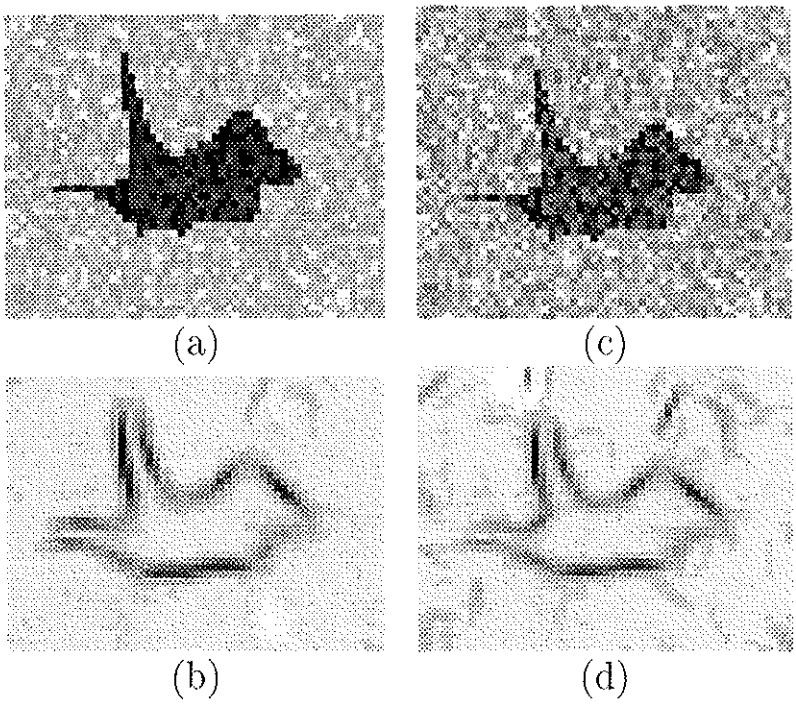


Figure 5: Results of processing noisy images with CORT-X 2. Uniform random noise was added to every pixel in the original image. The original image (left column) had pixels with activity levels between 0.0 and 1.0. Uniform random noise with pixel values ranging between 0.0 and 1.0 was scaled by C and added to the clean image prior to processing by CORT-X 2 with results shown in the right column. In (a,b), random noise between 0.0 and 0.5 ($C = 0.5$ or 50% noise) was added. In (c,d), noise between 0.0 and 1.0 ($C = 1.0$ or 100% noise) was added.

coding.

Coarse coding of the 2-D images used a spatial averaging method. Spatial averaging consists of convolving the original image I with a function Ψ and then sampling the resultant image with delta functions spaced every T pixels: $\delta(x - nT, y - kT)$. For simplicity, in 1-D this is

$$(I * \Psi) \cdot \sum_{n=-\infty}^{\infty} \delta(x - nT). \tag{1}$$

If the Fourier transform of I is $\hat{I}$, and that of Ψ is $\hat{\Psi}$, then the Fourier transform of equation (1) is

$$(\hat{I} \cdot \hat{\Psi}) * \frac{2\pi}{T} \sum_{k=-\infty}^{\infty} \delta(\Omega - k\Omega_s), \tag{2}$$

where $\Omega_s = 2\pi/T$, and T is the sampling period in pixels. If Ω_N is the highest frequency in the image, then for the image to be uniquely determined by its samples, we must have by the Nyquist sampling theorem that

$$\Omega_s = \frac{2\pi}{T} > 2\Omega_N. \tag{3}$$

Two simple spatial averaging functions Ψ are: (1) uniform averaging of the input image so that all pixels in a window of some width are summed and divided by the number of pixels in the window; (2) Gaussian averaging of the input image so that a normalized, Gaussian weighted sum of all pixels is taken over a window of some width. Both approaches were investigated in this paper.

Method (1) has the problem that uniform averaging is a rectangular filter in the space domain and a sinc function in the frequency domain which introduces high frequency aliasing ("ringing") in the resultant image. The Gaussian function of method (2) is a "smoother" low pass filter and so does not suffer from this problem. A Gaussian is also an eigenfunction of a Fourier transform, which simplifies calculation.

To best set the standard deviation σ of the Gaussians, we define two standard deviations away from the Gaussian midpoint to be essentially zero. The cutoff frequency of such a low pass filter is then $\pi/2\sigma$, which by equation (3) yields at equality:

$$\sigma = \frac{T}{2}. \tag{4}$$

Thus, the zero point of each Gaussian just touches the center of the next Gaussian. Figure 6 summarizes the preprocessing: 6(a) shows the output of CORT-X 2, 6(b) the centered log polar transform of (a), 6(c) depicts Gaussian coarse coding according to equation (4), and 6(d-f) show coarse coding down to 16×16, 8×8, and 4×4 pixels.

6 Recognition using Fuzzy ARTMAP

A simplified version of the Fuzzy ARTMAP network discussed in Carpenter *et al.* (1992) is used here consisting of a Fuzzy ART module (Carpenter, Grossberg, and Rosen 1991) ART_a and a field of output nodes F^b linked together by an associative memory F^{ab} that is called the *Map Field*. Figure 7 shows

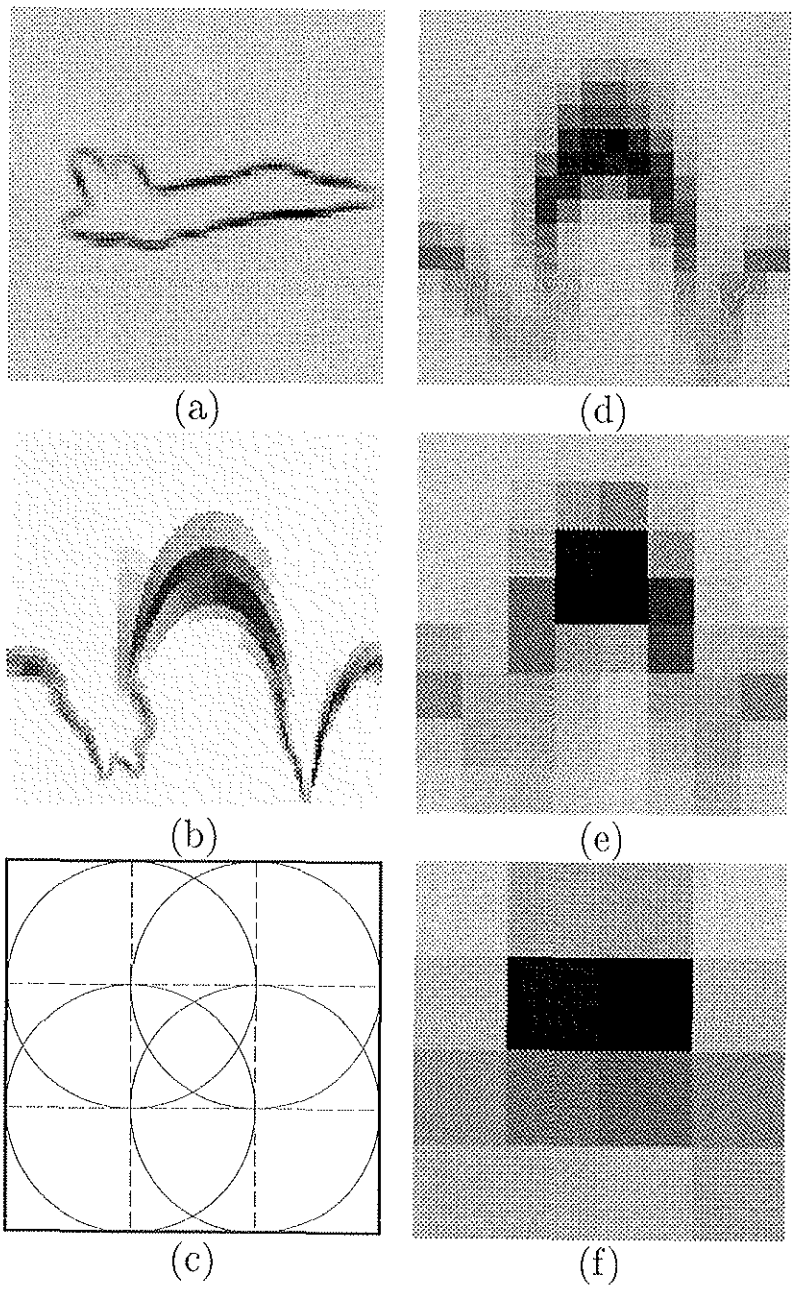


Figure 6: Preprocessing summary. (a) Output of CORT-X 2 preprocessing. (b) Centered log-polar image. (c) Gaussian coarse coding pattern. (d-f) Coarse coding reduction from 128×128 pixels down to 16×16, 8×8, and 4×4 pixels.

a block diagram of Fuzzy ARTMAP, and Figure 8 shows a flow chart with equations. In supervised learning mode, Fuzzy ARTMAP receives a sequence of input pairs $(\mathbf{a}_p, \mathbf{b}_p)$ where $\mathbf{b}_p$ is the correct output class given the analog input pattern $\mathbf{a}_p$. The ART_a module classifies analog input vectors $\mathbf{a}_p$ into categories and the Map Field makes associations from the ART_a categories to the outputs $\mathbf{b}_p$ in F^b. If $\mathbf{a}_p$ is categorized into an ART_a category that predicts an incorrect $\mathbf{b}_p$, the mismatch between actual and predicted $\mathbf{b}_p$ causes a memory search within ART_a via a mechanism called *match tracking*. Match tracking raises the ART_a vigilance parameter ρ_a by the minimum amount that will trigger a memory search. Since low vigilance leads to learning of large, coarse categories and high vigilance leads to learning of small, fine categories, match tracking sacrifices the minimum amount of category compression needed to correct each predictive error. Memory search by match tracking continues until a pre-existing ART_a category that predicts the correct ART_b category is found, or a new ART_a category is chosen after which learning takes place. Between learning trials, vigilance relaxes to its baseline vigilance $\bar{\rho}_a$. In test mode, input vectors $\mathbf{a}_p$ are classified by ART_a and the chosen category reads out its prediction to the Map Field. The index of the maximally activated node in the Map Field is taken to represent the predicted output class. (See Carpenter and Grossberg (1994) for more details.)

Fuzzy ARTMAP was modified to allow for on-line slow learning from ART_a F_2^a to the Map Field nodes. A maximal ART_a vigilance level, $\bar{\rho}_{max}$ is introduced (see Figure 8) such that an error at the Map Field triggers match tracking only if match tracking leads to a vigilance $\rho_a \leq \bar{\rho}_{max}$. If $\rho_a > \bar{\rho}_{max}$, learning takes place instead of memory search. By setting the Map Field learning rate β_{ab}, baseline $(\bar{\rho})$ and maximal $(\bar{\rho}_{max})$ vigilance levels appropriately, weights from F_2^a nodes to the Map Field approximate the conditional probability of the true class given the selected F_2^a category. A related approach to slow probability learning is described in Carpenter, Grossberg, and Reynolds (1993).

7 Simulation results

A computer simulation on the jet airplane database was done using the CORT-X 2 parameters in Table 1. The database was processed twice by CORT-X 2 using a larger and a smaller pair of oriented filters in order to compare recognition results at different scales. Coarse coding was done with both simple spatial averaging and Gaussian averaging, reducing the image down to 16×16, 8×8, and 4×4 pixels from an original size of 128×128. Except where mentioned, the simulations were run with the parameters shown in Tables 1 and 2.

The data were presented to the network in two different ways: (1) 2-D views were presented in the "natural" order in which they would appear if viewing the actual object in motion; (2) 2-D views were presented in random order. This was done to test whether presenting views in natural order helps recognition scores. Training in natural order consisted of 160 runs of from 1

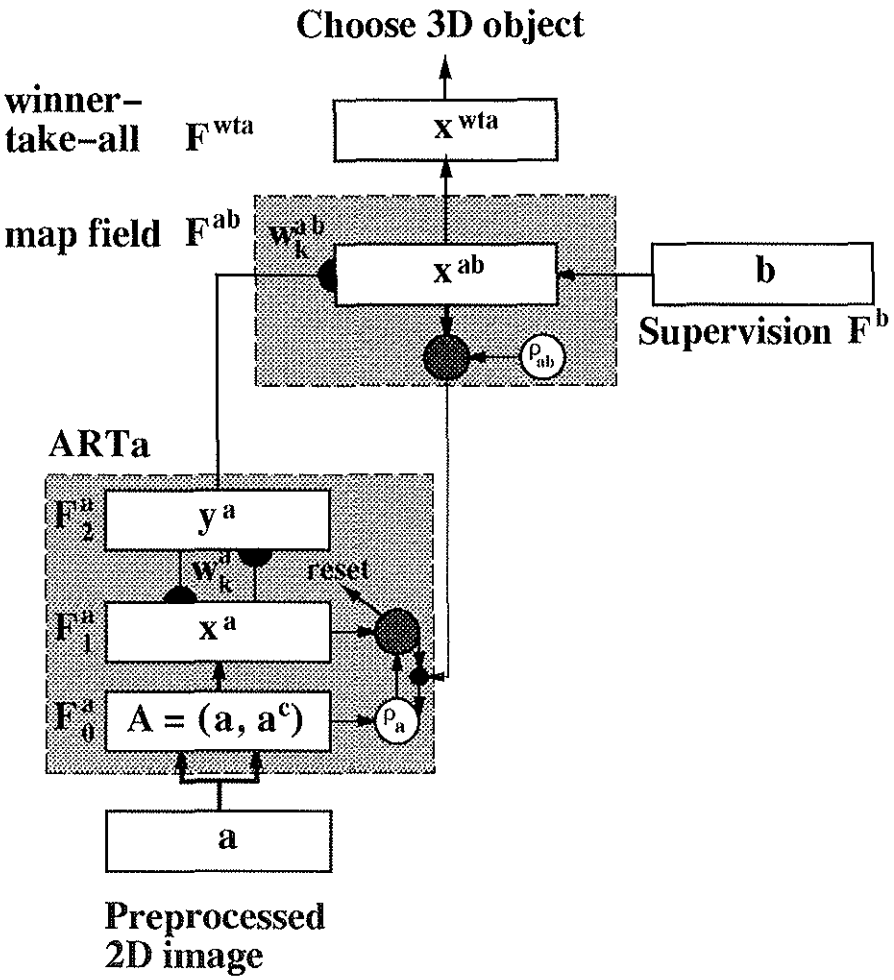


Figure 7: Fuzzy ARTMAP architecture. Each preprocessed 2-D input vector a is fed sequentially to the network as it becomes available. The inputs are complement coded which transforms the M-vector a into the $2M$-vector $\mathbf{A} = (\mathbf{a}, \mathbf{a}_j^c)$ at field F_0^a which is then fed into the input field F_1^a. A category node k is chosen at F_2^a which reads out its prediction to the Map Field via weights $\mathbf{w}_k^{ab}$. If the prediction is disconfirmed, a match tracking process is invoked in ART_a. Match tracking raises the ART_a vigilance ρ_a to just above the match ratio $|\mathbf{x}^a|/|\mathbf{A}|$. This triggers an ART_a search which activates either a different existing category, or a previously uncommitted category node at F_2^a. After the search process concludes, F^{wta} chooses the maximally activated node in F^{ab} as the 3-D object being viewed.

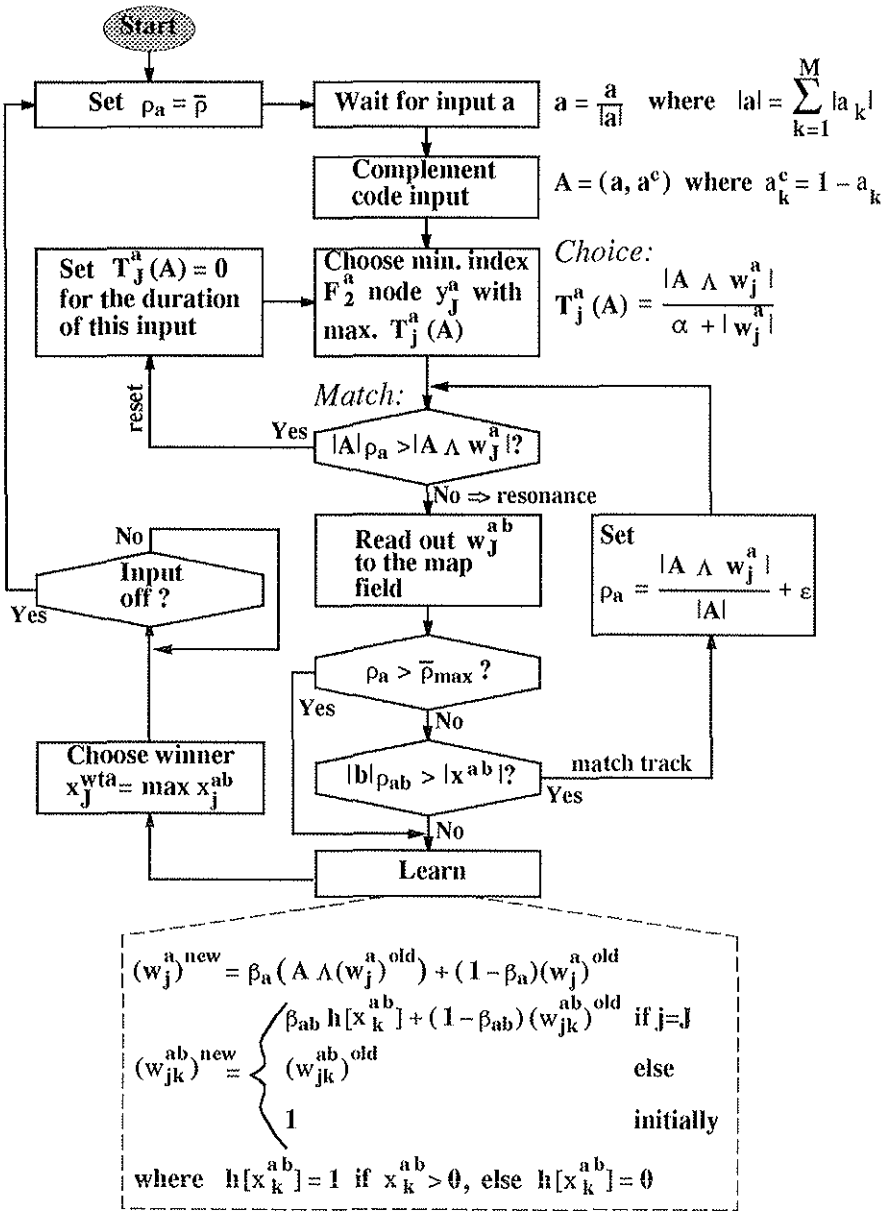


Figure 8: Fuzzy ARTMAP flow chart with reference to Figure 7. The operator $\wedge$ is defined as fuzzy AND (Zadeh, 1965) $(p \wedge q)_k \equiv \min(p_k, q_k)$.

Parameter	*Description*
$\kappa_{onC} = 7.0$	On-center magnitude
$\alpha_{onC} = 1.3$	On-center standard deviation
$\kappa_{offS} = 3.333$	Off-surround magnitude
$\alpha_{offS} = 1.875$	Off-surround standard deviation
$\overline{D} = B = 1.0$	Shunting values
$\overline{B} = D = 0.5$	Shunting values
$S = 0.2$	Spontaneous activity level
$A = 134$	Shunting decay
$\alpha_1 = \alpha_2 = 1.1$	Threshold contrast parameters
$\beta_1 = \beta_2 = \beta = .003$	Threshold noise parameters
$F = 0.5$	Complex cell scaling constant
$\epsilon = 0.1$	Hypercomplex cell divisive offset
$\mu = 5.0$	Hypercomplex cell convolution scaling
$\tau = 0.004$	Hypercomplex cell threshold
$\delta = 0.001$	Long range cooperation threshold
$\pi/8$	Oriented kernel orientation spacing
$(a_2, b_2)_{large} = (16, 8)$	Large set, large ellipse axis
$(a_1, b_1)_{large} = (10, 5)$	Large set, small ellipse axis
$(a_2, b_2)_{small} = (10, 5)$	Small set, large ellipse axis
$(a_1, b_1)_{small} = (6, 3)$	Small set, small ellipse axis
$G_1 = 2a_1/3$	Hypercomplex small kernel diameter
$G_2 = 2a_2/3$	Hypercomplex large kernel diameter
$U = 2a_2/5$	Multiple scale interaction kernel diameter
$O = 3a_2/5$	Long-range cooperation kernel length

Table 1: The parameter set used for CORT-X 2 in the simulations.

Parameter	*Description*
$\alpha = 0.6$	Fuzzy ART search order
$\beta_a = 1.0$	Fuzzy ART learning rate
$\beta_{ab} = 1.0$	Map Field learning rate
$\bar{\rho} = 0.1$	Baseline Fuzzy ART vigilance ρ_a
$\bar{\rho}_{max} = 1.0$	Maximum ART_a vigilance
$\rho_{ab} = 1.0$	Map Field vigilance

Table 2: The FuzzyARTMAP parameter set used for the simulations.

CORT-X 2 filter set	*Data presentation*	*Coarse code using spatial / Gaussian avg*		
		4x4	*8x8*	*16x16*
Small	Ordered	81.0/83.1	84.4/86.4	86.7/90.5
Small	Unordered	80.3/83.9	84.9/86.5	86.8/89.3
Large	Ordered	76.8/78.7	79.0/81.6	79.1/80.1
Large	Unordered	77.4/79.7	80.5/81.5	77.1/80.5

Table 3: Recognition results on a noise free database ($C = 0$). In the table, "Large" refers to the run with the larger set of CORT-X 2 oriented filters, "Small" refers to the run with the smaller set of filters. Views were presented either in natural order or in random order. Data was coarse coded from 128x128 down to 4x4, 8x8, or 16x16 using simple spatial averaging or Gaussian averaging. Recognition scores refer to the percent of 2-D views correctly associated with a 3-D object.

to 50 views over each object. Training in random order consisted of a series of 40 runs of 100 training set views over each object. Recognition scores are taken as an average of fifteen separate training-testing cycles.

7.1 Fast learning without noise

No clear advantage results from ordered presentation as compared to unordered presentation using noise-free data ($C = 0$) and fast learning, as shown by the results in Table 3. It can be seen that the smaller CORT-X 2 filter set resulted in better recognition performance overall and did better given more detail (less coarse coding).

7.2 Fast learning simulation with noise

The system was next tested with noisy data using additive white noise scaled by $C = 1.0$. Table 4 shows what percent of the additive noise survives processing by CORT-X 2 alone, and by CORT-X 2 and coarse coding together. The percent noise surviving these transformations was measured by the following formula:

$$\max_{\forall(x,y)} \left[\frac{\Psi(\mathbf{I}+\mathbf{N}) - \Psi(\mathbf{I})}{C} \right] \times 100, \tag{5}$$

where $\mathbf{I}$ is the image, $\mathbf{N}$ is the noise image, Ψ is the CORT-X 2 filter, $C > 0$ is the noise scaling parameter and (x, y) is the pixel index in the images. Table 4 represents the average results from ten measurements using equation (5). With such noise reduction, the recognition results shown in Table 5 were similar to those for the noise-free case in Table 3, except for some falling off of recognition scores at the lowest level of coarse coding (the 16×16 case).

Table 6 shows the number of nodes created by the network after training for the no noise (left entry) and noise (right entry) results reported above. Noise causes a small increase in the number of categories formed on average as the network attempts to correct a greater number of noise-induced errors during supervised training.

% noise surviving CORT-X 2 filtering and Coarse Coding:		
	Large CORT-X 2 filters (16x8, 10x5)	*Small CORT-X 2 filters (10x5, 6x3)*
	1.79	2.42
After Gaussian coarse coding from 128x128 down to:		
16x16	0.33	0.34
8x8	0.23	0.29
4x4	0.19	0.26
After spatial average coarse coding from 128x128 down to:		
16x16	0.40	0.40
8x8	0.28	0.30
4x4	0.21	0.28

Table 4: Percent of additive white noise surviving processing by CORT-X 2 and coarse coding.

CORT-X 2 filter set	*Data presentation*	*Coarse code using spatial / Gaussian avg*		
		4x4	*8x8*	*16x16*
Small	Ordered	80.1/83.3	84.5/85.9	84.2/89.1
Small	Unordered	79.4/83.2	83.9/86.4	84.3/88.0
Large	Ordered	76.6/79.4	79.3/80.8	75.8/79.3
Large	Unordered	76.0/79.7	78.4/80.7	75.5/79.0

Table 5: Recognition results on noisy data ($C = 1$) with fast learning ($\beta_{ab} = 1.0$). These results differ little from the noise-free results in Table 3 (no noise condition) with the exception of some consistent reduction in scores for the 16x16 coarse coding.

CORT-X 2 filter set	*Data presentation*	*Coarse code using spatial / Gaussian avg*		
		4x4	*8x8*	*16x16*
Small	Ordered	[172, 184] [165, 169]	[77, 73] [70, 73]	[34, 33] [33, 35]
Small	Unordered	[191, 198] [175, 179]	[76, 77] [73, 76]	[34, 35] [35, 36]
Large	Ordered	[168, 179] [160, 162]	[71, 68] [67, 71]	[31,33] [30, 31]
Large	Unordered	[183, 192] [169, 174]	[73, 75] [69, 72]	[32, 32] [33, 32]

Table 6: Average number of ART_a categories formed during training for the simulations of Table 3 (no noise) and Table 5 (noise). The format in the table is as follows: [spatial avg.]/[Gaussian avg.] = [No noise, Noise]/[No noise, Noise].

CORT-X 2 filter set	*Data presentation*	*Coarse code using spatial / Gaussian avg*		
		4x4	*8x8*	*16x16*
Small	Ordered	79.9/83.1	84.0/85.6	84.7/89.9
Small	Unordered	78.8/83.3	83.2/85.7	84.9/89.1
Large	Ordered	76.3/78.2	78.5/81.5	77.0/78.8
Large	Unordered	77.4/80.2	79.6/80.41	75.8/79.2

Table 7: Recognition results on noisy data ($C = 1$) with slow learning to the Map Field ($\beta_{ab} = 0.2$, $\bar{\rho}_{max} = 0.95$). Due to the low levels of noise surviving preprocessing, the recognition results here are not substantially different than those found using fast learning in noise in Table 5 except where noise was highest as in the 16x16 coarse coding. As noise increases, slow learning becomes more important for maintaining good recognition scores.

7.3 Slow learning simulation with noise

For the next set of computer simulations, the network was run on the noisy data using slow learning to the Map Field ($\beta_{ab} = 0.2$). Fast learning was still used within the ART_a module itself ($\beta_a = 1.0$). Note that for $\bar{\rho}_{max} = 1.0$, the results for slow learning and fast learning (Section 7.2) to the Map Field are equivalent. They are equivalent because with Map Field vigilance set to $\rho_{ab} = 1.0$ as in Table 2, the slightest mismatch at the Map Field will invoke match tracking and a new category will be created. To derive benefit from slow learning in the case $\rho_{ab} = 1.0$, we set $\bar{\rho}_{max} = 0.95$. Table 7 records the results using slow learning in large amplitude noise ($C = 1$). Where noise levels after preprocessing were very small, the results were approximately the same as in the fast learning case shown in Table 5. Slow learning begins to help when the noise level increases, as with the 16×16 coarse coding. Table 8 records the average number of categories formed for the noisy data case using fast learning and slow learning. Slow learning with $\bar{\rho}_{max} = 0.95$, caused approximately 10% fewer categories to be formed than with $\bar{\rho}_{max} = 1.0$, since noise-induced errors do not always cause the formation of a new category in the former case.

8 Voting versus view transitions

For the jet data set as processed by VIEWNET, it was found that the average overall length of an error sequence was 1.31 2-D views with a standard deviation of 0.57 views. Thus, when an error occurs, collecting evidence from (or voting over) two more views will usually be sufficient to correct the error. This can be done in VIEWNET by adding an integration field (F^{int}) between the Map Field (F^{ab}) and the winner-take-all field (F^{wta}) in Figure 7. The equation for the integrator field is stepped once each time ART_a chooses a category:

$$(x_k^{int})^{new} = \beta_{int} x_k^{ab} + (1 - \beta_{int})(x_k^{int})^{old}, \tag{6}$$

where x_k^{int} is an integrator node for the k^{th} object, β_{int} is the integration rate each time the equation is stepped, and x_k^{ab} is the k^{th} Map Field category. The

CORT-X 2 filter set	*Data presentation*	*Coarse code using spatial / Gaussian avg*		
		4x4	*8x8*	*16x16*
Small	Ordered	[184, 165] [169, 150]	[73, 67] [73, 66]	[33, 30] [35, 32]
Small	Unordered	[198, 180] [179, 163]	[77, 69] [76, 70]	[35, 32] [36, 33]
Large	Ordered	[179, 160] [162, 147]	[68, 61] [71, 66]	[33 , 30] [31, 29]
Large	Unordered	[192, 175] [174, 160]	[75, 69] [72, 67]	[32, 30] [32, 30]

Table 8: Average number of nodes formed during training for the simulations of Tables 5 (noise with fast learning) and 7 (noise with slow learning). The format in the table is as follows: [spatial avg.]/[Gaussian avg.] = [fast learning, slow learning]/[fast learning, slow learning]. It can be seen that slow learning reduced the number of nodes formed by approximately 10%.

maximal integration node is chosen by the winner-take-all field (F^{wta}) as the network's identification of the 3-D object.

Figure 9 shows the average recognition scores for voting with $\beta_{int} = 0.2$ over one, two, and three views under CORT-X 2 preprocessing with large and small scale filter sets and coarse coding to 4×4, 8×8 and 16×16 pixels using both Gaussian and spatial averaging. Voting over 3 frames improves recognition results by an average of ten percent with the best results being 98.5% correct for small scale filtered 16×16 Gaussian coarse coded data.

The advantage of voting over using 2-D view transitions is that given N 2-D views, the $O(N^2)$ cost for learning view transitions is avoided. To compare how well voting over view sequences does to using view transitions, the architecture described in Bradski, Carpenter, and Grossberg (1991) (Section 1) that incorporates 2-D views and 2-D view transitions for recognition was simulated.

In Figure 9, the black circles and squares represent the recognition scores using view transitions for preprocessing with the large and small scale CORT-X 2 filters respectively. Recognition scores from view transitions and from evidence accumulation are similar. Since evidence accumulation does not require the $O(N^2)$ nodes needed for learning 2-D view transitions, evidence accumulation over view transitions seems sufficient for this application.

9 Concluding Remarks

Using the smaller set of CORT-X 2 filters, a 3-D object recognition rate of approximately 90% may be achieved from single 2-D views alone without recourse to more elaborate methods of generating aspect graph models of the 3-D objects. When evidence integration or voting over a sequence of views is added, recognition rates reach 98.5% within three views. Voting over two views did as well as using view transitions on this database, but

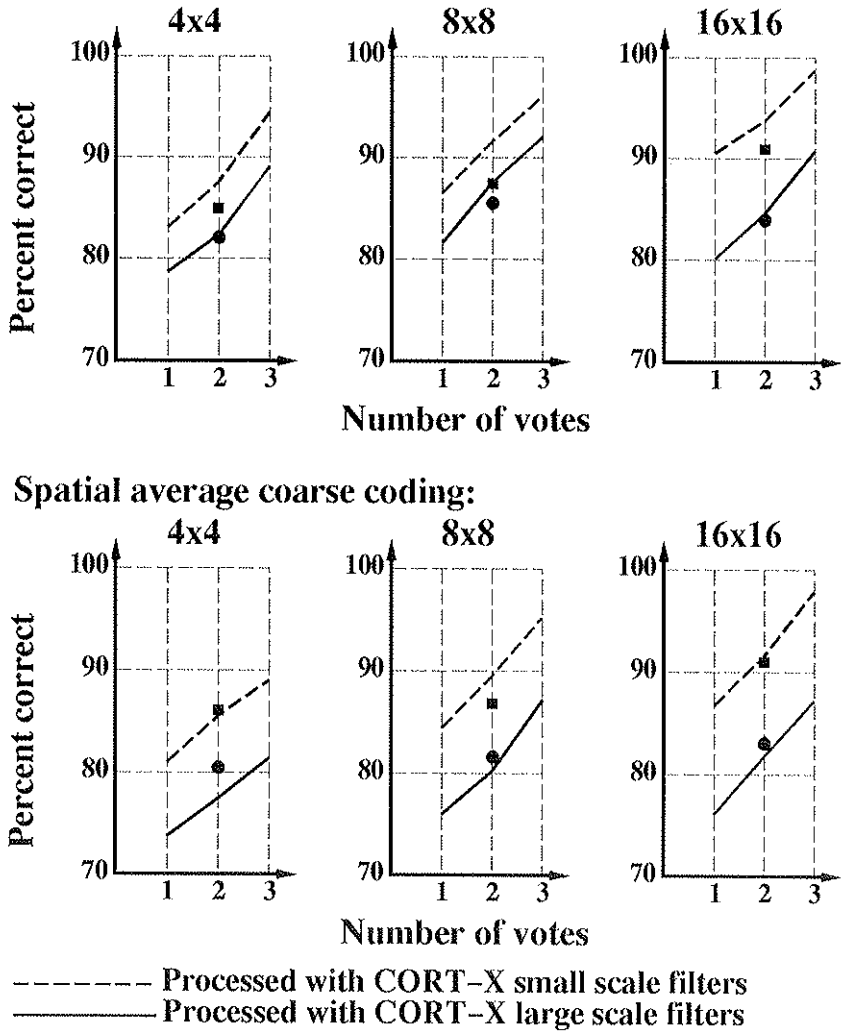


Figure 9: Recognition results for voting with an integration rate of $\beta_{int} = 0.2$. The graphs show the recognition results after gathering evidence over one, two and three 2-D views for data preprocessed using large (solid line) and small (dotted line) scale CORT-X 2 filters. Results from both Gaussian and spatial averaging coarse coding methods are shown where the images were reduced from 128×128 down to 4×4, 8×8 and 16×16 pixels. The circles and squares represent recognition scores resulting from using view transitions as discussed in Section 8.

without the drawback of needing to learn $O(N^2)$ view transitions given N, 2-D views. These recognition rates can be maintained even in high noise conditions using the preprocessing methods described here.

These high recognition rates were achieved by using a different preprocessor and supervised learning to create more optimal category boundaries than in the Seibert and Waxman studies. Seibert and Waxman (1992) used unsupervised clustering of coarse coded maximal curvature data to create general categories that unambiguously selected for the correct 3-D object only 25% of the time. In so doing, their network created 41 categories during training. To overcome the ambiguity of their general ART 2 categories, Seibert and Waxman used 2-D view category transitions to help identify the 3-D objects. Even if two 3-D objects shared the 2-D view categories of ART 2, they might not share the particular 2-D view category transitions that could then be used to distinguish one object from the other at the cost of needing to represent $O(N^2)$ transitions. Seibert and Waxman's network must then be able to represent possible cross-correlations between every categorical 2-D view in its view transition matrices, one for each object, even if no correlations are eventually found between some of the categories. Thus, their algorithm needed to represent the possible correlations between each of the 41 2-D view categories that were generated. The total number of correlations were then $(41^2 - 41)/2 = 820$, since transitions and their reverse are equivalent and there are no self-transitions. This is done for each object for a total representation of $820 \times 3 = 2460$ possible correlations. In actual practice, the view transition matrices were sparse. For example, 70 view transitions were actually learned during training for the F-16. In contrast, Tables 3 and 6 show that VIEWNET obtained its best recognition results over all three jets using a total of 33 2-D view categories without any representation of view transitions.

The Fuzzy ARTMAP architecture computes goodness of fit information that may be used to enhance its power in future applications. In particular, the match or choice equation in Figure 8 may be used to measure to the quality of the recognition. If VIEWNET recognizes a 3-D object, but its ART_a category prototype provides a poor fit to the input vector, then the goodness of fit information could be used to cause VIEWNET to collect more data before a final recognition decision is made. If VIEWNET is embedded in an active vision system, then a poorly fitting view could be used to trigger the system to move to get a better perspective.

A Appendix: CORT-X 2 equations

The equations for the CORT-X 2 filter as described in Section 3 are discussed below. Figure 2 shows the flow chart and filter kernels. Table 1 summarizes the parameters used in the simulations. Filter kernels G, U, and O are normalized to have area equal to one.

A.1 Step 1. Discounting the Illuminant

ON-C and OFF-C Network: The activation x_{ij} at node v_{ij} at position (i,j) obeys the shunting on-center off-surround equation:

$$\frac{d}{dt}x_{ij} = -Ax_{ij} + (B - x_{ij})C_{ij} - (x_{ij} + D)E_{ij}, \tag{7}$$

and $\bar{x}_{ij}$ obeys the off-center, on-surround equation:

$$\frac{d}{dt}\bar{x}_{ij} = -A(\bar{x}_{ij} - S) + (\overline{B} - \bar{x}_{ij})\overline{C}_{ij} - (\bar{x}_{ij} + \overline{D})\overline{E}_{ij} \tag{8}$$

where C_{ij}, $\overline{C}_{ij}$, E_{ij}, $\overline{E}_{ij}$ are discrete convolutions of the input with gaussian kernels of the form:
$K_{ij} = \sum_{p,q} I_{pq} K_{pqij}$ with $K_{pqij} = \kappa \exp\left\{-\alpha^{-2}\log 2[(p-i)^2 + (q-j)^2]\right\}$.

The on-center kernel of $\bar{x}_{ij}$ is the off-surround kernel of x_{ij}, and the off-surround kernel of $\bar{x}_{ij}$ is the on-center kernel of x_{ij}. Then $\overline{C}_{ij} = E_{ij}$, $\overline{E}_{ij} = C_{ij}$. Also in equations (7) and (8), $\overline{B} = D$ and $\overline{D} = B$. At equilibrium in the ON-C network,

$$x_{ij} = \frac{\sum_{(p,q)}(BC_{pqij} - DE_{pqij})I_{pq}}{A + \sum_{(p,q)}(C_{pqij} + E_{pqij})I_{pq}}, \tag{9}$$

and in the OFF-C network,

$$\bar{x}_{ij} = \frac{AS + \sum_{(p,q)}(DE_{pqij} - BC_{pqij})I_{pq}}{A + \sum_{(p,q)}(C_{pqij} + E_{pqij})I_{pq}}. \tag{10}$$

A.2 Step 2. CORT-X 2 Filter

Oriented receptive fields are elliptical with $y^2/a_s^2 + x^2/b_s^2 = 1$, where a_s is the major axis and b_s is the minor axis with $a_s \geq b_s$. Two sizes of receptive fields were used, indexed by the subscript s with 1 = small scale and 2 = large scale. Orientations are indexed below by subscript k.

Simple Cells: Simple cells of scale s with activation variable $x = x_{ij}$ and receptive field orientation k have outputs

$$S_{sL}(i,j,k) = \max[L_s(x,k) - \alpha_s R_s(x,k) - \beta_s, 0] \tag{11}$$

$$S_{sR}(i,j,k) = \max[R_s(x,k) - \alpha_s L_s(x,k) - \beta_s, 0] \tag{12}$$

where $L_s(x,k)$ and $R_s(x,k)$ are the left or right oriented receptive field inputs

$$L_s(x,k) = \frac{\sum_{(p,q)\in l_s(i,j,k)} x_{pq} w_{pq}}{\sum_{(p,q)\in l_s(i,j,k)} w_{pq}} \tag{13}$$

and

$$R_s(x,k) = \frac{\sum_{(p,q)\in r_s(i,j,k)} x_{pq} w_{pq}}{\sum_{(p,q)\in r_s(i,j,k)} w_{pq}}, \tag{14}$$

where w_{pq} is a weighting factor proportional to the area of a cell covered by the receptive field. L and R in S_{sL} and S_{sR} indicate that each receptive field is sensitive to the opposite direction-of-contrast from its companion. The ON and OFF networks have separate sets of simple cells with the ON simple cells denoted by S^+_{sL} and S^+_{sR}, and the OFF simple cells denoted by S^-_{sL} and S^-_{sR}.

Complex Cells: The complex cell output $C_s(x,k)$ is defined by

$$C_s(i,j,k) = F[S^+_{sL}(i,j,k) + S^+_{sR}(i,j,k) + S^-_{sL}(i,j,k) + S^-_{sR}(i,j,k)]. \tag{15}$$

Hypercomplex Cells (First Competitive Stage): The hypercomplex cells $D_s(i,j,k)$ receive input from the spatial competition among the complex cells:

$$D_s(i,j,k) = \max\left[\frac{C_s(i,j,k)}{\epsilon + \mu \sum_m \sum_y C_s(p,q,m) G_s(p,q,i,j,k)} - \tau, 0\right]. \quad (16)$$

Hypercomplex Cells (Second Competitive Stage): Hypercomplex cells $D_s(i,j)$ compute the competition among oriented activities $D_s(i,j,k)$ at each position. This process is simplified as a winner-take-all process

$$D_s(i,j) = D_2(i,j,K) = \max_k D_s(i,j,k), \quad (17)$$

where K denotes the orientation of the maximally activated cell.

Multiple Scale Interaction: The interaction between the small and large scales is defined by

$$B_{12}(i,j) = D_1(i,j) \sum_{p,q} D_2(p,q) U(p,q,i,j). \quad (18)$$

Long-Range Cooperation: The large detectors $D_2(i,j)$, are capable of responding across locations where pixel signal strength has been reduced by noise. Such boundary signals may, however, be poorly localized. To overcome this tradeoff between boundary completion and localization, large-scale cells interact cooperatively as

$$B_2(i,j) = D_2(i,j) \max\left[\sum_{p,q} D_2(p,q,K) O(p,q,i,j,K) - \delta, 0\right]. \quad (19)$$

CORT-X 2 Output: The final output of the CORT-X 2 filter is the sum of the multiple scale interaction and the cooperative process:

$$B(i,j) = B_{12}(i,j) + B_2(i,j). \quad (20)$$

Acknowledgements

Gary Bradski was supported in part by the National Science Foundation (NSF IRI 90-24877) and the Office of Naval Research (ONR N00014-92-J-1309).

Steve Grossberg was supported in part by the Air Force Office of Scientific Research (AFOSR F49620-92-J-0499), ARPA (AFOSR 90-0083 and ONR N00014-92-J-4015) and the Office of Naval Research (ONR N00014-91-J-4100).

References

Bowyer, K., Eggert, D., Stewman, J., & Stark, L. (1989). Developing the aspect graph representation for use in image understanding. In *Proc. 1989 Image Understanding Workshop*, pp. 831–849.

Bradski, G., Carpenter, G., & Grossberg, S. (1991). Working memory networks for learning multiple groupings of temporally ordered events: Application to 3-D visual object recognition. In *Proceedings of the IJCNN-91, Seattle, WA.*, Vol. 1, pp. 723–728. Piscataway, NJ: IEEE Service Center.

Carpenter, G., & Grossberg, S. (1987). ART 2: Self-organization of stable category recognition codes for analog input patterns. *Applied Optics, 26*, 4919–4930.

Carpenter, G., & Grossberg, S. (1994). Self-organizing neural networks for supervised and unsupervised learning and prediction.. In Cherkassky, V., Friedman, J., & Wechsler, H. (Eds.), *From Statistics to Neural Networks. Theory and Pattern Recognition Applications.* New York, NY: Springer-Verlag.

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

Carpenter, G., Grossberg, S., & Mehanian, C. (1989). Invariant recognition of cluttered scenes by a self-organizing ART architecture: CORT-X boundary segmentation. *Neural Networks, 2*, 169–181.

Carpenter, G., Grossberg, S., & Reynolds, J. (1993). Fuzzy ARTMAP, slow learning and probability estimation. Tech. rep. CAS/CNS-TR-93-014, Boston University, Boston, MA: Boston University.

Carpenter, G., Grossberg, S., & Rosen, D. (1991). Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system. *Neural Networks, 4.*

Carpenter, G., & Ross, W. (1993). ART-EMAP: A neural network architecture for learning and prediction by evidence accumulation. In *Proceedings of the World Congress on Neural Networks, (WCNN-93).*, Vol. III, pp. 649–656.

Chang, I., & Huang, C. (1992). Aspect graph generation for non-convex polyhedra from perspective projection view. *Pattern Recognition, 25*(10), 1075.

Eggert, D., & Bowyer, K. (1993). Computing the perspective projection aspect graph of solids of revolution. *IEEE Trans. on Pattern Analysis and Machine Intelligence, 15*(2), 109.

Gigus, Z., & Malik, J. (1988). Computing the aspect graph for line drawings of polyhedral objects. In *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, pp. 654–661.

Gigus, Z., & Malik, J. (1990). Computing the aspect graph for line drawings of polyhedral objects. *IEEE transactions on pattern analysis and machine vision*, *12*(2), 113.

Grossberg, S. (1993). 3-D vision and figure-ground separation by visual cortex. Tech. rep. CAS/CNS-TR-92-019, Boston University, Perception and Psychophysics, in press.

Grossberg, S., & Todorovic, 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.

Grossberg, S., & Wyse, L. (1991). A neural network architecture for figure-ground separation of connected scenic figures. *Neural Networks*, *4*, 723–742.

Grossberg, S., & Wyse, L. (1992). A neural network architecture for figure-ground separation of connected scenic figures. In Pinter, R., & Nabet, B. (Eds.), *Nonlinear Vision: Determination of Neural Receptive Fields, Function, and Networks* (1 edition)., chap. 21, pp. 516–543. CRC Press, Inc.

Koenderink, J., & van Doorn, A. (1979). The internal representation of solid shape with respect to vision. *Biological Cybernetics.*, *32*, 211–216.

Plantinga, H., & Dyer, C. (1990). Visibility, occlusion, and the aspect graph. *Int. J. Comput. Vision*, *5*(2), 137–160.

Ponce, J., & Kriegman, D. (1990). Computing exact aspect graphs of curved objects: Parametric surfaces. In *Proc. 8th National Conf. on Artificial Intell.*, pp. 1074–1079.

Rieger, J. (1990). The geometry of view space opaque objects bounded by smooth surfaces. *Artificial Intell.*, *44*, 1–40.

Rimey, R., & Brown, C. (1991). Hmms and vision: Representing structure and sequences for active vision using hidden markov models. Tech. rep., University of Rochester Computer Science Department TR No. 366.

Schwartz, E. (1977). Spatial mapping in primate sensory projection: analytic structure and relevance to perception. *Biological Cybernetics*, *25*, 181–194.

Seibert, M., & Waxman, A. (1992). Adaptive 3-D-object recognition from multiple views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*, 107–124.

Sripradisvarakul, T., & Jain, R. (1989). Generating aspect graphs for curved objects. In *Proc. IEEE Workshop Interpretation of 3-D Scenes*, pp. 109–115.

Stewmen, J., & Bowyer, K. (1990). Direct construction of the perspective projection aspect graph of convex polyhedra. *Computer vision, graphics, and image processing*, *51*(1), 20.

Zadeh, L. (1965). Fuzzy sets. *Information Control*, *8*, 338–353.