performs saccadic motion. Through the use of controlled saccadic motion, the rotational and translational values of motion can be determined after a few iterations of the algorithm. If the motion of the mobile entity changes smoothly along its trajectory, the algorithm will "track" the instantaneous direction of motion, and will provide continuous egomotion values for any type of path. It is important to note that our method simultaneously provides rotation and translation information, as opposed to methods that first determine rotation and then derotate an image in order to find the FOE and thus the direction of translation.

Although experiments were carried out using a mobile robot that moved on a flat floor (essentially 2-D motion), the algorithm is applicable to full 3-D motion. As was shown in Sections II and III, motion parallax occurs when the fixation point falls in any cross-sectional plane perpendicular to the instantaneous direction of translation (i.e., a plane when ϕ is constant). Therefore, the algorithm will converge to the proper direction of translation and will provide rotational information with respect to the visual sensor's coordinate basis. Also, note that this method does not assume that the forward pointing axis of the moving entity coincides with the instantaneous direction of motion. Such an assumption can be made for vehicles with conventional forward wheel steering with no wheel slippage. In this case, a similar but somewhat simpler algorithm can be used when the camera's rotational angle is known with respect to the forward direction of the vehicle [17].

In our experimentation, the limiting factor for calculating the egomotion parameters in real time is the optical flow determination. In order to determine the motion parallax around the line-of-sight, we tracked a small number of feature points (< 20) over several frames. This was performed on a Sun 4 workstation with a standard framegrabber and each flow determination step required 10 to 20 s. The evaluation of the saccadic control equation requires only a single multiply and two additions, which can be considered computationally inexpensive when compared to evaluating the nonlinear equations of standard 3-D motion and structure determination techniques used on optical flow. Furthermore, we have encountered a small percentage of error in our experiments. This error could be reduced by improving the angular resolution of our active camera. Also, the algorithm depends greatly on accurate tracking and accurate flow measurements, so errors in the tracking and flow algorithms need to be minimized.

Acknowledgment

This work was carried out during a 17-month research visit at Osaka University by the first author. Many thanks go to the members of the Tsuji Laboratory in the Department of Control Engineering, in particular, to Dr. Hiroshi Ishiguro.

References

[1] H.-H. Nagel, "Images sequences—Ten (octal) years—From phenomenology towards a theoretical foundation," in *Proc. Int. Conf. Pattern Recogn.*, Paris, France, 1986.

[2] S. Ullman, "Recent computational studies in the interpretation of structure from motion," in *Human and Machine Vision*, J. Beck, B. Hope, and A. Rosenfeld, Eds. New York: Academic, 1983.

[3] D. Lawton, "Processing translational motion sequences," *Comput. Graphics, Image Process.*, vol. 22, pp. 116-144, 1983.

[4] R. Dutta, R. Manmatha, E. Riseman, and M. Snyder, "Issues in extracting motion parameters and depth from approximate translational motion," in *Proc. DARPA Image Understanding Workshop*, Cambridge, MA, 1988.

[5] D. Ballard and C. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1982.

[6] R. Bajcsy, "Active perception," *Proc. IEEE*, vol. 76, no. 8, pp. 996-1005, 1988.

[7] P. J. Burt, "Smart Sensing within a Pyramid Vision Machine," *Proc. IEEE*, vol. 76, no. 8, pp. 1006-1015, 1988.

[8] R. Bajcsy, "Perception with feedback," in *Proc. DARPA Image Understanding Workshop*, Cambridge, MA, 1988.

[9] D. Ballard, "Reference frames for animate vision," in *Proc. Int. Joint Conf. Artificial Intell.*, Detroit, MI, 1989.

[10] M. Swain and M. Stricker, Eds., "Promising directions in active vision," in *Proc. Nat. Sci. Foundat. Active Vision Workshop*, Chicago, IL, 1991.

[11] J. E. Cutting, *Perception with an Eye for Motion*. Cambridge, MA: MIT Press, 1986.

[12] B. K. P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.

[13] D. Raviv and M. Herman, "Towards an understanding of camera fixation," in *Proc. IEEE Int. Conf. Robot., Automat.*, Cincinnati, OH, 1990.

[14] H. von Helmholtz, *Physiological Optics*, vol. 3, (Optic. Soc. Amer., 1924. New York: Dover, 1964, p. 295.

[15] J. J. Gibson, *The Perception of the Visual World*. Cambridge, MA: Riverside Press, 1950.

[16] J. J. Koenderink and A. J. Van Doorn, "Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer," *Optica Acta*, vol. 22, no. 9, pp. 773-791, 1975.

[17] M. Barth, H. Ishiguro, and S. Tsuji, "Computationally inexpensive egomotion determination for a mobile robot using an active camera," in *Proc. IEEE Int. Conf. Robot. Automat.*, Sacramento, CA, 1991.

An Invariant Pattern Recognition Machine Using a Modified ART Architecture

Narayan Srinivasa and Musa Jouaneh

***Abstract*—A novel invariant pattern recognition machine is proposed based on a modified ART architecture. Invariance is achieved by adding a new layer called F_3, beyond the F_2 layer in the ART architecture. The design of the weight connections between the nodes of the F_2 layer and the cells of the F_3 layer are similar to the invariance net. Computer simulations show that the model is not only invariant to translations and rotations of 2-D binary images but also noise-tolerant to these transformed images.**

I. Introduction

The problem of invariant pattern recognition has interested researchers for a long time. Casasent *et al.* [1] developed a model based on optical correlations to achieve invariance to position, rotation, and scaling of images. Cavanagh [2] proposed a model for size and position invariance in the visual cortex of the brain. Fukushima [3] and Fukushima and Miyake [4] developed a model based on several hierarchically organized processing stages to gradually free image processing from its spatial coordinates. Higher order threshold logic units were used for invariant image processing by Maxwell *et al.* [5]. Szu [6] used holographic coordinate

Manuscript received July 18, 1992; revised February 11, 1993.

N. Srinivasa is with the Department of Mechanical Engineering, University of Florida, Gainesville, FL 32611.

M. Jouaneh is with the Department of Mechanical Engineering, University of Rhode Island, Kingston, RI 02881.

IEEE Log Number 9209683.

transformations to achieve 2-D spectra that is invariant to translations, rotations and scaling.

Recently, Widrow *et al.* [7], [8] have used an invariance net as a preprocessor to a trainable classifier in order to develop an invariant pattern recognition machine. However, the invariance net is intolerant to noise and its computation time increases rapidly with an increase in image size. Khotanzad and Lu [9] and Yegnanarayana and Ravichandran [10] have used a filter in conjunction with a multilayered perceptron to achieve invariance to translations, rotations, and scaling. The invariant filter designed by them is based on determining the geometric moments of the input image. A previous generalization [11] of the ART1 architecture solves the invariance problem by using a Fourier–Mellin filter as a preprocessor. However, this method requires very expensive hardware such as scanned laser devices and reusable spatial light modulators. In our previous effort [12], we have used the invariance net [7], [8] as a preprocessor to the ART1 neural network to achieve invariant pattern recognition. However, it is highly intolerant to noise.

In this paper, a self-organizing neural network model is developed for the purpose of invariant pattern recognition. Invariance is achieved by adding a new postprocessing layer to an ART1 neural network called the F_3 layer and is based on the invariance net [7], [8]. Unlike the method proposed in [12], the new network has noise tolerant properties and is computationally more efficient. While the processing time required by the proposed network increases with an increase in image size, it would be ideal to use it for low resolution images as it is purely based on software.

This correspondence is divided into the following sections. Reasons for the choice of an ART architecture for invariant pattern recognition are delineated in Section II. Section III discusses in brief the working principle of the ART1 neural network. In section IV the overall structure of the proposed model is described with emphasis on the structure and working principle of the new processing layer. Section V discusses the computer simulations and the results obtained. A possible approach to achieve size invariance through the proposed model is discussed in Section VI. Concluding remarks are given in Section VII.

II. Rationale for the Choice of an ART Architecture

Neural networks can be classified into two types based on their learning methods: supervised and unsupervised neural nets. The backpropagation algorithm [13], the Functional-link net [14], Boltzmann machine [15], and multilayered perceptrons [16] are examples of neural networks which require supervision from an external agent for training. On the other hand, the adaptive resonance theory (ART) network [17]–[19], Kohonen's network [20], Anderson's network [21], and the adaptive bidirectional associative memory (ABAM) network [22] are examples of neural nets that do not require supervision for training.

Unsupervised learning networks are attractive since they do not need to be trained by exemplars before coding can proceed. The Kohonen and Anderson networks work well with both binary and analog inputs but cannot learn new associations once they are trained. The ABAM network, while conceptually simpler than ART network, is subject to pattern orthogonality constraints and pattern storage problems. In contrast, the ART network does not have the above limits on its performance. Furthermore, processing of inputs in real time is possible using "fast learning conditions" [17], and because every time a familiar pattern is presented, it directly activates an appropriate stored pattern. In addition, the ART network possesses additional properties that are desirable in a pattern classifier for use in engineering applications [23], [24]. These properties include:

1) Self-Scaling Property: This refers to its ability to treat mismatches in input patterns with few features as essential while suppressing the same mismatches as noise in input patterns with many features [17]. For image processing, this implies that depending on the resolution of the image presented, the network will be selective to certain features of it.

2) Self-Stabilizing Property: This refers to its ability to defend its fully committed memory capacity from being washed away by an incessant flux of new input patterns and to access a node in its memory without a search if a familiar input were to be presented.

3) Plasticity: This refers to its ability to recognize and store new input patterns in a nonstationary environment limited only by the total memory available. A proof of these properties is given in [25].

Properties 2) and 3) help not only in preserving previously learned images but also in continuing to learn new images without erasing the memories of prior images. All the above properties make the ART network an ideal for use as a pattern recognition machine for image processing. However, the network is not invariant to translations, rotations, and scaling and needs to be modified in order to achieve that.

III. Working Principle of the ART1 Neural Network

The architecture of the ART1 neural network is shown in Fig. 1. The network has two processing layers F_1 and F_2. The algorithm for the net is as follows:

Step 1) Present the binary input pattern as an N dimensional input vector I with each component I_i ($i = 1, 2, \cdots, N$) having a value 0 or 1.

Step 2) Bottom-up processing to obtain a weighted sum y_j for each node j in the F_2 layer:

$$y_j = \sum_{i=1}^{N} b_{ji} I_i \tag{1}$$

where b_{ji} is the bottom-up long term memory (*LTM*) trace connecting the ith component of I at F_1 to the jth node at F_2.

Step 3) Choose the node J with the largest value of y_j in the F_2 layer.

Step 4) Verify if I belongs to the Jth node of the F_2 layer as follows. I belongs to the Jth node if

$$\frac{\sum_{i=1}^{N} t_{Ji} I_i}{\sum_{i=1}^{N} I_i} > \rho \tag{2}$$

where ρ is the vigilance parameter and t_{Ji} is the top-down LTM trace. If I belongs to the Jth node then go to step 5); otherwise, go to step 6).

Step 5) Update b_{Ji} and t_{Ji} under fast learning conditions as follows:

$$b_{Ji}^{\text{new}} = \frac{t_{Ji}^{\text{old}} I_i}{0.5 + \sum_{i=1}^{N} t_{Ji}^{\text{old}} I_i} \tag{3}$$

and

$$t_{Ji}^{\text{new}} = t_{Ji}^{\text{old}} I_i \tag{4}$$

It should be noted that the LTM traces are initialized as

$$t_{ji} = 1.0; \quad b_{ji} = \frac{1}{1 + N} \qquad \text{for all } i \text{ and } j.$$

Step 6) Since I does not belong to the node that was most like it, send a reset wave to F_2 via the orienting unit to deactivate that

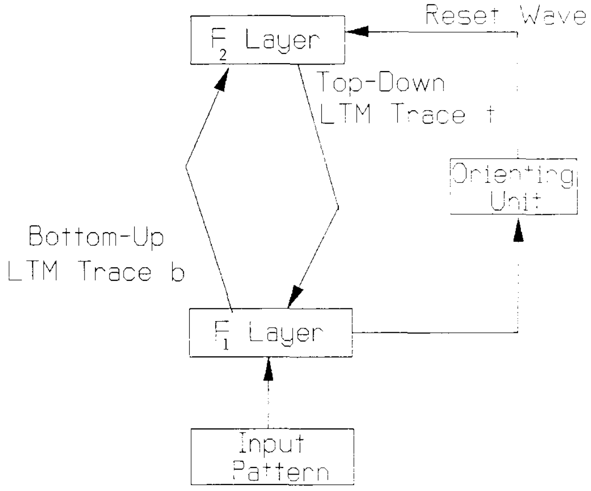


Fig. 1. The ART1 neural network architecture.

node and go back to step 2) to search for another node in the F_2 layer.

IV. Structure and Working Principle of the New Network

The new network contains three processing layers: the F_1 and F_2 layers of the ART1 neural net and a new layer called F_3. This layer is based on the invariance net [7], [8] and acts as a vehicle to generate invariance in the F_2 layer. To illustrate the working principle of the new network, consider a 4 × 4 binary image as an input to the F_1 layer of the ART1 neural net. The algorithm for the new net is as follows:

Step 1) The 4 × 4 binary image B is converted to a 16-dimensional input vector I and is input to the F_1 layer of the ART1 neural net.

$$I_i = B_{km} \qquad i = 1 \text{ to } 16 \tag{5}$$

where B_{km} are the elements of the image B and

$$k = n\left[\frac{i-1}{4}\right] + 1 \quad \text{and} \quad m = i - 4(k-1).$$

The operator $n[x]$ converts x into an integer. This input is then coded into a node J in the F_2 layer by following steps 1) through 6) of the ART1 net algorithm.

Step 2) Slab #1 of the F_3 layer is then activated. This slab contains 16 cells corresponding to the number of features in the vector I. Each cell further contains 16 units connected to the Jth node in the F_2 layer. Let the weight matrix for the top left cell be W_1. Then, the elements of W_1 are given by

$$w_{km} = t_{Ji} \qquad i = 1 \text{ to } 16 \tag{6}$$

where k and m are defined as in (5).

The weight matrix for all the cells of slab #1 can be constructed from W_1 as follows:

$$\begin{bmatrix} W_1 & T_{r1}(W_1) & T_{r2}(W_1) & T_{r3}(W_1) \\ T_{d1}(W_1) & T_{r1}T_{d1}(W_1) & T_{r2}T_{d1}(W_1) & T_{r3}T_{d1}(W_1) \\ T_{d2}(W_1) & T_{r1}T_{d2}(W_1) & T_{r2}T_{d2}(W_1) & T_{r3}T_{d2}(W_1) \\ T_{d3}(W_1) & T_{r1}T_{d3}(W_1) & T_{r2}T_{d3}(W_1) & T_{r3}T_{d3}(W_1) \end{bmatrix} \tag{7}$$

where the operator T_{di} represents translation of each row of the weight matrix W_1 by i pixels down and T_{ri} represents translation of each column of the weight matrix W_1 by i pixels to the right. Thus, slab #1 incorporates invariance to top-down and left-right translations within the binary image.

Step 3) Additional slabs are created to achieve invariance to different rotations. For illustrative purposes, a slab structure that is invariant to 90° rotations is considered. Slab #1 always corresponds to a rotation of 0°. So, the slab structure will contain an additional three slabs (one each for 90°, 180°, and 270° rotations).

For slab #2, the weight matrix for the top left cell will be $W_2 = R_{90°}(W_1)$ where $R_{90°}$ is an operator that rotates the elements of W_1 by 90°. The weight matrices for all the cells of slab #2 is obtained by replacing W_1 in (7) by W_2. Similarly, the weight matrices for slabs #3 and #4 are obtained by replacing W_1 by W_3 and W_4, respectively, where $W_3 = R_{180°}(W_1)$ and $W_4 = R_{270°}(W_1)$. The entire slab structure required to achieve invariance to all translations and 90° rotations is shown in Fig. 2.

Step 4) A cluster of 64 nodes are formed in place of node J in the F_2 layer. These new nodes have a one to one correspondence to cells of the four slabs formed in step 3). The top-down LTM trace for each new node is obtained from the weight matrix of its corresponding cell. For example, to obtain the top-down LTM trace of a new node L at F_2 corresponding to the top-left cell of slab #2 we have

$$t_{Li}^{\text{new}} = w_{km} \tag{8}$$

where w_{km} are the elements of W_2 and k and m are defined as in (5).

The bottom-up LTM trace for node L is derived as

$$b_{Li}^{\text{new}} = \frac{I_i'}{0.5 + \sum_{i=1}^{16} I_i'} \tag{9}$$

where the elements I_i' are obtained as follows:

$$I_i' = R_{90°}(B) = B_{km}' \qquad i = 1 \text{ to } 16 \tag{10}$$

where B_{km}' are elements of the matrix $R_{90°}(B)$ and k and m are defined as in (5). The bottom-up and top-down LTM traces for all the new nodes at F_2 can be derived using the above procedure.

Step 5) All the elements of the weight matrices in the F_3 layer are reset to zero, and the program goes back to step 1) to process the next input.

As mentioned before, the concept of manufacturing invariance at F_3 is the same as the invariance net developed by Widrow *et al.* [7], [8]. However, there are some major differences between the invariance net and the modified ART network presented in this paper. First, the ADALINES [7], [8] (analogous to the cells of each slab in the new network) in each slab of the invariance net produce a single bit output from each slab and ultimately map the binary image to a different image. The slabs of the F_3 layer produce a cluster of nodes in the F_2 layer that act as feature detectors themselves. Second, the weight matrix of the top-left ADALINE of each slab is chosen randomly in the invariance net and hence produces invariant maps that are totally intolerant to noise. On the contrary, the weight matrix for the top left cell of each slab in the new network is based on the LTM traces that coded the first appearance of an input at the F_2 layer. This helps produce highly noise-tolerant recognition codes. And third, the invariance net is computationally more intensive than the proposed F_3 layer. To illustrate this, consider an input image with $N \times N$ pixels. Also, assume that the user requires the network to be insensitive to M rotations. Then the total number of weight connections required by an invariance net is MN^4.

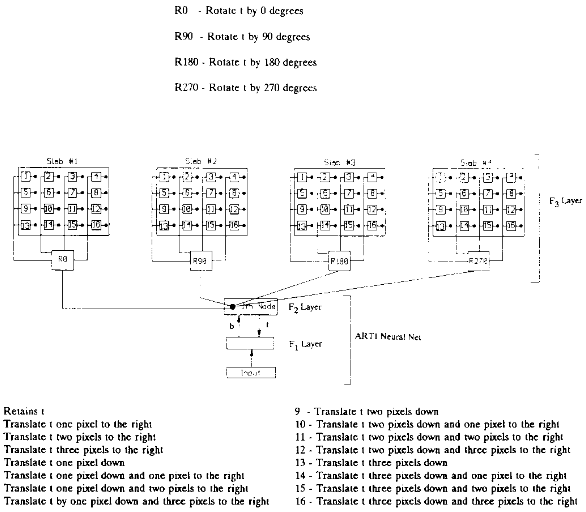

R0 - Rotate t by 0 degrees
R90 - Rotate t by 90 degrees
R180 - Rotate t by 180 degrees
R270 - Rotate t by 270 degrees
Slab #1
Slab #2
Slab #3
Slab #4
F_3 Layer
R0
R90
R180
R270
jth Node
F_2 Layer
b
t
F_1 Layer
ART1 Neural Net
Input
1 - Retains t
2 - Translate t one pixel to the right
3 - Translate t two pixels to the right
4 - Translate t three pixels to the right
5 - Translate t one pixel down
6 - Translate t one pixel down and one pixel to the right
7 - Translate t one pixel down and two pixels to the right
8 - Translate t by one pixel down and three pixels to the right
9 - Translate t two pixels down
10 - Translate t two pixels down and one pixel to the right
11 - Translate t two pixels down and two pixels to the right
12 - Translate t two pixels down and three pixels to the right
13 - Translate t three pixels down
14 - Translate t three pixels down and one pixel to the right
15 - Translate t three pixels down and two pixels to the right
16 - Translate t three pixels down and three pixels to the right

Fig. 2. Pattern of weight connections between the F_2 node that codes the binary input and the cells of the F_3 layer in order to achieve invariance to all translations and 90° rotations.

In contrast, the total number of weight connections required by the new network is MN^2. While this number is large, it would still be suitable for low resolution images.

V. Computer Simulations and Results

The modified ART architecture has been implemented in C on a VAX 11/750 station. For the purpose of simulating the modified ART architecture, the F_3 layer was structured to be invariant to all translations and 90° rotations. An array of 8 × 8 pixels was used as the image size for the binary input with the shaded squares having a 1 value and the nonshaded ones having a 0 value. This implies that the E_3 layer is made up of four slabs with each slab consisting of 64 cells. An input set consisting of four images "P," "M," "K," and "U" was presented to the modified ART architecture. The four input patterns were coded into four different categories as shown in simulation I of Fig. 3 for a vigilance greater than 0.5. Subsequently, a set of translated and rotated versions of these four images was presented. These images were again coded into the same four categories as shown in simulation II of Fig. 3. This indicates that the modified ART architecture is invariant to 90° rotations and translations. Also, the new network retains its self-stabilizing property by directly accessing the previously established categories of these four images.

In order to test the noise handling ability of the modified ART architecture, noise ranging from 8 to 30 percent (percentage measured as the ratio of noisy pixels to the total pixels contained in the uncorrupted image) was introduced to rotated and translated versions of the four images. It was found that the modified ART architecture was able to directly access previously established codes for each image without any errors. These results are summarized in simulations III, IV, and V of Fig. 3 and indicate that the modified ART architecture is noise-tolerant.

In order to test the modified ART architecture for its plastic properties, an unfamiliar set of inputs "N" and "O" were presented. These inputs were coded into two new categories. A subsequent presentation of rotated and translated versions of "N" and "O" resulted in direct access to their newly formed categories. These simulations indicate that the modified ART architecture is truly plastic to unfamiliar inputs. It should be noted that the order of presenting these images does not affect the coding process. For example, there can be a situation in which the first instance of an image encountered by the modified ART architecture is a rotated and translated version of its upright self. The new network has been tested for these cases and has behaved perfectly with no mistakes.

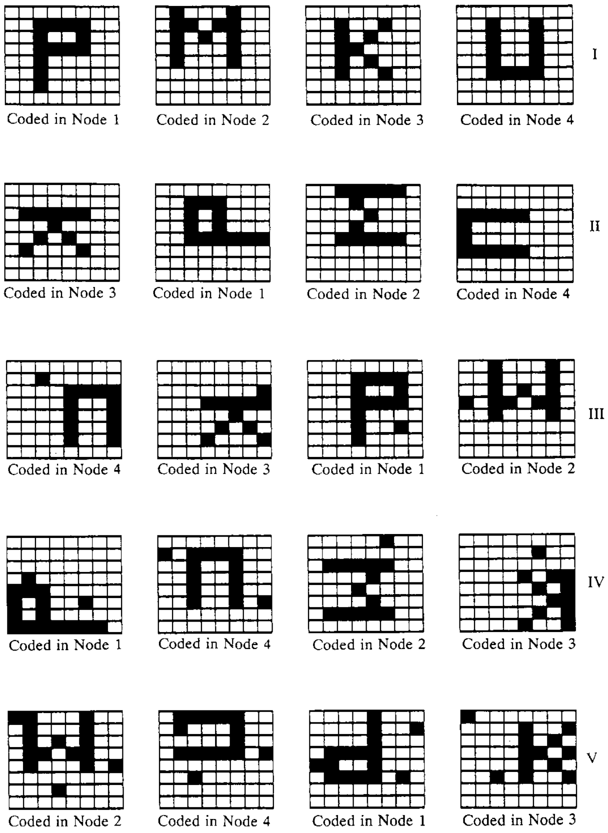
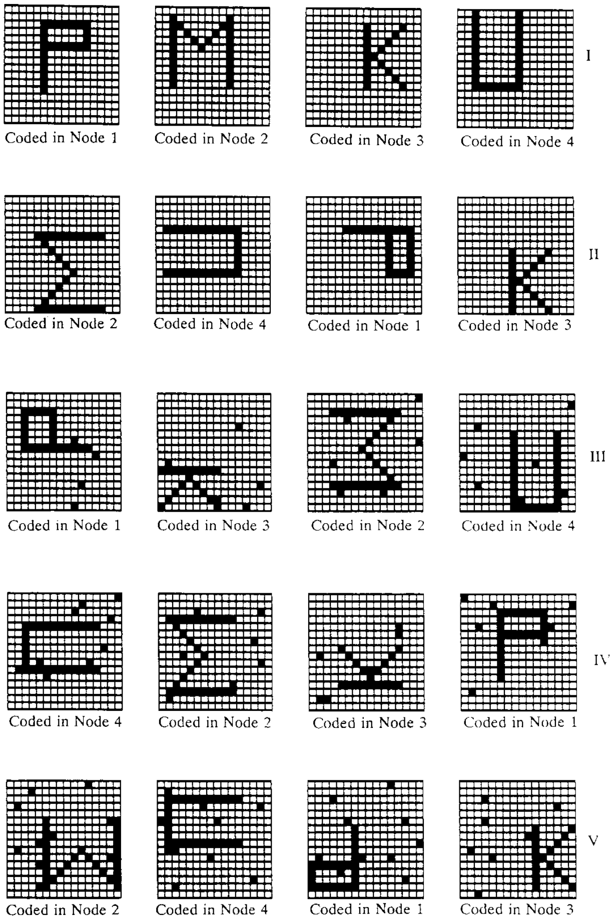


Fig. 3. Outputs of the modified ART architecture for an 8 × 8 images of "P," "M," "K," and "U" during simulations I through V. The noise level ranged from 8 to 30 percent.

For the purpose of testing the self-scaling property of the modified ART architecture, the simulations I through V conducted for an 8 × 8 image, were repeated for a 16 × 16 image. It was found that the modified ART architecture was able to correctly recognize all the twenty images (this includes images from all five simulations) independently as shown in Fig. 4. Also, the amount of noise tolerated improved from 30 percent for an 8 × 8 image to 45 percent for a 16 × 16 image. These results indicate that the modified ART architecture is more tolerant to mismatches in an image when the number of features that describe them increases. Thus, it can be seen that the modified ART architecture retains the self-scaling property of an ART architecture.

VI. Invariance to Size Through the Modified ART Architecture

In order to achieve invariance to size using the new network we propose a multislab structure as shown in Fig. 5. The inner most slab is similar to the slab structure discussed in the previous sections. All the other slabs within the multislab structure are invariant to a unique size of the input and are connected to the inner most slab independently. The total number of slabs within the multislab



Coded in Node 1 Coded in Node 2 Coded in Node 3 Coded in Node 4

Coded in Node 2 Coded in Node 4 Coded in Node 1 Coded in Node 3

Coded in Node 1 Coded in Node 3 Coded in Node 2 Coded in Node 4

Coded in Node 4 Coded in Node 2 Coded in Node 3 Coded in Node 1

Coded in Node 2 Coded in Node 4 Coded in Node 1 Coded in Node 3

Fig. 4. Outputs of the modified ART architecture for a 16 × 16 images of "P," "M," "K," and "U" during simulations I through V. The noise level ranged from 17 to 45 percent.

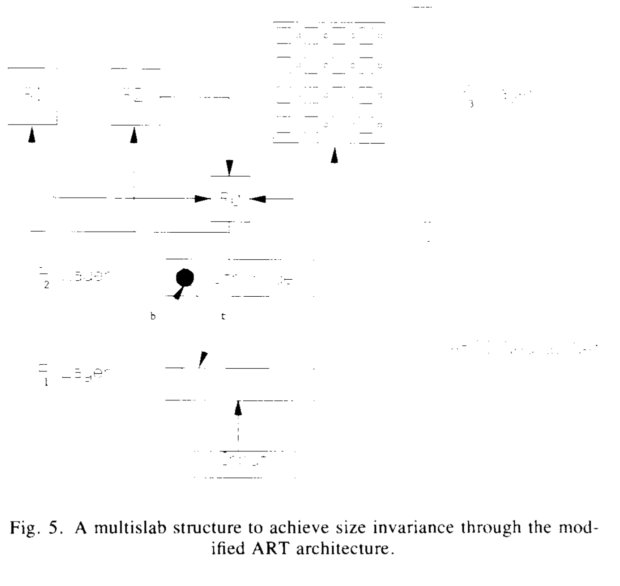


Fig. 5. A multislab structure to achieve size invariance through the modified ART architecture.

structure is equal to the sum of the total number of unique sizes (each slab is sensitive to a unique size) and the inner most slab.

These additional slabs are made invariant to different sizes by the nature of their weight connections to the nodes of the F_2 layer. The design of these connections is again based on the invariance net. The amplitude of weight connections between the nodes of F_2 and each slab are scaled in inverse proportion to the square of the linear dimension of the input image size. These additional slabs produce a set of nodes at F_2 with each node being sensitive to a unique size and position of the input image. The total number of nodes within a cluster at F_2 will now equal the total number of cells within the inner most slab at F_3 multiplied by the product of the number of the orientations and sizes that the user desires.

VII. Conclusion

An invariant pattern recognition machine is developed based on an ART architecture. A multislab layer called F_3 was added to an ART1 neural net to act as a vehicle to generate invariance in the F_2 layer. Each slab is made invariant to all translations but a single rotation only. The weight connections between the nodes of the F_2 layer and the cells within the slabs of the F_3 layer was based on a modified form of the invariance net. The new network was tested by feeding rotated and translated versions of 2-D binary images to an ART1 neural network. Results show that the new network is not only invariant to rotations and translations, but is also highly noise-tolerant and achieves invariant properties without altering the essential properties of an ART architecture. While the new network is computationally more efficient than the invariance net, there is an increase in processing time with an increase in resolution of the image. However, the proposed network is ideal to use for low-resolution images as it is purely based on software, and computers with powerful computational capabilities are rapidly becoming more affordable.

References

[1] D. Casasent and D. Psaltis, "Position, rotation and scale invariant optical correlations," *Appl. Opt.*, vol. 15, pp. 1793–1799, 1976.

[2] P. Cavanagh, "Size and position invariance in the visual system," *Perception*, vol. 7, pp. 167–177, 1978.

[3] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by a shift in position," *Biolog. Cybern.*, vol. 36, pp. 193–202, 1980.

[4] K. Fukushima and S. Miyake, "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position," *Pattern Recogn.*, vol. 15, pp. 455–469, 1984.

[5] T. Maxwell, G. L. Giles, and Y. C. Lee, "Nonlinear dynamics of artificial neural systems," in *Neural Networks for Computing*, J. S. Denker, Ed. New York: Amer. Instit. Phys., 1986.

[6] H. Szu, "Holographic coordinate transformations and optical computing," in *Optical and Hybrid Computing*, H. Szu, Ed. Bellingham, WA: *Int. Soc. Optic. Eng.*, The SPIE, vol. 634, 1986.

[7] B. Widrow, R. G. Winter, and R. A. Baxter, "Layered neural nets for pattern recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-36, pp. 1109–1118, July 1988.

[8] —, "Learning phenomena in layered neural network," in *Proc. 1st Int. Conf. Neural Networks*, San Diego, CA, 1987.

[9] A. Khotanzad and J. H. Lu, "Classification of invariant image representations using a neural network," *IEEE Trans. Acoust., Speech, and Signal Process.*, pp. 1028–1038, June 1990.

[10] B. Yegnanarayana and A. Ravichandran, "A two stage neural network for translation, rotation and size invariant visual pattern recognition," *Proc. ICASSP.*, May, 1991.

[11] G. A. Carpenter and S. Grossberg, "Invariant pattern recognition and recall by an attentive self-organizing ART architecture in a non-stationary world," in *Proc. 1st Int. Conf. Neural Networks*, San Diego, CA, 1987.

[12] N. Srinivasa and M. Jouaneh, "A neural network model for invariant

pattern recognition," *IEEE Trans. Signal Process.*, vol. 40, pp. 1595-1599, June 1992.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, vol. 1. Cambridge, MA: MIT Press, ch. 8, 1986.

[14] Y. H. Pao, *Adaptive Pattern Recognition and Neural Networks*. Reading, MA: Addison-Wesley, 1989.

[15] D. J. Sobajic, "Artificial neural nets for transient stability assessment of electric power systems," Ph.D. dissertation, Systems Engineering Department, Case Western Reserve Univ., Cleveland, OH, 1987.

[16] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, pp. 4-22, 1987.

[17] G. A. Carpenter and S. Grossberg, "The ART of adaptive pattern recognition by a self-organizing neural network," *Computer*, vol. 21, pp. 77-88, 1988.

[18] ——, "ART2: Self organization of stable recognition codes for analog input patterns," *Appl. Opt.*, vol. 26, pp. 4919-4930, 1987.

[19] ——, "ART 3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures," *Neural Networks.*, vol. 3, pp. 129-152, 1990.

[20] T. Kohonen, "Self-organized formation of topologically correct maps," *Biolog. Cybern.*, vol. 43, pp. 56-69, 1982.

[21] J. Anderson, "A simple neural network generating an interactive memory," *Mathemat. Sci.*, vol. 14, pp. 197-220, 1972.

[22] B. Kosko, "Adaptive bidirectional associative memories," *Appl. Opt.*, vol. 26, pp. 4947-4960, 1972.

[23] P. Kolodzy, "Multidimensional machine vision using neural networks," in *Proc. 1st Int. Conf. Neural Networks*, San Diego, CA, 1987.

[24] N. Srinivasa and M. Jouaneh, "An investigation of surface roughness characterization using an ART2 neural network," in *Sensors, Controls, and Quality Issues in Manufacturing*, PED vol. 55. Metals Park, OH: ASME, 1991, pp. 307-318.

[25] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput. Vision, Graphics, Image Process.*, vol. 37, pp. 54-115, 1987.

Classification-Based Segmentation of ZIP Codes

Yousuf Saifullah and Michael T. Manry

***Abstract*—In this paper a system for the segmentation of unconstrained handwritten ZIP codes is presented. A binarization algorithm is described that utilizes the bimodal nature of the input 256 gray level histogram. ZIP code images are scaled to a standard size for further processing. Objects that are not comparable to the size of numerals are classified as noise and removed. An algorithm that segments broken or touching characters is presented. Several trial segmentations of each ZIP code are produced. A character classifier is used to determine which trial segmentation is most likely to be correct.**

I. Introduction

Many handwritten character recognition systems [1]-[17] are presently in service or under development. Applications include mail sorting, automatic reading of forms such as tax forms, robotic vision, microfilm reading, and signature verification. The great variety of applications shows the critical importance of this subject. Rising labor costs have greatly increased the demand for automatic handling of these types of data.

In the real world, we encounter two types of handwritten characters: constrained and unconstrained. Constrained characters are written within specific boundaries, while unconstrained characters are not. The technology for reading constrained characters is highly developed, but that for reading unconstrained characters [3]-[10] is still under development because of its inherent difficulties.

The U.S. Postal Service has been interested in automatic mail sorting [8]-[13], [18]-[21] for many years. The complete operation of mail sorting can be broken down into several components; 1) location of the ZIP code on the package [18]-[21], 2) segmentation of a ZIP code image into constituent digits [4], [14], and 3) recognition of the ZIP code digits [3]-[12]. In order to facilitate research into segmentation of a ZIP code image into its constituent digits, the Postal Service Office of Advanced Technology has made available a database of ZIP code images, entitled "United States Postal Service Office of Advanced Technology Handwritten ZIP Code Database (1987)." These ZIP code images consist of arrays of varying size that are stored in 1 byte/pixel (256 gray level) format. Both five digit and nine digit ZIP codes are included. Each image has background and object (character) regions, plus noise or interference. The noise can be broken down into 1) random noise, 2) touching or overlapping parts of other numerals and characters, and 3) bars generated when the customer underlines the ZIP code.

In this correspondence we present a system of algorithms for the segmentation of a ZIP code image into its constituent digits. In Section II, the binarization and scaling algorithms are discussed. Noise removal algorithms are described in Section III. In Section IV, a method is given for distinguishing five digit ZIP codes from nine digit ZIP codes. A method for generating several trial ZIP code segmentations is described in Section V. A classification-based approach, for determining which trial segmentation is best, is given in Section VI.

II. Binarization and Scaling

Since we need to consider only two kinds of information in our ZIP code for the recognition task, "background" and "object," a binarization of the gray level image is the logical first step in processing. Binarization of images can be achieved using histogram approaches. Generally, in this approach, one searches for a threshold that separates background pixels from object pixels. ZIP code images often have bimodal histograms. Since background pixels are more likely than object pixels, background pixels produce the dominant mode. A natural choice for the thresholding point should be the lowest point in the valley between two modes, as seen in Fig. 1. Unfortunately the object mode is not always distinguishable from the valley.

There are many binarization methods, but they have problems. The *iterative threshold selection algorithm* [22], [23], which assumes that the histogram is bimodal and that background and object pixels are equally likely, is a poor choice because background pixels are much more likely than object pixels in ZIP code images. The *discriminant analysis method* [24] works well when a distinct contrast between the two classes exists, otherwise the optimization results tend to mix object and background pixels to a great extent. We want to generalize our algorithms so that they can work on low contrast images also, so *discriminant analysis* is not a good choice. *Entropy-based techniques* [25], [26] are not suitable because they tend to blur objects, resulting in a loss of loops and sharp contours. We have attacked these problems by using a cumulative histogram method.

Manuscript received October 17, 1991; revised February 3, 1993. This work was supported by the State of Texas through the Advanced Technology Program.

The authors are with the Department of Electrical Engineering, University of Texas at Arlington, TX 76019.

IEEE Log Number 9209684.

0018-9472/93$03.00 © 1993 IEEE