PERGAMON

Pattern Recognition 34 (2001) 841–853

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Shape recognition using an invariant pulse code and a hierarchical, competitive neural network[☆]

N. Rishikesh, Y.V. Venkatesh*

Computer Vision and Artificial Intelligence Laboratory, Department of Electrical Engineering, Indian Institute of Science, Bangalore-560 012, India

Received 4 May 1999; received in revised form 4 January 2000; accepted 4 January 2000

Abstract

The paper deals with the invariant recognition of patterns, and aims at developing (i) their pulse-coded representation; and (ii) an algorithm for their recognition. The proposed pattern encoder utilizes the properties of complex logarithmic mapping (CLM) (computed with reference to the *center of gravity*, CoG, of the shape), which *maps* rotation and scaling in its domain to shifts in its range. The encoder, then, invokes a pulse-encoding scheme similar to the one proposed by Dodwell [1] in order to handle these shifts, thereby generating pulse-codes invariant to scaling, rotation, and shift in the input shape. These pulses are then fed to a *novel* multi-layered neural recognizer which (i) invokes template matching with a distinctly implemented architecture; and (ii) achieves robustness (to noise and pattern deformation) by virtue of its overlapping strategy for code classification. The proposed encoder–recognizer (E–R), which is hardware implementable by a high-speed electronic switching circuit, can add new patterns on-line to the existing ones. The E–R is illustrated with experimental results. *While human visual system has been the main motivation to the proposed model*, no claim, *however*, *has been made on its direct biological plausibility.* © 2001 Pattern Recognition Society. Published by Elsevier Science Ltd. All rights reserved.

Keywords: Complex logarithmic mapping; Invariant code; Neural networks; Pattern recognition; Pulse coding of shape; Shape representation; Template matching

1. Introduction

Choice of a data structure, and its application to represent data play an important role in data analysis. For instance, representing an image in the transform domain paves the way for either image compression or spectral feature extraction or both. In computer vision, there has arisen a need to design a representation scheme to facilitate pattern recognition. Motivated by the remarkable characteristics of the human vision system (HVS), attempts have been made to model it, and to discover pattern representation in the human brain. Neuro-anatomical and psycho-physical experiments have revealed that visual processing for perception is distributed in the human brain. However, the coding of image information in the visual cortex, its relationship with the retinal input, and the nature of the feedback among the higher and lower functions of the visual pathway are still largely unknown.

In this paper, we deal with the problem of pattern (or shape) recognition, which is one aspect of image analysis. This problem can be decomposed into two stages: encoding for representing a contour; and matching the generated code with the known or stored models/templates of the patterns in the class of objects under consideration. The representation scheme presented here corresponds to an invariant pulse-encoding procedure. The coding procedure employs the complex logarithmic mapping (CLM) and a pulse encoder to achieve its novel

[☆] An earlier, brief version of this paper appeared in [15].

* Corresponding author. Tel.: + 91-80-3092572; fax: + 91-80-3600444/3600683.

E-mail addresses: rishi@ee.iisc.ernet.in (N. Rishikesh), yvvele@iris.ee.iisc.ernet.in (Y.V. Venkatesh).

PII: S0031-3203(00)00037-6

characteristic of generating pulses invariant to scaling, rotation and shift in patterns. The output of the pulse encoder acts as the input to a pattern recognition scheme, which is a multi-layered neural architecture capable of "adding" new patterns on-line. The complete block diagram representation of the approach is shown in Fig. 1.

The paper is organized as follows:

1. Section 2 describes the CLM and its mathematical properties, and also provides a brief survey of the literature on the CLM's applications to pattern (shape) encoding.
2. Section 3 examines the shape recognition scheme proposed by Dodwell [1]. The proposed encoding scheme employs some of the ideas found in Ref. [1].
3. Section 4 discusses the strategy employed by the proposed coder to generate invariant pulses. The output of the coder is shown to be a unique set of pulses, associated with the given pattern, which is invariant to rotation, scale and shift.
4. Section 5 presents the proposed recognition scheme, which is adaptive in nature, in the sense that it is able to add new patterns on-line. The proposed algorithm is (a) similar to template matching in the pulse domain, but distinct from template matching in its implementation; and (b) amenable to high-speed hardware implementation, in view of the facts that (i) the input is in the form of pulses; and (ii) the proposed strategy is efficient.
5. Illustrative experimental results are given in Section 6.
6. Section 7 concludes the paper with a summary of the new framework for pattern representation and recognition.

2. Complex logarithmic mapping

The proposed scheme for coding patterns utilizes the properties of the CLM to map rotation and scaling in its domain to shifts in its range. The CLM is described by $w = \log z$ or $w = \log \rho + j(\theta + 2k\pi)$ where w and $z = \rho e^{j(\theta + 2k\pi)}$ are complex variables, with ρ and θ denoting the polar co-ordinates of z, and k, an integer. This definition indicates that rotation and scaling in the z-domain would mean shifts along the θ-axis and log ρ-axis, respectively, in a $\theta - \log \rho$ plot. These properties of the CLM hold only when the mapping-origin is *well-defined* with respect to the contour. We employ the *center of gravity* (CoG) of the contour as the mapping-origin, since it offers a simpler method of ensuring invariance of the mapping with respect to shifts in the contour position.

See Refs. [2–4] for typical transform-based techniques to handle pattern shifts. And, for a template-matching approach, see Ref. [5]. These techniques are found to be time-consuming and sensitive to noise.

It should be added here that the role of the the CLM in modelling some aspects of the HVS has been of considerable interest to the computational vision community. The space-variant nature of the mapping, which projects the retinal image to the striate cortex, has triggered a flurry of models. Schwartz [6,7] approximates the retino-striate mapping using CLM, and suggests that both the primary and secondary visual cortex, the inferior pulvinar, and the somatotopic cortex represent a CLM of the sensory receptor surface onto a central neural surface. Further, he argues that the complex-log mapping provides an accepted model for this retinal-striate mapping in primates at both the local (hypercolumn) and global (retinotopic) representation scales.

3. Dodwell's encoding scheme

It has been found that the explicit times at which action potentials occur in the human brain are also important to the functioning of the neurons. In [8], such a time-based representation (or *temporal code*) is believed to be responsible for the functioning of the movement-sensitive visual neurons, and for the spatial localization of the sounds in the barn owl. Another model, distinct from the temporal model, refers to the *firing rate of neurons* [9] which is supposed to contain information on the "whats", even while the average firing rate remains unchanged.

However, the motivation for the present work arose from Dodwell's shape recognition model [1] (see also Refs. [10,11]) which employs a temporal contour coding scheme as the representation framework.

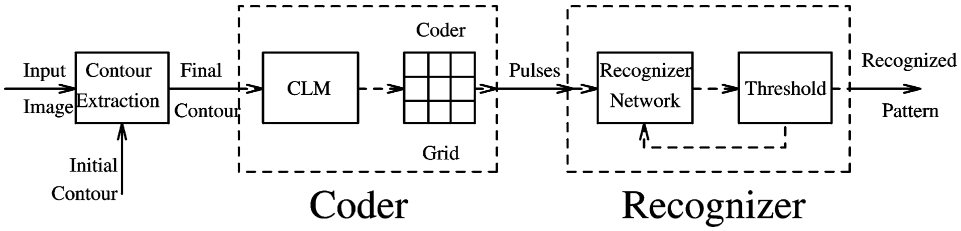

Fig. 1. Block diagram representation of the proposed approach.

Dodwell's model embodies a two-dimensional array of units, referred, in the sequel, as the coder grid. The units in the coder grid are connected in chains, so that each unit has at most two neighbors, one from which it receives excitation, and the other to which it "transmits" excitation. Each horizontal (vertical) chain is so organized that, whenever a horizontal (vertical) contour is projected onto the grid, the units in the grid which are thereby activated all lie on one chain.

Fig. 2 represents a typical coder grid and only horizontal chains are explicitly shown. Let the dimensions of the coder grid be $r \times c$, where r represents the number of rows and c represents the number of columns. Let the units belonging to the left-most column of the grid (represented by A in Fig. 2) be called as the end-set of the grid. The end-set units have the following properties:

- They are the originators of excitation. They transmit excitation to their neighbors but do not receive from any unit.
- Direction of transmission is from the end-set towards the output of the coder (shown by an arrow in Fig. 2).
- The end-set units fire simultaneously.

The other units in the grid transmit this excitation to their respective neighbors along the direction of transmission after a constant time which is equal to

(a) T_p seconds, if they are not activated by contour projections, and
(b) T_a $(> T_p)$ seconds, otherwise.

Consider the case where none of the units on the coder grid are activated by contour projections. Here, firing of the end-set A induces a sweep, or scan, of the array at a constant rate along each chain which will reach the coder output simultaneously. The output, then, will be a single large pulse of amplitude r units, occuring at a constant time $c{*}T_p$ seconds after end-set firing.

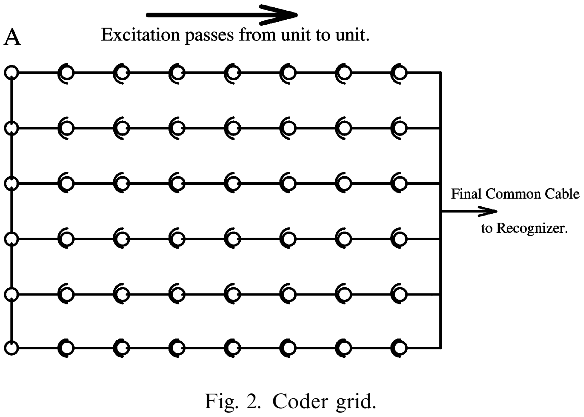

Fig. 2. Coder grid.

On the other hand, in the case where some of the units in the grid are activated by contour projections, the transmission periods along different chains of the grid will be altered depending on the number of units activated in that chain. For instance, consider a simple case where a horizontal contour of length $n < c$ is projected on *one horizontal* chain of the grid. This activates n units on that chain, and hence n units of that chain will induce delays of T_a seconds while $(c - n)$ units will induce delays of T_p seconds. The coder's output, then, is an initial pulse, of amplitude $(r - 1)$ units occuring at a time $c{*}T_p$ seconds after excitation from the end-set, followed by a signal of unit amplitude occuring at a time $(c - n){*}T_p + n{*}T_a = c{*}T_p + n{*}(T_a - T_p)$ seconds after excitation from the end-set. It may be noted that the initial pulse is due to the unactivated $(r - 1)$ horizontal chains and the delayed one (the delay being $n{*}(T_a - T_p)$) is as a result of the activated horizontal chain.

3.1. Two functionally perpendicular scans

The coder grid incorporating just the horizontal sweep is disadvantageous because it generates *similar* outputs for some patterns, which, on common sense grounds, are expected to be discriminable. For example, any parallelogram of the same height, with two horizontal sides (of constant length) generates the same output, as it *also* happens with the pattern of the "diamond" and any pair of non-horizontal straight lines of the same vertical extent. This difficulty is resolved in Ref. [1] by postulating an array, in which units on the same chain are activated by vertical contours falling on the recognizer. The second array is thus functionally perpendicular to the first. The coder, hence, has two separate outputs which may be treated as corresponding to horizontal and vertical sweeps/scans of the coder grid.

Fig. 3 illustrates the coder grid function; Fig. 3(a), a typical coder grid, with units (colored black) activated corresponding to presence of contour points; and Fig. 3(c), the pulse-code sequences corresponding to the horizontal and vertical scans of the grid. It may be noted that $t = 0$ in Fig. 3(c) corresponds to the initial pulse corresponding to the unactivated channels. Fig. 3(b) shows the units in the coder grid activated with the contour rotated, and Fig. 3(d) the corresponding pulse-code sequences. Considerable differences between the sequences in Figs. 3(c) and (d) confirm that the *present* grid-encoding is *not invariant* with respect to rotation of the contour points.

3.2. Properties of the pulse-code

Some of the distinct properties of the pulse-code generated by the coder grid may be listed as follows:

- The maximum amplitude of any pulse obtained with horizontal (vertical) scan is equal to the number of rows (columns) in the coder grid.

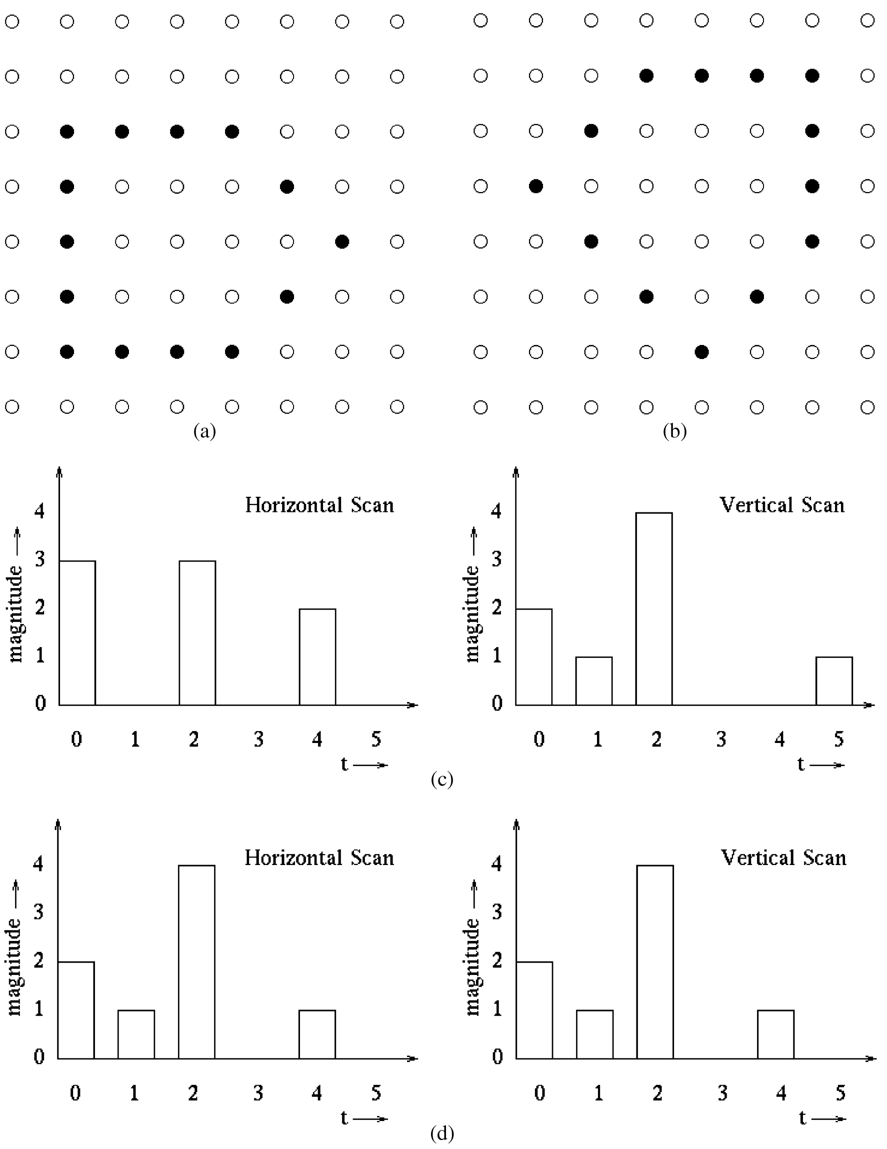

Fig. 3. (a) Operation of the coder grid: grid with activated (colored black) units for contour points. (b) Operation of the coder grid: grid with activated (colored black) units for rotated contour points. (c) Illustration of coder grid's function: pulse-codes corresponding to the horizontal and vertical scans of the grid shown in Fig. 3(a). (d) Illustration of coder grid's function: pulse-codes corresponding to the horizontal and vertical scans of the grid shown in Fig. 3(b).

- This maximum amplitude occurs when all rows (columns) have the same number of activated units. The position on the time-axis of this pulse will depend on the exact number of activated units.
- The maximum effective pulse width with respect to the horizontal (vertical) scan[1] is equivalent to the number of columns (rows) in the coder grid.
- The coder's output is not affected by translation in either axis.

However, it is found that Dodwell's [1] coder output is *unsatisfactory* for the following reasons:

1. It is not invariant to rotation and scaling of the contour.
2. Even a slight change in the input contour will distort the output code, thereby making it impossible to recognize the contour correctly.

[1] Effective pulse width is defined as the time at which the last pulse of non-zero amplitude occurs.

3. As the contour is fed directly to the coder, the coder array size will be very high, thereby yielding a longer pulse.

4. Proposed encoding scheme

In an attempt to avoid the disadvantages associated with (i) Dodwell's coder grid; and (ii) the various shift-handling strategies discussed in Section 2, and in order to provide a invariant pulse code for shape, we propose an application of Dodwell's coder grid in handling the shift after the CLM. Another attractive feature of the pulse-coding scheme presented below is its amenability to hardware implementation using high-speed electronic circuitry.

In the proposed scheme, the input is to be a set of points corresponding to the contour (shape) of interest. This, in turn, necessitates the extraction of the contour from a given image. A contour extraction algorithm (similar to the one presented in Ref. [12]) can be used for this purpose. The resulting two-dimensional coordinates of the contour points are fed to the Coder (Fig. 1) which maps them using the CLM with respect to the CoG of the contour. The use of the CoG as the base for the CLM enables shift invariance at this stage itself. As mentioned earlier, in order to achieve *complete invariance*, we need merely to handle the shifts that arise as a result of rotation and/or scaling of the (input) contour.

Having analyzed the disadvantages associated with the various shift-handling strategies (Section 2), and having observed that Dodwell's encoder grid is invariant to shift in both the axes (Section 3), we now propose an application of Dodwell's coder grid in handling the shift after the CLM. Explicitly, after mapping the contour using CLM, which converts rotation and scale in its domain to shift in its range, we input the CLM coefficients to a coder grid similar to that presented above, which is shift-invariant. This generates pulses which are *completely invariant* to shift, scaling and rotation in the input pattern. Thus, the goal of coding shape into invariant pulses is accomplished by (i) mapping the input shape using CLM; and (ii) feeding the CLM output to a coder grid, which is shift-invariant and generates pulse output. In order to avoid ambiguity, two functionally perpendicular pulse sequences obtained by vertical (column-wise) and horizontal (row-wise) scans of the grid are used for the representation (see Section 3).

The advantages of the proposed coder are:

1. The encoded pulses are invariant to rotation, scale and shift of the input shape.
2. Pulse outputs facilitate fast recognition with the help of high-speed (nano-second range) hardware circuits.
3. The size of the array used in the coder (and hence the size of the pulse) is small compared to that of Dodwell's coder since the mapping used is logarithmic.

5. Recognition of pulses

The problem of recognition of pulses can be formulated as follows: Given, two sequences of pulses, whose amplitude and the time of occurrence contain information about the class to which the pattern belongs, we need to design a recognizer to classify the pulses, and hence the patterns.

Some desirable properties of such a recognizer are: (i) the time taken for recognition should be small; (ii) it should be able to accommodate a new pattern on-line, in case there is no match for it in the database; and (iii) the recognizer should be robust to the noise in input pulses caused by the finiteness of the coder grid. (This noise leads to improper invariance of the codes with respect to various instances (rotated, scaled versions) of a given contour.) See Ref. [13] (time-delay neural network), and [1] (Uttley's classifier) for some of the approaches available in the literature. We now propose a new scheme for recognition of the pulse-code generated by the coder.

5.1. Proposed recognizer

Apart from the desire to overcome the limitations of existing schemes for pattern (encoding and) recognition (described above), another motivation for the proposed recognizer arose from the physiological findings of the cerebral cortex of mammals. It is now known that the striate surface of the cerebral cortex represents a processing sub-unit, called a *hypercolumn* by Hubel and Wiesel [14]. Each hypercolumn (which contains tens of thousands of cells) can be thought of as a cluster of many constitutent columns of cells, each one extending from the surface of the cortex down through to the white matter below. Thus, each hypercolumn is a small block of cells, organized into column sub-units, and concerned with many different types of features. However, it appears that no reference in the literature (on hypercolumns) has been made to the processing of temporal features by these hypercolumns.

In order to design a neural network which has the desirable properties specified above, our attempt is to hypothesize such an architecture for the recognition of the temporal codes of shapes. To this end, a novel neural-based network with competitive layers (reminiscent of hypercolumns) is described for recognizing the pulses generated by the coder. The proposed approach is similar to template matching in the sense that the sequence of pulses corresponding to the unknown pattern is matched with the 'templates' of pulses in the columns[2] of the network. However, the proposed recognition scheme is

[2] See explanation below.

quite distinct from the standard template matching scheme on the following grounds:

1. An overlapping strategy is used for pulse classification in order to achieve robustness. This is a sort of memory-based system architecture. Instead of having a single unit decide the label of the pattern on the basis of a feature, two units use the feature along with another feature. In effect, the decision is 'distributed' which contributes to robustness. It is expected that larger the overlap, better will be the robustness.
2. A decision at one stage influences the decision on the succeeding stage for handling ambiguities efficiently.
3. Lateral inhibition and a competitive, layered network have been used for reducing the execution time. In order to simplify the diagram, the figure does not show the lateral connections explicitly. Competitiveness in behavior of the elements implies that lateral inhibition has been employed.
4. Table 1 gives *explicitly* the weights to be used.

5.2. Recognizer architecture

The proposed recognizer utilizes a multi-layered neural architecture, shown in Fig. 4, for its classification purposes. Each layer of the neural network is a competitive layer, in which neurons compete among themselves to win. The competition, with respect to the input fed to that layer, is effected by means of a lateral inhibition scheme, which is the key to the speed of the network.

Another interesting feature of the network arises from its columnar[3] structure. Each and every column of the network corresponds to one particular pattern. The number of layers in the network is one less than the sequence length (L_{inp}) of the pulses, and the number of columns is equal to the number of patterns to be stored. If the two input sequences are not of equal length, then the one with a smaller length is padded with zeroes to make it equivalent to the other.

The structure of each neuron in the network is shown in Fig. 5. As shown, the neurons have a set of inputs (with weights associated with each of them) and an output. The two inputs to the network correspond to the pulse sequences obtained by row- and column-wise scans of the coder grid (see Section 3). Note that, for simplicity, other lateral inhibitory inputs have been omitted. Their inherent structure is to be identified for shape recognition.

In order to achieve robustness in recognition of the patterns, the inputs fed to the different layers of the network are overlapped. That is, if the layer t receives pulses t and $t+1$ of the two input sequences, then layer $t+1$ receives pulses $t+1$ and $t+2$ of the input sequences. In fact, the network is designed to have $L_{inp} - 1$ layers in order to accommodate this overlap of the inputs presented to the various layers. These constitute four

Table 1
Weights of a recognizer neuron

Notation	Pulse number	Scan
w^p_{ct}	t	Column
$w^p_{c(t+1)}$	$t+1$	Column
w^p_{rt}	t	Row
$w^p_{r(t+1)}$	$t+1$	Row

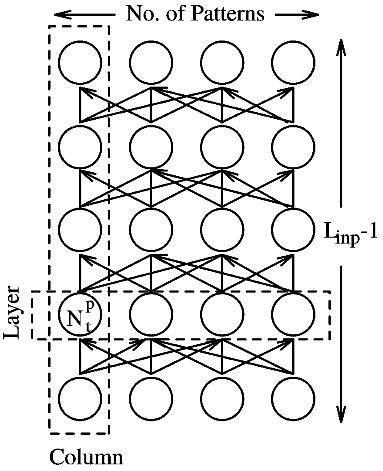

Fig. 4. Recognizer network's architecture (in which, for simplicity, no lateral connections are shown).

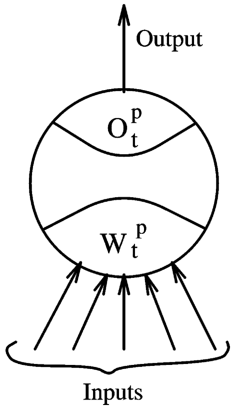

Fig. 5. Structure of a single neuron in the recognizer's network (for simplicity, lateral inhibitory inputs have not been shown).

[3] A column of neurons corresponds to a set (whose cardinality is equivalent to the number of layers) of neurons, taken one from each layer of the network.

inputs to each of the neurons. In the case of the first layer, the weight vector is four dimensional, whereas in the higher layers, it is five dimensional. In the case of layers other than the first layer, the fifth input comes from the output of the winner neuron of the previous layer. Lastly, O_t^p is the output associated with the neuron in the tth layer and pth column (which is propagated to the next layer when the neuron "fires"), and $\mathbf{w}_t^p$ is the weight vector associated with it. (The neuron of Fig. 5 has five inputs (*excluding the lateral signals for inhibition*), which indicates that it belongs to the inner layers of the network.)

The components and values of the weight vector associated with the neuron in the pth column of the tth layer of the network are listed in Table 1.

The above table gives the notation for the weights and their values. For example, w_{ct}^p takes as its value the amplitude of the tth pulse of the column-wise scan with respect to the pth pattern. The first layer has only the above-listed four weights. The other layers have five weights, and hence a five-input neuron, with the fifth weight represented by w_{ot}^p, and output O_t^p, is associated with the neuron under consideration. The fifth input to these neurons is the output from the neuron which had fired in the previous layer. The output associated with each neuron is the *pattern number* of the pattern p whose pulse amplitudes are stored as weights in the neuron. For implementation, the pattern number could be p itself.

5.3. Recognition algorithm

The notation used is as follows:

L_{inp}	input sequence length
N_t^p	neuron in the tth layer of the pth column
O_t^p	output associated with N_t^p
$\mathbf{w}_t^p$	weight vector associated with N_t^p
$\mathbf{i}_t$	input vector presented to the neurons in layer t
N_t^k	winner neuron of the layer t of the network.

A similar notation applies to the weights and output of the neuron. The confidence measures are represented by $\mathscr{C}_p$, with the suffix showing that a confidence value is associated with each of the patterns associated. The components of the weight vector are represented as in Section 5.1.

The recognition algorithm runs as follows:

1. Initialize the network with a few patterns by assigning as many number of columns of neurons, the number of layers being one less than L_{inp}.

   Assign the weights and outputs associated with each of the neurons as mentioned in Section 5.1.
2. Initialize all the Confidence Measures $\mathscr{C}_p$ to zero.
3. Feed the new input pattern to be recognized as the input to the network. (This requires a wait of one time-period initially, because the first layer needs the first and second pulses. From the second layer onwards, the pulses are processed as they come.)
4. Find the winner neuron in the first layer by calculating the distance measure as the Euclidean distance between the weight vector and the input vector. The winner neuron N_1^k alone *fires* and transmits its output to the next layer.

   Subtract the distance measure of individual neurons from the confidence measure corresponding to the pattern number stored in it (i.e., its output).
5. Calculate the *weighted distance* between the input vector and the weight vector of the neurons in the other layers:

   $$(\lambda_1 \|\tilde{\mathbf{w}}_t^p - \tilde{\mathbf{i}}_t\|^2 + \lambda_2 \delta(O_{t-1}^k, w_{ot}^p))^{1/2}, \qquad (1)$$

   where λ_1 and λ_2 are weighting parameters, with $\lambda_1 + \lambda_2 = 1$, and $\lambda_1 \geqslant 0.5$ and $\lambda_2 \leqslant 0.5$, thereby giving more weightage to the input vectors. This is done with a view to avoid propagation of noise to the higher layers caused by noise in the lower layers. $\tilde{\mathbf{i}}_t$ corresponds to the four-dimensional component of the five-dimensional input vector $\mathbf{i}_t$, excluding the one from the previous layer's winner. $\tilde{\mathbf{w}}_t^p$ is the corresponding four dimensional weight vector excluding w_{ot}^p. $\delta(x, y)$ is defined as:

   $$\delta(x,y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases} \qquad (2)$$

   Declare the neuron which has the minimum of this weighted distance as the winner.

   Propagate this winner's output forward to the next layer, and repeat the step till the final layer is reached.

   Subtract this weighted distance (in (1)) of individual neurons from the confidence measure corresponding to the pattern number stored in it.

   After the final layer's computation is completed, divide the confidence measures by the number of layers in the recognizer. (This is to be done in order to have the same threshold values for different lengths of input sequence.)
6. Find the largest two confidence measures among $\mathscr{C}_p$'s, and label them as $\mathscr{C}_i$ and $\mathscr{C}_j$ ($\mathscr{C}_i < \mathscr{C}_j$). If $\mathscr{C}_j > \mathscr{T}_{\mathscr{C}}$ *and* if $(\mathscr{C}_j - \mathscr{C}_i) > \mathscr{T}_d$, then declare that the new pattern's pattern number is j. Else, create a new column of neurons, and assign their weights (as defined in Section 5.1) appropriately with data from the present input.

   Generate a new pattern number, and treat it as the output to those neurons and the fifth weight of inner layers.

The threshold parameter $\mathscr{T}_{\mathscr{C}}$ is the lower bound on the largest of the confidence measures, for a pattern to be declared recognized. On the other hand, $\mathscr{T}_d$ is the lower bound on the difference between the two largest confidence measures. (Note that $\mathscr{T}_{\mathscr{C}} \leqslant 0$ and $\mathscr{T}_d \geqslant 0$.) In view of the fact that two inequalities are to be satisfied before arriving at a labelling decision, patterns which are ambiguous are discriminated if the ambiguity is *sufficiently* discernible, thereby demonstrating robustness of the scheme.

6. Results of shape recognition

The simulation studies on the coder and the recognizer, conducted on a *HP*9000/715 work-station, involve two different types of experiments. One of them tests the adaptability of the recognizer as new patterns come in. The other evaluates the ability of the recognizer to recognize correctly the rotated, scaled and shifted versions of the stored patterns. In this experiment, the percentage accuracy of recognition and the role played by the ratio of grid size to image size are studied.

The coding and recognition algorithms have been tested on the contours of 20 Indian states whose contours were extracted from the Indian map. This problem has been chosen because the shapes are irregular, and hence make the recognition quite complex. A few of the contours under consideration are shown in Fig. 6. Fig. 7 shows a rotated version of one of the contours (MP) shown in Fig. 6.

The CLM's of the untransformed contour, scaled version of the contour and rotated version of the contour are shown in Fig. 8. Fig. 8(a) shows the CLM of the untransformed version of the contour corresponding to Madhya Pradesh (MP of Fig. 6). From Fig. 8(b), which brings forth the changes that the CLM undergoes when the contour is scaled, it can be observed from the vertical axis that scaling amounts to shift along the vertical axis. While scaling up the contour leads to a positive shift, scaling down induces a negative shift along the vertical (log-magnitude) axis.

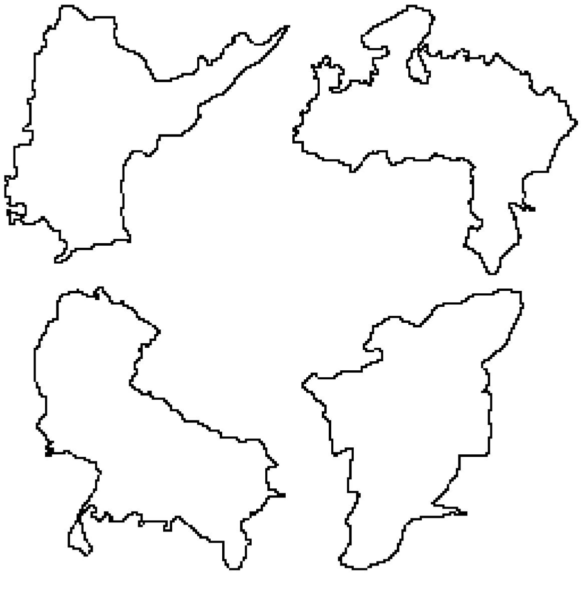

Fig. 6. Typical contours used in recognition experiments (clockwise from top left: AP, MP, TN and UP).

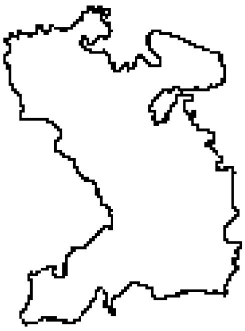

Fig. 7. Rotated version of one of the contours (MP) of Fig. 6.

A more interesting case arises with a rotated contour, whose CLM is shown in Fig. 8(c). As the angle varies from 0 to 2π radians, rotation of the contour *does not* mean shift of the CLM along the horizontal(angle) axis. Instead, a sort of *warping* takes place in the CLM output. Explicitly, a portion of the CLM is *cut* from the *back portion* and *pasted* at the *front*. This can be observed by noting the shapes of the CLM's shown is Figs. 8(a) and (c). This complexity in handling rotation, therefore, demands more than just shift invariance. But, the coder grid to which the CLM output is fed is capable of achieving it. Note that this invariance property can be attributed to the coder's dependency only on the "sweeping scan" along the entire set of rows(or columns) in generating the pulse-codes (see Section 4).

Having analyzed the effect of scaling and rotation on the CLM, we now present the actual pulse codes for the untransformed and rotated versions of a contour. Fig. 9 shows the pulse-codes for two different contours in Fig. 6 (Pattern 1 of Fig. 9 corresponds to UP of Fig. 6 and Pattern 2 corresponds to MP of Fig. 6). Fig. 10 shows the pulse-code for the (rotated MP of Fig. 6) contour shown in Fig. 7. Comparing Fig. 10 with the pulse-code for pattern 2 in Fig. 9, it is observed that the pulse-code for the rotated version is similar to that of the original pattern, thereby demonstrating the invariance property.

In all the experiments reported below, the value of λ_1 is set to a very high value, and $\lambda_2 = 1 - \lambda_1$. This is done in order to emphasize the importance of the input patterns rather than any possible noisy previous decisions. Typically, $\lambda_2 \in (0.001, 0.01)$. This apparently small value is

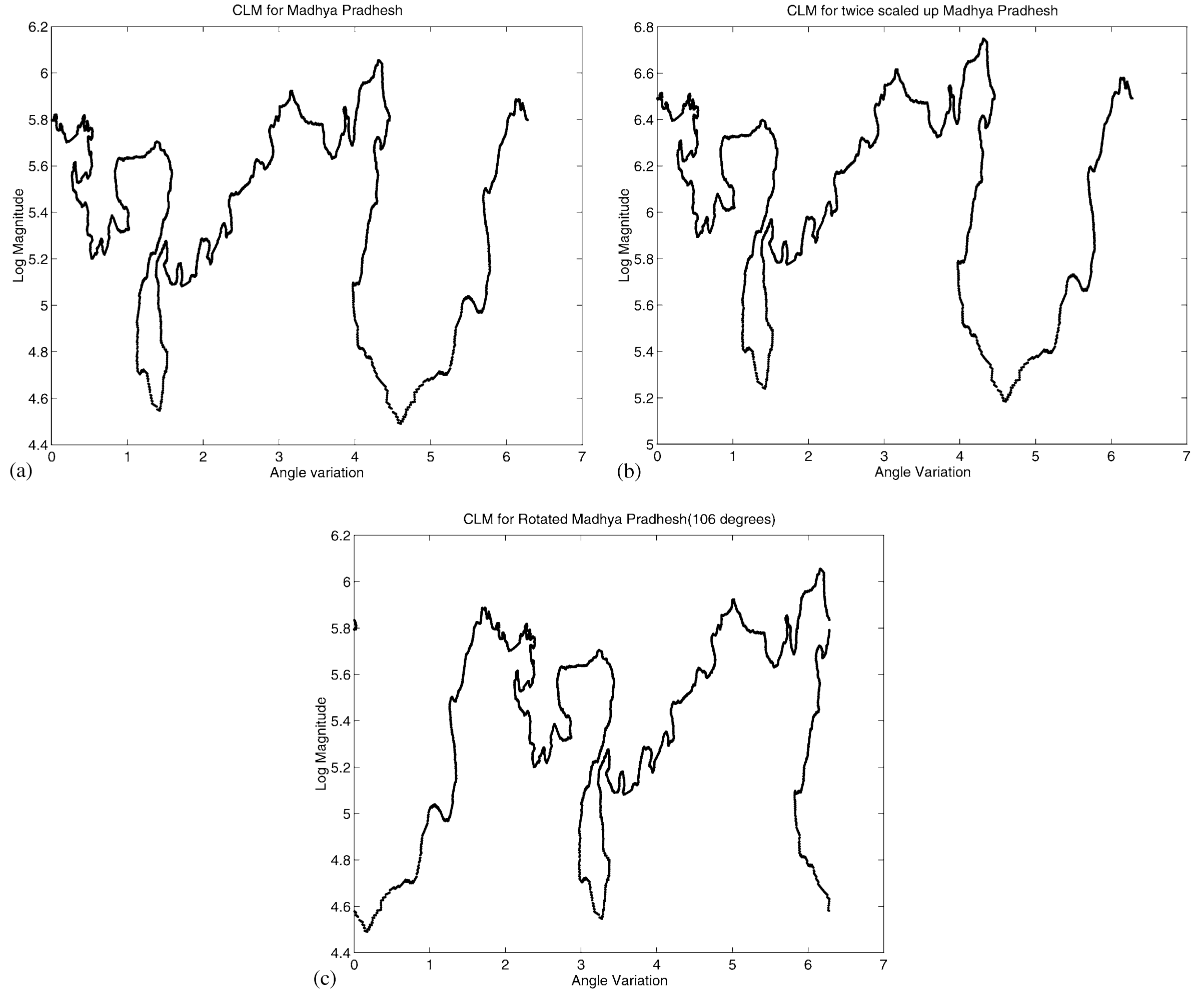

Fig. 8. (a) CLM of the untransformed version of the contour corresponding to Madhya Pradesh (MP of Fig. 6). (b) CLM of twice scaled up version of the contour corresponding to Madhya Pradesh (MP of Fig. 6). (c) CLM of rotated (106°) version of the contour corresponding to Madhya Pradesh (MP of Fig. 6).

chosen because the confidence measures ($\mathscr{C}_p$'s of Section 5.3) vary approximately from -0.5 to -3.0, and even a small value of λ_2 can resolve an ambiguity. On the other hand, if λ_2 is set to a high value, then it leads to undesirable results by assigning more weightage to previous decisions.

It may also be noted that the confidence measures ($\mathscr{C}_p$'s of Section 5.3) are nonpositive. This is evident from Step 5 of the algorithm given in Section 5.3, where we subtract the weighted distances from the zero-initialized confidence measures. This effectively implies that the threshold $\mathscr{T}_{\mathscr{C}}$ will be nonpositive, and $\mathscr{T}_d$ will be nonnegative (as evident from Step 6 of Section 5.3). Typical parameter values are: $\mathscr{T}_{\mathscr{C}} \in (-1.2, -0.7)$; and $\mathscr{T}_d \in (0.01, 0.03)$.

6.1. Test on adaptability

In the first experiment, the recognizer has, initially, a small number (say, five) of patterns stored in it. New and old patterns, selected randomly, are then presented to the recognizer after coding. Ideally, the recognizer has to add new patterns by increasing the number of columns, and should recognize the transformed versions of the old patterns correctly. However, due to the variation in the pulse code for transformed versions of the patterns (or due to the similarity of an old pattern to the newly arrived pattern), the recognizer may add transformed versions of old patterns (or recognize the new pattern to be an old one). In order to achieve a larger number of correct additions and a smaller number of wrong

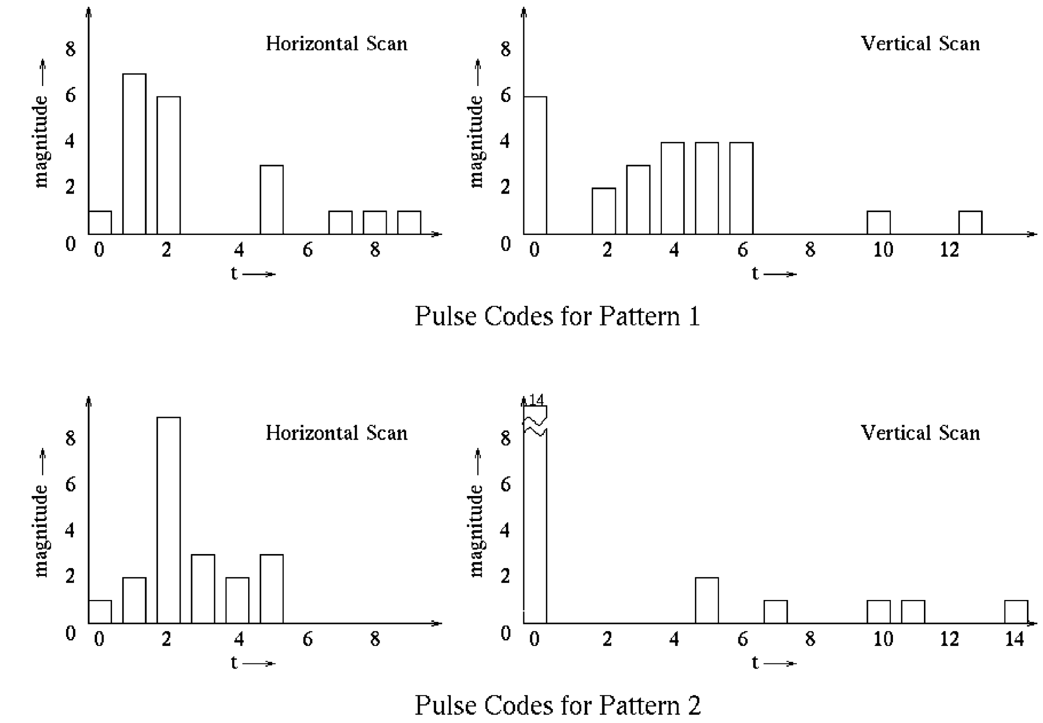

Fig. 9. Vertical and horizontal pulse-codes for two different patterns.

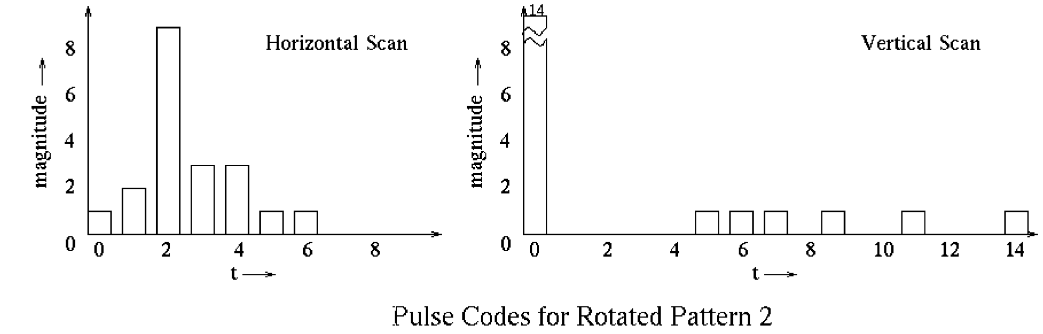

Fig. 10. Pulse-codes for a rotated version of pattern 2 corresponding to Fig. 9.

additions, we need to choose the threshold parameters correctly.

If the threshold parameters $\mathscr{T}_{\mathscr{C}}$ and $\mathscr{T}_d$ are very high, then the percentages of (i) correct additions (of new patterns); and (ii) wrong additions (of transformed old patterns) increase. (It should be noted here that, as the percentages of correct additions correspond to new patterns and that of wrong additions correspond to transformed old patterns, the sum of the two is not 100.) On the other hand, if these threshold parameters are kept low, then these two percentages decrease. Both these cases are undesirable, because we need a high percentage of correct additions and low percentage of wrong additions. Hence, a trade-off between the percentage of correct addition of new patterns and the incorrect addition of transformed old patterns has to be made by choosing the parameters appropriately.

The experiment is conducted over 1000 iterations with the new patterns presented randomly along with rotated, scaled and shifted versions of old patterns. (Once a new pattern is presented and added, it becomes old, thereby enabling it to be used for future test with transforms.) For

Table 2
Recognition results with image size = 128 × 128. Number of iterations over which % accuracy was calculated = 1000

	Grid size	% accuracy (With pulse overlap)				% accuracy (Without pulse overlap)
Grid size	Image size	Shift (S)	Rotation (R)	Scale (C)	SRC	SRC
20 × 20	0.156	100	71.2	70.6	74.1	68.9
30 × 30	0.234	100	74.6	74.8	77.6	71.8
40 × 40	0.333	100	79.9	79.2	80.6	77.5
50 × 50	0.390	100	82.7	81.5	86.2	79.2
60 × 60	0.468	100	85.6	84.7	89.4	85.8

Table 3
Recognition results with image size = 750 × 750 (to illustrate the effect of (grid size/image size))

	Grid size	% accuracy (over 1000 iterations)			
Grid size	Image size	Shift (S)	Rotation (R)	Scale (C)	SRC
50 × 50	0.066	100	78.6	77.8	81.1
100 × 100	0.133	100	89.4	84.4	90.1
150 × 150	0.200	100	90.5	88.2	92.0
200 × 200	0.266	100	92.3	89.2	94.5
250 × 250	0.333	100	94.4	92.9	96.6
300 × 300	0.400	100	96.0	94.2	97.8

the experiment conducted, the parameter values are: $\mathscr{T}_{\mathscr{C}} = -0.84$ and $\mathscr{T}_d = 0.028$. The percentage of correct additions (to the pattern set) is found to be 91.6%, and the percentage of wrong additions is 8.6%. The percentage of correct recognitions among the recognized cases is found to be similar to that obtained in the second experiment described below (Section 6.2).

6.2. Recognition accuracy

Recall that the code grid is of size $r \times c$, where r represents the number of rows and c, the number of columns. The typical grid sizes chosen for experimentation are given in Tables 2 and 3. The second experiment is carried out on a recognizer capable of recognizing all the patterns under consideration. In other words, in this experiment it is assumed that, instead of the new patterns, shifted, scaled or rotated versions of the stored patterns are presented to the recognizer. The randomly chosen patterns are subjected to the operations of rotation (0–360°), scaling (0.5–2.0) and shift (− 30–30 pixels) randomly.

The transformed test pattern is first fed to the coder and then to the recognizer. In this case, the tuning of the parameter $\mathscr{T}_{\mathscr{C}}$ is achieved by observing the number of wrongly added patterns. This is because, in this experiment, no patterns are to be added. For the experiments conducted, the parameter $\mathscr{T}_{\mathscr{C}}$ is set to -0.90, and $\mathscr{T}_d$ is not taken into consideration by setting it to zero.

The results pertaining to recognition accuracies are summarized in Tables 2 and 3 as follows. Here the columns corresponding to accuracy refer to the nature of transformation used: S (shift), R (rotation), C (scaling), and SRC (combination of all the three: S, R, and C).

From the above tables, we observe the following:

1. Increase of grid size leads to improved recognition accuracy. This is a consequence of the fact that the quantization error occuring in the process of fitting the patterns to the grid reduces when the grid size increases.
2. With a larger image size, even at a lower ratio between grid size and image size, the recognizer gives better results. This is because the log-mapping's rate of increase in magnitude is considerably slower than that of its argument. As a result, the coder provides a comparatively better resolution even with smaller grid size at larger image sizes, which, in turn, implies better recognition. (This can be seen by comparing the recognition accuracies and the corresponding grid sizes presented in Tables 2 and 3.)
3. The accuracy obtained with shifted patterns is always 100 because of the center-of-gravity-based mapping, which takes care of the shift implicitly.

6.3. Overlap of pulse inputs

In order to analyze the effects of overlap in the pulses fed to each layer (see Section 5), experiments were conducted *without* overlap in the pulses. This requires the following changes in the architecture and algorithm for the recognizer (cf. Section 5):

- Each unit receives two pulse inputs (one each from the horizontal and vertical scans) rather than four pulse inputs (two each from the horizontal and vertical scans). The units in layers other than layer one will have an extra input from the previous layer, as mentioned in Section 5.2. The weights assigned to the neurons correspond to w^p_{ct}, w^p_{rt} of Table 1 and O^p_t (see Section 5.2 for further details).

- As the vectors $\tilde{\mathbf{w}}^p_t$ and $\tilde{\mathbf{i}}^p_t$ of Eq. (1) are two dimensional (rather than four dimensional as in Section 5.2), normalization has to be done in order to compare these results with those obtained *with* overlap in inputs.
- The number of layers in the architecture would be equal to L_{inp} rather than being equal to $L_{inp} - 1$.

The tests for adaptability and recognition accuracy were conducted for such networks without overlap in the pulse inputs. With other parameters remaining constant, apart from the above changes, the test for adaptability yielded the following result: while the percentage of correct additions remained relatively invariant to the overlap of pulses, the percentage of wrong additions to the network was affected severely. The percentage of wrong additions was found to be 33.2 (cf. Section 6.1).

The analysis with respect to recognition accuracy for networks without overlapping pulse inputs is given in Table 2 *adjacent* to the results corresponding to networks with overlap. It is found that there is a deterioration in the recognition accuracy if overlaps in the pulses are absent. Hence, it can be concluded that overlap in the pulses makes the recognizer robust to noise caused by a finite coder grid (see Section 5.2).

6.4. Breaks in contours

In practice, contour breaks, which are caused by improper illumination and specular reflection, affect segmentation processes (like edge-linking), and hence lead to erroneous recognition of shape from contours. It has been found that the performance of the encoder-recognizer pair to contours with varying amounts of breaks at a single point deteriorates. However, by increasing the coder grid size, some improvement in the performance of the recognizer has been observed for the same broken contours.

6.5. Advantages of the recognizer

The advantages of the proposed recognizer are:

1. New patterns can be added to the system on-line without any major effort.
2. The recognizer is made robust to noise in the code by employing confidence measures with appropriate thresholds.
3. A decision taken in the previous layers influences the decisions taken in the subsequent layers, thereby increasing the overall robustness of the approach.
4. A further improvement in robustness is achieved by providing overlaps in the inputs fed to the various layers of the network.
5. Higher speed is realizable due to parallelism and lateral inhibition between the neurons in the layers.

7. Conclusions

For an invariant recognition of patterns, we have proposed a combination of (i) complex logarithmic mapping (CLM); (ii) pulse encoding of the resultant output; and (iii) recognition of the sequence of pulses so generated. The encoder converts the CLM output to a sequence of pulses. The CLM and pulse encoding ensure conversion of the pattern to a set of pulses which are invariant to rotation, scaling and shift of the pattern. A multi-layered neural architecture is then employed to recognize patterns, invoking a scheme *similar to* template matching, but *quite distinct from* those of the literature with respect to its implementation. Some important features of the proposed approach are robustness and the ability to add new patterns on-line to the existing ones. Examples are given to illustrate the proposed coding and recognition strategies.

Acknowledgements

The authors wish to express grateful thanks to the reviewers for their valuable suggestions and critical remarks. Thanks are also due to Mr. D. Ramasubramanian and Mr. R.B. Lokesh for their help in the creation of some of the diagrams.

References

[1] P.C. Dodwell, Visual Pattern Pecognition, Holt, Rinehart and Winston Inc, New York, 1970.

[2] J.K. Brousil, D.R. Smith, A threshold-logic network for shape-invariance, IEEE Trans. Computers EC-16 (1967) 818–828.

[3] D. Casasent, D. Psaltis, Position, rotation and scale-invariant optical recognition, Appl. Opt. 15 (1976) 1793–1799.

[4] H.J. Reitboeck, H. Altmann, A model for size- and rotation-invariant processing in the visual system, Biol. Cybern. 51 (1984) 113–121.

[5] L. Massone, G. Sandini, V. Tagliasco, Form-invariant topological mapping strategy for 2D shape recognition, Computer Vision, Graphics Image Process. 30 (1985) 169–188.

[6] E.L. Schwartz, Spatial mapping in the primate sensory projection: analytic structure and relevance to perception, Biol. Cybern. 25 (1977) 181–194.

[7] E.L. Schwartz, Topographic mapping in the primate visual cortex: history, anatomy and computation, in: D.H. Kelly (Ed.), Visual Science and Engineering: Models and Applications, Marcel Dekker Inc, New York, 1994.

[8] J.J. Hopfield, Pattern recognition computation using action potential timing for stimulus representation, Nature 376 (1995) 33–36.

[9] D. Ferster, N. Spruston, Cracking the neuronal code, Science 270 (1995) 756–757.

[10] P.C. Dodwell, Shape recognition in rats, British J. Psychol. 48 (1957) 221–229.

[11] P.C. Dodwell, A coupled system for coding and learning in shape discrimination, Psychol. Rev. 71 (1964) 148–159.

[12] Y.V. Venkatesh, N. Rishikesh, Modelling active contours using neural networks isomorphic to boundaries, Proceedings of the International Conference on Neural Networks (ICNN-97), Vol. III: Houston, TX, USA, 1997, pp. 1669–1672.

[13] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K.J. Lang, Phoneme recognition using time-delay neural network, IEEE Trans. Acoustics, Speech Signal Proces. 37 (1989) 328–339.

[14] D.H. Hubel, T.N. Wiesel, Brain mechanisms of vision, Scientific American 241 (3) (1979) 150–162.

[15] N. Rishikesh, Y.V. Venkatesh, An invariant pulse coder for 2-D shape recognition, presented at the Special Invited Session, International Conference on Information, Communications, and Signal Processing, ICICS'97, Singapore, 1997, pp. 1552–1556.

About the Author—N. RISHIKESH received the Bachelor of Engineering (B.E.) Degree from Madurai Kamaraj University, Madurai, India, in 1995, and Master of Science (M.Sc. Res) from the Indian Institute of Science, Banaglore, India, in 1997. Currently, he is a doctoral student, working on biological information processing. His research interests include Neural Networks, Computer Vision, and Computational Neuroscience.

About the Author—Y.V. VENKATESH obtained his Ph.D. from the Indian Institute of Science (Bangalore, India) for a dissertation on the stability analysis of feedback systems. He was an Alexander von Humboldt Fellow at the Universities of Karlsruhe, Freiburg, and Erlangen, Germany; National Research Council (USA) Fellow at the Goddard Space Flight Center, Greenbelt, Maryland; Visiting Fellow at the Australian National University; Visiting Professor, Institute for Mathematics, Technical Univeristy of Graz, Graz, Austria, to name a few. His research monograph on the stability and instability analysis of linear and nonlinear time varying feedback systems has appeared in the Springer-Verlag Lecture Notes in Physics. His present work is on pattern recognition using artificial neural networks, signal reconstruction from partial information, modelling human vision, and image fusion. He is a Professor at the Indian Institute of Science, Bangalore, India, and presently the Chairman of the Division of Electrical Sciences. He is a Fellow of the Indian National Science Academy, Indian Academy of Sciences, and the Indian National Academy of Engineering; and a Senior Member of the IEEE. He is on the Editorial Board of the new International Journal of Information Fusion.