

An Invariant Pulse-Coder for 2-D Shape Recognition

N. Rishikesh & Y.V. Venkatesh,
Computer Vision and Artificial Intelligence Laboratory,
Department of Electrical Engineering,
Indian Institute of Science,
Bangalore - 560 012, India.

Abstract: We present some novel schemes for (i) pulse coding for invariant representation of shape; and (ii) a neural architecture for recognizing the encoded shape. The proposed shape encoder utilizes the properties of complex logarithmic mapping (CLM) which transforms rotation and scaling (in its domain) to shifts (in its range). In order to handle this shift, in an attempt to provide increased speed of operation, the encoder converts the CLM output to a sequence of pulses. These pulses are fed to a novel multi-layered neural recognizer which (i) invokes template matching with a distinct architecture; and (ii) achieves robustness (to noise and shape deformation) by virtue of its overlapping strategy for code classification. The proposed encoder-recognizer system, which is hardware implementable by a high-speed electronic switching circuit, and can add new shapes on-line to the existing ones, is illustrated by examples. Shortage of space precludes a full description of the results. For details, see [1].

1 Introduction

We deal with the problem of 2-D pattern (or shape) recognition, which is one aspect of image analysis. This problem can be decomposed into two stages: invariant code generation 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 scheme proposed in this paper for coding shapes utilizes the properties of complex logarithmic mapping (CLM) which exhibits some desirable properties: the CLM transforms rotation and scale in its domain to shifts in its range. It is this shift that has to be handled in order to obtain a completely invariant code. Though the literature [2, 3, 4, 5] presents some methods of handling this shift, the results suffer from a number of disadvantages of which the large execution time and high noise sensitivity are prominent. In this paper, we present a novel method for handling the shift, which facilitates the generation of invariant pulse codes of shape. The output of the pulse encoder acts as the input to a shape recognition network.

2 Pulse Encoding of Shape

The motivation for the present work arose from Dodwell's model [6], in which a two-dimensional array (coder grid) of units is used for pulse-encoding of patterns. In the coder grid, shown in Fig. 1, one set of units in the grid (the end-set A) has only one neighbor each on a chain, to which they transmit excitation. The elements of set A have the property that their outputs are simultaneous: if one unit fires, they all fire. All other units in the grid transmit this excitation to their neighbors after constant time delays, T_p , if they are not activated from the receptors. When a unit is activated in the grid by the occurrence of a contour point on the receptors, the time for transmission of excitation is increased to T_a . Thus the occurrence of contours on the receptors will alter transmission times along the chains of the grid, and hence, correspondingly, the coder's output.

The coder's output is not affected by translation in either of the axes. The coder generates similar outputs for some patterns which, on common sense grounds, are expected to be discriminable. This difficulty is resolved in [6] by postulating an independent 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, and the coder has two separate outputs. However, it is found that Dodwell's [6] coder output is unsatisfactory for the following reasons: (a) It is not invariant to rotation and scaling of the contour. (b) Even a slight change in the input contour distorts the output code, thereby making it impossible to recognize the contour correctly. (c) As the contour is fed directly to the coder, the coder array size will be very high, thereby yielding a longer pulse sequence.

3 Proposed Encoding Scheme

In our proposed scheme, two-dimensional coordinates of the contour points are fed to the Coder which maps them using the CLM with respect to the Center-of-Gravity (CoG) of the contour. The use of the CoG as the base for the CLM results in shift invariance at this

stage itself. As CLM transforms rotation and scaling in its domain to shift in its range, this shift is the one that has to be handled now. To handle this shift, we propose an application of Dodwell's method after the CLM. That is, after mapping the contour using CLM, we input the CLM coefficients to a coder grid similar to that described above, which is shift-invariant. 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. In essence, the visual pattern has been pulse-coded in such a way that the output of the coder is a unique set of series of pulses, which are invariant to all translation, rotation and scaling. *The novelty in the proposed approach lies in (i) the use of Dodwell's grid structure to handle the shift in the CLM output; and (ii) coding the input shape as invariant pulse sequences.*

The advantages of the proposed coder are: (a) Invariance to rotation, scaling and shift of patterns, (b) A pulse-output that facilitates fast recognition with help of high-speed circuits, (c) A comparatively small length of pulses (cf., [6]), due to the use of log-mapping; and (d) Insensitivity of the recognition algorithm to small breaks in the contour.

4 Recognition of Pulses

We now propose a scheme for recognition of the code generated by the coder. To this end, we first formulate the problem of recognizing the coder output.

Problem Formulation: Given, two sequences of pulses, whose amplitude and the time of occurrence contain information about the class to which the pattern belongs, design a recognizer to classify the pulse sequences, and hence the patterns.

Some desirable properties of such a recognizer are: (a) The time taken for recognition should be small; (b) The recognizer should be (i) able to accommodate new-patterns on-line, in case there is no match for it in the database; and (ii) robust to noise arising from either the input signal or the finiteness of the size of the array of the coder.

Though there are some approaches available in the literature for handling this recognition problem, like the Time-Delay Neural Network [7], and Uttley's recognizer used by Dodwell [6], they suffer from a number of disadvantages (see [1]).

5 Proposed Recognizer

In an attempt to design a recognizer which has the desirable properties specified earlier, a novel neural-based network with competitive layers 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 columns¹ of the network. However, the proposed recognition scheme is quite distinct from the standard template matching scheme on the following grounds: (a) An overlapping strategy is used for pulse classification in order to achieve robustness, (b) A decision at one stage influences the decision on the succeeding stage for handling ambiguities efficiently, and (c) Lateral inhibition and competitive layered network have been used for reducing the execution times.

5.1 Recognizer Architecture

The proposed recognizer utilizes a multi-layered neural architecture, shown in Fig. 2, 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 columnar² 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. 3. 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 Sec. 2). In order to achieve robustness in the recognition of 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 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. For 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 t th layer and p th column

¹See explanation below.

²A 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.

(which is propagated to the next layer when the neuron "fires"), and \mathbf{w}_i^p is the weight vector associated with it. (The neuron of Fig. 3 has five inputs, 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 p th column of the t th layer of the network are listed below:

Table 1
WEIGHTS OF A NEURON

Notation	Pulse Number	Scan
w_{ct}^p	t	Column
$w_{c(t+1)}^p$	$t+1$	Column
w_{rt}^p	t	Row
$w_{r(t+1)}^p$	$t+1$	Row

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 t th pulse of the column-wise scan with respect to the p th pattern. The first layer has only the above-listed four weights. The other layers have five weights, and hence a five-input neuron. The fifth weight is represented as w_{ot}^p , which is assigned the value of the output O_t^p , associated with the neuron under consideration, and this is the *pattern number* of the pattern p whose pulse amplitudes are stored as weights in the neuron.

5.2 Recognition Algorithm

The notations used are as follows: L_{inp} : input sequence length; N_t^p : neuron in the t th layer of the p th column; \mathbf{i}_t : input vector presented to the neurons in layer t ; and N_t^k : winner neuron of the layer t . The confidence measures are represented by C_p . 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 Sec. 5.1.

2. Initialize all the Confidence Measures 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.

5. Calculate the *weighted distance* between the input vector and the weight vector of the neurons in the other layers:

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

where λ_1 and λ_2 are weighting parameters, with $\lambda_1 + \lambda_2 = 1$, and $\lambda_1 \geq 0.5$ and $\lambda_2 \leq 0.5$, in order to give more weightage to the input vectors. This is done to avoid propagation of noise to the higher layers due to noise in lower ones. $\tilde{\mathbf{i}}_t$ 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:

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

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 sequences.)

6. Find the least two confidence measures among C_p 's, and label them as C_i and C_j ($C_i < C_j$).

If $C_i > \mathcal{T}_c$ and if $(C_i - C_j) > \mathcal{T}_d$, then declare that the new pattern's pattern number is i .

Else, create a new column of neurons and assign their weights (as defined in Sec. 5.1) appropriately with data from the present input. Generate a new pattern number, and treat it as the output of those neurons and as the fifth weight of inner layers.

(The threshold parameter \mathcal{T}_c is the lower bound on the least of the confidence measures, for a pattern to be declared recognized. On the other hand, \mathcal{T}_d is the lower bound on the difference between the two lowest confidence measures.)

6 Results of Shape Recognition

The simulation studies on the coder and the recognizer, conducted on a HP9000/715 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.

The coding and recognition algorithms have been tested on the problem of recognition of 20 Indian states when they are given as separate contours. This problem has been chosen because the shapes are irregular, leading to considerable complexity. A few of the contours are shown in Fig. 4. 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 noise previous decisions. Typically, $\lambda_2 = 0.001$.

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 additions, we need to choose the threshold parameters correctly.

The experiment is conducted over 500 iterations with the new patterns presented randomly along with rotated, scaled and shifted versions of old patterns. For the experiment conducted, the parameters are set as: $\mathcal{T}_C = -0.84$ and $\mathcal{T}_d = 0.028$. The percentage of correct additions (to the pattern set) is found to be 92.6%, and the percentage of wrong additions is 5.6%.

Recognition Accuracy

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 only shifted, scaled or rotated versions of the stored patterns are presented to the recognizer. The randomly chosen patterns are subjected to rotation (0 - 360 degrees), scaling (0.5 - 2.0) or shift (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 \mathcal{T}_C is achieved by observing the number of wrongly added patterns. For the experiments conducted, the parameter \mathcal{T}_C is set to -0.90 , and \mathcal{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. From these tables, we observe that: (a) An increase of grid size leads to improved recognition accuracy; and (b) With a larger image size, even at a lower ratio of grid to image size, the recognizer

gives better results.

To summarize, the advantages of the proposed recognizer are: (a) New patterns can be added to the system on-line without any major effort. (b) The recognizer is made robust to noise in the patterns by employing confidence measures with appropriate thresholds. (c) The decision taken in previous layers influences the decisions taken in the subsequent layers, thereby increasing the overall robustness of the approach. (d) A further improvement in robustness is achieved by providing overlaps in the inputs fed to the various layers of the network. (e) 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 which are invariant to rotation, scaling and shift of the pattern. Improved pattern recognition is then achieved by a multi-layered neural architecture which is robust with respect to noise, and is capable of adding new patterns on-line.

References

- [1] N. Rishikesh and Y.V. Venkatesh, "Invariant Pulse Coding for Recognition of Patterns," Submitted to *Biological Cybernetics*, April 1997.
- [2] H.J. Reitboeck and J. Altmann, "A model for size- and rotation-invariant processing in the visual system," *Biological Cybernetics*, Vol. 51, pp. 113-121, 1984.
- [3] J.K. Brousil and D.R. Smith, "A threshold-logic network for shape-invariance", *IEEE Transactions on Computers EC-16*, pp. 818-828, 1967.
- [4] D. Casasent and D. Psaltis, Position, rotation and scale- invariant optical recognition, *Applied Optics*, Vol. 15, pp. 1793-1799, 1976.
- [5] L. Massone, G. Sandini, and V. Tagliasco, "Form-invariant topological mapping strategy for 2D shape recognition," *Computer Vision, Graphics and Image Processing*, Vol. 30, pp. 169-188, 1985.
- [6] P.C. Dodwell, *Visual Pattern Recognition*, Holt, Rinehart and Winston Inc., New York, 1970.
- [7] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. J. Lang, "Phoneme recognition using time-delay neural networks", *IEEE Transactions on Acoustics, Speech and Signal Processing*, Vol. 37, No. 3, pp. 328-339, March 1989.

Table 2 : RECOGNITION RESULTS WITH IMAGE SIZE = 128 × 128

Grid Size	Grid Size ImageSize	% accuracy (over 1000 iterations)			
		Shift(S)	Rotation(R)	Scale(C)	SRC
20 × 20	0.156	100	71.2	70.6	74.1
30 × 30	0.234	100	74.6	74.8	77.6
40 × 40	0.333	100	79.9	79.2	81.5
50 × 50	0.390	100	82.7	81.5	84.3
60 × 60	0.468	100	85.6	84.7	87.5

Table 3 : RECOGNITION RESULTS WITH IMAGE SIZE = 750 × 750

Grid Size	Grid Size ImageSize	% accuracy (over 1000 iterations)			
		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

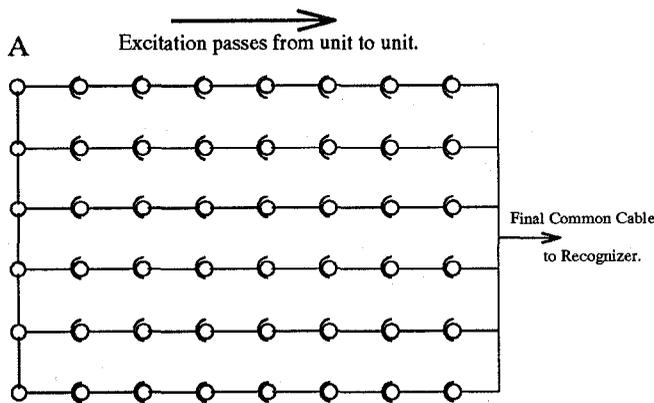


Fig. 1: The Coder Grid

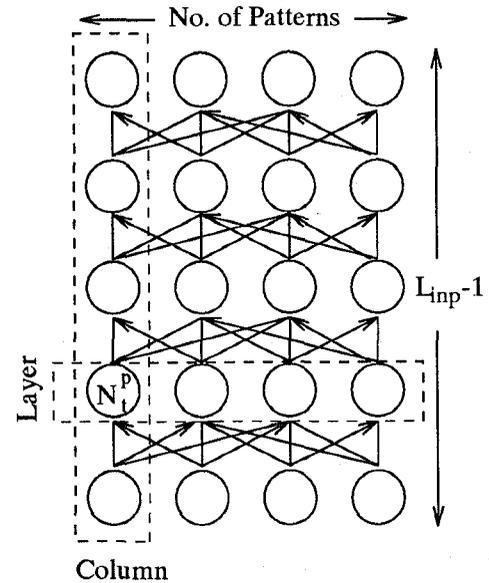


Fig. 2: The Recognizer Network

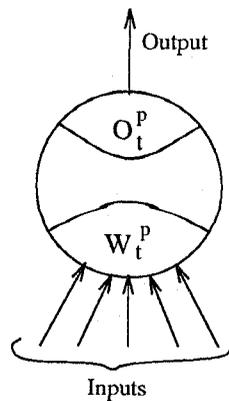


Fig. 3: The Recognizer Neuron

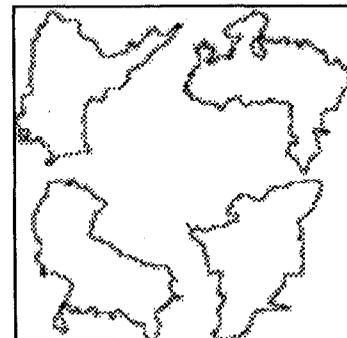


Fig. 4: Typical Contours