

# An Accurate Structural Approach to Pattern Recognition

Subhrajit Bhattacharya<sup>†</sup>

<sup>†</sup> Department of Mechanical engineering, IIT Kharagpur, INDIA.  
e-mail: subhrajit\_02@mech.iitkgp.ernet.in , subhrajit83@yahoo.co.in

## **Abstract**

In the present paper a simple, yet ingenious and accurate structural approach towards pattern recognition has been proposed, which can satisfactorily consider transformations on the image like translation, rotation, scaling and local & global deformations. The algorithms developed in the present work are inspired by an understanding and analysis of the way in which information about patterns are stored and accessed in a biological brain. The algorithm, though purely based on simple geometric and statistical tools, gives very satisfactory results. On the basic framework of the present algorithm application of Neural Network models and Genetic Algorithms can produce a much more fast and robust method for pattern recognition.

## **Keywords**

Structural Pattern Recognition; Graphics recognition; Optical character recognition

# 1 INTRODUCTION

Taking a brief look at the history of Pattern Recognition we may find that after an early quest on pure structural approach to pattern recognition, the problem has much been influenced by modern soft computing techniques [5] like Fuzzy Logic, Neural Networks and Genetic algorithm. Fuzzy Logic has been an important and effective tool for matching of patterns with considerable amount of deformations and variations. Many works used Fuzzy Sets and Fuzzy Logic as tools for Pattern recognition with highly satisfactory results [6-12]. With the development of neural computing tools, the application of Neural Network has vastly found its place in the field of Pattern Recognition [13-16]. Neural Network models along with Fuzzy sets have proved to be highly effective in increasing the speed and robustness of the process [17].

However, the present work proposes a purely structural approach. Even without the application of such soft computing tools the present work gives quite satisfactorily results. As a continuation of this work, there are scopes of making radical improvements on the performance of the present algorithm by application of such modern soft computing tools. As for example, though in the present work Fuzzy Logic hasn't been used actively, there are scopes for its use at several places in the algorithm where decisions have been made. This may definitely make considerable improvement on the performance. Moreover we may also try to make improvements on the robustness and speed of the present algorithm by feeding the results obtained from the algorithm to a suitably designed Neural Network. However these are out of the scope of the present work.

The basic principle of the present algorithm for pattern recognition is based on an understanding and analysis of the way in which information about patterns are generally stored and accessed in a biological brain. This analysis used here is more of a hypothesis, and its rigorous proof is out of scope of the present work. The following paragraph makes a brief discussion on the result of this analysis and tries to explain the manner in which our brain attempts to remember and recognize a pattern.

According to the present analysis, the biological brain can never remember a pattern exactly, i.e., it can't store information of an image pixel by pixel (here pixel may be defined as the smallest picture element visible to the biological eye distinctly). Rather, what it tries to do is to mark some characteristic points in the pattern whenever it encounters a new pattern. Then it tries to remember the patterns local to those characteristic points and their relative positioning in the whole pattern with respect to each other.

In the present algorithm for pattern recognition we approach in exactly the above mentioned manner. Whenever a new pattern is encountered, the algorithm tries to mark some characteristic points in the pattern, remembers the pattern local to those points, and the relative positioning of those characteristic points. In this way it develops a database of standard patterns.

Finally when the time for recognizing an unknown pattern comes, it searches for closest match for those characteristic points inside the new pattern on the basis of minimum deviation in Local Patterns. Once the closest matches for the characteristic points are obtained, the relative positioning of the points in the respective patterns are checked for consistency. The error in the comparison between the relative positioning, together with the errors in the Local Patterns give a measure of difference of that particular database pattern from the unknown pattern. The database pattern with minimum value of the difference gives the required match.

What precisely has been done in the algorithm is as follows. There are  $n$  Characteristic Points  $Q_1, Q_2, \dots, Q_n$  in a Database Pattern. The corresponding closest matches (minimum difference in Local Pattern) in the Unknown Pattern be  $S_1, S_2, \dots, S_n$ . In the process let the determined differences in the Local Patterns be  $e_1, e_2, \dots, e_n$  respectively. Now it is checked if the arrangement of the points  $Q_1, Q_2, \dots, Q_n$  among themselves matches with the arrangement of  $S_1, S_2, \dots, S_n$  among themselves.  $\epsilon_{u,v}$  is a measure of difference in this arrangement. The two measure of differences,  $e_i$  and  $\epsilon_{u,v}$ , together gives a measure of difference between the two patterns, which in turn is the decisive factor for the conclusion.

The following diagram shows how the seven points in a database pattern found their closest matches in the new pattern. They differ by their relative positioning and deformed local patterns:



*fig - 1*

The excellence of the algorithm lies in the fact that it attempts to exploit the method of pattern recognition in biological brain using a simple structural approach rather than creating the whole neural system artificially for vision and pattern recognition. Moreover the algorithm provides quite a few number of control parameters which gives lots of flexibilities to the working of the algorithm, making it suitable for a wide range of patterns.

## Symbols and Notations

$\Pi_A$	Pattern $A$ .
$A_\varepsilon(P)$	Local Pattern around point $P$ .
$[x]_y$	The value closest to $x$ which is a multiple of $y$ .
$\{R_i\}$	Set of all points $R_i$ , for all ‘relevant’ values of $i$ .
$a (\pm b)$	$a$ or $a\pm b$ . That is, $a-b$ or $a$ or $a+b$ .
$\Delta_{\Pi_A - \Pi_B}$	Difference between patterns $\Pi_A$ and $\Pi_B$ .
$\Delta_{\Pi_A, \{M_i\} - \Pi_B, \{N_i\}}$	Difference between pattern $\Pi_A$ with Characteristic Points $\{M_i\}$ and pattern $\Pi_B$ with corresponding match for Characteristic Points $\{N_i\}$ .

## Nomenclature of Parameters

$\varepsilon_0$	Local Pattern radius in Normalized scale
$\phi$	Exponential weight of Relative Positioning error over Local Pattern error
$\eta$	Selection proportion exponent
$\varepsilon_{thresh}$	Maximum allowable error in relative positioning
$p\%$	Minimum allowable proportion of points with high relative positioning error
$U_{threshold}$	Maximum allowable irregularity in differences with the members of a class
$r_0$	Minimum Search Circle radius
$k_r$	Search Circle centre eccentricity factor for radius

## List of captions for figures:

- Fig - 1* Overview of the main working principle of the present algorithm.
- Fig - 2* The CG-PD coordinate system for the shown pattern.
- Fig - 3* Generation of the 8-valued local pattern about the point  $P$  as shown (The odd and even sectors are shown in dark and light grey colors).
- Fig - 4* Searching around  $Q_i'$  in the Unknown Pattern to obtain  $S_i$  as the closest match for  $Q_i$ .

## 2 THE ALGORITHM

### 2.1 Initial definitions, hypothesis and preprocessing tools for the algorithm

#### 2.1.1 Representation of the 'raw' pattern

A pattern ( $\Pi$ ) of size  $m \times p$  pixels may be defined as a matrix of size  $m \times p$  which contains a single layer of intensity level information for the pixels. Let the intensity layer matrix be represented by  $\Pi$ . For multicolored patterns the calculations for the other layers will follow in similar fashion. The elements of that matrix  $\Pi$  are represented as  $\pi_{jk}$  ( $0 \leq \pi_{jk} \leq 1$ ) and stores the intensity information of the  $(j, k)^{\text{th}}$  pixel, where  $j$  and  $k$  are the discretized variables along the global axis directions  $X$  and  $Y$ . Henceforth a Database Pattern will be represented by  $\Pi_D$  and an Unknown/New Pattern by  $\Pi_U$ .

#### 2.1.2 Defining a Local Coordinate System – The CG-PD coordinate system

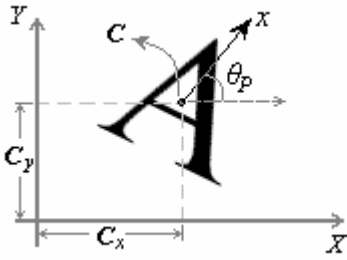


fig - 2

For reduction of time complexity of the algorithm and various other purposes, a coordinate system local to the pattern was needed to be defined. This coordinate system should be fixed to the pattern, i.e. the position of the points on the pattern in this local coordinate system should not change with the change in the global coordinate system.

Such property in local coordinate system is exhibited by the Centre of gravity – Principal Direction (CG-PD) Coordinate System [2].

The CG-PD coordinate system was determined for various patterns with different transformations on the intensity matrix, and the matrix of the complement elements was found to give best results. This is mainly because of a high intensity background for the patterns. The intensity matrix may be chosen directly instead for patterns with low intensity background.

Let that local coordinate system be defined by the origin  $C$ , and the pair of axis  $x$  and  $y$ .

Let  $\bar{\pi}_{jk} = 1 - \pi_{jk}$ .

Hence the origin of the CG-PD coordinate system is defined as,

$$C_X = \frac{\sum_{j=1}^m \sum_{k=1}^p j \cdot \bar{\pi}_{jk}}{\sum_{j=1}^m \sum_{k=1}^p \bar{\pi}_{jk}} \quad \text{and} \quad C_Y = \frac{\sum_{j=1}^m \sum_{k=1}^p k \cdot \bar{\pi}_{jk}}{\sum_{j=1}^m \sum_{k=1}^p \bar{\pi}_{jk}} \quad (1)$$

And the direction  $\theta_p$  made by the local  $x$  axis with the global  $X$  axis is given by,

$$\tan 2\theta_p = \frac{2I_{XY}}{I_{YY} - I_{XX}} \quad (2)$$

where,  $I_{XX} = \sum_{j=1}^m \sum_{k=1}^p (k - C_Y)^2 \bar{\pi}_{jk}$ ,  $I_{YY} = \sum_{j=1}^m \sum_{k=1}^p (j - C_X)^2 \bar{\pi}_{jk}$

and  $I_{XY} = \sum_{j=1}^m \sum_{k=1}^p (j - C_X)(k - C_Y) \bar{\pi}_{jk}$  are second moments of pixel intensities [2].

It may be noted that from the above definition two values of  $\theta_p$  in  $(-\pi, \pi]$  are obtained. By convention, the one about which the second moment of pixels intensities on the top-right (first) quadrant is less than that on the bottom-left (third) quadrant may be chosen.

Hence, the transformation from global  $X$ - $Y$  coordinate system to the local  $x$ - $y$  coordinate system is given by,

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta_p & \sin \theta_p \\ -\sin \theta_p & \cos \theta_p \end{bmatrix} \begin{bmatrix} X - C_X \\ Y - C_Y \end{bmatrix} \quad (3)$$

### 2.1.3 Normalization of a Pattern – Normalized CG-PD coordinate system

In order to make the patterns (both the database and the unknown patterns) independent of scale, the coordinates of points on the patterns in their CG-PD coordinate system need to be divided by some normalization factors.

These normalization factors along  $x$  and  $y$  directions may be satisfactorily defined as,

$$f_x = \sqrt{\frac{\sum \sum x^2 \cdot \bar{\pi}_{xy}}{\sum \sum \bar{\pi}_{xy}}} \quad \text{and} \quad f_y = \sqrt{\frac{\sum \sum y^2 \cdot \bar{\pi}_{xy}}{\sum \sum \bar{\pi}_{xy}}} \quad (4)$$

where, the double-summation is done over the whole of the pattern  $\Pi$ .

Hence, finally the normalized coordinates are given by,

$$\begin{bmatrix} x_N \\ y_N \end{bmatrix} = \begin{bmatrix} x/f_x \\ y/f_y \end{bmatrix} \quad (5)$$

It can be noted that the Normalization done in the above mentioned method will work satisfactorily even in presence of unwanted noises in the pattern.

### 2.1.4 Local Pattern around a point 'P' in the pattern

For defining the local patterns around a point P:

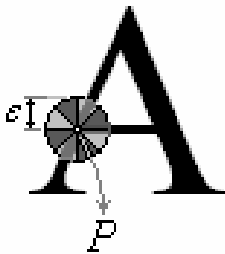


fig - 3

- A circle of radius  $\varepsilon$  is taken around the point P,
- The circle is divided into 8 sectors,
- The weighted mean of the intensities of the pixels lying within each sector are determined,
- The weights being any suitable monotonically decreasing function of the distance of the pixel from P.

The sectors of the circle are created with respect to the global coordinate system, i.e. the 8 lines forming the sectors are aligned at angles of  $0^\circ, 45^\circ, \dots, 315^\circ$  with the global  $X$  axis. This is done to reduce computational complexity.

As a measure of the local pattern around a point  $P$ , we obtain 8 values (the 8 weighed means corresponding to the 8 sectors) corresponding to the point:

$$\therefore \text{Local Pattern } (P) = A_\varepsilon(P) = [P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8]$$

The radius  $\varepsilon$  will logically depend on the scale of the pattern. It can be satisfactorily chosen to be  $\varepsilon = \varepsilon_0 \cdot \sqrt{f_x f_y}$ , where  $\varepsilon_0$  is a constant value.

### 2.1.5 Rotation Quantization Hypothesis

The reason for taking 8 sectors with respect to the global coordinate system and how that can account for the rotation effects in the local pattern can be explained by a hypothesis.

The hypothesis: *'effect of rotation gets quantized as the size or resolution of the pattern decreases, i.e. it is sufficient to check only a few angles of rotation for considering the rotational effects while matching two patterns when the size of the patterns are sufficiently small or the patterns are themselves quantized'*.

This can be demonstrated easily by considering small patterns. Let's consider a 1x1 pattern (i.e. a pixel) which has only one step of rotation possibility (i.e. the pixel itself). For a 2x2 pattern it is sufficient to consider 4 steps of rotation ( $0^0$ ,  $90^0$ ,  $180^0$  and  $270^0$ ). For 3x3 pattern 8 steps are sufficient. And so on.

Hence, it may be argued, by taking sufficiently small value for  $\varepsilon_0$  the 8-valued Local Pattern Vector can satisfactorily account for rotation effects in the local patterns.

The reason behind taking 8 sectors (i.e. maximum 8 steps of rotation consideration) is that it is the highest number of equally spaced sectors that can be handled in a computationally efficient way without any floating point operation.

## 2.2 Remembering the Pattern – creating the database

Each pattern in the database ( $\Pi_{D,l}$ ) is hence stored as,

- i. The coordinates of some characteristics points in the normalized CG-PD coordinate system of the pattern. Let these points be called  $Q_1, Q_2, \dots, Q_n$ . (The characteristic points may be chosen to be some equally spaced points in the Normalized CG-PD coordinates of the pattern.)
- ii. And, 8-element vectors corresponding to each of these characteristic points which define their Local Patterns in the Normalized CG-PG coordinate system, i.e.  $A_\varepsilon(Q_1), A_\varepsilon(Q_2), \dots, A_\varepsilon(Q_n)$ .

## 2.3 Comparing a database pattern with an unknown pattern

### 2.3.1 Preprocessing the Unknown Pattern

Though this process is somewhat computationally expensive, it needs to be done only once and need not be performed before comparing with each of the patterns in the database.

The following steps are performed to pre-process the unknown pattern  $\Pi_U$ .

- i. Determination of the CG-PD coordinate system. This includes calculation of  $C_X, C_Y$  and  $\theta_p$ .

- ii. Determination of the Normalization factors  $f_x$  and  $f_y$ . Hence determine the Normalized CG-PD coordinate system.
- iii. Determination of Local Pattern of some randomly selected points distributed over the whole of the unknown pattern. Ideally we should have determined Local Pattern of all the points in the pattern. But for reducing computational complexity we can satisfactorily restrict our search operation (described next) to around 50% of all the points in the unknown pattern, selected randomly.

### 2.3.2 The Search Process

Next, the closest matches of the Characteristic Points of the Database Pattern ( $\Pi_{D,l}$ ) are needed to be searched for inside the Unknown Pattern ( $\Pi_U$ ). Moreover, In order to consider the rotational effect in the local patterns, searches need to perform by rotating the Local Patterns suitably. This is the most time-expensive process. Hence, in order to minimize the time complexity of the search process we make two hypotheses:

#### *Hypothesis – I :*

It can logically be argued that if the two patterns are exactly similar (i.e. one can be superimposed on another after suitable rotation, translation and scaling) then the coordinates of a particular characteristic point in the Normalized CG-PD coordinate system of either of the patterns will be the same. This is very evident.

Hence we may say, if the Unknown Pattern differs from the Database Pattern moderately, then the coordinates of a particular Characteristic Point in the Database Pattern in its Normalized CG-PD coordinate system will have its closest match in the Unknown Pattern somewhere near the same coordinate in the Normalized CG-PD system of the Unknown Pattern. So it will be satisfactory if we restrict our search within a circle of radius ' $r$ ' near the coordinate of the Characteristic Point we are searching for. In fact, this provides a better result than searching over the whole Pattern because this eliminates the possibility of error due to similarity in Local Pattern at two or more places in the Pattern.

#### *Hypothesis – II :*

Similarly it can also be argued that if the two patterns are exactly similar then the angle of rotation on the database Local Patterns required while performing the search is

$\theta_\Delta = \theta_{P, \Pi_U} - \theta_{P, \Pi_D}$ . Again assuming moderate deviation of the angle of rotation of Local Patterns from this value, we restrict our rotational search to a few quantized values of angle around  $\theta_\Delta$  that can be achieved by just rotation of the Local Pattern vectors. In practice, the search for rotation in local patterns is done by angles  $[\theta_\Delta - 45^0]_{45}$ ,  $[\theta_\Delta]_{45}$  and  $[\theta_\Delta + 45^0]_{45}$ , where  $[x]_{45}$  represents the value closest to  $x$  which is a multiple of 45. The reason behind choosing 45 is that while finding the Local Pattern the circle around a point was divided into 8 sectors.

Now the search process will be described with a particular Database Pattern  $\Pi_{D,l}$  in the Unknown Pattern  $\Pi_U$ . At a time only one angle of rotation in the Local Patterns is considered. Let's say the search is being done by considering a rotation of  $[\theta_\Delta (\pm 45^0)]_{45}$  in the Local Patterns. Let the set of Characteristic Points in the Database Pattern  $\Pi_{D,l}$  be  $\{Q_1,$



$Q_2, \dots, Q_n$  Let the coordinates of the  $i^{\text{th}}$  point ( $Q_i$ ) in the Normalized CG-PD coordinate system of the Pattern be  $(x_{N,i}, y_{N,i})$ .

The following steps are performed to find the closest match of a characteristic point  $Q_i$  in the Unknown Pattern  $\Pi_U$  assuming a rotation of  $[\theta_\Delta (\pm 45^\circ)]_{45}$  in Local Pattern:



fig -- 4

- i. The point  $Q_i'$  with coordinates  $(x_{N,i}, y_{N,i})$  in the Normalized CG-PD coordinates of Unknown Pattern  $\Pi_U$  is marked.
- ii. All the points whose Local Patterns have been determined in the Preprocessing step and which lie inside a search circle of radius  $r = r_0 + k_r |CQ_i|$  (measured in the Normalized CG-PD units, and  $|CQ_i|$  being the dist between  $C$  and  $Q_i'$  in  $\Pi_U$ ) around  $Q_i'$  in  $\Pi_U$  are considered.

- iii. Next, the Local Pattern of each of these points are compared with the Local Pattern of  $Q_i$  rotated by  $[\theta_\Delta (\pm 45^\circ)]_{45}$  degrees. This rotation can be conveniently done by just pushing the Local Pattern Vector  $A_\varepsilon(Q_i)$  by  $([\theta_\Delta (\pm 45^\circ)]_{45} / 45)$  elements. Hence rotation of  $A_\varepsilon(P) = [P_1 P_2 P_3 P_4 P_5 P_6 P_7 P_8]$  by 90 degrees may give something like  $[P_7 P_8 P_1 P_2 P_3 P_4 P_5 P_6]$ .

The comparison can be done on the basis of the value of summation of the squares of the differences in the corresponding elements of the two vectors, i.e., difference between  $A_\varepsilon(M)$  and  $A_\varepsilon(N)$  is the dot-product  $(A_\varepsilon(M) - A_\varepsilon(N)) \cdot (A_\varepsilon(M) - A_\varepsilon(N))$ .

- iv. The point with the minimum value of the sum-square-difference is considered as the closest match for the point  $Q_i$  in  $\Pi_U$ . Let's call this point  $S_{i, [\theta_\Delta (\pm 45^\circ)]_{45}}$ .

Hence corresponding to a particular point  $Q_i$  and for a particular rotation considered in the Local Pattern a point  $S_i$  is obtained in  $\Pi_U$  which have the closest match in Local Pattern with  $Q_i$  with that particular rotation being considered.

Hence, corresponding to the  $n$  Characteristic Points  $Q_i$  ( $i=1$  to  $n$ ) in  $\Pi_{D,l}$ , 3 sets of  $n$  points in  $\Pi_U$  are obtained (for each of the 3 angles of rotation). Let these 3 sets be called  $\{S_i\}_{[\theta_\Delta -45^\circ]_{45}}$ ,  $\{S_i\}_{[\theta_\Delta]_{45}}$  and  $\{S_i\}_{[\theta_\Delta +45^\circ]_{45}}$  for the three angles of rotation respectively.

For demonstration of the next step of calculation, the set  $\{Q_i\}$  ( $i=1$  to  $n$ ) and any one of the sets of corresponding closest match points  $\{S_i\}$  ( $i=1$  to  $n$ ) are considered.

Let the difference (i.e. the sum-square-difference) in the Local Patterns in the corresponding points of the two sets be  $e_1, e_2, \dots, e_n$  respectively.

### 2.3.3 Checking the relative positions

Now, the two sets of  $n$  points,  $\{Q_i\}$  and  $\{S_i\}$ , which are the corresponding points of the two patterns are needed to be tested for consistency in relative positioning in the two Patterns. For this purpose, the properties of Complex Numbers can be utilized so that the effects of translation, rotation and scaling in the positioning of the patterns can be easily considered without even taking the CG-PD system as our coordinate system. This process, which allows dealing with the global coordinates of the points rather than the CG-PD coordinates, reduces the time complexities to a great extent.

Let the points in  $\{Q_i\}$  be represented by complex numbers  $q_1, q_2, \dots, q_n$  and those in  $\{S_i\}$  be represented by  $s_1, s_2, \dots, s_n$  in some reference coordinate system of real & complex axes (these need not be the CG-PD coordinate systems, and  $q_i$  &  $s_i$  can be represented in independent coordinate systems. For example, the global coordinate system of the screen may be chosen to represent the complex numbers).

Now, using the elementary properties of complex numbers, it can be stated that if the two patterns are exactly similar and the corresponding points in  $\{S_i\}$  are obtained by just suitable affine transformation of points in  $\{Q_i\}$  (i.e. one can be superimposed on another after suitable rotation, translation and scaling), then the quantity,

$$\rho_{u,v} = \frac{s_u - s_v}{q_u - q_v} \quad (6)$$

is constant for all  $u$  and  $v$ ,  $u \neq v$ .

But because of difference in the patterns, different values of  $\rho_{u,v}$  are obtained. The measure of this error in positioning can be expressed by a convenient non-dimensional complex expression (or rather a vector with two elements),

$$\varepsilon_{u,v} = f\left(\frac{|\rho_{u,v}| - |\bar{\rho}|}{\Delta|\rho|}\right) + i \cdot f\left(\frac{Arg(\rho_{u,v}) - Arg(\bar{\rho})}{\Delta Arg(\rho)}\right) \quad (7a)$$

where,  $i = \sqrt{-1}$ ;  $\bar{\rho} = \frac{1}{n(n-1)} \sum_u \sum_v \rho_{u,v}$ ,  $u \neq v$ ;  $\Delta|\rho| = \sup(|\rho_{u,v}|) - \inf(|\rho_{u,v}|)$  and

$\Delta Arg(\rho) = \sup(Arg(\rho_{u,v})) - \inf(Arg(\rho_{u,v}))$ , where  $\sup(x)$  is supremum and  $\inf(x)$  is infimum of the set of all possible values of  $x$  (i.e. all  $\rho_{u,v}$  for all possible  $u$  and  $v$  in this case),

and,  $f$  is a suitable monotonically increasing even function. We chose  $f(a) = a^2$ .

The real and complex parts of  $\varepsilon_{u,v}$  give the errors in the relative spacing and relative orientations of the points respectively. It can be noted that  $\rho_{u,v} = \rho_{v,u}$  and  $\varepsilon_{u,v} = \varepsilon_{v,u}$ .

Note: It may be mentioned here that the error  $\varepsilon_{u,v}$  has been defined complex just for the mere convenience of including both the errors in relative spacing and orientation into a single variable. Its real and complex parts are as such quite independent. Hence, here  $\varepsilon_{u,v}$  may be viewed more as a vector with two elements  $\{\text{Re}(\varepsilon_{u,v}), \text{Im}(\varepsilon_{u,v})\}$  than a complex number.

### 2.3.4 The Elimination

On the basis of some threshold values of Local Pattern errors  $e_k$  and the relative positioning errors  $\varepsilon_{u,v}$ , some of the points are rejected using a convenient rejection procedure. The number of accepted points be  $n_{accept}$ .

As rejection procedure the following rule was used,

*if* ( $e_k > e_{thresh}$ ) *or* (Number of  $v$ 's satisfying  $(\text{Re}(\varepsilon_{k,v}) > \text{Re}(\varepsilon_{thresh})$  *or*  $\text{Im}(\varepsilon_{k,v}) > \text{Im}(\varepsilon_{thresh}))$  is more than  $p\%$  of  $n$ ) then reject the  $k^{\text{th}}$  point.

where,  $p$  and  $e_{thresh}$  are real constants and  $\varepsilon_{thresh}$  is a complex constants.

After the elimination process,  $\varepsilon_{u,v}$  is redefined for all accepted  $u$  and  $v$  as,

$$\varepsilon_{u,v} = f\left(\frac{|\rho_{u,v}| - |\bar{\rho}|}{\Delta|\rho|}\right) + i \cdot f\left(\frac{\text{Arg}(\rho_{u,v}) - \text{Arg}(\bar{\rho})}{\Delta \text{Arg}(\rho)}\right) \quad (7b)$$

with,  $\bar{\rho} = \frac{1}{n_{accept}(n_{accept} - 1)} \sum_{u,v} \rho_{u,v}$ ,  $u \neq v$ ;  $\Delta|\rho| = \sup(|\rho_{u,v \text{ accepted}}|) - \inf(|\rho_{u,v \text{ accepted}}|)$  and

$\Delta \text{Arg}(\rho) = \sup(\text{Arg}(\rho_{u,v \text{ accepted}})) - \inf(\text{Arg}(\rho_{u,v \text{ accepted}}))$ .

### 2.3.5 The Difference Calculation

Once the values of the errors  $e_k$  and  $\varepsilon_{u,v}$  for the selected points are obtained, the difference between the patterns with respect to the sets of corresponding points  $\{Q_i\}$  in  $\Pi_{D,l}$  and  $\{S_i\}$  in  $\Pi_U$  can be given by two expressions,

$$\Delta_{\Pi_{D,l},\{Q_i\} - \Pi_U,\{S_i\}}^{\text{Re}} = \frac{1}{n_{accept}(n_{accept} - 1)} \left( \sum_{u,v \text{ accepted}; u \neq v} [(\text{Re}(\varepsilon_{u,v}))^\phi \cdot (e_u e_v)^{(1-\phi)}] \right) / \left( \frac{n_{accept}}{n} \right)^{\frac{\eta}{2}} \quad (8a)$$

and,

$$\Delta_{\Pi_{D,l},\{Q_i\} - \Pi_U,\{S_i\}}^{\text{Im}} = \frac{1}{n_{accept}(n_{accept} - 1)} \left( \sum_{u,v \text{ accepted}; u \neq v} [(\text{Im}(\varepsilon_{u,v}))^\phi \cdot (e_u e_v)^{(1-\phi)}] \right) / \left( \frac{n_{accept}}{n} \right)^{\frac{\eta}{2}} \quad (8b)$$

where,  $\phi$  and  $\eta$  are a constants.

The expressions in (8a) and (8b) respectively gives the differences with respect to the relative spacing and relative orientations of the points  $\{Q_i\}$  and  $\{S_i\}$ .

Considering both the above said differences to have equal importance, and that they are not on the same scale, the final measure of the difference can be given by their product,

$$\Delta_{\Pi_{D,l},\{Q_i\} - \Pi_U,\{S_i\}} = \Delta_{\Pi_{D,l},\{Q_i\} - \Pi_U,\{S_i\}}^{\text{Re}} \cdot \Delta_{\Pi_{D,l},\{Q_i\} - \Pi_U,\{S_i\}}^{\text{Im}} \quad (9)$$

And finally, the difference between  $\Pi_{D,l}$  and  $\Pi_U$  is obtained as the minimum among the three values of  $\Delta$  corresponding to the three angles of rotation in Local Pattern,

$$\Delta_{\Pi_{D,l} - \Pi_U} = \min \left( \Delta_{\Pi_{D,l},\{Q_i\} - \Pi_U,\{S_i\}_{[\theta_{\Delta} - 45]_{45}}}, \Delta_{\Pi_{D,l},\{Q_i\} - \Pi_U,\{S_i\}_{[\theta_{\Delta}]_{45}}}, \Delta_{\Pi_{D,l},\{Q_i\} - \Pi_U,\{S_i\}_{[\theta_{\Delta} + 45]_{45}}} \right)$$

### 2.3.6 The final Decision Making

Let database be divided into  $C$  classes ( $C_1, C_2, \dots, C_C$ ) and let the  $i^{\text{th}}$  class has  $c_i$  patterns in the database. Let the  $j^{\text{th}}$  pattern of the  $i^{\text{th}}$  class be called  $\Pi_{D,i,j}$ ,  $i=1$  to  $C$ ,  $j=1$  to  $c_i$ .

Then the difference between  $\Pi_U$  and the  $i^{\text{th}}$  class  $C_i$  can be defined as [1],

$$\Delta_{C_i - \Pi_U} = \min \left( \Delta_{\Pi_{D,i,1} - \Pi_U}, \Delta_{\Pi_{D,i,2} - \Pi_U}, \dots, \Delta_{\Pi_{D,i,c_i} - \Pi_U} \right) \quad (10)$$

And a class  $C_i$  is rejected if the comparison with patterns of that class gives highly irregular predictions, that is,

$$\text{If } \frac{1}{c_i(c_i - 1)} \sum_{a=1}^{c_i} \sum_{b=1}^{c_i} \left| \Delta_{\Pi_{D,i,a} - \Pi_U} - \Delta_{\Pi_{D,i,b} - \Pi_U} \right| > U_{threshold}, \text{ then reject } C_i.$$

Among the accepted  $C_i$ , the one with minimum value of  $\Delta_{C_i - \Pi_U}$  is selected as the class of pattern with closest match.

### 3 ESTIMATION OF COMPLEXITIES

#### 3.1 Time Complexity

On performing some simple analysis it can be easily found that the approximate time complexities in recognizing an unknown pattern of size  $m \times p$  by comparison with a database of  $L$  patterns, are,

- Preprocessing step:
  - Determination of Normalized CG-PD coordinate system:  $O(mp)$ . Each step consisting of approx  $O(10)$  multiplications and  $O(10)$  additions (integer or floating point depending on the nature of  $\pi_{jk}$ ).
  - Determination of Local Pattern of Characteristic Points (50% of total points in the pattern):  $O(mpe^2)$ . Each step consisting approx  $O(1)$  division and  $O(1)$  additions (floating point).
- Comparison with  $L$  Database Patterns, each with  $n$  Characteristic Points:
  - The total Search Process for all the patterns:  $O(Lnr^2)$ . Each step containing approx 3 sets of  $O(10)$  addition/subtractions and  $O(10)$  multiplications (all floating point) and 1 comparison.
  - Checking relative position and difference calculation:  $O(Ln^2)$ . Each step consisting approx 4 sets of  $O(10)$  multiplications and  $O(10)$  addition/subtractions (all floating point).

It can be noted that except the preprocessing step, the complexities don't depend much on the original size of the pattern ( $m \times p$ ). However it depends on  $n$ , which can be logically chosen to be proportional to  $m \times p$ .

#### 3.2 Space Complexity

For storing the data after preprocessing the unknown pattern:  $8mp$  floating point values.  
For storing the  $L$  database patterns in the memory:  $O(10.Ln)$  floating point values.  
Hence, with the present day computers, the space complexity of the algorithm isn't a major problem.

## 4 RESULTS

The algorithm was implemented using Microsoft Visual Basic 6.0 with the following values of the parameters:  $\varepsilon_0 = 0.5$ ,  $\phi = 0.5$ ,  $\eta = 1.5$ ,  $p = 80\%$ ,  $\varepsilon_{thresh} = 0.4 + 0.4i$ ,  $U_{threshold} = 0.5$ ,  $r_0 = 0.1$  and  $k_r = 0.6$ . The value of  $n$  was on an average 49 for each pattern.

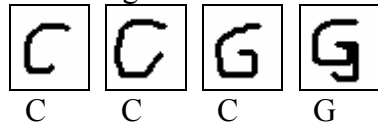
### 4.1 Testing with English alphabets in binary intensity level

Description of the database:

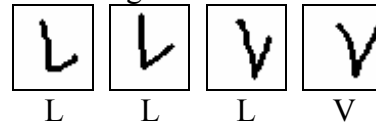
- There were 25 classes of Patterns in the database corresponding to 25 English alphabets (Capital). The database contained a total of 70 entries, i.e. on an average 2.8 patterns corresponding to each class.
- The size of the raw patterns (database and unknown) were 39 x 42 pixels.
- The images used only two intensity levels, namely 1 and 0.

The following are some of the characters used for testing the ability of the algorithm in resolving close looking different characters. The pictures of the characters used for test are shown and the corresponding recognition by the program is written below each of them:

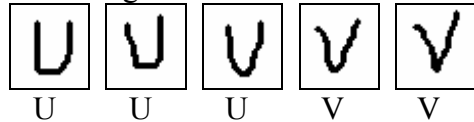
Resolving between 'C' and 'G':



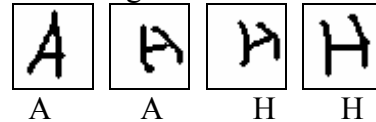
Resolving between 'L' and 'V':



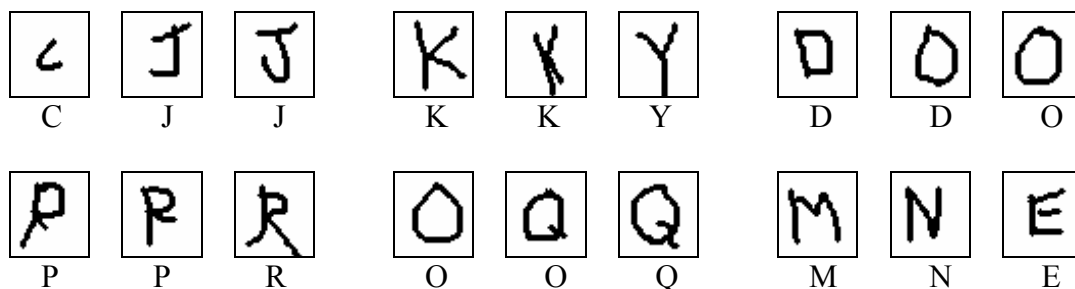
Resolving between 'U' and 'V':



Resolving 'H' and 'A':



Some other example of results obtained from the program:



The time taken by the Visual Basic program for recognizing a new pattern (consisting preprocessing time and the time for comparison with all the 70 database patterns) on a AMD Athlon 2000 machine with 128MB RAM and running Microsoft Windows XP was found to be approximately 3.6 seconds.

## 4.2 Testing with a database of 4 patterns

The test was performed with,

- A database of 4 grayscale patterns as shown below:



Pattern 1



Pattern 2



Pattern 3



Pattern 4

- The size of the raw patterns (database and unknown) were 39 x 42 pixels.

It was then tested with the following unknown Patterns, and the results of closest match obtained are as written below each pattern:



Pattern 1



Pattern 4



Pattern 4



Pattern 3

It may be noted that the matches that the program gives for the unknown patterns with the ones in database pattern are in quite agreement with what our ‘common sense’ tells us!

## 5 CONCLUSIONS

The present work provides a robust structural approach to the problem of pattern recognition. The algorithm tries to mimic the way in which human brain attempts to recognize patterns. Hence, the results from the algorithm are closer to what predicted by a human brain.

The algorithm tested for 25 English capital alphabets gave highly satisfactory results which makes it suitable for application like OCR. The test with the database of 4 patterns demonstrated the ability of the algorithm in mimicking human ‘common sense’.

Among the drawbacks, the time complexity of the algorithm makes it unsuitable for real time processes. Moreover, as the algorithm depends mainly on ‘Characteristic Points’ in the patterns, the algorithm fails more often with patterns which have lesser characteristic features (for example the alphabets ‘I’ or ‘O’).

However, further developments on the present algorithm may be attempted in order to reduce the time complexity and increasing the robustness of the algorithm. This may be attempted by optimizing the various parameters in the algorithm and using better Search process for finding best Local Pattern match. The search process may be attempted to be improved by Genetic Algorithms.

The working of the algorithm may be greatly improved by implementation of a suitably designed Neural Network model for the algorithm. A possible working principle of the NN model may be that the coordinates of the Characteristic Points in the unknown pattern in its CG-PD coordinate system and the corresponding Local Patterns are fed into the model through the input layer, and the outputs from the NN will be the differences with the database patterns. This will reduce the complexity of the processes of Search, Relative position check, Elimination and Difference calculation to a great extent. But this will require the NN model to be a dynamic one so that new patterns may be incorporated into the database.

## ACKNOWLEDGEMENT

I would like to thank Prof. Suman Chakravorti, Dept. of Mechanical Engineering, IIT Kharagpur, for providing me an opportunity to study on and think over Pattern Recognition by assigning the term project for the Computer Graphic and Solid Modeling course. This work is an outcome of that term project only.

I would also like to thank my class mates Mr. Saurabh Gupta and Mr. Dhananjay Neeraj for handling the other half of the term project which dealt with image processing filters, hence letting me concentrate on this problem of Pattern Recognition.

## REFERENCES

- [1] Abraham Kandel, *Fuzzy Techniques in Pattern Recognition*, John Wiley (1982).
- [2] E.P.Popov, T.A.Balan, *Engineering Mechanics of Solids*, Prentice-Hall of India (2002).
- [3] Erwin Kreyszig, *Advanced Engineering Mathematics*, John Wiley & Sons Inc.
- [4] N.G.Das, *Statistical Methods*, M.Das & Co. (1997).
- [5] N. R. Pal, *Soft Computing for Pattern Recognition*, Fuzzy Sets and Systems 103 (1999) 197-200.
- [6] W.Pedrycz, *Fuzzy sets in pattern recognition: Accomplishments and challenges*, Fuzzy Sets and Systems 90 (1997) 171-176.
- [7] S. Pal, D.K. Dutta Majumder, *Fuzzy Mathematical Approach to Pattern Recognition*, Wiley, New York, 1986.
- [8] J.C. Bezdek, *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, New York, 1981.
- [9] B. Shukhat, *Fuzzy patterns recognition*, Proc. ISUMANAFIPS' 95, College Park, MD, USA, 17 20 September, 1995, pp. 788-791.
- [10] J.C. Bezdek, S.K. Pal (Eds.), *Fuzzy Models for Pattern Recognition*, IEEE Press, New York, 1992
- [11] W. Pedrycz, *Fuzzy sets in pattern recognition*, Pattern Recognition, 2/3 (1990), 121-146.
- [12] J. A. Drakopoulos, B. Hayes-Roth, *tFPR: A fuzzy and structural pattern recognition system of multi-variate time-dependent pattern classes based on sigmoidal functions*, Fuzzy Sets and Systems 99 (1998) 57-72.
- [13] C. G. Looney, *Pattern Recognition using Neural Networks*. Oxford University Press, 1999.
- [14] A. S. Pandya and R. B. Macy, *Pattern Recognition with Neural Networks in C++*, IEEE Press, 1995.
- [15] Perantonis, S.J., Lisboa, P.J.G., Translation, rotation, and scale invariant pattern recognition by higher-order neural networks and moment classifiers. IEEE Trans. Neural Networks 3(2) (1992) 241–251.
- [16] R. Srinivasan, C. Wang, W.K. Hob, K.W. Limb, *Neural network systems for multi-dimensional temporal pattern classification*, Computers and Chemical Engineering 29 (2005) 965–981.
- [17] Rui-Ping Li, Masao Mukaidono, I. Burhan Turksen, *A fuzzy neural network for pattern classification and feature selection*, Fuzzy Sets and Systems 130 (2002) 101 – 108.