# ALGORITHMS OF RECOGNITION OF THE PALM OF A PERSON INVARIANT TO AFFINE TRANSFORMATIONS

Raed Fathi Sahawneh Faculty of Science and Information Technology, Irbid National University- Jordan, sahawneh\_115@yahoo.com

## ABSTRACT

This article deals with the algorithms of recognition based on computing estimations. Algorithms of recognition of the palm of a person have been developed on the basis of this approach invariant to affine transformations.

Keywords: Algorithm, Affine Transformation, Recognition, Person's Palm.

#### **INTRODUCTION**

Let us have a given object set  $\Omega = \{w_i\}$ , This set has to be split into subsets (classes)  $\Omega_i$ ,  $i = \overline{1, m}$ ,  $\Omega = \bigcup_{i=1}^{m} \Omega_i$ . All objects are described by the vector of attributes  $x = (x_1, ..., x_i)$ . We shall name this vector the vector of non-derived (or prime or basic) attributes. Each of the attributes can assume values from some set A (alphabet). This can be {0,1} attribute non-present or present respectively; {0,1, ..., n} - degree of evidence of the attribute has different gradations; a set of real numbers from a segment [a, b]; a set of symbols or lines of symbols, etc.

Splitting of the set  $\Omega$  is not given complete, only some information on the classes  $\Omega_i$  is given in the form of the set  $\Omega^*$ , where  $\Omega^* = \bigcup_{i=1}^m \Omega^*_i$ ,  $w_{i,j} \in \Omega_i - \text{known}$  (given) representatives of class  $\Omega_i$ .

$$\Omega^* \in \Omega$$
,  $\Omega^*_i \in \Omega$ 

The sets  $\Omega_i^*$  are finite and let  $|\Omega_i^*| = n_i$  be the number of representatives of i-th class.

Such prior information on splitting  $\Omega$  for the recognition algorithm is given by the 3D structure

$$T = \{x_{i,jr}\}, i = \overline{1, m}, j = \overline{1, n_i}; r = \overline{1, l}$$

 $x_{i,jr}$  – r-th element of the vector of attribute of the description of j-th representative of the i-th class. Such 3D array of prior information can be presented in the form of the two-dimensional array (the so-called learning table  $T_{N,1}$ ). Each line of the table is a vector of attributes, whose

lines are set as follows: lines with numbers from 1 to  $n_i$  be descriptions of objects of class  $\Omega_i$ , lines with numbers  $n_1 + n_2$  – descriptions of objects of class  $\Omega_1$ , etc.

## METHODOLOGY

Let us examine the full set of attributes  $(x_1, ..., x_l)$ , distinguish the system of subsets of the set of attributes  $(S_1, ..., S_k)$  (a system of the algorithm reference sets). Let  $|S_1| = q_i, q_j < 1$ , each subset constitute  $S_i$  of attributes from the complete set of  $q_i$ . Let  $|S_i| = (S_{i,1}, ..., S_{i,q})$ , we shall call derived attributes (or secondary) and pack with two-dimensional structure  $S = \{S_{i,j}\}, i = \overline{1, k}; j = \overline{1, q_i}$ .

Introduction of derived attributes is caused by the fact, that the main information for recognition is often contained not in separate attributes, but in their combination. Such combination of prime attributes we have called as derived attribute.

Informally the algorithm of recognition based on estimation now can be formulated as follows: vector of attributes of the object w that has to be identified is compared with the learning table  $T_{N,1}$  taking into account the system of reference subsets  $S_1, ..., S_k$  to take the decision to which class  $\Omega_i$  the object has to be attributed.

Prior to the formal description of the algorithm we shall formulate its structure of data as stated above.

Recognition algorithm can be described now more formally in the above said denotations and structures.

It is necessary to take a decision on attributing object w, described by the vector of attribute  $y = (y_1, ..., y_j)$  to a certain class  $\Omega_i$  proceeding from Tand S.

1. For each representative  $w_r(r = \overline{1, n_1})$  of class  $\Omega_i$  and reference subset  $S_i$  we formulate estimation of proximity of object w described  $y = (y_1, ..., y_j)$  that has to be recognized (input object).

$$O_1(\mathbf{y}, \mathbf{x}_i, \mathbf{s}_i, \Omega_j) = O_1\left((\mathbf{y}\mathbf{s}_{i1}, \dots, \mathbf{y}\mathbf{s}_{iq_i})(\mathbf{x}_j\mathbf{s}_j, \dots, \mathbf{x}_{ji}\mathbf{s}_{qi})\right)$$

2. Then we formulate estimate of the input vector in all representatives of j-th class of *i-th* reference subset

$$O_{2}(y, s_{i}, \Omega_{j}) = O_{2}(O_{1}(y, x_{i}, s_{i}, \Omega_{j}), \dots, O_{1}(y, x_{iq_{i}}, s_{i}, \Omega_{j}))$$

Table	1
-------	---

Denotation	Explanation
L	Set of values (alphabet of presentation of the objects' description to
	the vector of attributes)
Ι	Length of vector of attributes of the objects' description (number of
	non-derived attributes)
$x = {x_i}; i = \overline{I, L}$	Vector of attributes of the objects' description $x_i \in L$

m	Number of classes of recognition, that is, number of subsets $\Omega_1$ of the
	set $\Omega$
n	Number of representatives of class $\Omega_1$ in the learning table $n_i = \ \Omega_i^*\ $
$T = \{t_{ij_r}\};$	Algorithm learning table, prior information on splitting the set $\Omega$
$i = \overline{1, m}; j = 1, n;$	
$r=\overline{1,I}$	
К	Number of reference subsets of the algorithm (number of derived
	attributes)
$q_i; j_i = \overline{1, K}$	Number of components of the vector of description of the object
	included in i-th derived attribute $q_i =  S_i $
$S = {s_j}; i = \overline{l, K};$	Two-dimensional structure of description of indices the vector of
$j = \overline{1, q_1}$	attributes included in non-derived attributes

3. We formulate estimation of the input vector in all reference subsets for each *j*-th class

$$O_3(\mathbf{y}, \Omega_j) = O_3((O_2(\mathbf{y}, \mathbf{s}_i, \Omega_j), \dots, O_2(\mathbf{y}, \mathbf{s}_k, \Omega_j)))$$

Proceeding from estimations  $O_3(y, \Omega_j)$  of the input vector in each  $W_j$  class a decision is taken on attributing object  $w^1$  to one of the classes  $\Omega_j$ ,  $j = \overline{1, m}$  on refusal from recognition.

The rule of solution can assume various forms. Recognition object  $w^1$  can be attributed to the class to which maximum estimation  $O_3$  corresponds, or if it exceeds estimations in all other classes at least by a specified threshold value p, or if its ratio to the sum of all other estimations will not be below some threshold  $p_i$ , etc. The choice of the rule of solution and its parameters depends on the concrete task of recognition, on prior information and practical experience in solving such problems.

Now we shall formulate the described structure of the data of recognition algorithm in terms of the programming language (Pascal).

Object of program	Denotation	Explanation
Ι		Number of non-derived attributes
m		Number of recognition classes
k		Number of derived attributes
$x[\overline{1,I}]$	$\{x_i\}$	Vector of attributes
$n[\overline{1,m}]$	n <sub>j</sub>	Number of representatives in class
$T[\overline{1, m}, \overline{1, n_{j}, \overline{1, I}}]$	$\{l_{i_r}\}$	Learning table
$q[\overline{1,k}]$	$q_i$	Number of necessary attributes in
$S[\overline{1,k},\overline{1,q_1}]$	S <sub>ij</sub>	Two-dimensional array of indices of derived attributes formation

Table 2

Let us denote F\_01, F\_02, and F\_03 functions, that recurrently determine partial estimations  $O_1, O_2$  and  $O_3$  respectively, and G\_01, G\_02 and G\_03 – finite formation of

estimations. Below is recognition algorithm in the form of a program module that forms an array  $O[\overline{1,m}]$  of the estimations of input object in each of the classes  $\Omega_i$ . Proceeding from the values of this array a decision is taken on attributing input object to a certain class.

Let us estimate the necessary number of operations in terms of computing functions  $O_1, O_2$  and  $O_3$ .

Suppose, that  $N = \sum_{i=1}^{n} n_i$  - total volume of learning table, that is, total number of vectors in T.

Internal cycle with  $J_1$  is performed q|i| times. Then, with complete i = i0, when the cycle with j is performed k times, internal cycle with  $j_1$  is performed:

 $n_{i_0} \sum_{j=1}^k q_i$  times

then we shall have the following estimations:

01 - is calculated N  $\sum_{i=1}^{k} q_i$  times.

02 - is calculated N \* k times.

03 - is calculated m \* k times.

Now we shall fill with the concrete content the notion of proximity and unveil the operations:  $O_1$ ,  $O_2$  and  $O_3$ .

Let there be given a certain set M. The real function d(a, b) is called the function of distance (metric) provided, that:

- 1. d(X, Y) > 0 for all X and Y from M.
- 2. d(X, Y) = 0 then and only then when X = Y.
- 3. d(X, Y) = d(X, Y)
- 4. d(X, Y) < d(X, Z) + d(Z, Y) for all X, Y and Z with M&.

In practice the most often occurring cases are when such data can be analyzed that can be presented in n-dimensional Euclidean space  $E_n$ .

Numerous different metrics have been elaborated for this space that is widely used in solving practical problems. Table 3 presents some of them.

1	Euclidean distance	$d(X, Y) = \left[\sum_{i=1}^{n} (x_i - y_i)^2\right]^{1/2}$
2	L <sub>i</sub> – norm absolute value	$d(X, Y) = \sum_{i=1}^{n}  x_i - y_i $
3	Supremum norm	$d(X, Y) = S_{n,p}\{ x_i - y_i \}$
4	1 <sub>p</sub> - norm	$d(X, Y) = \left[\sum_{i=1}^{n}  x_i - y_i ^p\right]^{1/p}$

 Table 3: Different Metrics

5 Mahalanobis distance 
$$d(X, Y) = (X - Y)^{T}W(X - Y)$$

Euclidean metric is most often used. Supremum-norm is the simplest in calculations. Mahalanobis distance is often called the generalized Euclidean distance. This distance is invariant with regards to the non degenerate linear transformation W - scattering matrix is determined as follows:

$$W = n(X - Y)(X - Y)^{T/2}$$

Rather often there are other heuristic measures used that are not distances in the sense of the said definition but are used practically. Among them, for example, is the Jeffries- Matsushita measure

$$d(X, Y) = \left[\sum_{i=1}^{n} (\sqrt{x_i} - \sqrt{y_i})\right]^{1/2}$$

and the measure known under the name of the coefficient of divergence

$$d(X,Y) = \left[ (\sum_{i=1}^{n} (\frac{x_i - y}{x_i + y})^2) / n \right]^{1/2}$$

These measures have been borrowed from the probability theory and mathematical statistics.

Thus, we have considered some of the examples of the distance function between vector I  $E_n$ . In all cases with  $X_1 = Yd(X, Y) = 0$ . That is, if the vectors coincide, the distance between them is equal to zero.

Now let us consider the notion of resemblance of the set members. This notion to some extent is opposite to the notion of distance, that is, the greater is the distance between the members, the lesser they are similar and vice versa, the lesser is the distance between the set members, the more similar they are.

Formally similarity measure is defined as the real function D(X, Y) that satisfies the following conditions:

1. 
$$0 < D(X, Y) < 1, when X \neq Y.$$

2. 
$$D(X, Y) = 1$$
 when  $X = Y$ .

3. D(X, Y) = D(X, Y).

If the vector elements take values o or 1, then their similarity measure is often called the coefficient of association. There is a great number of types of the coefficients of association. Let us examine some of them making some denotations.

Let 
$$n_1$$
 be the number of equalities  $x_i = y_i = 1$ ;  
 $n_0$  - number of equalities  $x_i = y_1 = 0$ ;  
 $n_{01}$  - number of equalities  $x_i = 0$ ;  $y_i = 1$ ;

 $n_{10}$  - number of equalities  $x_i = 1$ ;  $y_i = 0$ ;

for all  $i = \overline{1, n}, x_i \in X, y_i \in Y$ .

Table 4 shows examples of the coefficient of similarity presented in the introduced terms.

$D(X, Y) = n_1(n_1 + n_{01} + n_{10})$
$D(X, Y) = (n_1 + n_0)/n$
$D(X, Y) = n_1/n$
$D(X, Y) = 2n_1 / [2n_1 + n_{10} + n_{01}]$
$D(X, Y) = 2(n_1 + n_0)/(n + n_1 + n_0)$
$D(X, Y) = n_1 / (n_1 + 2(n_{10} + n_{01}))$
$D(X, Y) = (n_1 + n_0)/(n + n_{10} + n_{01})$

 Table 4: Coefficient of Similarity

Coefficient of correlation or similar to it coefficients is often used as the measure of similarity. Though these coefficients do not correspond to the definition because they take values from the segment [-1, 1].

One of such coefficients:

$$D(X,Y) = \sum_{i=1}^{n} x_i y_i / \left[\sum_{i=1}^{n} x_i^2 * y_i^2\right]^{1/2}$$

is invariant to the linear transformation.

The norm in  $E_n$  and some threshold P are often used in the problems of recognition to determine the measure of similarity as follows: D(X, Y) = 1 provided that  $||X, Y|| \le P$ , otherwise D(X, Y) = 0.

There are multivariate measures of similarity. Two vectors X and Y are regarded as similar if inequalities of the type  $|x_i - y_i| < a_i$  are made at least P times, that is,

$$D(X,Y) = \begin{cases} 1, \text{ if } \sum_{i=0}^{\infty} (|x_i - y_i| \le a_i) \ge P\\ 0, \text{ in other case} \end{cases}$$

for i = 1,  $n(|x_i - y_i| < a_i) = 1$ , if inequality is made, otherwise this expression will be equal to zero.

In this case the function of similarity takes only two values, 0 or 1. The measure of similarity can be determined by gradation of proximity, that is:

$$D(X,Y) = \frac{[\sum(|x_i - y_i| \le a_i)]}{n}, \text{ or } D(X,Y) = \sum(|x_i - y_i| \le a_i)$$

In the latter case X and Y are completely similar (coincide) when D(X, Y) = n. Such determination of measure corresponds to the introduced value, but can be used in practice.

We shall describe the presented neural elements in the programming language Pascal by the following data types:

NEd $(x, n, p, w)$	NEd $(x, n, p)$
Type NE_d= record	Type _WE= record
d, n, out: word;	d, n, out: word;
X: array [1,,n] of real;	p, A of real;
W: array [q,,n] of real;	X: array [l,,n] of real;
end;	end;

out - value of "neuron output" (function of activation);

A – value of neuron accumulator (storage, partial or total weighted sum);

n – number of NE inputs;

p - threshold;

X – array vector of input data;

W – array vector of weight coefficient;

D – number of gradations of neuron output value.

Traditional patterns of computing with functioning neural networks and neural elements do not envisage a fixed processing procedure. That is why calculations are assigned in a nonprocedural form. In case of program realization of NE using traditional sequential computers calculations are made sequentially.

Let it be a Pascal-like construction:

A1: A:=0 For i:-1v to n do A: = A + X [i] \* W [i] Out: = F (A, p, d)

Function F determines value of NE output depending on the weighted sum A, threshold P and the number of neuron output gradations d. The weighted sum A, is formed sequentially in the cycle. Execution of this fragment requires n addition operator, n multiplication operators and performance of threshold function  $F_i$ .

Access to the vector elements in  $A_1$  is sequential, that is, the vector arrives for the processing component by component  $(x_1, ..., x_n)$ , in case of parallel realization of the algorithm the vector arrives in parallel, that is, access to all elements of input vector is simultaneous.

Algorithms NE (sequential, parallel and sequential-parallel) will be presented in the form of the graph G(V, E) where V – node of the graph, E – arcs. In our case V – input data or operations, G – information links between operations or input data and operations. We shall describe such graphs by a multilevel structure as follows:  $< m(n_i, ..., n_m), \{(i_1, ..., j_1) - (i_2, ..., j_2)\} >,$ 

where m – number of levels;  $n_i$  – number of nodes on the i-th level (i = l, m);

 $(i_1, ..., j_1) - (i_2, ..., j_2)$  – arc from the node of the graph located on the  $i_1$ -th level and has on this level its serial number  $j_1$  to the node with the serial number  $j_2$  of  $i_2$ -th level.

Let us examine algorithm  $A_1$  filling with a concrete content the functions F and A mentioned in it.

On the lowest level of calculations in algorithm  $A_2$ , that is in the cycle with the deepest location of  $(j_1)$  similarity of input vector x to the  $i_1$ -th representative of i-th class is calculated with respect to j-th attribute. The sum of coincidences of input vector X with vector from table Ti,j is formed and the measure of similarity 0 or 1 is determined in accordance with threshold P[i]. Each derived attribute has its threshold - P[i].

We shall consider realization of such cycle in a greater detail.

Suppose we have two vectors X and Y.

{cycle in classes  $T_i$ },

{cycle in representatives of  $classT_i(T_{ij})$ },

{cycle in reference subsets (derived attributes)},

{cycle in j-th reference subset },

{formation of estimation of similarity of input vector for the *i*-th representative of the *i*-th class},

{formation of estimation of similarity for the *i*-th class},

{taking a decision on attributing object described by vector X to class  $i_0$  }.

Symbol # denotes 'additive inversion in module 2' (mod 2).

 $x_i$  and  $y_i$  (i = l, ..., n) take values 0 or 1. The vectors are regarded as similar if their components coincide ( $p \le n$ ) not less than p. If p is equal to n, then the vectors coincide completely, otherwise there is a partial coincidence. We shall define the measure of similarity by function:

$$F:\sum_{i=1}^{n}(x_{i}\#y_{i}) \geq P$$

that takes value 1 if inequality is performed, value 0 if inequality is not performed. Symbol # denotes operation 'additive inversion in module 2'

The defined function p can be realized on the n-input neural element with the activation function:

$$\sum_{i=1}^{n} z_{1} \ge P, \text{ where } z_{1} = x_{i} \# y_{i}$$

We shall denote it by the expression NE(x, y, n, p), fig.1a shows the diagram of such neural element.



Fig. 1 a. Diagram of neural element.

Fig. 1b showing diagram of this element presents formation of inputs. In our case symbol # means 'additive inversion in module 2'. In a general case this symbol corresponds to the operation that lies at the base of the measure of space of the attributes' values (distance or similarity).



Fig. 1b. Diagram of neural element with the revealed formation of inputs.



Fig. 1c. Developed parallel diagram of summing input values in neural element.

Fig. 1c shows a developed parallel diagram of summing input values in neural element. This diagram of the incomplete binary tree that is performed in [log n] steps [a] means ceiling a. Thus, one step of operation of NE (actuation of NE) is equal to [log n] steps of comparisons with the threshold and one step of execution of (#) operation. The above said includes the case of synchronous parallel work of NE.

Let us consider an example.

Suppose a set  $\Omega$  presents images of printed digits (0,1, ...,9) and is split into 10 subsets  $\Omega_i$ . For practical realization and testing the recognition algorithm we shall form the 'real material', that is, noisy and 'slightly broken' images of digits, that is, we shall subject the 'ideal' standard images of digits to the effects of impediments and, therefore, the formed images will be the material for recognition.

We shall examine the bit images, that is, each pixel of the image is equal to 1 or 0 (blackand-white binary images). The effect of impediment will be determined as follows: matrix  $n_1 * n_2$ , whose k elements are equal to 1 and the rest are equal to 0, with unities placed evenly in the matrix that we shall call processes intensity impediments 100K  $n_1 * n_2$  (uniform intensity impediment P%). With image matrix O and impediment matrix Z we shall form the noisy image matrix R by the function:

$$R = (0, Z) \mod 2, r_{ij} = (O_{ij}, Z_{ij}) \mod 2$$

Thus, images presented by matrices  $R = \{r_{ij}\}, i = 1, n, j = 1, n$ , will be the material for recognition.

We have  $n_1 = 8$ ,  $n_2 = 8$ , then l = 8 \* 8 = 64 of the length of the vector of non-derived attributes, m = 10- the number of classes for recognition. Let  $n_1 = n = 5$ , i = 1, ... 10 - the number of representatives of each class. Learning table T was formed via program. We shall select column lines of matrix R for the reference subsets. Then K = 8 - 8 = 16.

Reducing the two-dimensional problem of recognition to the one-dimensional problem that we have discussed we shall make a transition.

We shall present matrix R by vector  $X = \{x_i\} = 1, ..., 64$ , matrix 'elongates' into vector line by line starting from the left upper element [1] from left to right and from top to bottom). Thus the learning table T is formed and index sets  $S_i, S_1, ..., S_8$ - lines of matrix R, R9...R16 – columns R, Ti, is the matrix representing *j*-th representative of *i*-th class. First level neurons are NE( $S_j(x)#S_j, T_j$ )8,8). Let the class of neurons determine similarity of input vector to the *r*-th representative of *i*-th class relative to  $S_j$ . Second level neurons determine estimation of similarity of input X to  $T_j$  in all  $S_j$  and, finally, third level neurons determine estimation of similarity of X to  $\Omega_j$ .

### CONCLUSIONS

Recognition algorithms based on computing estimation have been considered and the possibility of using these algorithms for solving the problem of identification of a person by the image of his palm has been shown. The synthesized algorithms of recognition of the palm are invariant to affine transformations.

# REFERENCES

- 1. Zhuravlev Yu. I., Nikiforov V. V. Algorithms of recognition based on computing estimations. Cybernetics. 1976. N3, p. p. 1-11.
- Gorelik A. L. and Skripkin V. A. Methods of recognition. M., Higher School publishers, 1977. - p. 222.