Antonio DONNARUMMA (*), Maurizio ORLANDO (**), Giovanni PODDA (**)

(*) Istitute of Mechanical Engineering, University of Salerno, Italy.
(**) Departement of Production Systems and Economy, Politechnic of Torino, Italy.

AN HYBRID SYSTEM FOR THE RECOGNITION OF "QUASI-PLANAR" OBJECTS, BASED ON FOURIER DESCRIPTORS, BY MEANS OF NEURAL AND THE DISTANCES METHOD.

Summary. This paper presents an object recognition hybrid system capable of recognizing quasi-flat objects with invariance with respect to translation, rotation and scale. Objects which present themselves in different configurations (such as pliers and callipers, which may be opened or closed) are also recognized. This system is obtained by merging conventional techniques of image-processing such as boundary detection and Fourier descriptors with neurocomputing techniques and the distances method.

Introduction: object description

This paper deals with the recognition of objects for which the thickness is small in comparison with the other two dimensions (pliers, callipers, spanners placed on a flat surface), allowing them to be identified with sufficient precision by a planar view .
In order to characterize such objects the polar contour descriptor, described in recent literature [1], [4] and used by the authors [6], [7], [9], is employed.
The digitalized image of the object under consideration comes from a video camera and is shown on a two-level display. Its contour is found by a following contour algorithm [13], starting from the first point where the raster scan meets the object (the highest point on the left of the displayed image).
The origin of the curvilineard abscissa s on the contour is assumed in this first point; the distance between contiguous 4-connected pixels is taken as 1 unit, the distance between diagonally contiguous pixels is taken as $\sqrt{2}$ units. The barycentre is assumed as the pole.
Each point of the contour is therefore defined by the curvilinear abscissa and by its distance from the pole $\mathbf{d=d(s)}$.

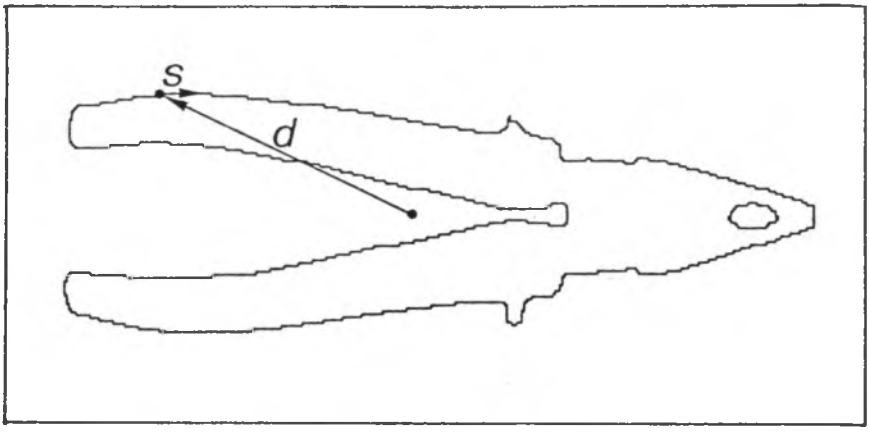


Fig. 1. Reference system for the polar diagram **d=d(s)** of the image contour

The translation invariance is already inherent in this approach as every image is described with respect to the barycentre of its contour. The scaling invariance is achieved for two images by extracting 256 equidistant samples from the **d=d(s)** diagram of each image and changing the distance d(s) of an image by a factor equal to the square root of the ratio between the areas subtended by the two diagrams.
Rotational invariance can be achieved by considering the diagrams as circular lists and rotating them until the best match is found.
This procedure can, by itself, be considered a recognition system [6], [9].
A second way for obtaining rotational invariance consists in orienting each object according to its main axes of inertia: in this case the contour description starts at a point, where one of these axes intersects the contour in the maximum (or minimum) distance from the pole [11].
In the present paper the rotational invariance is achieved by carrying out the Fast Fourier Transform (FFT, [19]) of the diagram **d=d(s)** after sampling, scaling, and calculating its module.
This procedure can be crucial in regard to the intrinsic description of the contour, via the 'sampled values' because of the presence of high frequencies which may not be well represented by the values assumed by the transformation components.
The first problem is resolved by using a sampling frequency about ten times higher than the maximum one. As for the second problem, it has been resolved by checking that for all the examined objects, the eleventh and higher modules are significantly neglegible, and by deciding not to use components higher than the sixteenth for the recognition.
With these limitations the procedure, while allowing precise recognition of the examined objects on the basis of FFT, can not be generalized for reconstruction of the object. Therefore for reconstruction extraction of geometrical descriptors is preferable [12].
On the basis of the FFT we tackled the problem of the objects recognition by using two different approaches:

1) the neural paradigm

2) the fuzzy sets theory.

1. Approach by means of the neural paradigm

The task of recognizing an object on the basis of Fourier descriptors was handled by an adaptive artificial supervised neural network (Multilayered Back Propagation Network, M.B.P.N.).
This was not an a-priori choice: as far as the authors are concerned, there is no a general theory which allows the designer to choose the network architecture, for the particular problem to be solved, on the basis of rigorous criteria. However, the many experiences reported in literature [3], [5], [8] and [10] show that the M.B.P.N. architecture is well suited for problems concerning visual pattern recognition and classification (after an appropriate preprocessing of the patterns).

1.1. The M.B.P.N.: special features

The strength point of the B.P.N. network lies in the possibility to build inside it the mapping function f between the input vectors Xi (i=1,2,3..n) and the output vectors Yi (i=1,2,..m). Xi belong to a subset of an Euclidean space with n dimensions whereas the vectors Yi belong to a mapped subset f[A] of an m-dimensional Euclidean space (1).

$$f : A \subset \mathbf{R}^n \Rightarrow f[A] \subset \mathbf{R}^m \qquad (1)$$

The building process for the function f (i.e. the training of the network) takes place on the basis of an assigned learning rule [2], an assigned activation function S and the repeated presentation of K correct association (the "training set") between Xi and Yi (2).

$$(X_1,Y_1),(X_2,Y_2),\ldots,(X_K,Y_K) \text{ ove } X_i \subset A \subset \mathbf{R}^n ; Y_i \subset f[A] \subset \mathbf{R}^m \qquad (2)$$

The network learns by memorizing the partial corrections ΔW_{ij} in its processing elements (p.e.) at each presentation and only at the end of the K presentations does it updates the weights based of the average of the partial corrections ΔW_{ij} (eq. 3).

$$W_{ji,new} = W_{ji,old} + \frac{1}{K} * \Sigma_K \Delta w_{ji} \qquad (3)$$

The network is trained with N cycles of K presentations. The values N and K depend on the required performance as well as on the information environment. The values of N and K are stated by the project team leader.
A further strenght point of the M.B.P.N., inherent in the rule for determining the changes in the weights (eq. 4, 6, 7 and 8), is that the global error (eq. 5) generally diminishes more and more as the learning process goes on (some exceptions exist and are documented in [5]).
The learning algorithm is based on the formulas shown below, where ΔW_{ij} is the change in the weight connecting the generic element i with the generic element j of the adjacent layer, α (by definition positive) is the learning rate, y0 and y0' are, respectively, the desired and the actual answers of the network, F is the error (mean square error, m.s.e.) calculated over the K presentations of the training set, f(Si) is the gradient of the transfer function for the i-th element, dj(L) is a dummy variable useful for defining the recurrence formulae.

$$\Delta W_{ji} = -\alpha \frac{\partial F}{\partial W_{ji}} \qquad (4)$$

$$F = \frac{1}{K}\Sigma_K[\frac{1}{2}\Sigma_o(y_o - y_o')^2] \quad (5)$$

$$\Delta W_{ji}(L) = -\alpha\, y_i(L-1)\, d_j(L) \quad (6)$$

$$dj(L) = (y_{oj} - y_{oj}') * f'(S_j) \quad \Rightarrow \text{(output layer)} \quad (7)$$

$$dj(L) = f'(S_j) * \Sigma_r [d_r(L+1) * W_{rj}] \quad \Rightarrow \text{(hidden layer L.} \quad (8)$$

The equations (6), (7), and (8) above are based on condition (4), which imposes that the changes ΔWij applied at the weights Wij must be proportional to, and have opposite sign from the derivative of the error F with respect to the weight. The proportionality factor α (learning rate) can be modified during the training by the operator.

1.2. Applied training strategies

It is helpful to consider the p actual synaptic connections (i.e. weights) of the network as vector elements (the vector of the weights) in a p-dimensional space. Each weight vector is associated with a value of the global error F of the network in a (p+1)-dimensional space. The error points form a (p+1)-dimensional hypersurface (error hypersurface) which, so to say, spans over the p-dimensional space of the weight vectors.
Generally, nothing is known of the topology of this hypersurface. However the recent technical literature [5] and [8] reports the results of extensive experimentations, on the basis of which one can make some general considerations which can be used as guidelines for the practical training of the network.
- The hypersuface can contain quasi-flat, extended zones characterized by a very small slope. In these zones learning is slow and therefore, in order to accelerate the convergence speed, it is advisable to increase the learning rate value α. On the other hand it has to be kept in mind that an increase of α also increases the downhill step along the error hypersurface and therefore bears the risk of "jumping" over a minimum.
- In most practical cases it has been found that the error hypersurface has a manifold of global minima (corresponding to the different permutations of the same weight vector element values)
- The error hypersurface is ragged and therefore has a lot of local minima. The presence of local minima can be a serious obstacle for reaching a global minimum: in fact, if one enters into the "attraction basin" of a local minimum during the training, the condition (eq. 4) which imposes progress in the direction of decreasing error prevents to exit from this zone.
It can be determined (however with some uncertainty) whether a minimum is local or global. The values of the weights are randomly changed and the resulting behaviour of the network is studied. A random perturbation of the weights translates into a random move on the error hypersurface. In the first case, if the basin is not too large this may suffice for leaving this "valley" and learning can be successfully continued until another minimum is encountered. In the second case the minimum remains stationary even after several perturbations of the weights and it can be assumed that a global minimum or a probable optimum condition for this network topology has been reached.

As already mentioned in the previous paragraph the training cycle number N and the number K of correct association examples depend on the specific application. In the presented case the target was to evaluate the ability of the network to learn "by heart" the whole set of examples

presented ("associative memory") as well as the ability of generalizing knowledge supplied via a reduced set of examples ("intelligent network"). In fact it is this last aspect which makes the use of an approach based on non-traditional logic (such as the neural paradigm or the fuzzy logics) so interesting [14], [15], [16], [17].

1.3. Experimentation of the system

The recognition system was tested with a set of 18 bilevel digital images. Six of them represent a calliper, six pliers and six a spanner. The objects were taken at various distances and in different orientations. The calliper and the pliers were taken with various degrees of opening.
For each image the moduls of the first 16 components of the FFT [19] of the normalized polar contour diagram (fig. 3, 4 and 5 on the next pages) were calculated. The network is trained by presenting these moduls to the input layer. Accordingly the network has to classify the object as a calliper, pliers or spanner.
Therefore the network has an input layer of 16 elements and an output layer of three elements. Also, it has one hidden layer of 12 neurons. The activation function S is a parametrized sigmoid. Each layer is fully connected to the next higher-level layer, resulting in 228 weights (fig. 2).

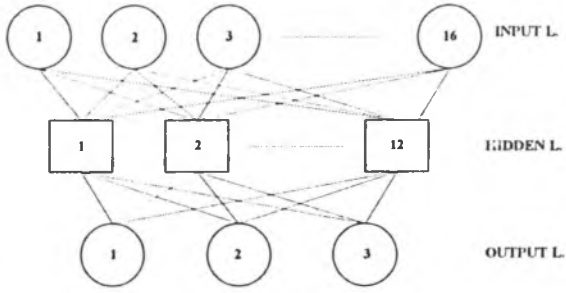
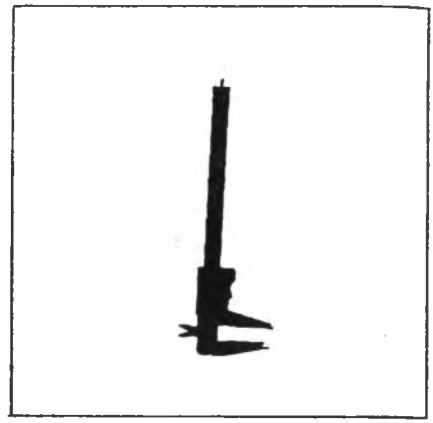
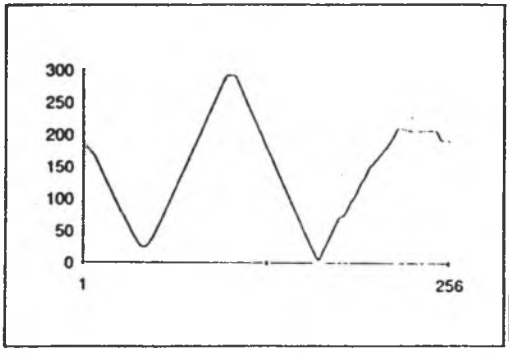
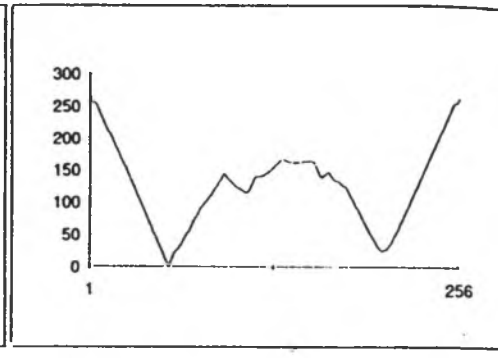


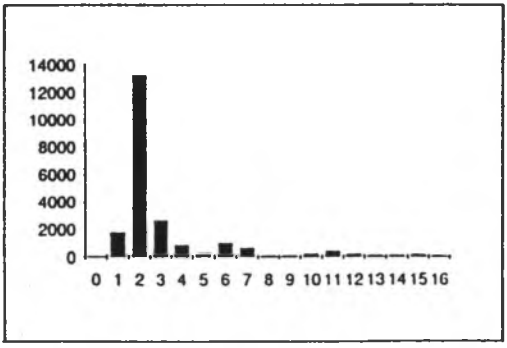
Fig. 2. The layout of the network built, trained and tested in this paper

The EXPLORENET package under Windows 3.1 was used for building training and testing the network. This package can run independently (version 3000) as well as (version 3001) with the special high-performance hardware (BALBOA 860 HNC board with 10 MB RAM) installed on a P.C. Zenith Z-320/SX in the A.V. laboratory of the D.S.P.E.A. of the Politechnic of Torino.
Explorenet implements 21 network architectures [18], among which is also M.B.P.N. (multilayered B.P.N.), as well as a G.U.I. providing icons and customable windows, which hel us monitor the evolution of the network once it has been built (activation status of the elements, values of the weights, value of the m.s.e. etc.) and to control the training by actions (also programmable via "task") on the parameters and run time flags (learning rate, training enabling on/off, statistic enabling on/off etc.).









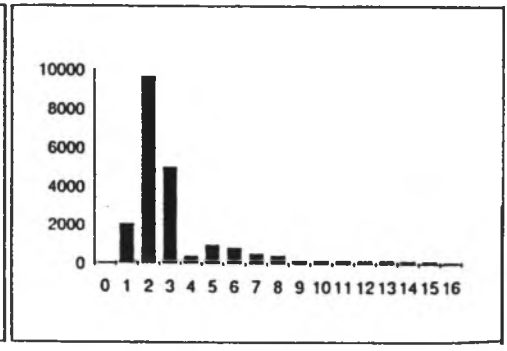


Fig. 3. Polar diagrams and FFT modules of the calliper

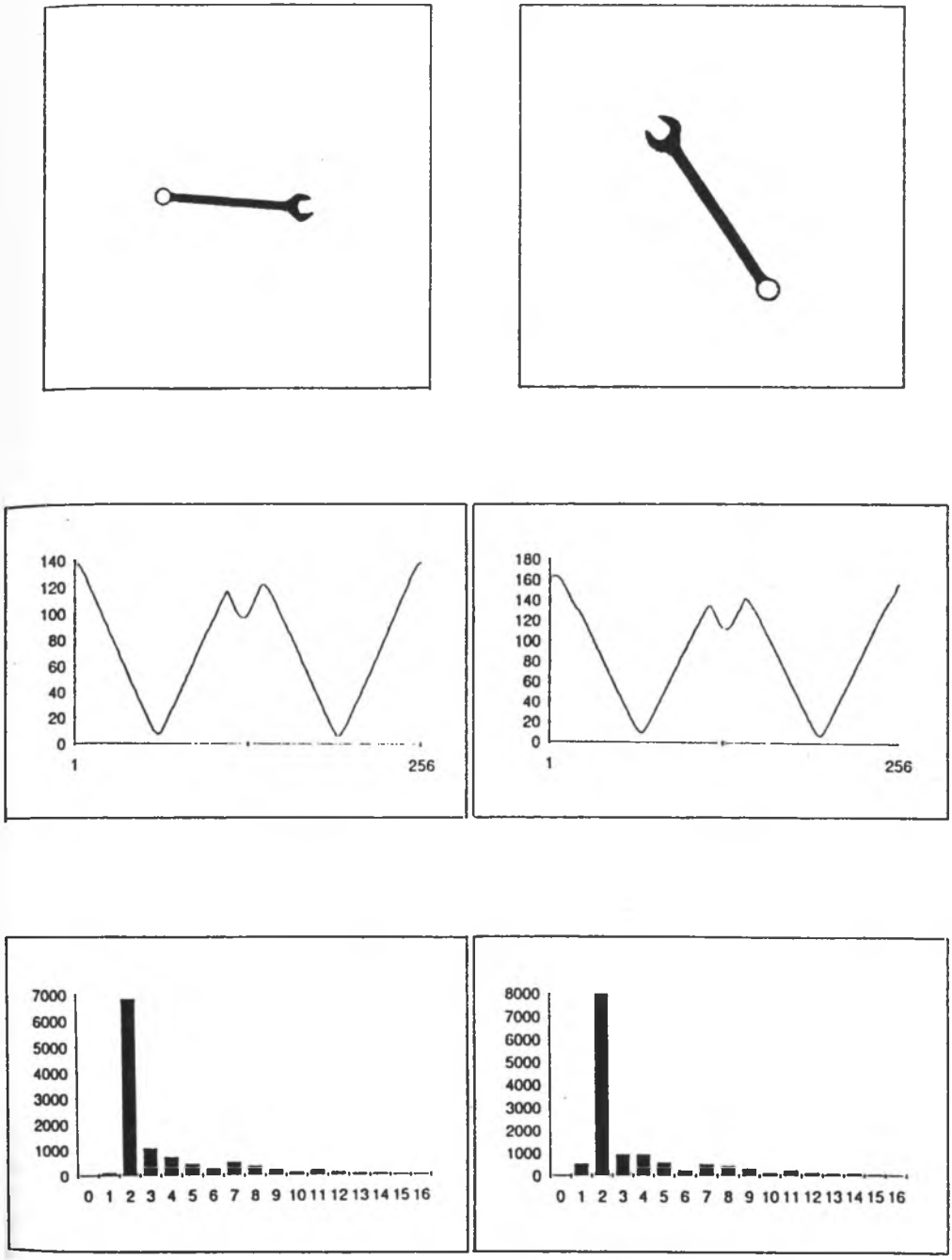
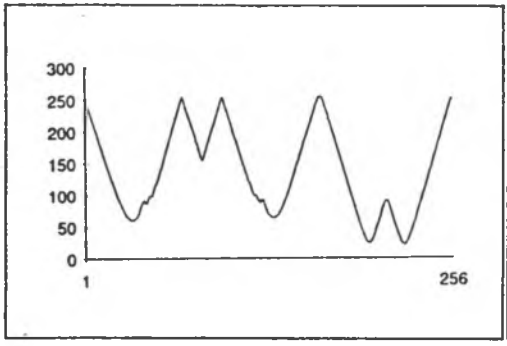
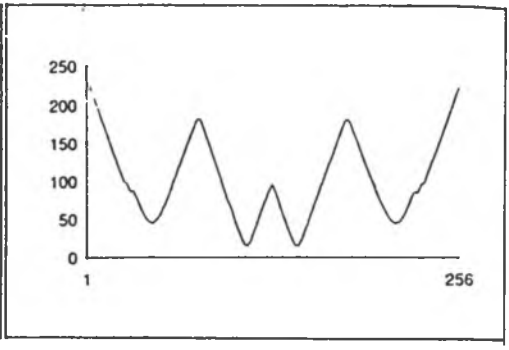
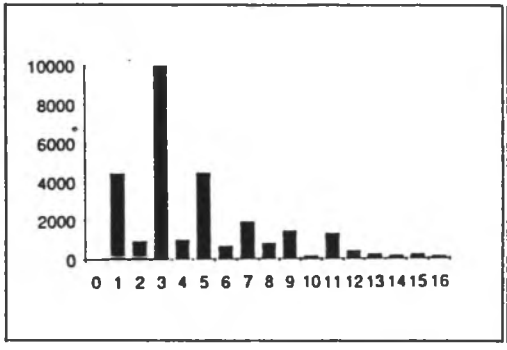


Fig. 4. Polar diagrams and FFT modules of the spanner







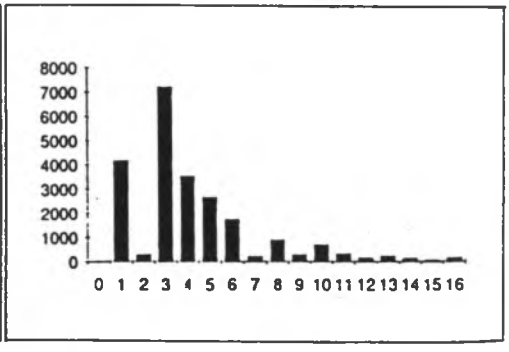


Fig. 5. Polar diagrams and FFT modules of the pliers

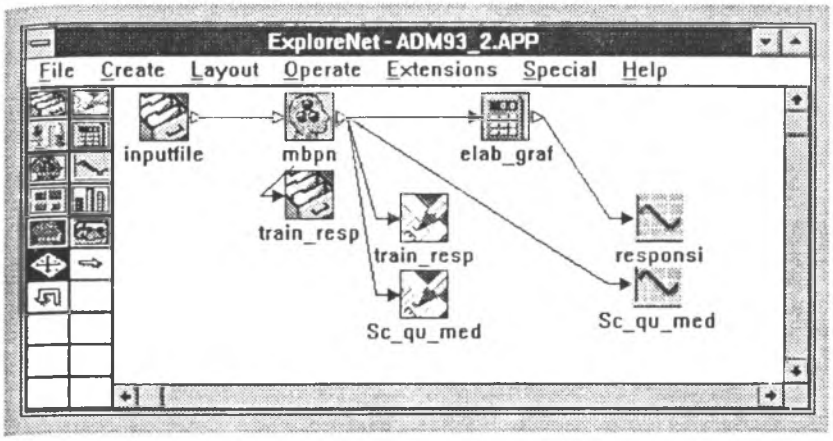


Fig. 6. Data flow to and from the tested M.B.P.N network: the Explorenet screen

The input data are supplied to the network ("mbpn" icon, type "network" [18]) via ASCII file ("input" icon, type "file"), and the responses are stored in another ASCII file ("train_resp" icon, type "file"). The designer can follow the evolution of the network on the display.The evolution of the network is shown in numerical as well in graphic format; the windows display the values entered in input, the desired response and the actual response of the network ("train_resp" icon, type "form" and "responses" icon, type "graph"). In addition there are two windows showing the m.s.e. at each presentation cycle ("Sc_qu_med" icon, type "form" and "Sc_qu_med" icon, type "graph").
The weights were initialized random in the range $-4 < Wij < +4$. The initial learning rate α was set to 0.75. For the activation function S, a sigmoid, with a limited range between 0 and 1, with a value of 0.5 and a slope of 1 at the origin was selected.

1.4.Training of an associative memory

In the first phase of the experimentation the network was trained on all the examples (associative memory).

cicli	0-1000	1000-3000	3000-6000	-----	6000 ⇒
α	0.75	4.0	0.75	-----	0.75
Δw_{ij}	come da eq. (4),(6),(7),(8)			$-0.1 < \Delta w_{rnd} < +0.1$	.....

Table. 1 Training history of an associative memory

The training started from a random point in a quasi flat zone of the error hypersurface. This clearly appeared seeing that the error was decreasing very slowly, in a linear mode ("Sc_qu_med" window). In order to increase the convergence speed the learning rate value α was then set to 4. After about 3000 presentation cycles the m.s.e. started to diminish considerably and the learning rate was again set to its initial value of 0.75. After around 6000 cycles the m.s.e. became stable. As it is impossible to define the nature of the minimum, the

application was paused and the weights were modified with randomly selected values out of a small range around zero. Then the application was continued. After about 10000 cycles the m.s.e. became stable at very low values (around 2%). Random weights changes, even large ones, did not lead to an exit of the attraction basin of this minimum. This was therefore considered to be a global one and the training was stopped. The weights were saved in a file.

1.5. Training of an intelligent network

In the second phase of the experimentation the network was trained so as to acquire a certain ability for generalization, hence to respond correctly even if presented with objects slightly different to those on which it had been trained. Therefore a metodology as described below was used.

1) The set of 18 examples was subdivided into two sets of 9 examples each. The 9 examples of the first set (the "training set") were directly used in the training phase as well as in the evaluating phase, whereas the 9 examples of the second set (the "training test set") were directly used only in the network evaluation phase.

2) The network was trained only on the training set. During the training the m.s.e. was monitored. Every 100 presentation cycles, the training was paused (using the appropriate "training enabling on/off flag [18]) and the network was presented with the examples of the training test set. At the same time the associated m.s.e. was observed. It was found that the errors associated with the training set diminished with the progress of the application, whereas those associated with the training test set increased again after an initial decrease.
This happens when the knowledge of the network starts to become too specialized on the particular examples and therefore loses its generalization capabilities. Hence the training was stopped at the minimum of the m.s.e. curve associated to the training test set.
In the evaluation phase all examples were presented and the relative responses were stored.
Notice that the global error stays under 3% ("Sc_qu_med" window): the network responds to all presentations with remarkable precision.

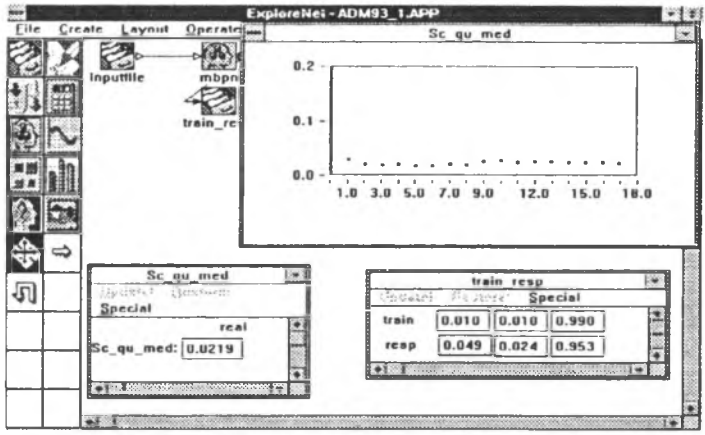


Fig. 7. Final application phase

2. Approach by means of the fuzzy sets theory

2.1. Distance between sets

We recall [5] that one speaks of distance between two elements X and Y of a set I and indicates with d(X,Y) an application of I x I on the set R+ of non-negative real numbers fulfilling the conditions (9), (10) and (11) as shown below:

$$d(X,Y) = d(Y,X) \tag{9}$$

$$\forall\, X,Y,Z \in I\ ,\quad d(X,Z) \leq d(X,Y) + d(Y,Z) \tag{10}$$

$$d(X,Y) = 0 \quad \Leftrightarrow \quad (X=Y) \tag{11}$$

The concept of distance can be extended to sets in various ways. In particular the Hamming distance between two sets is:

$$A = \{x, c_A(x)\} \qquad \forall\, x \in U \tag{12}$$

$$B = \{x, c_B(x)\} \qquad \forall\, x \in U \tag{13}$$

where cA(x) and cB(x) are the membership functions of the sets A and B, and U is the universe under consideration. The distance, as stated, is characterized by the following relations:

$$d(A,B) = \sum_{i=1}^{n} | c_A(x_i) - c_B(x_i) | \tag{14}$$

if U is finite and of power n

$$d(A,B) = \sum_{i=1}^{\infty} | c_A(x_i) - c_B(x_i) | \tag{15}$$

if U is infinite and countable and if the series is convergent

$$d(A,B) = \int_U | c_A(x) - c_B(x) | \tag{16}$$

if U is a continuum provided its integral is finite, or convergent if U is also infinite.

2.2. Application

For the purpose of separating the three classes of objects we first used the distance between the elements of the subsets consisting of the first 16 harmonics of the FFT of the polar diagram of the contour of the three objects in question (fig. 3,4 and 5).
$FFT_{i,samp}$, $FFT_{i,comp}$ shall be the moduls of the i-th component of the FFT for the sample object and the one compared with it.

We also consider the distance d(FFTi,samp , FFTi,comp) and the one obtined by dividing the two elements by FFTi,samp., thereby arriving at:

$$\frac{d(1,FFT_{i\,confr})}{FFT_{i,camp}} = \frac{|FFT_{i\,camp}-FFT_{i\,confr}|}{FFT_{i,camp}} \qquad (17)$$

If FFTi,comp >FFTi,samp, the distance between the two elements is larger than unity. If we now consider the complement of the distance (eq. 17) as the membership function of the compared element to the sample set, the condition d>1 means a negative membership function, which is not compatible with the fuzzy logic rules. We can get rid of this problem by setting c(xi)=0 if $d \geq 1$ writing:

$$d \geq 1 \quad \Rightarrow \quad c(x_i) = 0 \qquad (18)$$

where d id the first member of (18)

The interpretation of (10) seems obvious: the element for which $d \geq 1$ does not belonging to the sample set and therefore c(xi)=0.

With the limits described above we can define the membership function c(x) of the compared element setting the membership function for the sample element to unity. Then the Hamming distance between the sample set and the compared set becomes:

$$d(I_{camp}, I_{confr}) = \sum_{i=1}^{16} [1 - c(x)] \qquad (19)$$

formally equal to (17) but so as to avoid terms with an absolue value greater than unity.

How the sample set assimilates the set of the means of the modules of the FFT for the six images of the calliper, the six of the pliers and the six of the spanner. Here it can be noted in particular that the pliers and the calliper are in closed configuration as well as in open configuration.
Tables 2,3,and 4 show the Hamming distances, the membership function and the fuzziness indices for the three object classes referring to the calliper, pliers and spanner samples respectively. One realizes the differentiating power of the Hamming distance which is especially clear for the spanner. This results from the fact that the pliers and the calliper have been treated in open and closed configuration as pointed out above. In conclusion, the Hamming distance appears as a differentiating element of a certain effectiveness in quasi-flat object classification problems, based on a FFT of the d=d(s) diagram of the object contour.

3. Guidelines for the continuation of the research

We recognize the following guidelines for the continuation of our research:

- improvement of the selection of the sample as well as that of the compared element. This is in order to define the validity limits in the choice of the Hamming distance as a discriminating factor between object classes;

- setting up of a comparison methodology between the performance of a classification system based on the neural approach and an analogous system based on the fuzzy approach;
- evaluation of the use of descriptors based on the two dimensional Fourier Transform of the rough two level images in order to avoid the calculation of the d=d(s) diagram.

Object	Hamming dist. **H**	Memb. function	Fuzz. ind. δ
Calliper1	4.656	0.709	0.291
Calliper2	5.328	0.666	0.333
Calliper3	2.640	0.835	0.165
Calliper4	6.400	0.600	0.400
Calliper5	6.592	0.587	0.412
Calliper6	3.920	0.754	0.245
Spanner1	7.200	0.550	0.450
Spanner2	7.000	0.562	0.437
Spanner3	7.872	0.492	0.507
Spanner4	8.288	0.481	0.518
Spanner5	7.376	0.538	0.461
Spanner6	7.568	0.526	0.473
Pliers1	9.776	0.388	0.611
Pliers2	10.51	0.342	0.657
Pliers3	10.10	0.369	0.631
Pliers4	8.128	0.492	0.508
Pliers5	8.816	0.449	0.551
Pliers6	9.296	0.418	0.581

Table 2. Sample calliper: Hamming distances, membership functions, fuzz. indices

Object	Hamming dist. **H**	Memb. function	Fuzz. ind. δ
Calliper1	8.296	0.481	0.518
Calliper2	8.480	0.470	0.530
Calliper3	7.392	0.538	0.462
Calliper4	7.760	0.515	0.485
Calliper5	8.640	0.460	0.540
Calliper6	8.128	0.492	0.508
Spanner1	2.534	0.842	0.158
Spanner2	2.370	0.851	0.148
Spanner3	3.168	0.802	0.198
Spanner4	3.428	0.786	0.214
Spanner5	1.112	0.930	0.069
Spanner6	1.668	0.896	0.104
Pliers1	11.49	0.282	0.718
Pliers2	11.87	0.257	0.742
Pliers3	11.74	0.266	0.734
Pliers4	9.920	0.380	0.620
Pliers5	10.53	0.342	0.658
Pliers6	10.67	0.338	0.667

Table 3. Spanner sample: Hamming distances, membership functions, fuzz. indices

Object	Hamming dist. H	Memb. function	Fuzz. ind. δ
Calliper1	9.931	0.371	0.620
Calliper2	7.193	0.550	0.450
Calliper3	9.304	0.418	0.581
Calliper4	11.49	0.282	0.717
Calliper5	11.35	0.290	0.709
Calliper6	7.757	0.515	0.484
Spanner1	10.67	0.333	0.666
Spanner2	10.67	0.333	0.666
Spanner3	12.12	0.242	0.757
Spanner4	12.12	0.242	0.757
Spanner5	11.22	0.298	0.701
Spanner6	11.35	0.290	0.709
Pliers1	5.552	0.653	0.347
Pliers2	6.803	0.574	0.425
Pliers3	7.757	0.515	0.484
Pliers4	4.649	0.709	0.290
Pliers5	4.880	0.695	0.305
Pliers6	5.984	0.626	0.374

Table 4. Pliers sample: Hamming distances, membership functions, fuzz. indices

4. REFERENCES

[1] A. Pugh, (editor), "Robot Sensors-Vision". IFS publications Ltd, UK North-Holland, 1986.

[2] D.E. Rumelhart e al., "Parallel didtributed processing: explorations in the microstructure of cognition". voll. 1 & 2 ,MIT press, Cambridge M.A., 1986.

[3] B.G. Batchelor, D.A. Hilland, D.G. Hodgson, "Automated ,Visual Inspection", IFS publications, Ltd, UK North-Holland, 1987.

[4] R.C. Gonzales, P. Wintz, "Digital Image Processing". Addison Wesley, Reading, Mass., 1987.

[5] R.H.Nielsen, "Neurocomputing". Addison Wesley, 1990.

[6] G. Graziano, M. Orlando, G.Podda, "Individuazione di una forma generica mediante algoritmo polare", "Atti del VII Convegno Nazionale A.D.M"., vol. 1, pag. 33 - 45, Trento 1991.

[7] M. Cantamessa, M. Orlando, G. Podda, "Riconoscimento di figure piane con invarianza per traslazione, rotazione e scala mediante reti neurali", "Atti del VII Convegno Nazionale A.D.M", vol. 1, pag. 47 - 59, Trento 1991.

[8] J. Hertz, A. Krogh, R.G.Palmer, "Introduction to the Theory of Neural Computation". Addison Wesley, 1991.

[9] G.Graziano, M. Orlando, G.Podda, "Recognition of concave shapes with internal holes", "Proceedings. of CIM'92 International. Conferance". vol. 1, pag. 133 - 139, Zakopane, Poland, 1992.

[10] T. Redarce, Y. Lucas, "Practical implementation of vision systems in a CIM environment", "Computer-Integrated Manufacturing Systems", vol. 5 n.1, 1992.

[11] G.N. Bebis, G.M. Papadourakis, "Object Recognition Using Invariant Object Boundary Representations and Neural Network Models", "Pattern Recognition", vol 25, n.1, 1992, pag. 25 - 44.

[12] P.J.V. Otterloo, "A Contour Oriented Approach to Shape Analysis", Prentice Hall, 1991.
[13] G. Podda, "Algoritmo di contorno per inseguimento", "Atti del VII Convegno Nazionale A.D.M", vol. 1, pag. 23 - 31, Trento, 1991.
[14] J.C. Bezdek, S.K. Pal, "Fuzzy Models for Pattern Recognition". IEEE Press, New York, 1992.
[15] A. Fadini, "Introduzione alla teoria degli insiemi sfocati". Ed. Liguori, Napoli 1980.
[16] A. Donnarumma, "Uber Einen Entscheidungs Operator: Das C-Calculus", "Proceedings osf the International Conference EVAD", pag. 40 - 49, Praga.
[17] A. Donnarumma, E. Santoro, "Konstruktions Processe: das Model Risiko-Entscheidung", "Proceedings osf the International Conference EVAD", pag. 31 - 39, Praga.
[18] "HNC Explorenet Manual" e "HNC Balboa 860 Manual".
[19] O. Brigham, "The Fast Fourier Transform". Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
[20] F. Persiani, A. Donnarumma, E. Del Giudice, A. Piazzi, "L' impiego di reti neuronali e logiche fuzzy evolutive nella progettazione meccanica", "Proceedings of 1st International Scientific Conference on Achievement in the Mechanical and Material Engineering", vol. 2, pag. 75 - 84, Gliwice, Poland, 1992.
[21] A. Donnarumma, M. Pappalardo, "Sull' impiego di logiche polivalenti per la classificazione automatica di figure piane", " Proceedings of the 5-th "Congreso International de espresion grafica di Diseño Industrial", vol. 1, Principado de Asturias, España, 2-4 de Junio 1993.
[22] A. Donnarumma, "Analisi dei sistemi semplici e complessi", "Lamiera", n. 3/83, pag. 68 - 72.

EIN HYBRIDES SYSTEM ZUR ERKENNUNG QUASI PLANARER OBJEKTE MITTELS FOURIER DESCRIPTOREN, MIT ANWENDUNG DER NEURONALEN NETZE UND DER DISTANZ METHODE.

Inhaltsangabe. Ein hybrides System zur Erkennung quasi planarer Objekte mit Invarianz in Bezug auf die Translation, Rotation und Skala wird dargestellt. Die Objekte können verschiedene Gestaltungen (zum Beispil: teil - oder ganz geöffnete Zangen oder Kaliber) vorstellen.
Das System stammt aus der Anwendung der neuronalen Netze auf konventionelle Verfahren des Image Processing (Randaufnahme, Fourier Descriptoren) ab.
Ausserdem, werden die dargestellten Objekte mittels einer Methode klassifiert, die sich auf der Hammingdistanz zwischen der Menge der Module der ersten i Oberwellen der F.F.T. des Mustersignals und der entsprechenden, auf die zu klassieren Signal bezogenen Menge, gründet.
Die Hammingdistanz wird zur Bestimmung des Zugehörigkeitsgrades des verglichenen Signals der von Muster definierten Menge, benutzt.
Schreitet solcher Grad den vorbestimmten Wert über, so wird der Objekt als der oberbeschriebenen Klasse zugehörend, erkennt.

Revised by: Wojciech Cholewa