Available online at www.sciencedirect.com
SCIENCE DIRECT®

Image and Vision Computing 25 (2007) 240–247

www.elsevier.com/locate/imavis

Distributed recursive learning for shape recognition through multiscale trees

Luca Lombardi [a,*], Alfredo Petrosino [b]

[a] Dipartimento di Informatica e Sistemistica, Universitá di Pavia, Via Ferrata 1, 27100 Pavia, Italy
[b] Dipartimento di Scienze Applicate, Universitá 'Parthenope' di Napoli, Via A. De Gasperi 5, 80131 Napoli, Italy

Received 12 October 2004; received in revised form 16 January 2006; accepted 31 January 2006

Abstract

The paper reports an efficient and fully parallel 2D shape recognition method based on the use of a multiscale tree representation of the shape boundary and recursive learning of trees. Specifically, the shape is represented by means of a tree where each node, corresponding to a boundary segment at some level of resolution, is characterized by a real vector containing curvature, length, symmetry of the boundary segment, while the nodes are connected by arcs when segments at successive levels are spatially related. The recognition procedure is formulated as a training procedure made by a Fuzzy recursive neural network followed by a testing procedure over unknown tree structured patterns. The proposed neural network model is able to facilitate the exchange of information between symbolic and sub-symbolic domains and deal with structured organization of information, that is typically required by symbolic processing.

Keywords: Multiscale tree representation; Syntactic shape recognition; Fuzzy neural networks; Recursive learning

1. Introduction

Syntactic pattern recognition [13,14] represents a possible meaningful step in the designing of an artificial vision system. Pattern classes contain objects, such as geometric figures, with an identifiable hierarchical structure that can be described by a formal grammar. The idea behind is the specification of a set of pattern primitives, a set of rules in the form of a grammar that governs their interconnection and a recognizer (an automaton) whose structure is determined by the set of rules in the grammar. The patterns could be not only structured (usually tree structured), but each pattern primitive could possess a sub-symbolic nature and possibly a fuzziness degree, measuring the inherent vagueness and the imprecise nature of patterns, is attached to it.

For shape recognition a lot of approaches have been reported in literature [42]. On the basis of the fact that coarse-to-fine strategies have been successfully used in a variety of image processing and vision applications (see for instance [37]), including stereo matching, optical flow computation, etc. a pattern may be also represented at various resolution levels by a graph of primitives and their relations. In such a case, production rules describe the evolution of the object primitives at increasing resolution levels.

Multiscale image representations have attracted much attention from researchers lately [15,36,41], due to their intuitive description of structures present in the image, the reduction of complexity provided by the decomposition of a complicate problem in simpler ones, and the improvement in computation time. Most studies include the use of the Gaussian filter as the basis to achieve a multiscale description [7], due to the fact that the Gaussian is the only linear filter that combines isotropy, homogeneity and causality, i.e. no new features appear when the scale increases, when applied to 1D signals, while causality is not necessarily accomplished for 2D signals. Furthermore, the Gaussian filter produces a displacement in the position of main features of the image such as local extrema or zero crossings [4,22]. For these reasons, non-linear filters have been explored as the basis for a multiscale description. Anistropic diffusion [31], Gabor filters [9], B-splines [44] or wavelets [19] have demonstrated their uselfulness in multiscale representation.

To construct a compact and intuitive multiscale image representation, one of the best options is to use a tree structure. These representations are acquiring increasing importance

* Corresponding author.
E-mail addresses: luca.lombardi@unipv.it (L. Lombardi), alfredo.petrosino@uniparthenope.it (A. Petrosino).

0262-8856/$ - see front matter © 2006 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2006.01.022

in recent years [5,24]. Approaches to organise data and associate information with the elements of the structures include pattern tree [6], model-feature graph [8]. Systems have been developed to represent contours, divided in segments that are progressively joined when the scale parameter is increased [17]. Wavelets have also been used to generate a tree representation, as in [2,34], where the wavelet tree is proposed as a way to identify textures in an image by recursively extracting feature values, and segmentation is then carried out using a multichannel classification algorithm. The structures suffer from the disadvantage that mismatches at a coarse scale cause errors from which it is impossible to recover since the algorithms usually proceed by sub-dividing the corresponding coarse elements into sub-elements. We follow the approach firstly introduced in [3] that tries to overcome these disadvantages. The shape contours, extracted by a heat-diffusion process, are represented at all scales by a sequence of concave/convex segments identified by the inflection points along the curves.

Starting from these premises, the objective of the present work is to develop a general recognition system in which the a priori knowledge for any particular case could be introduced via a training process. Typically, to introduce a priori knowledge in the process, recognition is carried out by a correspondence process between the calculated tree for a specific image and a model which incorporates all the information available about the image. This model is also tree shaped and obtained by a training process. The model is typically rigid and the correspondence process may lack of flexibility, adaptibility to the environment changes.

We propose a hybrid model for syntactic object recognition based on the use of recursive learning, as introduced in [12], capable to process streams of structured data by neural networks, where the temporal processing which takes place in recurrent neural networks is extended to the case of graphs by connectionist models. The model uses symbolic grammars to build and represent syntactic structures and neural networks to rank these structures on the basis of the experience. Recursive neural networks are initialised with prior knowledge about training structured data; this ability makes recursive neural networks useful tools for modeling tree automata [16,27], where prior knowledge is available.

Specifically, we propose a *fuzzy recursive neural network* to deal with the recursive nature of data and the uncertainty of their description. Several researchers have considered the possibility of integrating the advantages of fuzzy systems with those not less known of neural networks, giving rise to *fuzzy neural networks* (see for instance [18]). These kinds of studies are very interesting in all the application domains where the patterns are strongly correlated through structure and the processing is both numerical and symbolic, without discarding the component of structure that relates different portions of numerical data and the imprecise and incomplete nature of the data. We demonstrate, by testing the model on an airplane shape data set, whose reference shapes are depicted in Fig. 1, the effectiveness of the model, the completely parallel nature of it, and the particularly encouraging performance.

The paper is organized as follows. Section 2 describes the adopted multiscale tree representation along with its advantages with respect to other shape representation. The adopted fuzzy recursive neural network model is reported and discussed in Section 3, where the reader is also introduced to the new learning paradigm of recursive learning through data structure. In Section 4 two other, more classical, approaches to deal with shape representation and recognition are reported for comparison. Tests on an airplane data set are reported in Section 5, while Section 6 describes some hints to be addressed for further investigation about the reported shape recognition system.

2. Multiscale tree representation

We are interested in describing the patterns with a representation which takes into account both their structure and the sub-symbolic information. To derive tree representations of planar shapes from images, a full Gaussian pyramid of images, taken at various levels of resolution, is first constructed. After the application of an edge detector and a contour following procedure to all resolution levels, each object boundary present in the scene is decomposed into a sequence of feature primitives, i.e. curve segments. The boundary decomposition procedure, detailed in [3], is based on the analogy with a heat-diffusion process acting on a physical object with the same shape as the given digital object. By assigning a non-zero value to all contour pixels of the digital object, an isotropic diffusion process propagating from each element to its neighbours towards the interior of the object is performed. In formula:

$$I_{t+1}(p) = I_t(p) + D\left(\sum_{q\in N(p)} (I_t(q) - I_t(p))\right) \tag{1}$$

where $I_t(p)$ represents the value of the pixel p at time t and D is the diffusion coefficient that describes the sharing factor of the local value of each pixel content among all its neighbours $N(p)$, in the number of 9. After a number of steps the contour elements that preserve high values correspond to local

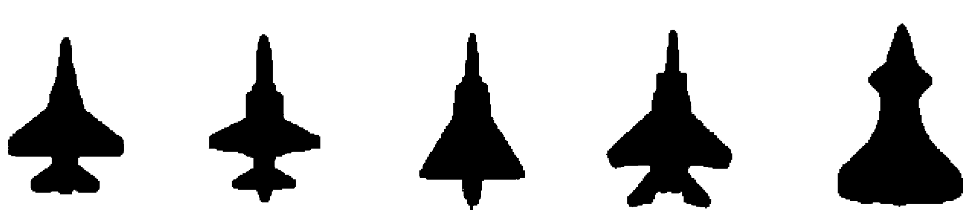

Fig. 1. Airplane reference shapes.

convexities and those in which a sharp decrement is produced correspond to local concavities.

These boundary descriptions at different scales induce a tree structure. Each node in the tree corresponds to a segment (concave, convex, etc.), connecting segments at consecutive levels that are spatially related. The children of a node correspond to consecutive segments at the next level of resolution that can be seen as one global segment at that level, giving a more detailed description of the same portion of the boundary as the parent node. The simblings of a node correspond to a given orientation of the curve boundary, while the leaves of the tree correspond to the segments at the finest level of resolution. Curvature values corresponding to the labels like ⟨very_concave⟩ (values up to 1210), ⟨concave⟩ (values between 1210 and 1430), ⟨straight⟩ (values between 1430 and 1510), ⟨convex⟩ (values from 1510 up to 2200) and ⟨very_convex⟩ (values from 2200) are associated to the segments along with corresponding attributes, like the segment length and a measure of the symmetry, providing a quantitative description in terms of geometric features as well as a qualitative description of the spatial arrangement of the segments. In Fig. 2(d) the constructed tree representation of Fig. 2(c) is given.

Specifically, each node is characterized by a 3D real feature vector, coming from the curvature segment it represents. Firstly, the temperature values obtained on the border of the shape after a given number of iterations are measured at each pixel; assuming the object is thermally insulated from the background, a set of thresholds is chosen so as to associate values exceeding these thresholds with a shape-related code words. The second and third attributes are computed as follows, letting $f(l)$ be the curvature function

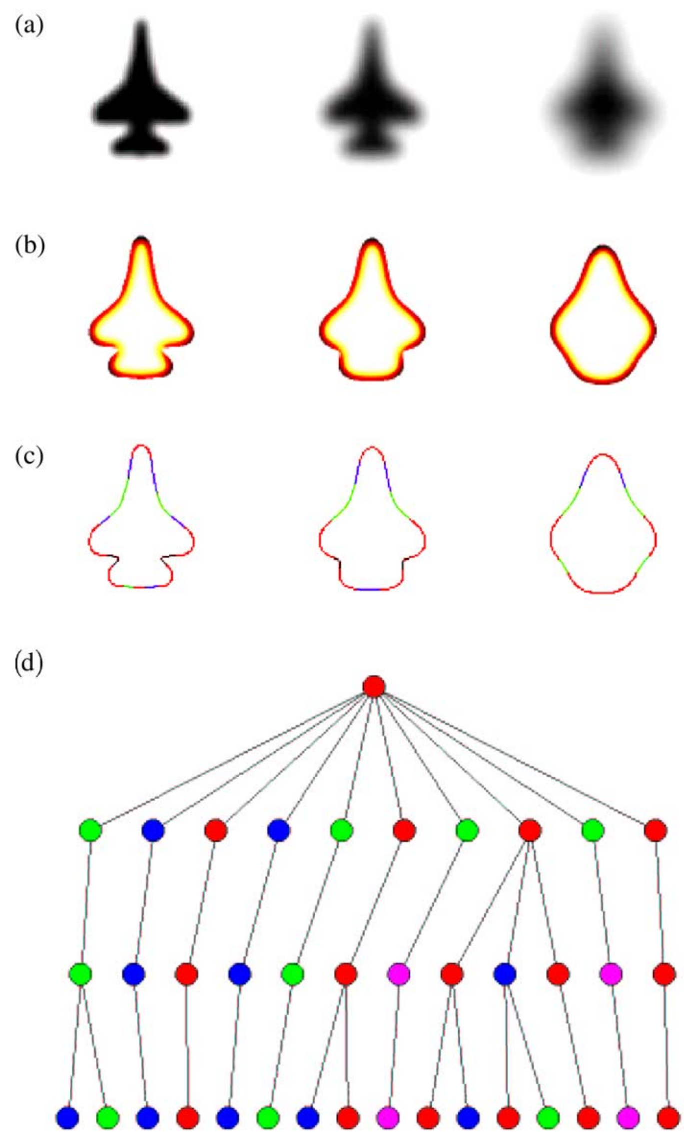

Fig. 2. An airplane shape (a) at three levels of the Gaussian pyramid, (b) after the diffusion process, (c) coloured to enhance extracted contours features and (d) the corresponding multiscale tree representation.

along a segment c:

$$L_c = \int dl \quad (2)$$

$$S_c = \int_0^{L_c} \left(\int_0^s f(l)dl - \frac{1}{2} \int_0^{L_c} f(l)dl \right) ds \quad (3)$$

From the above formulae, L_c gives the total length of the segment, while S_c represents the degree of symmetry. If $S_c = 0$ the segment is intended to be symmetric, while if S_c gets positive or negative value, the segment is intended to be inclined to the left or to the right, respectively. These attributes, normalized, are used in the learning procedure to measure the similarity or dissimilarity of a pairing between segments of two different given shapes.

3. Recursive neural networks

To process data represented by multiscale trees, a computational model based on neural units is adopted [12]. In particular, the model realizes mappings from directed ordered acyclic graphs (DOAGs) (in our case ordered trees) to n-dimensional vectors using recursive neural networks. Recursive neural networks are characterized not to possess explicit feedback connections; the recurrent processing is then driven by the inherent recursive structure of the patterns in the training domain. Consider that instances in the training domain are structured patterns of information described by *ordered* r-ary trees. Here, by an *ordered* r-ary tree we mean an r-ary tree where for each vertex a *total order* on the edges leaving from it is defined. A total order can be induced on the tree nodes by topologically sorting the nodes with a linear order $<$, such that $v < w$ if there exists an edge (v,w). Informally speaking, the encoding of a given tree into a distributed network is achieved by recursively setting the previously computed representation for the direct subtrees together with the representation of the label attached to the root. By starting this process from the leaves to the tree root, a representation of the whole tree is achieved and can be mapped to a particular user-specified output (structured or not).

To define the dynamics of the network, the *generalized shift operator* is adopted [12]. For $k = 1,\ldots,r$, the generalized shift operator is denoted by q_k^{-1} and is associated to the kth child of a given vertex, such that when applied to a node x_v returns the variable attached to the kth child of that node, with x_v denoting the value of a tree vertex labelled v.

Denoting uniformly labelled r-ary trees by the boldface uppercase letters corresponding to the label space of the tree, i.e. $\mathbf{Y}$ denotes a tree with labels in Y, labels are accessed by vertex subscripts, i.e. $\mathbf{Y}_v$ denotes the label attached to vertex v. Given a tree structure $\mathbf{Y}$, the tree obtained by ignoring all node labels will be referred to as the *skeleton* of $\mathbf{Y}$, denoted by skel $(\mathbf{Y})$. Two structures $\mathbf{Y}$ and $\mathbf{Z}$ can be distinguished because they have different skeletons, i.e. $\mathrm{skel}(\mathbf{Y}) \neq \mathrm{skel}(\mathbf{Z})$, or they have the same skeleton but different vertex labels. The class of trees defined over the local universe domain Y and skeleton in $\#^{(1,o)}$, i.e. the set of graphs whose vertices have in-degree 1 and maximum out-degree o, will be denoted as $Y^{\#(1,o)}$. In the following, the training set comprises couples $(\mathbf{U}, \mathbf{Y})$, where $\mathbf{U} \in U^{\#(1,o)}$ are input trees with symbols s_k in Σ, each converted into a vector $\mathbf{U}_k$ in R^n, attached to nodes; the same is true for $\mathbf{Y} = \tau(\mathbf{U}) \in Y^{\#(1,o)}$ which represent output trees.

Since $\tau(\cdot)$ is a binary transduction between input and output trees, $\tau \subseteq U^{\#(1,o)} \times Y^{\#(1,o)}$ the aim of any supervised learning algorithm is to estimate the transduction $\tau(\cdot)$. In particular, we consider transductions from $\mathbf{U} \in U^{\#(1,o)}$ to $\mathbf{Y} \in Y^{\#(1,o)}$, which are: (a) *IO-isomorph*, i.e. $skel(\tau(\mathbf{U})) = skel(\mathbf{U})$; (b) *causal*, i.e. $\tau(\mathbf{U})_v$ only depends on the sub-tree of $\mathbf{U}$ induced by v and its descendants, $\forall v$.

Such an IO-isomorph transduction $\tau(\cdot)$ admits a *recursive state representation*:

$$\mathbf{X}_v = f(\mathbf{X}_{\mathrm{ch}[v]}, \mathbf{U}_v) \quad (4)$$

$$\mathbf{Y}_v = g(\mathbf{X}_v, \mathbf{U}_v) \quad (5)$$

where $\mathbf{X}_{\mathrm{ch}}[v]$ is a fixed size array of labels attached to the (ordered) children of v and

$$f: X^r \times \mathcal{U} \rightarrow X \quad (6)$$

$$g: X \times \mathcal{U} \rightarrow Y \quad (7)$$

According to the total order induced on the tree nodes, the states are updated following a recursive message passing scheme such that a state label $\mathbf{X}_v$ is updated after the state labels corresponding to the children of v in the order defined, as instance, by any reversed topological sort of the tree nodes.

Here, we adopt a Fuzzy recursive neural network defined on the basis of radial basis functions (RBF). Recurrent radial basis functions, firstly introduced in [11] and adopted in [20] to model fuzzy dynamical systems, can be extended to process structures by the following parametric representation:

$$X_{i,v}^b = e^{-\alpha_{i,v}/\sigma_{i,v}^2} \quad (8)$$

$$\alpha_{i,v} = \sum_{k=1}^{r} ||q_k^{-1}\mathbf{X}_v - C_i^k||^2 + ||\mathbf{U}_v - \bar{C}_i||^2 \quad (9)$$

where $X_{i,v}^b$ denotes the output of the ith radial basis function unit, $i = 1,\ldots,p$. The weight matrices $C_i^k \in R^m$ and $\tilde{C}_i \in R^n$, $i = 1,\ldots,p$, are the same for every node v, i.e. the transduction $\tau(\cdot)$ is stationary. The fuzzy state vector $\mathbf{X}_v$ is obtained using an additional layer of units, with weight matrix $W \in R^{m \times p}$, which realizes a normalized sum of the p radial basis functions. The addition of a third layer of just one output neuron allows the computation of the fuzzy membership of any given string.

The recursive state neurons are the same as indicated in the Eq. (9), where here the fuzzy state transistion is modelled by radial basis functions. These functions are very well-suited for implementing Fuzzy frontier-to-root tree automata (FFRTA) state transitions. This means that the model we are reporting is able to process symbolic data in a sub-symbolic manner with the same computational power of an automaton demanded to make this. By considering that each state transition automaton is given by

$$X_{i,v}^{b} = \prod_{k=1}^{r} e^{-\|q_k^{-1}\mathbf{X}_v - C_i^k\|^2/\sigma_{i,v}^2} e^{-\|\mathbf{U}_v - \tilde{C}_i\|^2/\sigma_{i,v}^2} \tag{10}$$

and due to the property that a multi-dimensional radial basis functions can be decomposed in 1D radial basis functions the Eq. (10) can be rewritten as:

$$X_{i,v}^{b} = \prod_{k=1}^{r}\prod_{j=1}^{m} e^{-(q_k^{-1}X_{j,v} - C_{i,j}^k)^2/\sigma_{i,v}^2} \prod_{l=1}^{n} e^{-(U_{l,v} - \tilde{C}_{i,l})^2/\sigma_{i,v}^2} \tag{11}$$

where m stands for the number of states and n corresponds to the number of inputs. The encoding of an FFRTA looks as follows.

Here, the product

$$\prod_{k=1}^{r}\prod_{j=1}^{m} e^{-(q_k^{-1}X_{j,v} - C_{i,j}^k)^2/\sigma_{i,v}^2} \prod_{l=1}^{n} e^{-(U_{l,v} - \tilde{C}_{i,l})^2/\sigma_{i,v}^2} \tag{12}$$

in Eq. (11) for each radial basis function i provides the fuzzy degree to reach the state S_i when the fuzzy membership functions are modelled as radial basis functions and the fuzzy AND operation used in Eq. (11) i is the algebraic product in fuzzy theory. The adoption of this operation is for computational convenience, either due to its property to transform 1D radial basis functions into a multi-dimensional one and to simplify the learning rules for such kinds of networks. As stated before the radial basis functions are combined by a fuzzy OR operation which is selected to be the bounded sum operation in fuzzy theory. Since this summation would give values larger than one, which are not fuzzy values, a succeeding normalization is needed to keep the values in the range [0,1].

This proposed network construction algorithm can implement any FFRTA with m states and n input symbols using a three-layered network with neurons at the input layer which behave as fuzzy singleton fuzzifiers, p radial basis units which act as fuzzy logical Λ on the first layer, p being the number of explicitly specified transitions, m units which act as fuzzy logical Λ on the second layer and n_w $(r+1)$-order weights with $n_w \leq 3\ nmr$ [16,33]. Fig. 3 is a pictorial representation of the computation involved in a fuzzy recursive neural network with radial basis functions and the output neuron which computes the maximum. Note that if the Eq. (12) is decomposed again, the product

$$\prod_{l=1}^{n} e^{-(U_{l,v} - \tilde{C}_{i,l})^2/\sigma_{i,v}^2} \tag{13}$$

directly leads to the insertion of a new neuron layer (just above the input layer) which is demanded to the computation of the Gaussian membership functions corresponding to the classical linguistic variables *small*, *medium*, *high*, etc. usually characterizing the fuzzy degree of input symbols, while the third layer is requested to combine (according to the arcs in the fuzzy automation) these fuzzy degrees together with those attached to states.

The supervised learning problem in recursive neural networks is solved in the usual framework of error minimization, by searching the parameters. by gradient descent techniques. The gradients can be efficiently computed using the BackPropagation Through Structure (BPTS) algorithm, an extension of Backpropagation through time that unrolls the recursive network in a larger feedforward network, following the ordered tree structure [39].

4. Comparisons: flattened representation and tree matching

With the aim to make consistent comparisons with other well assested techniques reported in literature, in the following

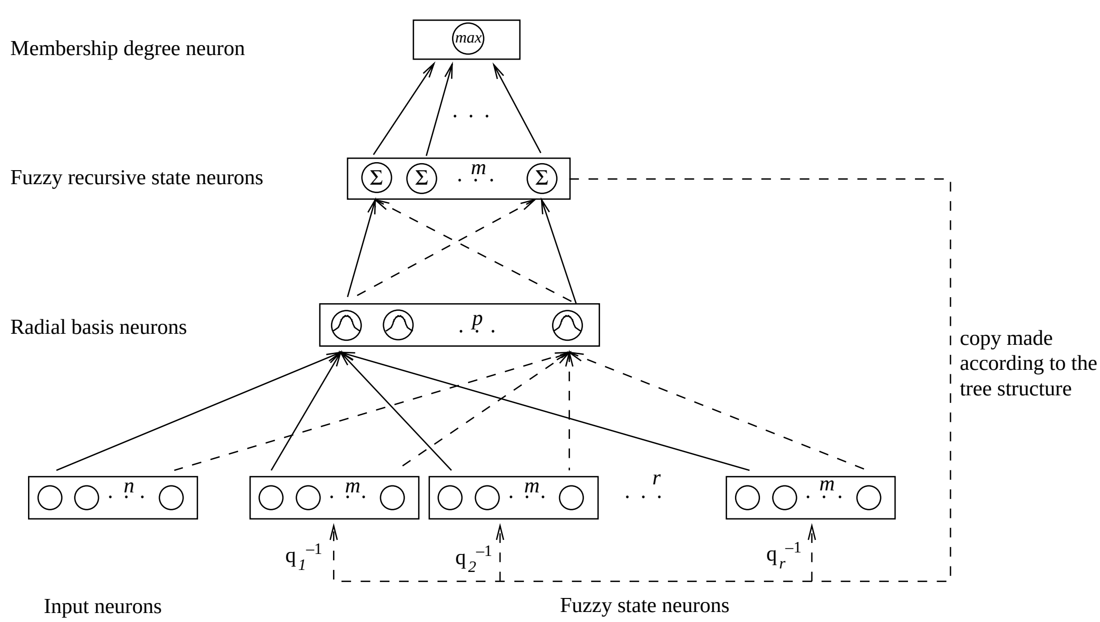

Fig. 3. The fuzzy recursive network with RBF neurons.

two different strategies are described and thus adopted: a flattened shape representation learned by standard artificial neural networks like Multilayer Perceptrons and, based over the tree shape representation described above, a tree matching algorithm that realizes a non-exact match between the tree representative of a class and that related to the shape at hand.

4.1. 1 Flattened representation

Previous to the calculation of the flattened representation, the object boundary is firstly detected, by applying a linear smoothing filtering and a strongly noise independent segmentation procedure, based on the image entropy optimization between foreground and background [25]. The boundary is then coded by using elementary but salient features, like the variational angle sequence [30] or, as typical, as Euclidean distances of selected boundary points from the centroid (see Fig. 4).

The sequence of shape features is modelled as a Circular Auto-Regressive (CAR) process [26], which is a parametric equation that expresses each sample of an ordered set of data samples as a linear combination of a specific number of previous samples plus an error term. Since the sequence is assumed to be circular, then it is invariant to rotation and translation. The form of the model is:

$$y_i = \alpha_0 + \sum_{k=1}^{m} \alpha_k y_i - k \qquad \forall\, i = 0,\ldots,n-1 \tag{14}$$

where y_i is the current primary feature; y_{i-k} is the feature detected k times before the current features; $\alpha_0,\ldots,\alpha_m$ are the unknown CAR coefficients; m is the model order. Let us indicate with $\underset{\sim}{\beta} = [\alpha_0,\ldots,\alpha_m]$ the least square error (LSE) estimate of the CAR model. To improve the representativity of this solution the following method is adopted. If n is fixed, the number of line segments in the shape polygonal approximation, then $p = \lfloor (B/n) \rfloor$ pixels will lie on the contour between two end points and $(p-1)$ polygonal approximations of the shape through model (4) are possible, depending from the starting point. The sequences of primary features generated for each of them may be slightly different. To overcome the problem we adopt an iterative scheme consisting in solving $p-1$ systems each having n equations and $m+1$ unknowns. It is based on the consideration that the $p-1$ sequences are obtained in some order (clockwise or counterclockwise); firstly, the CAR vector for the first sequence of primary features is computed, then for two sequences, three sequences and so on. The process is repeated $p-1$ times, according to the number of sequences or polygonal approximations. In particular, denoted with $\underset{-}{\beta}_j$ the solution of the jth system thus constructed, $j = 1,\ldots,p-1$, let us define

$$\varepsilon_j = \frac{\overline{\beta}^j_T \overline{\beta}_{j-1}}{||\overline{\beta}_j||^2 ||\overline{\beta}_{j-1}||^2}\; j = 2,\ldots,p-1 \qquad \text{and}$$
$$\varepsilon_1 = \frac{1}{||\overline{\beta}_1||^2} \tag{15}$$

a measure of similarity. The final feature vector $\overline{\beta}^*$ is the solution of the sth system, where $s = argmin_{0\le j\le p-1}\{\varepsilon_j\}$. We shall refer to this way of proceeding as the multi-polygonal auto-regressive model (MPARM).

The sequences of CAR parameters are lastly fed into a multilayer perceptron (MLP) trained to classify the shapes with the conjugate-gradient (CG) algorithm [35]. The process is repeated for the overall set of reference images. After learning, an unknown shape passing through all the previous stages is classified as correct or not from the MLP with frozen weights.

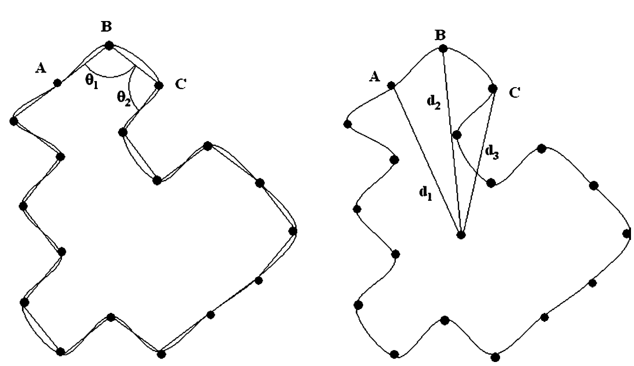

Fig. 4. The adopted vector representation.

4.2. 2 Tree matching algorithm

For comparison, we have applied a tree matching algorithm, where the model of the sample tree representing the reference shape is rigid. Many authors have considered the problem of tree matching [10,23,28,32,38,43]. Tree matching is realized by a series of simple operations (deletion, insertion and edition) applied to individual nodes; a cost value is assigned to each operation so that the sequence that converts one tree into the other with minimal cost can be calculated.

We consider the algorithm firstly described in [40] by Ueda and Suzuki and adopted in [1], where it is also proposed an efficient dynamic programming matching algorithm for 2D object recognition.

Our case includes attributed ordered trees, i.e. labelled ordered trees T_i where attributes, like concavity/convexity, length, resolution level, are associated to all nodes. In this case, the aim is to determine the best matching between the nodes of two trees, here denoted as T_1 and T_2 corresponding to the class representative tree and a sample tree. We fix the following constraints:

1. a node belonging to T_1 may map to a node of T_2 at any level;
2. the mapping has to preserve the left-to-right order of the nodes;
3. for any leaf, exactly one of its ancestors (or the leaf itself) is in the mapping.

It follows that the number of matched nodes is less or equal than the minimum number of leaves between T_1 and T_2. Moreover, if a node is in the mapping, none of its ancestors nor descendants is. This means that for any segment of a shape boundary at the fine levels of resolution there must be one and only one mapped segment at some resolution level that covers it.

Specifically, the problem is reduced to find the set $M_i = \{i_h, j_h\}$ that satisfies the constraints and minimizes the total cost function

$$\mathrm{Cost}(T_1,T_2) = \min_M \sum_h d(i_h,j_h) \quad (16)$$

where d is a function that weights all the attributes of each node.

5. Experimental results

A set of experiments has been designed to settle the evaluation of the multiscale tree based object recognition algorithm. The task faced has been the recognition of five different shaped airplanes (Fig. 1). The data set was formed by changing each airplane in size, orientation and position. Specifically, the objects were 5° rotated and translated in random positions within the image boundaries. The size of each airplane was varied from 0.25 to 1.25 times the size of the original. By doing so, 5040 shapes were generated. The data were consequently subdivided in 3600 shapes for training and 1440 shapes for testing.

In all the experiments, the application of the boundary description procedure sets the diffusion coefficient D to 0.0625 and the number of steps N to 80 after extensive experiments as reported in Ref. [3]. For each image, once the boundary decomposition is obtained, the tree structure representing the multi-resolution profiles shown in Fig. 2(c) is constructed as may be seen in Fig. 2(d).

Table 1 summarizes the results obtained using recursive neural networks on 8-ary multiscale ordered trees. Fuzzy recursive nets, as designed by us and described in previous sections, trained with BPTS had 25, 20, 15, 10 and 5 state units as shown in the table, while in all the experiments the learning rate was initially fixed to 0.02 and the rejection threshold to 0.5. The table shows the accuracies percentage for the testing set computed as the percentage of the correct classification number over the total number of patterns and the rejection rate % as the percentage of the number of rejected patterns over the total number of patterns for learning set. Although different numbers of state neurons were adopted, the recursive neural networks appear to have similar performances, which have been better than those achieved by using other methods on the same data set, like tree matching by dynamic programming (see [1]) and those achieved by adopting shape recognition methods based on flattened representation (see Table 2). The degraded performance is mainly due to the strong similarity of the airplane shapes a_1 and a_2, denoting that tree matching algorithms fail with respect to these typical problems. Less degradation in performance is visible by adopting classical MultiLayer Perceptrons (MLP) over flattened shape representations as those described in Section 4.1.

Table 1
Experimental results in terms of recognition accuracy and rejection

Network	Estimated accuracy (%)	Rejection rate (%)
25 State neurons	98.78	0.10
20 State neurons	98.78	0.11
15 State neurons	98.37	0.13
10 State neurons	97.16	0.15
5 State neurons	96.64	0.25

Table 2
Experimental results in terms of recognition accuracy and rejection

Model	Estimated accuracy (%)
FRNN	98.78
Tree matching	92.12
MLPI	97.84
MLPII	97.14

FRNN, fuzzy recursive neural network; MLPI, MLP applied to CAR parameters extracted from the angular variation sequences; MLPII, MLP applied to CAR parameters extracted from the Euclidean distance.

6. Conclusions

This paper reported an algorithm for the recognition of 2D objects jointly using a multiscale tree representation and recursive learning of the tree representation. The boundary of each object has been described in terms of segments which correspond to nodes on a tree and, for each resolution level, a set of nodes fully describes the image at that level. The tree representation is thus further processed by Recursive Neural Networks in order to obtain a fixed size vector, which can be used to define topological relations between the curvature segments describing the shape boundaries of the objects. Already the preliminary results, achieved on the data set of car silhouettes [21], produced good retrieval of exact classes, validating the promising properties of the proposed scheme. In the present paper, we reported the design a Fuzzy Recursive Neural Network, based on radial basis functions, and investigated the application of the proposed technique (multiscale tree learning) to a more complex task, the airplane shape recognition, also in cases of the presence of occluded objects. We confide that occlusion should be better dealt with recursive neural networks than tree matching algorithms, due to the neural network tolerance to noise and invariance, together with a multiscale tree representation; both characteristics should assure that the general tree structure could be learned also in the case the occlusion partially hides salient features. Further investigations could be:

- improve learning efficiency by iterative cascade correlation learning;
- extension to deal with occlusions and deformable objects;
- understandability of the learned prototypes.

The applications under study comprise *queries-by-example* in image databases and the integration of the shape recognition module in visual attention systems, like road sign recognition, where colour information is not sufficient, and a structure analysis, made in sub-symbolic and parallel manner, is required.

Acknowledgements

We thank Marco Maggini from University of Siena for having provided BPTS software. We also thank the anonymous referees for the comments and remarks received.

References

[1] V. Cantoni, L. Cinque, C. Guerra, S. Levialdi, L. Lombardi, 2-D object recognition by multiscale tree matching, Pattern Recognition 31 (10) (1998) 1443–1454.
[2] T. Chang, C.-C.J. Kuo, Texture analysis and classification with tree-structured wavelet transform, IEEE Transactions on Image Processing 2 (4) (1993) 429–441.
[3] L. Cinque, L. Lombardi, A. Rosenfeld, Evaluating digital angles by a parallel diffusion process, Pattern Recognition Letters 16 (1995) 1097–1104.
[4] J.J. Clark, Authenticating edges produced by zero-crossing algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (1) (1989) 43–57.
[5] C. Di Ruberto, Recognition of shapes by attributed skeletal graphs, Pattern Recognition 37 (1) (2004) 21–31.
[6] P.S. Burt, Attention mechanism for vision in a dynamic world, Proceedings of the 11th International Conference on Pattern Recognition (1988) 977–987.
[7] P. Burt, E. Adelson, The laplacian pyramid as a compact image code, IEEE Transactions on Communications 31 (4) (1983) 532–540.
[8] C.R. Dyer, Multiscale image understanding, in: L. Uhr (Ed.), Parallel Computer Vision, Academic Press, New York, 1987, pp. 171–213.
[9] J. Fdez-Valdivia, J.A. Garcia, J. Martinez-Baena, X.R. Fdez-Vidal, The selection of natural scales in 2D images using adaptive gabor filtering, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (5) (1998) 458–469.
[10] A.J. Fitch, A. Kadyrov, W.J. Christmas, J. Kittler, Fast exhaustive combinatorial versus continuous optimization, Proceedings of 16th International Conference Pattern Recognition 3 (2002) 903–906.
[11] P. Frasconi, M. Gori, G. Soda, Representation of finite state automata in recurrent radial basis function networks, Machine Learning 23 (1996) 5–32.
[12] P. Frasconi, M. Gori, A. Sperduti, A general framework for adaptive processing of data structures, IEEE Transactions on Neural Networks 9 (5) (1998) 768–786.
[13] K.S. Fu, Syntactic Methods in Pattern Recognition, Academic Press, New York, 1974.
[14] F.S. Fu, B.K. Bargava, Tree systems for syntactic pattern recognition, IEEE Transactions on Computer C-22 (12) (1973) 1087–1099.
[15] J.M. Gauch, Image segmentation and analysis via multiscale gradient watershed hierarchies, IEEE Transactions on Image Processing 8 (1) (1999) 69–79.
[16] M. Gori, A. Petrosino, Encoding nondeterministic fuzzy tree automata into recursive neural networks, IEEE Transactions on Neural Networks 15 (6) (2004) 1435–1449.
[17] C. Guerra, 2-D object recognition on a reconfigurable mesh, Pattern Recognition 31 (1) (1998) 83–88.
[18] J.-S.R. Jang, C.-T. Sun, E. Mizutani, Neuro-fuzzy and Soft Computing, Prentice Hall PTR, Upper Saddle River NJ, 1997.
[19] J.-S. Lee, Y.-N. Sun, C.-H. Chen, Multiscale corner detection by using wavelet transform, IEEE Transactions on Image Processing 4 (1) (1995) 100–104.
[20] C.-F. Juang, C.-T. Lin, A recurrent self-organizing neural fuzzy inference network, IEEE Transactions on Neural Networks 10 (4) (1999).
[21] L. Lombardi, A. Petrosino, Object recognition by recursive learning of multiscale trees, in: V. Di Gesú, F. Masulli, A. Petrosino (Eds.), Lecture Notes in Computer Science, 2955, Springer, Berlin, 2005, pp. 255–262.
[22] Y. Lu, R.C. Jain, Behaviour of edges in scale space, IEEE Transactions on Pattern Analysis and Machine Intelligence 11 (4) (1989) 337–356.
[23] B.T. Messmer, H. Bunke, Efficient subgraph isomorphism detection: a decomposition approach, IEEE Transactions Knowledge Data Engineering 12 (2) (2000) 307–323.
[24] F. Mokhtarian, Silhouette-based isolated object recognition through curvature scale space, IEEE Transactions on Pattern Analysis and Machine Intelligence 17 (5) (1995) 539–544.
[25] J.N. Kapur, P.K. Sahoo, A new method for grey level image thresholding using the entropy of the histogram, Computer Vision, Graphics and Image Processing 29 (1985).
[26] R. Kashyap, R. Chellappa, Stochastic models for closed boundary analysis: representation and reconstraction, IEEE Transactions on Information Theory 27 (5) (1981) 109–119.
[27] C.W. Omlin, C.L. Giles, Constructing deterministic finite-state automata in recurrent neural networks, Journal of the ACM, 43(6) (1996) 937–972.
[28] B.J. Oomen, K. Zhang, W. Lee, Numerical similarity and dissimilarity measures between two trees, IEEE Transactions on Computer 45 (12) (1996) 1426–1434.
[30] N.R. Pal, P. Pal, A.K. Basu, A new shape representation scheme and its application to shape discrimination using a neural network, Pattern Recognition 26 (4) (1993) 543–551.
[31] P. Perona, J. Malik, Scale-space and edge detection using anisotropic diffusion, IEEE Transactions on Pattern Analsis and Machine Intelligence 12 (7) (1990) 629–639.
[32] M. Pelillo, K. Siddiki, S.W. Zucker, Matching hierarchical structures using association graphes, IEEE Transactions Pattern Analysis Machine Intelligence 21 (11) (1999) 119–1105.
[33] A. Petrosino, A systematic approach to the representation of fuzzy symbolic knowledge in recursive neural networks, Technical Report DSA—University 'Parthenope' of Naples, 2005.
[34] O. Pichler, A. Teuner, B.J. Hosticka, A comparison of texture feature extraction using adaptive Gabor filtering, pyramidal and tree structured wavelet transforms, Pattern Recognition 29 (5) (1996) 733–742.
[35] M.J.D. Powell, Restart procedures for the conjugate gradient method, Mathematical Programming 12 (1977) 241–254.
[36] B.K. Ray, K.S. Ray, Scale-space analysis and corner detection on digital curves using a discrete scale-space kernel, Pattern Recognition, 30(9) 1463–1474.
[37] A. Rosenfeld Multiresolution Image Processing and Analysis 1984. IEEE Transactions. Pattern Analysis Machine Intelligence, 15(4) (1993) 337–352. Journal of ACM, 43(2) (1996) 937–972.
[38] M. Safar, C. Shahabi, X. Sun, Image retrieval by shape: a comparative study, Proceedings of IEEE International Conference on Multimedia and Expo 1 (2000) 141–144.
[39] A. Sperduti, A. Starita, Supervised neural networks for the classification of structures, IEEE Transactions on Neural Networks 8 (3) (1997) 714–735.
[40] N. Ueda, S. Suzuki, Learning visual models from shape contours using multi-scale convex/concave structure matching, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(4) (1993) 337–352.
[41] K.L. Vincken, A.S.E. Koster, M.A. Viergever, Probabilistic multiscale image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (2) (1997) 109–120.
[42] D. Zhang, G. Lu, Review of shape representation and description techniques, Pattern Recognition 37 (1) (2004) 1–19.
[43] J.T.-L. Wang, K. Zhang, K. Jeong, S. Shasha, A system for approximate tree matching, IEEE Transactions on Knowledge Data Engineering 6 (4) (1994) 550–571.
[44] Y.-P. Wang, S.L. Lee, Scale-space derived from B-splines, IEEE Transactions on Pattern Analysis and Machine Intelligence 20 (10) (1998) 1040–1055.