MATCHING OF STRUCTURAL SHAPE DESCRIPTIONS WITH HOPFIELD NET

JAYANTA BASAK, SANTANU CHAUDHURY*, S. K. PAL, and D. DUTTA MAJUMDER

Electronics and Communication Sciences Unit
Indian Statistical Institute, Calcutta 700 035 India

** Department of Electrical Engineering*
Indian Institute of Technology, Delhi
Hauz Khas, New Delhi 110016, India

Structural description of objects comprised descriptions of the parts and spatial relations between the parts. This paper presents a Hopfield net based scheme for matching structural shape descriptions. The current formulation of the matching scheme is general enough to take care of partial mismatch between the individual parts and spatial constraints between these parts. In addition, a transformation of the shape descriptions has been suggested with which shape descriptions containing asymmetrical spatial constraints between the parts can be matched using symmetric interconnection weights for the Hopfield net. The Hopfield net based formulation has been extended to consider the problem of finding the best match of the test shape descriptions with one of the stored prototypes. The matching scheme has been experimentally applied for recognition of hand-tools and symbols. In both cases, the network produced encouraging recognition results.

Keywords: Shape matching, neural network, spatial constraints.

1. INTRODUCTION

An object can be described in terms of the description of its parts and spatial relations between the parts. Such a description explicitly represents the spatial organisation of the primitives forming the objects. Object descriptions formed in this way are categorised as *structural object descriptions*.[1] Structural descriptions are rotation and translation invariant. Only when parts of an object undergo changes and/or are physically removed do structural descriptions need modification. Because of these invariant characteristics objects as well as higher level entities, like real world scenes or images, can be effectively represented using structural descriptions. If the constituent objects of a scene are also described in the same way, we get a hierarchic structural description of the scene.

Structural descriptions are useful for many computer vision problems like shape matching,[1,2] stereo matching,[3] etc. Structural descriptions of 2-D or 3-D shapes are relatively less affected due to occlusion, distortion or deformations. Hence, recognition or matching schemes based on structural descriptions are robust against occlusion and distortions.[4] Structural object descriptions can be equivalently represented in terms of directed or undirected graphs in which the set of nodes represent the primitives and the arcs represent the spatial relations between the primitives.

International Journal of Pattern Recognition and Artificial Intelligence Vol. 7 No. 2 (1993) 377–404

Attributes associated with the nodes and the arcs provide the descriptive power to the graphs. These graphs are commonly referred to as attributed relational graphs.[5] Recognition schemes based on inexact matching of the graphs have also been developed.[6] Structural description of the images (left and right) can be used in a stereo matching procedure.[3] The number of features or primitives involved in such a matching scheme will in general much smaller than the traditional feature (*viz.* edge points corners) based methods. Also, the matches obtained with the structural descriptions will be more reliable. But *a priori* knowledge of the problem domain is required for construction of a structural description of an image in terms of the known primitives and spatial relations. It is clear from the above discussion that development of effective computational schemes for matching structural descriptions are of prime importance for solving a large class of computer vision problems.

Structural object descriptions can be matched by finding a mapping between the primitives such that the spatial constraints are preserved between them. This is nothing but a constraint satisfaction problem which can be solved using brute force backtracking tree search. Several look-ahead operators have been suggested[1] for increasing efficiency of the tree search algorithm. But, in the worst case, its time complexity is exponential to the size of the descriptions involved. In the case of inexact matching, when a mapping is to be found so as to maximize the match between the common portions of the two structural descriptions, the problem becomes harder.[1] In this case, it is a constraint optimisation problem. Hopfield nets[7–9] provide an efficient computational paradigm for solving these types of problems.[10–12]

The Hopfield net is a recurrent network with all the neurons connected to each other via weighted links. The neuronal units can have either binary or continuous valued activation functions. Each unit, in addition to the inputs from other neurons, can receive external input or bias. The network response is dynamic. Corresponding to a new input, the output of the neurons are calculated and are fed back. In the next iteration outputs are calculated using the modified inputs. This process is repeated. When the network is stable, successive iterations produce smaller and smaller outputs changes until eventually the outputs become constant. The network is stable if the weights are symmetric and there are no self-exciting loops. Symmetric links between the neurons imply that the weight w_{ij} between any two neurons n_i and n_j along the forward path is the same as that of the link along the reverse path (i.e. w_{ji}). A function is defined for these networks which normally decreases each time the network changes state (i.e. each time outputs of the neurons change their value). This function is called the energy function of the network. Eventually, this function reaches a minimum when the network reaches a stable state. The basic characteristic of the network to settle to a minimum of its energy function through iterative updating of its units, which can take place in parallel, can be effectively harnessed for obtaining a good, if not optimal, solution to the structural matching problem at various levels of complexity. Despite the fact that the Hopfield nets are not guaranteed to provide the optimal solutions, rapid computational capability of the network provide an important mechanism for tackling the computational complexity involved in solving the structural matching problems. If implemented in hardware, the solution is not expected to take more than a few time constants. Also, the time required for convergence changes little with the size of the problem.

In this paper a scheme for solving the structural matching problem with Hopfield nets has been presented. In the present formulation, the Hopfield net can be considered as a two-dimensional array in which output of the (i,j)th unit represents the degree of similarity between the ith primitive and the jth primitive of two shapes. Weights associated with the links indicate, on the basis of the spatial constraints between the primitives, the compatibility between the possible matches represented by the corresponding units. This architecture is similar to that presented in Ref. 12. But, in their work, Lin *et al.* have not considered that case of asymmetric spatial relations like bottom, right, left, etc. These relations are specified with a definite sense. For example, if a primitive A is to the left of a primitive B, then it is obvious that B is not having the same spatial relation with respect to A. Other works[10,11] also have not tackled this problem. For the asymmetric spatial relations, weights of the network will not remain symmetric and consequently convergence is not guaranteed under all possible conditions. In this paper, a transformation for the structural descriptions has been proposed to circumvent this problem. With this transformation all types of spatial relations can be used in the structural descriptions. In addition, the energy function of the net has been formulated in the most general fashion to take care of all the problems of inexact matching of the descriptions. Also, the technique has been extended to tackle the problem of simultaneous matching of one description with a number of stored protypical descriptions. The present scheme can also take care of partial mismatch between the parts and spatial relations.

2. MATCHING PROBLEM

In this section, we have considered the problem of finding a match between a candidate shape and the prototype so that the correct correspondence can be established between similar primitives of the two. Individual primitives of the shapes are considered to be low-level geometric tokens, like lines, arcs, subparts of different shapes, etc., which are characterised by a set of property-value pairs. Correspondence can be established between those primitives of the two shapes which possess an approximately equivalent set of property-value pairs. In addition, spatial constraints between the primitives in one shape must be preserved among the matched primitives in the other shape. Spatial constraints or the relational features between the primitives can also be characterised by a set of property-value pairs and accordingly matching criterion can be formulated. Also, in many cases, two shapes do not match completely; only portions of the given shapes may match. The neural network based matching scheme proposed in this paper has been motivated by the above mentioned considerations. For matching structural descriptions of the two shapes, having N primitives each, an artificial neural net with $N \times N$ neural units have been used. Each neural unit represents a possible match between individual primitives of the two shapes under consideration. This representation is natural because each primitive of the candidate shape can be mapped onto any one of the primitives of the prototype. The lowest energy state of the network, in accordance with an appropriately defined energy function, will represent the best mapping between the primitives of the candidate and prototype shapes, which satisfies all the problem dependent constraints.

Let us assume that the energy function has the following form:

$$\mathcal{E} = \mathcal{E}_1 + \mathcal{E}_2 + \mathcal{E}_3 + \mathcal{E}_4 + \mathcal{E}_5 \,. \tag{1}$$

Each term on the right hand side of Eq. (1) represents the contribution due to individual constraints. The term $\mathcal{E}_1$ ensures that any primitive in the candidate shape cannot match with more than one primitive in the prototype and vice versa. Hence, $\mathcal{E}_1$ is defined in the following form:

$$\mathcal{E}_1 = A/2 \sum_p \underbrace{\sum_i \sum_j}_{i \neq j} V_{ip} V_{jp} + B/2 \sum_i \underbrace{\sum_p \sum_q}_{p \neq q} V_{ip} V_{iq} \,. \tag{2}$$

Here, A and B are positive constants;

V_{ij} represents the state of the (i,j)th neuron, and hence it can take values between 0 and 1.

The first sum in $\mathcal{E}_1$ is zero if and only if there is no more than one non-zero entry in each column. The second term is zero if the same condition holds good for each row.

The term $\mathcal{E}_2$ guarantees that the total number of primitive matched is equal to N. So, $\mathcal{E}_2$ has the following form:

$$\mathcal{E}_2 = C/2 \left(\sum_i \sum_p V_{ip} - N \right)^2 . \tag{3}$$

C is a positive constant. When two shapes containing N primitives are perfectly matched, this term becomes zero. If the shapes contain different numbers of primitives, say X and Y, the parameter N must be set to $\min(X, Y)$ because the network can find match for the maximum of $\min(X, Y)$ primitives. Even in the case for partial match, a minimum value of this term ensures the maximum possible number of matches between the primitives. This problem can also be tackled by including dummy primitives in the descriptions containing a lesser number of primitives.

As mentioned before, primitives of the candidate and prototype shapes are characterised by a set of possible attributes or properties and values of these attributes. Two primitives can be considered to be matched if they have a nearly identical set of attributes and similar values for those attributes. This criterion is captured by the $\mathcal{E}_3$ term of the energy function which is given by

$$\mathcal{E}_3 = D/2 \sum_i \sum_p V_{ip} \sum_{a=1}^{\mathcal{A}} \sum_{b=1}^{\mathcal{B}} W_i(a) (\mathcal{P}(i,a,b)(1 - \mathcal{C}(p,a,b)) + \mathcal{C}(p,a,b)(1 - \mathcal{P}(i,a,b))) \,. \tag{4}$$

Here, D is a positive constant,

$W_i(a)$ is the priority or weight of the attribute a for the primitive i (this is assumed to be known *a priori*),

a is the variable representing individual attributes assumed to be indexed from 1 to $\mathcal{A}$,

b is the index for individual elements of the possible value sets of the attributes. The value sets have been assumed to be finite and discrete. Cardinality of the largest set is assumed to be $\mathcal{B}$

$\mathcal{P}(i, a, b)$ is the selector function which can be either 1 or 0.
$\mathcal{P}(i, a, b) = 1$ if the primitive i of the prototype possesses attribute a with the value b,
$\mathcal{P}(i, a, b) = 0$ otherwise.

$\mathcal{C}(p, a, b)$ is the corresponding selector function for the primitives of the candidate shape.

This term will become zero when property-value pairs for the matched primitives of the candidate as well as the prototype shapes are exactly identical. In the case of inexact match, if we expect a partial match of the primitives of the prototype shape among the primitives of the candidate this term can be modified into

$$\mathcal{E}_3 = D/2 \sum_i \sum_p V_{ip} \sum_{a=1}^{\mathcal{A}} \sum_{b=1}^{\mathcal{B}} W_i(a)(\mathcal{P}(i, a, b)(1 - \mathcal{C}(p, a, b))) \,. \tag{5}$$

Here, for example, mismatch of a primitive (say, v) of the candidate shape (i.e. $\mathcal{C}(v, a, b)$ being 1 and $\mathcal{P}(v, a, b)$ being 0) does not affect the value of the term. Similar modification can be incorporated when only consistent matches of the primitives of the candidate shape is being looked for.

But, $\mathcal{E}_3$ being zero does not ensure correct match because none of the terms of the energy function so far considered guarantees existence of identical spatial constraints between the matched primitives of the candidate and the prototype shapes. This condition is imposed by the term $\mathcal{E}_4$ of the energy function.

$$\mathcal{E}_4 = E/2 \sum_i \sum_j \sum_p \sum_q V_{ip} V_{jq} \sum_{t=1}^{T} W(t) \big(\mathcal{RR}(i, j, t)(1 - \mathcal{CC}(p, q, t)) + \mathcal{CC}(i, j, t)(1 - \mathcal{RR}(p, q, t))\big) \,.$$

Here, E is a positive constant,

t is the index for relational attributes,

$W(t)$ is the priority of the tth spatial relation,

$\mathcal{RR}(i, j, t)$ is the selector function which takes values either 0 or 1,
$\mathcal{RR}(i, j, t) = 1$ if the tth relation exists between the ith and jth primitives of the prototype.
$\mathcal{RR}(i, j, t) = 0$ otherwise.

$\mathcal{CC}(p, q, t)$ is the corresponding selector function for the primitives of the candidate shape.

It is obvious from the above formulation of the term $\mathcal{E}_4$ that only binary relational constraints have been taken into account. Also, as described for $\mathcal{E}_3$, this term becomes zero only for perfect match between the shapes. Modifications similar to those for the term $\mathcal{E}_3$ can also be incorporated here. This term can be easily modified in the following way to accommodate k-ary relational constraints:

$$\mathcal{E}_4 = E/k \sum_{i_1}\sum_{i_2}\cdots\sum_{i_k}\sum_{p_1}\sum_{p_2}\cdots\sum_{p_k} V_{i_1p_1}V_{i_2p_2}\cdots\sum_{t=1}^{T_k} W(t)(\mathcal{RR}(i_1, i_2, \ldots, i_k, t)$$
$$(1 - \mathcal{CC}(p_1, p_2, \ldots, p_k, t)) + \mathcal{CC}(p_1, p_2, \ldots, p_k, t)(1 - \mathcal{RR}(i_1, i_2, \ldots, i_k, t)))\,.$$

If the constants A, B and C are sufficiently large as compared to D and E, the low energy state is expected to give a good match between the candidate and prototype descriptions.

Based on the energy function it is straightforward to determine the input bias and connection weights for the neural network. The coefficients of the quadratic terms correspond to the weights in the connection matrix and those of the linear terms correspond to the input bias to the neurons. The connection weights are given by,

$$T_{ip,jq} = \underbrace{-A\delta_{pq}(1-\delta_{ij})}_{\text{inhibitory connections within each row}} \overbrace{-B\delta_{ij}(1-\delta_{pq})}^{\text{inhibitory connections within each column}} \underbrace{-C}_{\text{global inhibition}}$$
$$\underbrace{-E\sum_{t=1}^{T} W(t)(\mathcal{RR}(i,j,t)(1-\mathcal{CC}(p,q,t)) + \mathcal{CC}(i,j,t)(1-\mathcal{RR}(p,q,t)))}_{\text{inhibition by the binary constraints between the primitives}} \quad (6)$$

where, $\delta_{ij} = 1$ if $i = j$
$= 0$ otherwise.

The input bias to each neuron is given by

$$I_{ip} = \underbrace{C_N}_{\text{excitation bias}} \underbrace{-D/2\sum_{a=1}^{A}\sum_{v=1}^{V} W_i(a)(\mathcal{P}(i,a,b)(1-\mathcal{C}(p,a,b)) + (\mathcal{C}(p,a,b)(1-\mathcal{P}(i,a,b))}_{\text{bias by the attributes of the primitives}}\,. \quad (7)$$

In this mapping it is interesting to note that the attributes of the primitives contribute to the input bias of the neurons and the relational constraints contribute to the inhibitory connections between the neurons.

It is clear from the above formulation of the structural matching problem that the weights associated with the links connecting the neurons are determined by the nature of the relational constraints. For symmetric spatial relations, weights will be symmetric. But if we consider asymmetric spatial relations, *viz.* right, left, above, etc., $\mathcal{RR}(i,j,t) \neq \mathcal{RR}(j,i,t)$ and $\mathcal{CC}(p,q,t) \neq \mathcal{CC}(q,p,t)$. Consequently connection weights will be no longer symmetric. But the Hopfield net is not guaranteed to converge to the minimum energy state for asymmetric connection strengths. Although it has been observed for practical simulation problems that if the difference between T_{ij} and T_{ji} is sufficiently low and the gain is also not high, then the network ultimately converges to a minimum energy state, and the time taken for convergence is very high. Hence, it is required to identify a modified strategy so that shape descriptions involving asymmetric spatial relations can be reliably matched with the Hopfield net using the above formulation. In the next section, a scheme is suggested for mapping the problem to the Hopfield net, which has symmetric interconnection weights even for shape descriptions involving asymmetric spatial relations.

3. SHAPE MATCHING WITH ASYMMETRIC SPATIAL RELATIONS

As discussed in the previous sections, it is clear that with the current problem formulation the Hopfield net cannot be effectively used in most of the shape matching problems because asymmetric spatial relations, *viz.* right, left, up, etc., between parts of the objects are very common in general. In this section we describe a technique for transforming structural descriptions of the shapes. With this transformation, the shapes described in terms of asymmetric spatial relations can be matched efficiently using the Hopfield net with symmetric interconnection weights.

A structural description of a shape can be equivalently represented by an attributed relational graph.[5] The nodes of this graph represent individual primitives and directed arcs are the representation of the spatial relations between the primitives. A node n_i and an arc e_{jk} are associated with attribute sets A_{n_i} and $A_{e_{jk}}$ respectively. The set A_{n_i} contains attributes describing geometric properties of the primitives and the elements of the set $A_{e_{jk}}$ describe the spatial relations between the primitives linked by the arc e_{jk}. Asymmetric spatial relations between the primitives can be most naturally represented using this graph. Symmetric relations between the nodes n_j and n_k can also be represented in this framework by associating the same relational attributes with both the edges, e_{jk} and e_{kj}. Consider the example of a simple attributed, relational graph shown in Fig. 1.

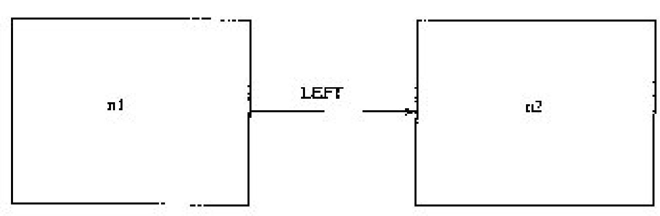


Fig. 1. Example of an asymmetric spatial relation.

For the given example, the directed arc between n1 and n2 indicates that the primitive n2 is to the left of n1. Now, let us replace the directed arc between n1 and n2 by an undirected edge connecting the two nodes. This edge can be associated with an attribute which would indicate the type of the spatial association between the nodes. Consequently, both the sets, $A_{e_{12}}$ and $A_{e_{21}}$, will have identical elements. To encode the exact sense of the spatial relation, additional attributes must be included in the attribute set of the corresponding nodes. Since, the primitive n1 is to the left of n2, A_{n1} will have an additional attribute which will be indicative of the spatial relation "Left" while A_{n2} will have an additional attribute indicative of "Right", i.e. the spatial relation of n2 with respect to n1. In this fashion, all the directed edges between the nodes in an attributed relational graph can be replaced by undirected symmetric edges. Edges will be associated with the types of spatial relations. Types of spatial relations, henceforth the ith type of the spatial relation being indicated by the symbol T_i, will not have any sense of direction or asymmetricity attached to them. For each type of spatial relations, two relational attributes (the kth relational attribute being indicated by r_k) will be considered for indicating the exact sense of the spatial relation between the nodes. If a relational attribute r_k indicates an asymmetric spatial relation of the node n_i with node n_j, then we say that the spatial relation of the node n_j with the node n_i can be indicated by the dual of the relational attribute r_k represented as $\overline{r_k}$. For the given example, if r_k = Left then $\overline{r_k}$ = Right. The transformation technique discussed here is a general purpose procedure by which any attributed relational graph can be transformed into an undirected graph.

Let us consider an attributed relational graph $G = (V, E)$, where V is the set of nodes and E is the set of directed edges. The graph G can be transformed into an undirected graph $G_T = (V_T, E_T)$ by following the procedure described above. Now, in the graph G_T, corresponding to each node n_i, the modified attribute set A_{ni}^T will be $A_{ni} \cup Ar_{ni}$. The set Ar_{ni} contains additional relational attributes. Each element of Ar_{ni} is a tuple (r_k, c_k) where r_k is the relational attribute for the kth type of the spatial relation (the dual of the kth relation can be indicated by using $\overline{r_k}$ or by associating a different index, say j, with it) and c_k is the number of nodes with which the node n_i has the kth spatial relation. The count c_k must be maintained because a node have the same relation with more than one node. The graph G_T can be effectively used for representing structural descriptions if and only is the transformation T is unique. Otherwise, a transformed graph G_T can ambiguously represent more than one description. Consequently, any matching strategy which uses the transformed graph for representing structural descriptions is bound to make inconsistent conclusions. In the following, we formally establish the properties of the transformations and identify the conditions under which the technique becomes an effective representational scheme.

Theorem 3.1. Transformation T cannot produce any other attributed graph other than G_T from G.

Proof. The proof is straightforward because G always has a unique undirected structure and for any particular node, the total number of arcs of a particular type going to or coming from that node is always fixed. □

Now, we need to show that there exists an inverse transformation procedure which will produce a directed attributed relational graph which is the same as that of the original graph G. To start with we present the inverse transformation algorithm.

ALGORITHM Inverse-transform(G_T)

```
for all nodes n_i ∈ V_T
   for all edge-types T_i associated with the edges coming out from the node n_i
      if edges of the type T_i have not yet been associated with directions
         call label(n_i, T_i, f)
         /*f is a flag; it is set true by the procedure label
         if the labeling is successfully accomplished */
      endif
   endfor
endfor
```

Procedure label(n_i, T_i, f)

1. Put in *choice set* the tuple (r_i, c_{r_i}), $(\overline{r_i}, c_{\overline{r_i}}) \in Ar_{n_i}$

/* r_i and $\overline{r_i}$ are the relational attributes of a node corresponding to an edge of type T_i

2. for all edges (n_i, n_j) of type T_i emanating from n_i
   - 2.1 Choose an attribute, say r_i (or $\overline{r_i}$), from the *choice set* which has not yet been chosen for that edge
   - 2.2 if no attribute exists in the *choice set* which can be chosen for the edge (n_i, n_j)
     - 2.2.1 $f = failure$; return
   - 2.3 If the set Ar_{n_j} of the node n_j has a tuple, say $(\overline{r_i}, c_{\overline{r_i}})$ (or (r_i, c_{r_i})), with $c_{\overline{r_i}}$ (or c_{r_i}) ≥ 1,
     - 2.3.1 associate the direction from n_i to n_j (or from n_j to n_i) to the edge (n_i, n_j)
     - 2.3.2 Decrement $c_{\overline{r_i}}$ (or c_{r_i} of the node n_j
     - 2.3.3 Call label (n_j, T_i, f)
     - 2.3.4 If ($f = failure$) go to 2.1
     - 2.3.5 If ($f = success$) decrement c_{r_i} (or $c_{\overline{r_i}}$ as the case may be); go to 2
   - 2.4 else go to 2.1
3. end for
4. $f = success$; return.

Theorem 3.2. The algorithm inverse-transform produces a graph G' wherein all the spatial constraints specified in the graph G are preserved.

Proof. In the process of obtaining the graph G_T from the graph $G = (V, E)$, according to the transformation procedure described in this section, no nodes or edges are eliminated. Also, in the algorithm inverse-transform no edge or vertex is being eliminated in any step. Therefore, $|V'| = |V|$ and $|E'| = |E|$. In other words, according to both the graphs G and G' the set of nodes that are spatially related to each other are identical. Also, information about the type of the relations between the nodes is preserved in the transformed graph G_T and consequently in G'. In the process of transformation, only the edges are stripped of their directions and the algorithm inverse-transform reassociates the directions with the edges such that the sense of the asymmetric spatial relations are preserved. Let us assume that the algorithm produces a graph G' in which direction of at least one edge (n_i, n_j) is different from that in the graph G. Let the edge be of type T_i. The procedure *label* has associated the opposite direction with the edge (n_i, n_j) because corresponding to that direction the conditions specified in Step 2.3.3 of the procedure were satisfied by the tuples of the attribute set of the node n_j and further recursive calls to the procedure label for associating directions with the edges of type T_i continued successfully. The sequence of recursive calls got terminated only after associating directions with all the related edges (i.e. the edges belonging to the paths of type T_i in the graph G_T which pass through the node n_i) without violating the constraints in Step 2.3.3. These constraints can be satisfied only if there exists a sequence of nodes $(n_i, n_j, n_{j+1}, n_{j+2}, \ldots, n_k)$ with $c_{r_i} \geq 0$ and $c_{\overline{r_i}} \geq 0$ for each such node for compensating the choice of wrong direction for the arc (n_i, n_j). Again, for the given sequence of nodes the compensation must take place without affecting the directions of the other edges connected to the nodes. This implies that for those edges correct labels are available in the choice set. This can happen only when error can be compensated by interchanging directions of the edges (n_{l-1}, n_l) and (n_l, n_{l+1}) for all the nodes in the sequence. This is possible only if $n_i \equiv n_k$. In other words, all the above conditions will be true, in the case where there exists a directed cycle corresponding to a particular type of spatial relation in the original graph G. The algorithm inverse-transform, therefore, can associate the wrong direction to an edge in a cycle such that the direction of the cycle as a whole becomes oppositely defined. As a consequence, the spatial constraint between the primitives in the graph G is preserved in the graph G' in terms of spatial relations having an opposite sense. In the absence of a cycle, the initial assumption will turn out to be false because in the course of recursive calls, as described before, it will encounter a situation where either c_{r_i} or $c_{\overline{r_i}}$ will become zero and hence the initial wrong direction cannot be compensated. So ultimately the procedure label will backtrack and the initial mistake will be rectified. Therefore in the graph G' all the constraints between the primitives as specified in the graph G are preserved. □

As a consequence of the above theorem, asymmetric spatial relations can be taken care of without using asymmetric connection weights between the neurons. The relative significance of the relational constraints will be used for calculating symmetric interconnection weights. The asymmetric nature of the spatial relations will be instrumental in determining the input bias of the neurons. For the neurons involved, selector functions corresponding to the associated primitives will indicate the presence or absence of the relational attribute or its dual.

Because of measurement error, random structural alterations or occlusions, structural entities may not always match perfectly. This possibility has been already considered while formulating the energy function of the Hopfield net. But mismatch between relational attributes of the primitives has not been taken care of in that formulation. If the term $\mathcal{E}_4$ takes care of the difference between ordinary attributes of the primitives, then we need to include another term $\mathcal{E}_6$ in the energy function calculation for taking into account mismatch between the relational attributes of the primitives. In this case, a difference in the number of other primitives which satisfy a particular type of spatial relation with the primitives will be of significance. The term $\mathcal{E}_6$ is defined in the following way:

$$\mathcal{E}_6 = F/2 \sum_{i_c} \sum_{j_p} V_{ip} \sum_{t} W(t)(\sigma(c_t^{i_c} - c_t^{j_p}) + \sigma(c_t^{j_p} - c_t^{i_c})) . \quad (8)$$

Here, F is a positive constant,

t is the index for relational attributes,

$Wi(t)$ is the priority of the tth relational attribute,

$c_t^{i_c}$ is the number of tth relational attributes associated with the, ith primitive of the candidate shape (here, duals of the primitives have been considered to be appropriately indexed),

$c_t^{j_p}$ is the number of tth relational attributes associated with the jth primitive of the prototype shape

$\sigma()$ is a function which is defined in the following way:

$\sigma(Z) = Z$ if $Z \geq 0$

$\sigma(Z) = 0$ if $Z < 0$

The function $\sigma()$ has been formulated in the above fashion to take care of the following situations. When the function $\sigma(c_t^{i_c} - c_t^{j_p})$ in $\mathcal{E}_6$ becomes zero, it enables us to ignore relational constraints which are present in the input candidate structural description in addition to those which have been satisfied for the prototype under consideration.

4. SHAPE MATCHING WITH MULTIPLE PROTOTYPES

The problem of matching a prototype to a candidate structural description can be extended to the problem of finding the best match for a candidate structure in a library of multiple stored prototypes. Let there be M prototypes, each having a maximum of N primitives. Let us assume that the input candidate shape possesses N primitives only. So, as discussed in Sec. 2, we need to use $M \times N \times N$ neurons with which we can consider all possible matches between the primitives. These neurons are basically arranged as a stack of two-dimensional arrays, where each plane represents the matching with a single prototype. So, the neuron (i, l, p) is ON if the ith primitive in the candidate structure matches the lth primitive of the pth prototype.

The basic constraints involved in the process of shape matching can be mathematically formulated as follows:

(1) If $V_{ilp} = 1$ then $V_{jkq} = 0 \forall j, k$ and $q \neq p$. This condition inhibits matching of the primitives of the candidate shape to primitives of different prototypes.

(2) In a column or a row of a single plane of neurons, activation of only one neuron can become 1. This implies that one primitive of the candidate can match only one primitive of the prototype and *vice versa*.

To take care of the constraints, in the actual network implementation, shapes having a lesser number of the primitives are to be appended with dummy primitives. These dummy primitives can match with only other dummy primitives with minimum cost; but matching of the dummy primitives with any of the valid primitives incurs maximum contribution to the energy function. The use of dummy primitives enables us to tackle the problem of missing primitives in this framework. Based on the above mentioned constraints, the energy function for the net is redefined in the following way:

$$
\begin{aligned}
\mathcal{E} = {} & A/2 \underbrace{\sum_i \sum_j}_{i \neq j} \sum_l \sum_p V_{ilp} V_{jlp} + B/2 \sum_i \underbrace{\sum_l \sum_m}_{l \neq m} \sum_p V_{ilp} V_{imp} \\
& + C/2 \underbrace{\sum_p \sum_q}_{p \neq q} \left(\sum_i \sum_l V_{ilp} \right) \left(\sum_j \sum_k V_{jkq} \right) \\
& + D/2 \left(\sum_i \sum_l \sum_p V_{ilp} - N \right)^2 \\
& + E/2 \sum_i \sum_l \sum_p V_{ilp} \sum_{a=1}^{A} \sum_{b=1}^{B} W_i(a)(\mathcal{P}(i,l,a,b)(1 - \mathcal{C}(p,a,b))) \\
& + F/2 \sum_i \sum_j \sum_l \sum_m \sum_p V_{ilp} V_{jmp} \sum_{t=1}^{T} W_{ij}(t)(\mathcal{RR}(i,j,p,t)(1 - \mathcal{CC}(p,q,t)) \\
& + \mathcal{CC}(i,j,t)(1 - \mathcal{RR}(p,q,t))) \\
& + G/2 \sum_i \sum_l \sum_p V_{ilp} \sum_{t=1}^{T} W_i(t)(\sigma(c_t^{l_c} - c_t^{j_p}) + \sigma(c_t^p - c_t^i))\,.
\end{aligned}
$$

Symbols occurring in the above equation have the same meaning as those in Sec. 2. Based on the above equation we can calculate the interconnection weights and the input bias of the neurons of the Hopfield net.

5. EXPERIMENTATIONS AND DISCUSSIONS

To establish the effectiveness of the matching mechanism developed in the previous sections, we have considered the application of the technique for solving problems of different domains. In the following subsections, we have summarised our observations.

The network has been simulated on micro-VAX (supporting Ultra-32 operating system). The states of the neurons in the Hopfield net have been synchronously updated in the sequential simulation code. This is done because simulation of the asynchronous updating requires very small discrete time steps and consequently, the time required for convergence becomes very large. On the other hand, in synchronous updating, since the effect of updating the state of one neuron is communicated to others only after completion of the updating of all the neurons for a particular time instant, we can use relatively larger time steps. In the present case, for all the simulations, the size of the step was chosen to be 0.001. The initial states of the neurons are chosen in such a way that the total sum of the outputs of the neurons in the initial state is the same as that for the final state. The total output in the initial state must be uniformly distributed over all the neurons. As the network is updated, depending on the interconnection pattern, the output of some of the neurons becomes 1 and that of the others gradually diminishes. Since, in the final state, only N neurons corresponding to the consistent match of the N primitives of the input candidate shape will be finally ON, we have for the simple single-layered model: $\sum_i \sum_j V_{ip} = N$. Hence, initial states are set as

$$U_i(0) = U_{00} + \delta U_i \tag{9}$$

where $U_{00} = g^{-1}(1/N)$ and $|\delta U_i| \leq 0.1 U_{00}$. The term δU_i introduces a perturbation in the initial states of the neurons such that the network is prevented from getting stuck at initial configuration. In the case of the multiple-layered model, the total number of neurons becoming ON in the final state must be N. Hence, initial values can be chosen according to the following equation: $U_{00} = g^{-1}(1/M * N)$. The transfer function $g(\cdot)$ is chosen as

$$g(x) = 1/(1 + \exp(-x/T)) \tag{10}$$

where the parameter T controls the gain of the neurons. With this transfer function, $U_{00} = -T \log(N-1)$ for single layered networks and $U_{00} = -T \log(MN-1)$ for multiple-layered networks. If the value of T is large, the gain is small and the rate of convergence will be slow. On the other hand if T is chosen to be small, then the network is highly prone to get stuck at spurious local energy minima, thereby providing wrong results.

The quality of the experimental observations are definitely dependent on the interconnection weights among the neurons in the network. Interconnection weights in turn are dependent on the values of the parameters A, B, C, D, E, F, G. The parameters A, B, C, D determine the relative weightage of the domain constraints while the parameters E, F, G determine the relative significance of the spatial constraints among the primitives. If the values of A, B, C, D are much greater than those of E, F, G, mismatch will occur because spatial constraints are violated under such circumstances. On the contrary, if the values of E, F, G are comparatively higher, then matches are obtained wherein domain constraints are violated. For example, with such a choice of parameters, in the final result a single primitive of the test shape may be found to be in correspondence with more than one primitive of the prototype. After careful experimentation, the final parameters for the present implementation were chosen to be $A = 200$, $B = 200$, $C = 200$, $D = 200$, $E = 500$, $F = 500$, $G = 500$. All the weights of the attributes, relations and relational attributes were taken as unity.

5.1. Recognition of Planar Shapes

Structural shape descriptions are useful representations of a variety of shapes which can be easily decomposed into primitive parts having discernible spatial relations between them. Hand tools form a class of objects which can be efficiently represented using structural descriptions. In robotic applications, it should be possible for the vision system to identify these objects despite partial distortions in their shapes. Also, recognition systems must be capable of discriminating between several similar hand tools. The scheme proposed in this paper for matching structural descriptions can be effectively used for this problem. As a case study, we have considered the problem of finding the closest match for a given hand tool among a set of similar hand tools known *a priori*. For the hand tool, different kinds of hammers have been considered. Hammers have been considered to be composed of subparts, each of which can be characterised in terms of their shapes, relative size, and color. Parts have been found to have any one of the following shapes — triangle, rectangle, trapezium. Parts of similar shape can vary in size. With respect to the size, we have classified the parts into three distinct classes — small, medium and large depending on their relative areas. Since qualitative descriptors of the relative areas have been used, the descriptions have become scale invariant. Various types of joins among the parts have been used as spatial relation between the parts. Instead of specific numeric parameters, use of the basic nature of joins as relational features has enabled us to obtain invariant shape descriptions despite minor variations in the shape. Different types of joins considered for the present problem are the following:

$\langle T_1, X, Y \rangle$ join of one end of either the primitive X or Y with the primitive Y at the midpoint of Y or X in an orthogonal fashion. Exact sense of the relation will be captured by the relational label of the edge connecting the nodes which represent the primitives X and Y in the corresponding relational attributed graph of the shape under consideration.

$\langle T_2, X, Y \rangle$ join of the end point of the primitive X with that of Y.

$\langle T_3, X, Y \rangle$ placement of the primitive X or Y at the middle of the other between both the ends.

Corresponding to these types of the spatial associations between the primitives, labels of the edges of the attributed graphs used for representing the hammers are R_1, R_2, R_3. The duals of these relations are indicated as $\overline{R}_1$, $\overline{R}_2$, $\overline{R}_3$ respectively. The hand tools have been assumed to be built of seven types of primitives. They are:

$L1$ {(shape, rect), (size, medium), (color, black)}
$L2$ {(shape, rect), (size, medium), (color, yellow)}
$L3$ {(shape, rect), (size, small), (color, yellow)}
$L4$ {(shape, rect), (size, small), (color, black)}
$L5$ {(shape, triangular), (size, small), (color, yellow)}
$L6$ {(shape, trapezium), (size, small), (color, white)}
$L7$ {(shape, triangular), (size, small), (color, white)}

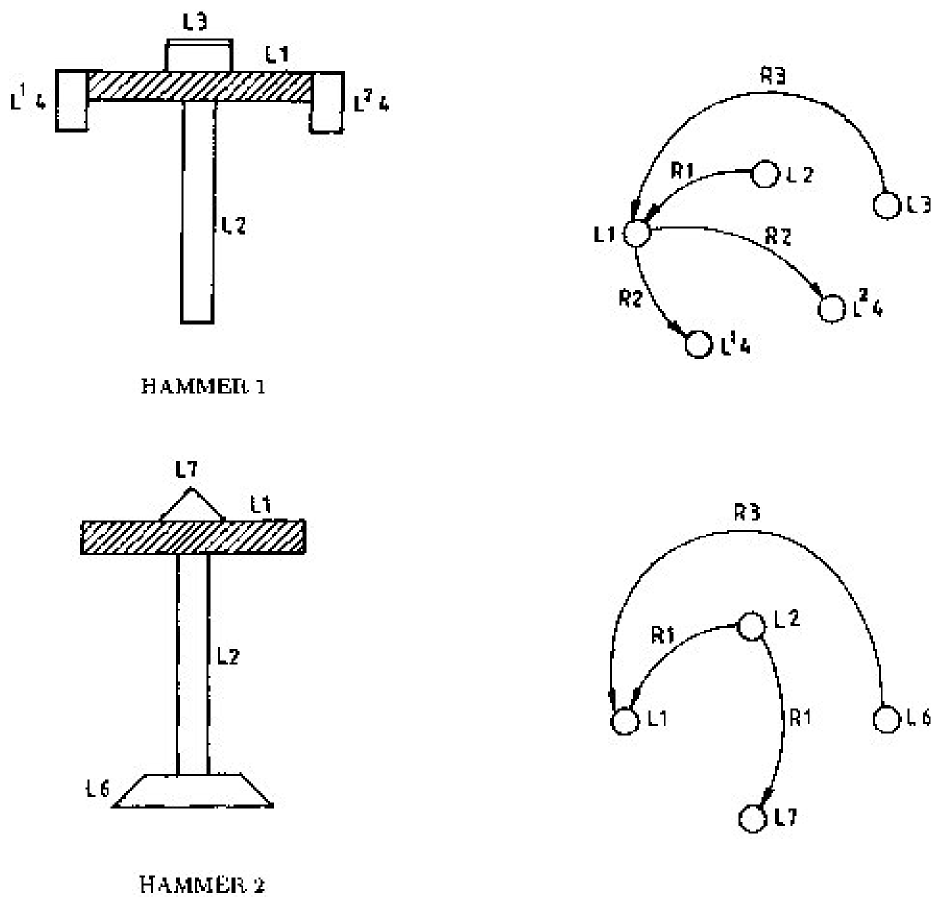


Fig. 2. Model hand tools.

Individual prototype and test shapes and the corresponding attributed relational graphs are shown in Fig. 2. In the figures and the tables, the ith instance of the jth type of the primitive has been indicated as $L^i j$. But, the single instance of a particular type of the primitive (say, kth type) has been denoted simply as Lk. Dummy primitives have been indicated as L^D.

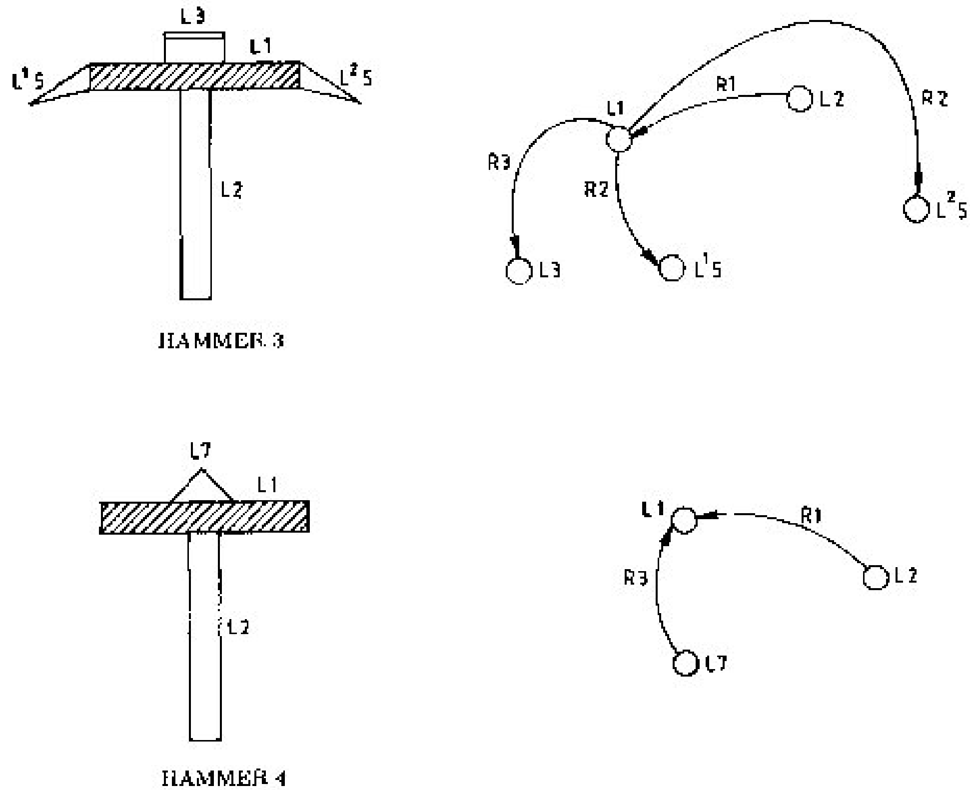


Fig. 2. Cont'd.

The Hopfield net based matching scheme has been applied for finding the correct match of distorted and undistorted instances of the hammers among the stored prototypes. In the first set of experiments, we tried to find a match for the undistorted Hammer 1. We have found (see Tables 1 and 2) that primitives of Hammer 1 matched with the correct counterparts of the stored model when simultaneous matching with all the models was attempted. Only multiple non-zero neuronal output in the same row has been observed for the different instances of the identical type of the primitives. In this case also, magnitude of the output of the neurons corresponding to the wrong matches decreased with the increase in the number of iterations (see Table 2). Matching sequentially with the individual prototypes has also yielded a maximum number of matched primitives for the correct model but there are instances of false matches with the primitives of the other models (see Table 3). But, in simultaneous matching using multi-layered network, there were no false matches with the primitives of the other objects because domain constraints had inhibited these mismatches. This is an important observation which indicates appropriateness of the present technique for matching even similar shapes. This technique was also used for finding match for distorted hammers (in fact, hammers with missing parts). These hammers are shown in Fig. 3. Even with distorted hammers correct recognition results were obtained (see Tables 4 and 5).

5.2. Recognition of Bengali Characters

Structural descriptions of the characters encode spatial relations between the individual primitives. The present technique of matching the structural shape descriptions can be therefore applied for the character recognition problems. As a case study we have considered the problem of recognition of Bengali characters. This problem also has not received much attention in the past.[13] Bengali characters have got features distinct from general Devnagari scripts. In general, as evident from Fig. 4, spatial constraints between straight or curved segments which meet at junctions characterise different Bengali characters. This feature can be effectively exploited by defining appropriate types of the spatial relations and making use of them for constructing the Hopfield net for matching.

Table 1. Matching Hammer 1 with multiple prototypes simultaneously (after 50 iterations).

Candidate	Prototype				
Hammer 1	Hammer 1				
	$L1$	$L2$	$L3$	L^14	L^24
$L1$	0.999	0.0	0.0	0.0	0.0
$L2$	0.0	1.0	0.0	0.0	0.0
$L3$	0.0	0.0	1.0	0.0	0.0
L^14	0.0	0.0	0.0	0.937	0.727
L^24	0.0	0.0	0.0	0.720	0.904

Match score for primitives of other shapes is zero.

Table 2. Matching Hammer 1 with multiple prototypes simultaneously (after 100 iterations).

Candidate	Prototype				
Hammer 1	Hammer 1				
	$L1$	$L2$	$L3$	L^14	L^24
$L1$	1.0	0.0	0.0	0.0	0.0
$L2$	0.0	1.0	0.0	0.0	0.0
$L3$	0.0	0.0	1.0	0.0	0.0
L^14	0.0	0.0	0.0	0.999	0.511
L^24	0.0	0.0	0.0	0.502	0.999

Match score for primitives of other shapes is zero.

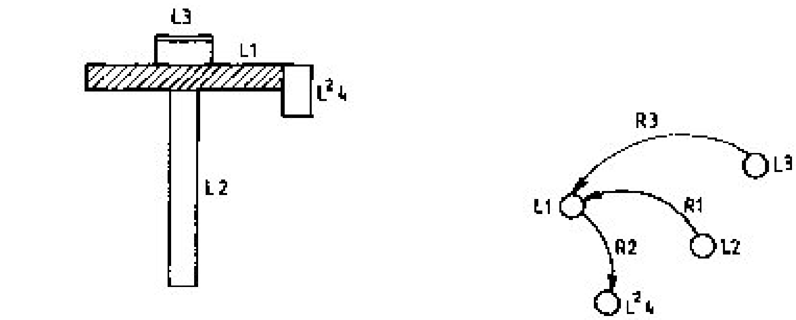
Table 3. Matching Hammer 1 with individual prototypes using single layered network (150 iterations)

Candidate	Prototype				
Hammer 1	Hammer 1				
	$L1$	$L2$	$L3$	L^14	L^24
$L1$	1.0	0.0	0.0	0.0	0.0
$L2$	0.0	1.0	0.0	0.0	0.0
$L3$	0.0	0.0	1.0	0.0	0.0
L^14	0.0	0.0	0.0	1.0	0.513
L^24	0.0	0.0	0.0	0.486	1.0

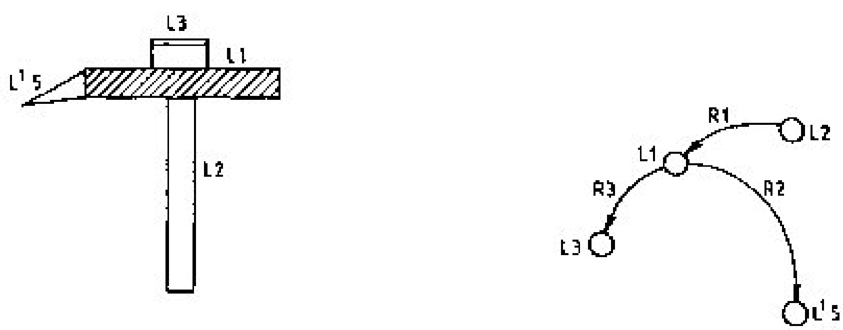
Candidate	Prototype				
Hammer 1	Hammer 2				
	$L1$	$L2$	$L6$	$L7$	L^D
$L1$	0.002	0.0	0.0	0.0	0.0
$L2$	0.0	1.0	0.0	0.0	0.0
$L3$	0.0	0.0	0.7	0.0	0.0
L^14	0.0	0.0	0.0	0.0	0.0
L^25	0.0	0.0	0.0	0.0	0.0

Candidate	Prototype				
Hammer 1	Hammer 3				
	$L1$	$L2$	$L3$	L^15	L^25
$L1$	1.0	0.0	0.0	0.0	0.0
$L2$	0.0	1.0	0.0	0.0	0.0
$L3$	0.0	0.0	1.0	0.0	0.0
L^14	0.0	0.0	0.0	0.0	0.0
L^25	0.0	0.0	0.0	0.0	0.01

Candidate	Prototype				
Hammer 1	Hammer 4				
	$L1$	$L2$	$L6$	L^D	L^D
$L1$	0.7	0.0	0.0	0.0	0.0
$L2$	0.0	1.0	0.0	0.0	0.0
$L3$	0.0	0.0	0.999	0.0	0.0
L^14	0.0	0.0	0.0	0.0	0.0
L^25	0.0	0.0	0.0	0.0	0.0



DISTORTED HAMMER 1



DISTORTED HAMMER 3

Fig. 3. Distorted hand tools.

Table 4. Matching distorted Hammer 1 (primitive L^14 missing) with multiple prototypes simultaneously — Case I (after 50 iterations).

Candidate	Prototype				
Distorted Hammer 1	Hammer 1				
	$L1$	$L2$	$L3$	L^14	L^24
$L1$	0.507	0.0	0.0	0.0	0.0
$L2$	0.0	0.999	0.0	0.0	0.0
$L3$	0.0	0.0	0.999	0.0	0.0
L^24	0.0	0.0	0.0	0.0	0.999
L^D	0.0	0.0	0.0	0.999	0.0

Match score for other shapes is zero.

Table 5. Matching distorted Hammer 3 (primitive L^25 missing) with multiple prototypes simultaneously — Case I (after 50 iterations).

Candidate	Prototype				
Distorted Hammer 3	Hammer 3				
	$L1$	$L2$	$L3$	L^15	L^25
$L1$	0.33	0.0	0.0	0.0	0.0
$L2$	0.0	0.999	0.0	0.0	0.0
$L3$	0.0	0.0	0.999	0.0	0.0
L^1S	0.0	0.0	0.0	0.981	0.959
L^D	0.0	0.0	0.0	0.0	0.723

Match score for other shapes is zero.

For the problem of Bengali character recognition, the types of the spatial relations considered are:

$\mathcal{T}_1(X,Y)$: X is connected to Y end-to-end and makes an angle less than or equal to 90 degrees; X is to the left of Y.

$\mathcal{T}_2(X,Y)$: The end of Y is connected to any point in the right half of X and Y makes an angle of less than 90 degrees with the left half of X.

$\mathcal{T}_3(X,Y)$: The end of Y is connected to any point in the right half of X and makes an angle of approximately 90 degrees.

$\mathcal{T}_4(X,Y)$: The end of X is connected to the end of Y and the angle between them is less than 90 degrees.

$\mathcal{T}_5(X,Y)$: The end of X is connected to any point in the right half of Y and the angle between them is less than 90 degrees. The convex part of X faces the right half of Y.

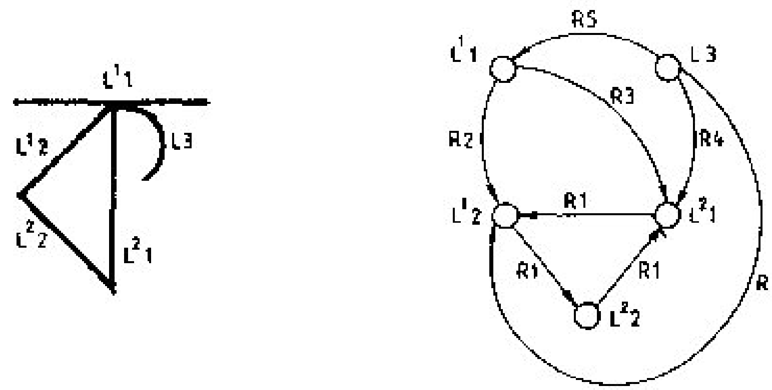
$\mathcal{T}_6(X,Y)$: Same as that of $\mathcal{T}_4$ except that the angle between X and Y is greater than 90 degrees.

Corresponding to these types of spatial relations appropriate relational labels and their duals can be specified in a manner similar to that in the previous section. The characters have been found to be built with the following primitives:

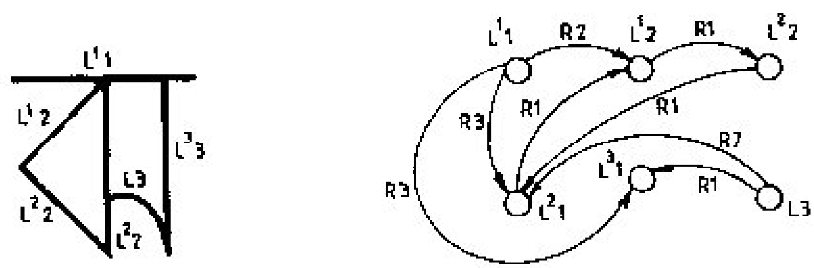
$L1$ {(shape, linear), (size, large)}
$L2$ {(shape, linear), (size, medium)}
$L3$ {(shape, arc), (size, medium)}
$L4$ {(shape, circular), (size, small)}

The prototype characters and their models have been shown in Fig. 4.

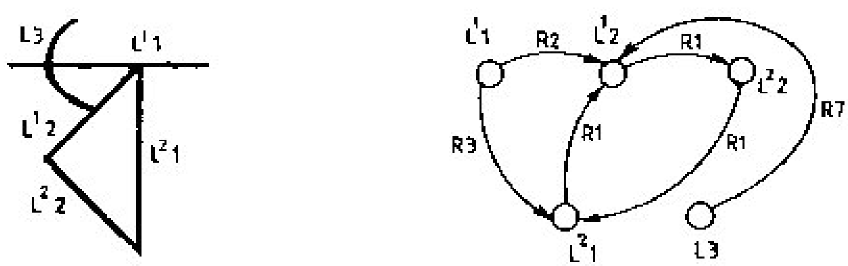
In this case, we tried to find the match for a known character (BCHAR1) in a set of characters specified in Fig. 4. With the multilayered network architecture we got the best results (see Table 6). Also, matching experiments were done with the distorted samples of BCHAR1 and BCHAR2 (shown in Fig. 5). In these distorted



BCHAR 1

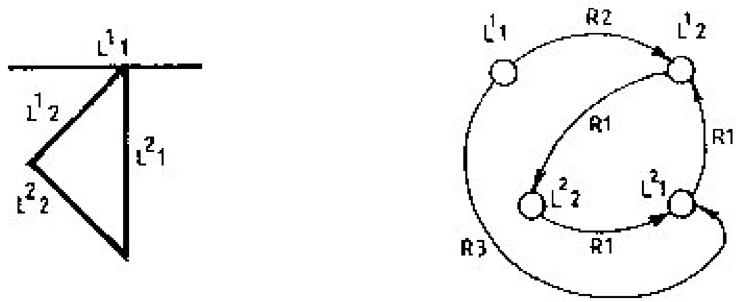


BCHAR 2

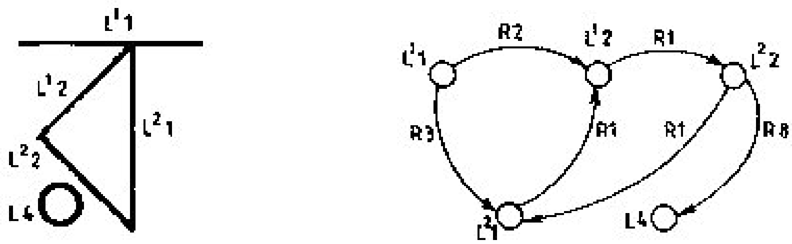


BCHAR 3

Fig. 4. Bengali characters.



BCHAR 4



BCHAR 5

Fig. 4. Cont'd.

samples individual primitives were modified. As a result, two additional types of primitives were found in these samples. They are:

$L5$ {(shape, arc), (size, small)}
$L6$ {(shape, arc), (size, large)}

In the test examples, because of modification in the shape of the character's primitives, some of the spatial relations between the primitives were changed. Results show that even though the characters considered were very similar, the network has correctly found the matches despite distortions (see Tables 7–10). In this case also a multi-layered network was used. Hence, this representation and matching scheme is potentially useful for developing character recognition systems.

Table 6. Matching Bengali character BCHAR1 with all the characters simultaneously (after 100 iterations).

Candidate	Prototype											
BCHAR1	BCHAR1						BCHAR2					
	$L^1 1$	$L^2 1$	$L^1 2$	$L^2 2$	$L3$	L^D	$L^1 1$	$L^2 1$	$L^3 1$	$L^1 2$	$L^2 2$	$L3$
$L^1 1$	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$L^2 1$	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$L^1 2$	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$L^2 2$	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$L3$	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
L^D	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0

Candidate	Prototype											
BCHAR1	BCHAR3						BCHAR4					
	$L^1 1$	$L^2 1$	$L^1 2$	$L^2 2$	$L3$	L^D	$L^1 1$	$L^2 1$	$L^1 2$	$L^2 2$	L^D	L^D
$L^1 1$	0.0	0.0	0.0	0.0	0.0	0.0	0.88	0.0	0.0	0.0	0.0	0.0
$L^2 1$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.53	0.0	0.0	0.0	0.0
$L^1 2$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
$L^2 2$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
L^3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
L^D	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Candidate	Prototype					
BCHAR1	BCHAR5					
	$L^1 1$	$L^2 1$	$L^1 2$	$L^2 2$	$L4$	L^D
	L1	L2	L3	L4	L5	L6
$L^1 1$	0.0	0.0	0.0	0.0	0.0	0.0
$L^2 1$	0.0	1.0	0.0	0.0	0.0	0.0
$L^1 2$	0.0	0.0	1.0	0.0	0.0	0.0
$L^2 2$	0.0	0.0	0.0	0.0	0.0	0.0
L^3	0.0	0.0	0.0	0.0	0.0	0.0
L^D	0.0	0.0	0.0	0.0	0.0	0.0

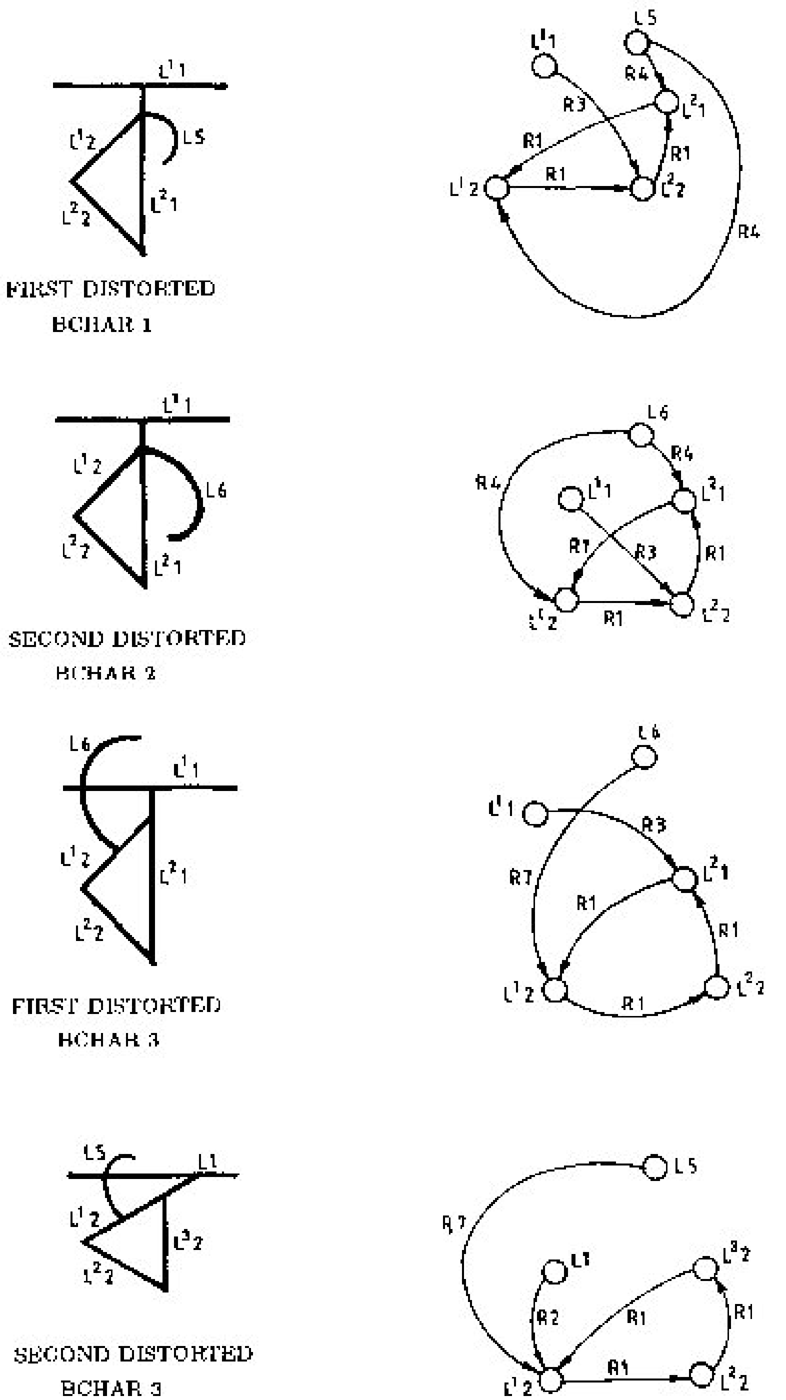


Fig. 5. Distorted Bengali characters.

Table 7. Match of first distorted BCHAR1 (after 150 iterations).

Candidate	Prototype					
TEST BCHAR1	BCHAR1					
	$L^1 1$	$L^2 1$	$L^1 2$	$L^2 2$	$L3$	L^D
$L^1 1$	1.0	0.0	0.0	0.0	0.0	0.0
$L^2 1$	0.0	1.0	0.0	0.0	0.0	0.0
$L^1 2$	0.0	0.0	1.0	0.0	0.0	0.0
$L^2 2$	0.0	0.0	0.0	1.0	0.0	0.0
$L5$	0.0	0.0	0.0	0.0	0.0	0.0
L^D	0.0	0.0	0.0	0.0	0.0	1.0

Match score is zero for primitives of all other characters.

Table 8. Match of second distorted BCHAR1 (after 150 iterations).

Candidate	Prototype					
TEST BCHAR1	BCHAR1					
	$L^1 1$	$L^2 1$	$L^1 2$	$L^2 2$	$L3$	L^D
$L^1 1$	1.0	0.0	0.0	0.0	0.0	0.0
$L^2 1$	0.0	1.0	0.0	0.0	0.0	0.0
$L^1 2$	0.0	0.0	1.0	0.0	0.0	0.0
$L^2 2$	0.0	0.0	0.0	1.0	0.0	0.0
$L6$	0.0	0.0	0.0	0.0	0.0	0.0
L^D	0.0	0.0	0.0	0.0	0.0	1.0

Match score is zero for primitives of all other characters.

Table 9. Match of first distorted BCHAR3 (after 150 iterations).

Candidate	Prototype					
TEST BCHAR3	BCHAR3					
	$L^1 1$	$L^2 1$	$L^1 2$	$L^2 2$	$L3$	L^D
$L^1 1$	1.0	0.0	0.0	0.0	0.0	0.0
$L^2 1$	0.0	1.0	0.0	0.0	0.0	0.0
$L^1 2$	0.0	0.0	1.0	0.0	0.0	0.0
$L^2 2$	0.0	0.0	0.0	1.0	0.0	0.0
$L6$	0.0	0.0	0.0	0.0	0.873	0.0
L^D	0.0	0.0	0.0	0.0	0.0	1.0

Match score is zero for primitives of all other characters.

Table 10. Match of second distorted BCHAR3 (after 150 iterations).

Candidate	Prototype					
TEST BCHAR3	BCHAR3					
	$L^1 1$	$L^2 1$	$L^1 2$	$L^2 2$	$L3$	L^D
$L^1 1$	1.0	0.0	0.0	0.0	0.0	0.0
$L^3 2$	0.0	0.032	0.0	0.0	0.0	0.0
$L^1 2$	0.0	0.0	1.0	0.0	0.0	0.0
$L^2 2$	0.0	0.0	0.0	1.0	0.0	0.0
$L5$	0.0	0.0	0.0	0.0	0.0	0.0
L^D	0.0	0.0	0.0	0.0	0.0	1.0

Match score is zero for primitives of all other characters.

6. CONCLUSION

In this paper, a scheme for matching structural descriptions using Hopfield net has been presented. The matching scheme is based on an appropriate formulation of the energy function for the Hopfield net. This energy function is general enough to accommodate possibilities of partial mismatches between the primitives and spatial relations. Also, the differential criticality factor can be associated with the attributes of the primitives and spatial relations. A transformation scheme has been presented for transforming structural descriptions containing asymmetric spatial relations so that the Hopfield net constructed with the transformed descriptions will not have asymmetric interconnection weights. Consequently, the present scheme can be applied for general structural description matching problems involving realistic asymmetric spatial relations like right, left, on, above, etc. The present scheme of matching a test description simultaneously with all the prototypes eliminates the possibilities of false matches among similar shapes. This fact has been established by the experimental results presented in this paper. Hence, the proposed methodology can be adopted for developing recognition systems in a variety of application domains like robot vision, document processing and navigation, etc.

ACKNOWLEDGEMENTS

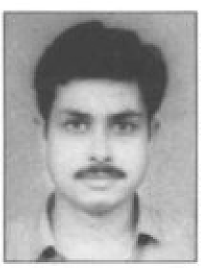
The authors, particularly, Jayanta Basak, Santanu Chaudhury and D. Dutta Majumder, acknowledge Nodal Centre for Knowledge-Based Computing for providing research facilities.

REFERENCES

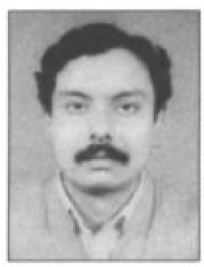
1. L. G. Shapiro and R. M. Haralick, "Organisation of relational models for scene analysis", *IEEE Trans. Pattern. Anal. Mach. Intell.* **4**, 6 (1982) 595–602.
2. L. G. Shapiro, J. D. Moriarty, R. M. Haralick and P. G. Mulgaonkar, "Matching three-dimensional objects using a relational approach", *Pattern Recogn.* **17** (1984) 385–405.
3. K. L. Boyer, A. J. Vayda and A. C. Kak, "Robotic manipulation experiments using structural stereopsis for 3d vision", *IEEE Expert* (1986) 73–94.

4. R. T. Chin and C. R. Dyer, "Model based recognition in robot vision", *ACM Comput. Surv.* **18** (1986) 69–108.
5. A. Eshera and K. S. Fu, "Recognising and locating partially visible objects", *IEEE Trans. Syst. Man Cybern.* **14** (1984) 398–412.
6. R. C. Bolles and R. A. Cain, "Recognising and locating partially visible objects", in *Robot Vision*, ed. A. Pugh, Springer, 1983.
7. J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities", in *Proc. Nat. Acad. Sci. USA* **79** (1982) 2554–2558.
8. J. J. Hopfield, "Neurons with graded response have collective computational properties like those of two state neurons", in *Proc. Nat. Acad. Sci. USA* **81** (1984) 3088–3092.
9. J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems", *Biol. Cybern.* **52** (1985) 141–152.
10. E. Mjolness, G. Gindi and P. Anandan, "Optimisation in model matching and perceptual organisation", *Neural Computation* **1** (1989) 218–229.
11. W. Li and M. Nasrabadi, "Object recognition based on graph matching implemented by a Hopfield style network", in *Proc. Int. Joint Conf. on Neural Networks II*, Washington, DC, 1989.
12. W.-C. Lin, F.-Y. Liao and T. Lingutla, "A hierarchical multiple-view approach to three-dimensional object recognition", *IEEE Trans. Neural Networks* **2** (1991) 84–92.
13. V. Govindan and A. P. Shivaprasad, "Character recognition — a review", *Pattern Recogn.* **23** (1990) 671–683.

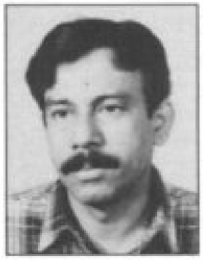
Received 6 January 1992; revised 27 October 1992.



Jayanta Basak completed his undergraduate course in electronics and telecommunication engineering at Jadavpur University in 1987. He received the M.E. degree in computer science and engineering from the Indian Institute of Science, Bangalore in 1989. He worked as a computer engineer in the National Centre for Knowledge Based Computing at the Indian Statistical Institute (ISI), Calcutta from 1989 to April 1992. After that he joined the Electronics and Communication Sciences Unit, ISI as a programmer. His current research interests are neural networks and computer vision.



Santanu Chaudhury obtained his B. Tech degree in electronics and electrical communication engineering in 1984 and his Ph.D. degree in computer science and engineering in 1989 from the Indian Institute of Technology (IIT), Kharagpur. Since January 1992, he has been an Assistant Professor in the Department of Electrical Engineering, IIT, Delhi. His previous experience includes working as a lecturer at IIT, Kharagpur, a Computer Engineer at the Indian Statistical Institute, Calcutta, and a Junior Scientific Officer at IIT, Kharagpur. His research interests are in the areas of computer vision, neural networks, and artificial intelligence. He is the author of more than thirty publications in refereed journals and conference proceedings.



Sankar K. Pal is a Professor in the Electronics and Communication Sciences Unit at the Indian Statistical Institute, Calcutta. He obtained his B.Sc. (Hons.) in physics and B.Tech., M.Tech., and Ph.D. in radiophysics and electronics in 1969, 1972, 1974, and 1979 respectively, from the University of Calcutta. In 1982 he received another Ph.D. in electrical engineering along with a DIC from Imperial College, University of London. In 1986 he was awarded a Fulbright Post-Doctoral Visiting Fellowship to work at the University of California, Berkeley and the University of Maryland, College Park. He received an NRC-NASA Senior Research Award to work at the NASA Johnson Space Center, Houston, Texas, in 1989. He received the 1990 Shanti Swarup Bhatnagar Prize in engineering sciences (which is the most coveted and highest award to a scientist in India) for his contribution in pattern recognition.

He served as a Professor-in-Charge of the Physical and Earth Sciences Division, Indian Statistical Institute during 1988–90. He was also a Guest Lecturer (1983–86) in Computer Science, University of Calcutta.

His research interests mainly include pattern recognition, image processing, artificial intelligence, neural nets, and fuzzy nets and systems. He is a co-author of the book *Fuzzy Mathematical Approach to Pattern Recognition*, John Wiley & Sons (Halsted), 1986 and a co-editor of the book *Fuzzy Models for Pattern Recognition*, IEEE Press, 1992. He has lectured at different U.S. and Japanese universities/laboratories on his research work. He is listed in *REFERENCE ASIA*, Asia's Who's Who of Men and Women of Achievements, vol. IV, 1989.

Dr. Pal is a Fellow of the IEEE and the IETE, a member of the editorial board of *IEEE Transactions on Fuzzy Systems*, the *International Journal of Approximate Reasoning*, and the *Far-East Journal of Mathematical Sciences*, a member of the reviewing board of *IEEE Computer Magazine* and *Mathematical Reviews* and an Executive Committee Member of ISFUMIP and IUPRAI. He is also a permanent member of the INDO-US Forum for Cooperative Research and Technology Transfer (IFCRTT).

Dwijesh Dutta Majumder received the M.Sc.(Tech.) from Calcutta University. In 1962 he received the Ph.D. degree in memory technology from the same university. In 1955 he joined the Electronic Computer Division of the Indian Statistical Institute (ISI). From 1972 until 1992 he was head of the Electronics and Communication Sciences Unit in ISI. He has published more than 300 research papers and six books in the fields of memory technology, speech and other types of pattern recognition, image analysis, computer vision, artificial intelligence, neural modeling, cybernetics, and knowledge based computing.

Dr. Majumder is chairman of the National Centre for Knowledge Based Computing, Professor Emeritus of ISI, Distinguished HCL Professor of IIT, and Emeritus Scientist of CIR. He is a fellow of INSA, INAE, I.A.Sc., IETE, and is a member of the Governing Boards of the International Association of Pattern Recognition, the World Organization of Cybernetics Systems and the International Fuzzy Systems Association.