PsyCOP—A Psychologically Motivated Connectionist System for Object Perception

Jayanta Basak and Sankar K. Pal, *Fellow, IEEE*

***Abstract*—A connectionist system has been designed for learning and simultaneous recognition of flat industrial objects (based on the concepts of conventional and structured connectionist computing) by integrating the psychological hypotheses with the generalized Hough transform technique. The psychological facts include the evidence of separation of two regions for identification ("what it is") and pose estimation ("where it is"). The system uses the mechanism of selective attention for initial hypotheses generation. A special two-stage training paradigm has been developed for learning the structural relationships between the features and objects and the importance values of the features with respect to the objects. The performance of the system has been demonstrated on real-life data both for single and mixed (overlapped) instances of object categories. The robustness of the system with respect to noise and false alarming has been theoretically investigated.**

I. Introduction

RECONGITION of objects in a scene is a very important task in the field of computer vision. The word according to Suetens *et al.* [1] refers to the task of finding and labeling parts of a two-dimensional image of a scene that correspond to the objects in the scene. In computational methodologies, normally, some models for each object (i.e., some general descriptions of each object) are established and then different parts of the scene are labeled according to the knowledge about the models [1], [2]. There exist various techniques based on heuristic search [3], generalized Hough transform [4], [5], relaxation labeling [6], association, and relational graph matching [7], [8], etc. One variation of the template matching scheme is generalized Hough transform [9], [10], where the boundary points of an object are transformed to the parameter space. In this technique the object can be found even when some portion of the data is missing. Whatever methodologies be used, they should be fast enough to handle real-life data, or at least the methodology should be efficiently implementable on a fast parallel machine. Moreover, the technique should be able to learn the object models under supervised or unsupervised mode.

Apart from computational methodologies, there exist various psychological studies in this field. The task of object recognition involves two primary problems. First, one needs to identify the object and the parts thereof which helps in applying the previously gained knowledge, i.e., the model-base to encounter new objects. Second, track or locate the objects properly. It has been mentioned in the computational theories of Kosslyn *et al.* [11], [12] that these two high-level tasks are processed in two different parts of the brain (the identification part is processed in the occipito-temporal region, while the localization part is processed in the occipito-parietal region). The necessary behavioral experiment in support of this separate processing zones has also been provided in literature [13].

There are various studies on the selective attentional mechanism in the literature of psychology [14]–[17]. The theory of selective attention states that different parts in a scene are attended at different times depending on the visual cues present in the scene. The mechanism involves the feedback through top-down paths which gates the receptive field of lower level neurons. The selective gating of the lower level signals takes place with the help of attention director. There exist two different theories of visual attention, namely, early and late selection theories. In early selection theory [14], [15], attention control occurs before recognition of an object (e.g., color, length, etc., of an object). On the other hand, the late selection theory [17] indicates that the attention control occurs after recognition of an object (e.g., reading alphanumeric texts, etc.). Bundsen [18] presented a unified theory of selective attention mechanism where both early and late selection processes occur.

Psychological studies reveal some properties of the cognitive behavior of the animals, while artificial intelligence (AI) formalisms try to extract out objects from digital images. One way to take the advantages of psychological findings and those of conventional algorithms is to use the connectionist framework of computation. The connectionist models (neural networks) have some basic advantages like robustness, adaptivity/capability of generalization, and scope for massive parallelism. Moreover, neural networks, believed to mimic the biological nervous systems (although in a very naive manner), provide a tempting computational paradigm in which the psychological findings can plausibly be incorporated in a better way.

Several connectionist models for object recognition have been developed so far. Cognitron [19] was developed to categorize input patterns by employing competitive learning techniques. But it fails to recognize patterns suffering from positional shift. To incorporate the property of position and scale invariance, a multilayered model, namely, neocognitron [20], was developed. The model uses two kinds of cells,

Manuscript received July 17, 1993; revised April 20, 1994 and July 27, 1994.

The authors are with the Machine Intelligence Unit, Indian Statistical Institute, Calcutta 700 035 India.

namely, S and C cells arranged in alternate layers. S-cells extract features at various stages, while C-cells ensure position and scale invariance. Shift invariance is achieved by tolerating the positional shift, a little, in each layer of C-cells, at a time. The model was extended to incorporate the property of selective attention [21] by using feedback pathways from the output layer to the input layer.

The power of neocognitron and its variation lies in the fact that the models are capable of tolerating error due to positional shift, scale change or deformation. But, the model is not capable of recognizing more than one object simultaneously. Whenever a mixture of patterns is provided to the network, it always recognizes one (most prominent one) of them. Moreover, the model does not consider the structural relationships between the features and the objects. The model is also incapable of tolerating the rotational variance. Recently, the model has been extended to segment and recognize cursive scripts [22] with the help of a "search controller" which assists to select a particular search area. A variation of the model has also been used to achieve rotation invariant object recognition [23].

Hinton [24], [25] used the idea of generalized Hough transform and extended it to dynamic Hough transform model. DHT deals with the problem of scale, position, and orientation invariance by considering a reference frame for the object and describing the features with respect to the object. Using cooperative and competitive computation, the reference frames of the object models were determined. In this model, the relative importance values of the features were not considered.

Mozer [26] developed a word perception model using the selective attentional mechanism for considering the relative positions. It is able to learn and recognize multiple letters, but it was applied only for word perception. Mozer [27] incorporated the mechanism of selective attention and also used the model to explain the phenomenon of neglect dyslexia in psychological patients. The performance of the model, however, is dependent on the orientation of the objects and therefore it may be difficult to use this model for industrial object recognition. Moreover, like neocognitron, no structural relationship between the features and the objects was considered.

Zemel *et al.* [28], [29] have developed a connectionist model using structural properties of the objects. Although the model is capable of learning the structural relations between the features and objects, it is not able to recognize multiple objects simultaneously.

Feldman presented the principles of connectionist computing, principle of stable coalition formation and winner-take-all network in [30]–[32]. Note that, the neural networks mainly involve the study of emergence of activations of the cells and weights of the links on the basis of mathematical modeling. On the other hand, AI-based techniques mostly deal with inference representation. For dealing with the recognition problem in visual domain, representation of knowledge in spatial domain has to be considered. This leads to the concept of structured connectionist models to develop visual recognition system. Sabbah [33] also used structured connectionist framework for origami object recognition.

Some of the connectionist models [19], [21] developed for object recognition, as discussed before, do not consider the task of simultaneous recognition of more than one object. Some of them consider the pose identification, but at the same time uses multiple copies of the same entity at each location [29], [28]. Some models [26], [27] consider pose identification with the help of selective attention mechanism, but do not consider the structural relationships between the features and objects. Although the neurological findings indicate the possibility of existence of two different channels for entity and pose identification, the existing connectionist models do not incorporate this fact in the strategy for recognition. The objective of the present investigation is not to explain the psychological behavior [11], [12] of cognition, but to develop an efficient connectionist system for object recognition by integrating cognitive findings with the generalized Hough transform technique.

In the present investigation, a connectionist system for object recognition has been developed considering the psychological fact that the identification and pose estimation of objects occur in two different regions of human brain. The system is named as PsyCOP which stands for a Psychologically motivated Connectionist system for Object Perception. The principles of structured connectionist computing, as discussed by Feldman [30]–[32], are used in implementation of the model. The idea of incorporating spatial information in the visual domain with a smaller number of neurons used in [31] has been exploited in the design of connectionist architecture. It has been found that separation of two channels for identification and localization, in the design of the connectionist system, leads to an architecture with reduced number of neurons. A two-stage learning paradigm is also designed, which is, in principle, the same as that presented in Basak *et al.* [34], [35]. The learning algorithm has some similarity with that presented in adaptive resonance theory [36]. The robustness of the model under noisy environment is also theoretically investigated.

II. Strategy for Recognition

In the proposed methodology, objects are localized by indicating their position and orientation. The position of an object denotes the position of the object centroid, and the orientation of the object is the orientation of the principal axis of the object with respect to some standard reference frame. The features are also attributed by the feature type, feature position, and orientation. Here, we have used polygonal approximations of two-dimensional (2-D) objects and the corners are used as features. Note that many other features could have been used; however, we have restricted to corner features only. The position and orientation of a corner feature is represented by the position of the corner and orientation of the angular bisector of the corner with respect to some standard reference frame. The reference frame is fixed for all features detected in the scene and the model produces output showing the position and orientation of an object with respect to the same reference frame.

It is to be noted in this respect that we have considered rigid objects only. For any rigid object, so far as the scale

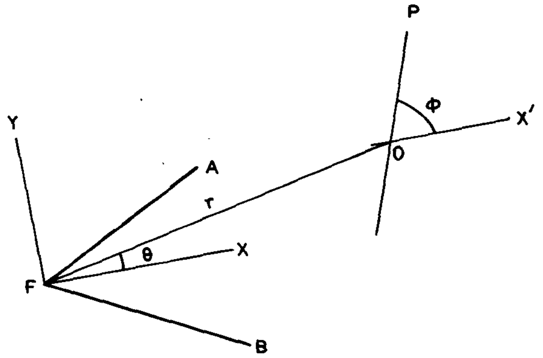


Fig. 1. F and O are the positions of the feature and object, respectively. The corner has an angle AFB. FX is the angular bisector. The feature reference frame is (FX, FY), where FY is normal to FX. The principal axis of the object is OP. (r, θ, ϕ) denotes the location of the object with respect to the feature reference frame.

remains unchanged, the relative position and orientation of an object with respect to a constituent feature remains unchanged. This has been illustrated in Fig. 1. Therefore, if the values of $(r,\ \theta,\ \phi)$ are stored for a particular feature-object combination, then the corresponding feature would be able to predict the position and orientation of the corresponding object in the scene. This is very similar to the idea of generalized Hough transform (GHT) technique [10], where each feature in the image (it was edge pixels in the original work) gives vote to the candidate objects to which it belongs specifying the object locations. The objects (along with their locations) which get cumulative votes greater than certain threshold are considered to be present in the scene.

Even if we consider rigid objects with fixed scale, GHT has a couple of disadvantages. First of all, the peak selection process has to be accurate. A high value of threshold may result in removal of peaks due to genuine objects and a low value of threshold may cause some spurious peaks to remain. Moreover, the required threshold to properly segment the peaks may be different in different regions in the accumulator space. Second, the size of the accumulator space drastically increases with the increase in the number of objects in the model-base. Moreover, the importance of different features with respect to the objects are not considered in GHT. Using connectionist framework of computation, such kind of problems can be dealt with in a better way. The design of such a connectionist system is also motivated by the psychological findings.

In the proposed system, two different channels (block diagram is shown in Fig. 2) have been used to represent the object identities and their locations. "An entity or an object has appeared at a particular location": this can be represented in the form of a coupling between two nodes, one representing the object identity and the other representing the location. Let us call the cells used to represent the "what it is" part as A-cells, and the cells used to represent the "where it is" as B-cells. In designing the actual system, modifiers and pi-connection between the links have been used which is discussed in the Appendix.

The model employes the technique of iterative hypotheses verification [37]. The input layer of the network consists of a set of neurons to represent the entire set of features that can appear in the objects to be recognized. For example, if we consider the corners as the features of the polygonal objects, then the entire range of the angles of the corners are divided into a number of slots, and each slot represents a particular feature. The relative locations of the objects with respect to the constituent features (r, θ, ϕ values) are stored in the links.

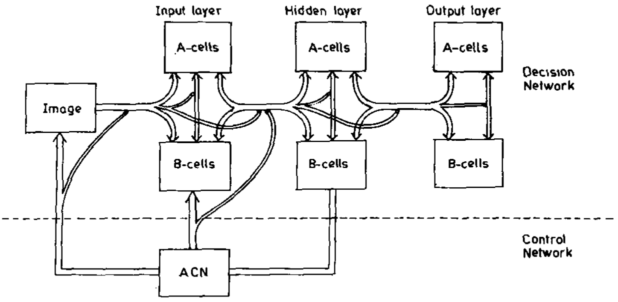


Fig. 2. Block diagram of PsyCOP. The decision network determines the location and identity of an object while the control network controls the selective attention mechanism.

Whenever a set of features and their positions are specified to the system, each feature instantiates a set of possible candidate objects and their respective locations depending on the $(r,\ \theta,\ \phi)$ values stored in the links. All these activations are represented in the form of stable coalitions where an activated B-cell representing the location gets connected through an activated link to an A-cell representing the entity (either feature or object). The activations corresponding to the object instantiations are grouped by the system to produce initial hypotheses.

After the formation of the initial hypotheses of the object categories, the iterative process of verification takes place when each candidate coalition feeds back its activation to the feature level. If the feature activation is less than the feedback activation, then the activation level of the candidate object category is decreased. On the other hand, if the feature activation level is greater than the feedback then the activation level of the candidate object category is increased. The system stabilizes when a match between the input and the feedback activation is achieved. Let us now describe the architecture of the system in detail.

III. Structure of the Model

The system consists of three different layers, namely, input layer, output layer, and hidden layer (Fig. 2). The input layer corresponds to the features, the output layer corresponds to the objects, and the hidden layer corresponds to the feature-object associations. The activation of an A-cell in each layer is either one or zero representing if the corresponding entity is present or absent, whereas the activation level of a B-cell represents the confidence about the presence of some entity at the corresponding location. The A-cells are arranged in linear arrays representing the maximum possible number of features in the input layer, the maximum possible number of objects that the system can learn in the output layer, and the maximum possible number of feature-object associations that

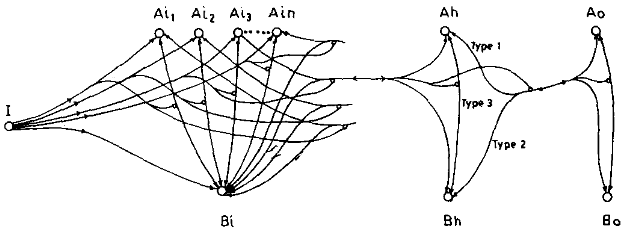


Fig. 3. Connections between the A and B cells of the input, hidden and output layers. "I" represents a feature from which the input nodes are activated.

have occurred in the hidden layer. The B-cells are arranged in the form of a three-dimensional (3-D) array (i.e., columns of B-cells are arranged over 2-D grid), where each cell specifies a particular position and orientation of an entity (feature or object or feature-object association).

The network has three different types of links (Fig. 3). Type 1 links connect the A-cells of one layer with the A-cells of another layer. Type 2 links connect the B-cells of one layer to the B-cells of another layer. The A-cells and the B-cells of the same layer are connected by the type 3 links. For example, the hidden layer A-cells representing the feature-object associations are connected by type 3 links with the hidden layer B-cells.

The type 3 links help in the formation of couplings between A and B cells. If a B-cell is activated, then the activated A-cells that can be accessed from that B-cell through type 3 links represent the entities present at the location corresponding to the B-cell. Similarly, if an A-cell is activated, the activated B-cells that can be accessed through type 3 links represent the locations where the entity corresponding to the A-cell is present. But there will be false alarming in the output whenever more than one object is present in the scene. For example, let two objects O1 and O2 be present in the scene at P1(x_1, y_1, ξ_1)[1] and P2(x_2, y_2, ξ_2), respectively. In that case the A-cell corresponding to O1 would be connected to both the B-cells corresponding to P1 and P2. Similarly, the A-cell for O2 would be connected to both the B-cells corresponding to P1 and P2. As a result, the network would infer that O1 is present at P1 and P2, and O2 is also present at both the locations. Consequently there will be a confusion in the output.

To prevent such a situation, type 3 links are selectively stimulated by using modifiers. An A-cell will be able to access B-cells through only those type 3 links which are stimulated by the modifiers. The type 3 links are modified by the conjunction of type 1 and type 2 links coming from the lower layer. A and B cells in each layer also get activated by the conjunctive or pi-connection from the lower layers.

The type 1 links, emanating from the input layer A-cells, and the type 2 links, emanating from the input layer B-cells, get conjunctively connected to form the type 12 links (note that the conjunctive or pi-connection between type 1 and type 2 links will be denoted by type 12 links). The type 12 links then branch out and enter the hidden layer A-cells and hidden layer B-cells as input, and connect to the hidden layer type 3 links as modifiers. Another branch of the type 12 link emanating from the input layer goes to modify the type 12 link emanating from the hidden layer. Similarly, the hidden layer type 12 links branch out to the output layer A-cells, B-cells, and output layer type 3 links.

Between the hidden and the output layer, there are two kinds of type 12 links. The bottom-up type 12 links carry the activation values of the hidden nodes to the output nodes, while the top-down type 12 links carry the activation values of the output nodes down to the hidden nodes. Type 1 links between hidden and output layer store the relative importance of the feature-object associations, while the type 2 links have some fixed weights. The type 1 links from the input layer to the hidden layer store the transformational offsets (r, θ, and ϕ values), whereas the type 2 links between input and hidden layer have some information regarding physical offsets. The details have been discussed in the next section.

Corresponding to each input A-cell (representing a feature), the hidden layer contains a number of hidden A-cells (which is equal to the number of objects to which the feature belongs, and each hidden A-cell represents a feature-object pair). Whenever the feature set corresponding to an object is mapped onto the input layer, each input coalition tries to activate hidden cells depending on (r, θ, ϕ) values stored in the links. As a result, hidden B-cells are activated within a cluster. To map the feature set onto input layer and to form the cluster of activations, a sequential scanning mechanism is employed which has much similarity to the selective attention mechanism described in the literature of psychology. The selective attention mechanism is realized with the help of a special network, namely, attention control network (ACN). The attention control network is coupled with the type 2 links from the input layer B-cells to the hidden layer B-cells. ACN also uses the mechanism of modifiers to selectively stimulate the type 2 links emanating from the zone of attention. The detailed mechanism of selective attention will be discussed in the next section.

IV. Object Recognition Mechanism

Let us now describe the overall process of recognition with the proposed network. The features (e.g., corners) extracted from the graylevel image are mapped onto the input layer sequentially with the help of attention control network (ACN). When the image (containing the input features) is scanned, the features in the zone of attention activate the input A-cells and the B-cells depending on the type, position, and orientation of the features. It is to be mentioned here that the feature extraction process should take care of the fact that at most one feature can appear in a region equal to the input grid size. The size of the zone of attention depends on the sparseness of the features. It is chosen to be greater than or approximately equal to the input grid size.

The input A-cells have special type of transfer functions which produce maximum output (unity) for certain range of input activation and zero for rest of the input. Note that the activation level of A-cells represent the presence or absence of some entity and not the confidence level about its presence.

[1] The symbol ξ is used to represent angle of orientation.

The input B-cells have two parts: one of them holds the activation value representing the confidence about the presence of a feature and the other has a special type of transfer function (radial basis function [38] tuned at certain orientation). One of the B-cells at the input grid location of a feature gets maximally activated depending on the orientation of that feature. A local competition mechanism within a column of B-cells helps the maximally activated cell to become winner and others lose their activation.

The type 3 links and the conjunctive connections of type 1 and type 2 links, connected to the input A and B cells, also get stimulated (guided by the ACN). As a result the activated A and B cells form coalitions through the stimulated type 3 links and also become able to send activation to the hidden layer through the stimulated type 12 links.

Although the A and B cells get selectively activated from the input, the connecting type 3 and conjunctive type 12 links seem to remain stimulated which may cause formation of false coalitions. To prevent this, the links have a special property where if either of the cells of a stimulated link remains inactive for sometime, the link loses the signal carrying capability unless it is further stimulated by some signal. The time over which the stimulated links lose their signal carrying capability is set in such a way that the link gets deactivated before the attention control network switches to the next zone of attention. (Let us term this property as attenuable LTM).

The hidden layer A and B cells corresponding to a feature are activated from the input coalition through the stimulated type 12 links. The locations of the hidden B-cells to be activated from an input coalition depends on the transformational offsets stored in the type 1 links and the physical offsets stored in the type 2 links between input and hidden layer. Whenever a hidden B-cell is activated, it competes with the other B-cells connected to the same A-cell, and the winner represents the approximate position of the feature-object pair. Once a hidden B-cell establishes itself as the winner for an input activation, it gets biased to that conjunctive connection coming from the input layer (biased connection) and do not take part in the competition process with other hidden B-cells so long as the scanning of the input image is not complete. The principle of biased connection is explained in Fig. 4. Due to the property of attenuable LTM, the links connected to hidden cells other than the winner get deactivated.

In this process of scanning the image, the features are mapped onto input layer and feature-object pairs and corresponding locations are activated in the hidden layer. Each input coalition tries to activate a hidden B-cell at a location corresponding to the object, and since the input coalitions are formed sequentially in the scanning process, a cluster of activations is formed in the hidden layer B-cells corresponding to an object.

Once the scanning of the image is over, the hidden layer A and B cells conjunctively send their activation values to the output layer through bottom-up links. Each output B-cell is connected to a group of hidden B-cells in proximity (as shown in Figs. 5 and 6). Let us term this group of cells as the purview of an output cell. Each output node collects the activation values from its purview in the hidden layer through the bottom-up links. Each output node (B-cell) has a negative self-feedback associated with it. As soon as an output B-cell gets activated, it sends back its activation value down to the hidden layer through the top-down links. Once the hidden B-cells receive feedback activation, the nodes which are activated from the same input pair start competing between themselves. We call this as selective competition process which has been explained in Fig. 7. (Note that the hidden B-cells compete within certain neighborhood during the scanning process of the image. In the settling process, however, the B-cells compete selectively, and this is not confined within neighborhoods.) The hidden A-cell coupled with the winner hidden B-cell (in the selective competition process), corresponding to an input coalition, represents the most likely object to which the corresponding feature belongs. The winner B-cell computes the difference between input activation (since there is biased connection, as mentioned before, the hidden B-cell cannot receive activation from any other input coalition) and the feedback activation and send the difference (we call this as differential support) to the output node through bottom-up link. Here, we like to mention once again that we are not considering the activation values of the A-cells, because the A-cells only modulate the signals by zero or unity, and the actual confidence about the presence of an entity is represented by the activation level of the corresponding B-cell.

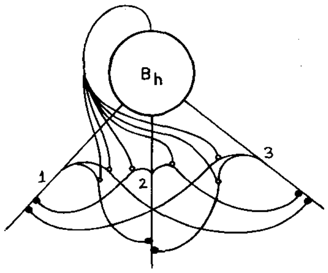


Fig. 4. A network for biased connection of the links to the node Bh. Each link inhibits the other links with the help of inhibitory modifiers. The inhibitory modifiers are in turn stimulated from the output of the node. Let, initially, link 1 carry the activation signal (link 1 has been stimulated by some other means). In that case the node gets activated and stimulates the interlink connections also. Since link 1 carries activation, it can inhibit the other links through the inhibitory modifiers. On the other hand, the other links cannot inhibit link 1, since they are not carrying any signal. As a result, the other links (except link 1) cannot carry signal to node Bh, unless link 1 is deactivated by some other means. In other words, the node Bh gets biased to the link 1.

In the output layer, B-cells having overlapped purview are activated due to the presence of a cluster in the hidden layer. The output B-cells compete within a small neighborhood and the winner represents the exact location of the object. Each hidden node gets biased to an output node from which it is getting maximum feedback (this happens with the same principle as the input bias). The differential activation is sent to the output node to which the hidden node is biased. Each output node receives differential support and negative feedback, and it updates its activation value. The same process repeats and the activation values of the output nodes stabilize when the differential support and negative self-feedback become equal. After stability, the activation levels of the output B-cells represent the confidence about the presence of the

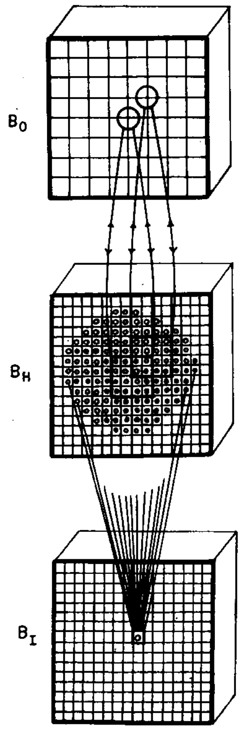


Fig. 5. Connections from the input layer B-cells to the hidden layer B-cells and from hidden layer B-cells to the output layer B-cells. Links from an input B-cell connect to the hidden B-cells over a cone.

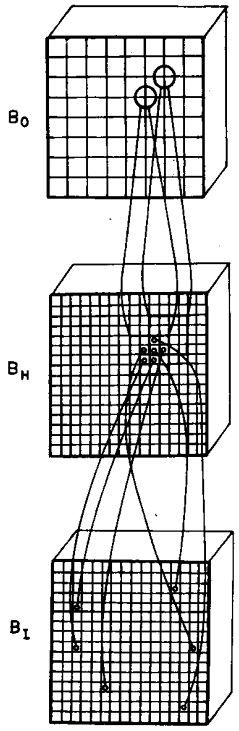


Fig. 6. The relative position of an object with respect to its constituent features. For each input node, at least one hidden node is activated. The hidden node activations are collected at the output layer B_o.

corresponding objects (determined by the activated A-cells coupled with them) at that locations.

Now let us discuss in detail about the instantiation of hidden nodes (transformation embedding), competition within neighborhood, selective attention, and finally the dynamic behavior of the network.

A. Instantiation of Hidden Nodes

Type 1 links store the actual values of positional and orientational offsets (r, θ, ϕ) between features (represented by input A-cells) and objects to which the features belong (represented by feature-object combinations, corresponding to hidden layer A-cells). Type 2 links between the input layer and the hidden layer have fixed weights and the weights decrease with the increase of physical offsets between the locations of input B-cells and hidden B-cells. Each type 1 and type 2 link can be viewed as a composition of three links, one to represent r, one to correspond θ, and the other to correspond ϕ. Let us denote the weights of the type 1 and type 2 links from input to hidden layer by $W1$ and $W2$, respectively. The values of $W2$ are mathematically given as

$$W2^* = \frac{\epsilon_*}{\epsilon_r^2 + \epsilon_\theta^2 + \epsilon_\phi^2} \tag{1}$$

where "*" stands for r, θ or ϕ, and $(\epsilon_r, \epsilon_\theta, \epsilon_\phi)$ represents the physical offsets.

Let the ith input A-cell and jth B-cell conjunctively activate the kth hidden A-cell and lth hidden B-cell. Then the activation received by the hidden B-cell (ubh) is given as

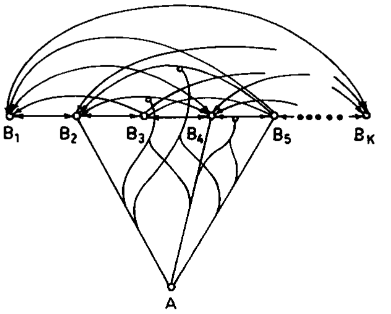


Fig. 7. The connections for selective competition. The cells $B2, B4$, and $B5$ compete among themselves because only those particular internode links are stimulated by modifiers.

$$\begin{aligned} ubh_l = {} & (W1_{ik}^r \cdot W2_{jl}^r + W1_{ik}^\theta \\ & \cdot W2_{jl}^\theta + W1_{ik}^\phi \cdot W2_{jl}^\phi) xa_i \cdot xb_j \end{aligned} \tag{2}$$

where xa and xb are the activation values of the input A and B cells, respectively (the activation received by hidden kth A-cell is equal to ubh). It is clear from (2) that the activation of the hidden layer B-cell would be maximum for which the stored transformations in $W1$ links perfectly match with the physical offsets represented by $W2$. In other words, this kind of activation helps the network to form a cluster of activations in the hidden layer B-cells at a particular location when an object is present at that location. The transformation values are also incorporated in other networks [29]. The way it has been incorporated here, however, is different from that used in [29].

The type 2 links carry activations from the input to the hidden layer over a cone, where the weights of the links decrease as the distance between the hidden nodes and the input node increase. If a hidden node has the same location as that of the input node, then according to the equation one of the weights would become infinitely large. To have a finite set of weights in such cases the hidden layer position and orientation can always be defined in such a way that there exists a minimum deviation by $(|\Delta x/2|, |\Delta y/2|, |\Delta\xi/2|)$ from the input layer.

B. Selective Attention

As described before, selective attention is a pseudo-parallel mechanism where different parts of an object are attended sequentially to get an idea about the object. This technique has been used in the proposed model. Features mapped onto the input layer are scanned sequentially to activate the proper hidden nodes.

In the scanning process each time a particular zone is attended. During scanning, features appearing in the zone of attention are mapped onto input layer, and input coalitions activate the hidden nodes. The size of the zone of attention approximately depends on the sparseness of the features. The zone of attention should be such that more than one feature belonging to the same object does not appear simultaneously. Whenever more than one feature belonging to the same object appears simultaneously, they would be mapped into the same location in the hidden layer. Since each hidden B-cell can take care of only one feature, there will be a collision. Each B-cell in the hidden layer is provided with a mechanism for collision detection which, in turn, helps in controlling the size of the zone of attention.

1) Attention Control Network: The attention control network (ACN) is coupled with the links from the input layer B-cells to the hidden layer B-cells. A link would be able to carry activation from the input layer to the hidden layer only when it is stimulated by the ACN. It is to be noted here that the mechanism of attention is applied only in the phase of initialization of the network. Once the network is initialized, the network is able to update the states of the nodes simultaneously.

The size of the zone of attention has a default value. Whenever a conflict arises in any hidden B-cell, it informs the ACN, and the ACN reduces the size of the zone of attention. The process of reducing the size of the attention zone is stopped whenever the signal from the hidden layer indicates that there exists no conflict. The size of the attention zone gradually increases as the ACN shifts the zone of attention.

2) Conflict Detection: Each hidden B-cell is able to detect a conflict whenever it arises between more than one nonzero signal coming from the input layer. To detect a conflict situation and consequently to inform ACN, each hidden node has a separate part, namely, conflict detector (CD). The CD functions in the following way.

Let $\boldsymbol{u} = [u_1, u_2, \cdots, u_l]$ be the input vector to a particular B-cell. If there exists no conflict then

$$u_i u_j = 0, \qquad \text{for all } i, j.$$

In other words

$$\max_{i \neq j}\{u_i u_j\} = 0.$$

The conflict detector of each hidden B-cell has a transfer function given as

$$v1 = f(\max_{i \neq j}\{u_i u_j\} - \varepsilon) \tag{3}$$

where $f(\cdot)$ is a step function which is given as

$$f(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise.} \end{cases}$$

A small positive constant ε is used for noise tolerance (ε may also be chosen as zero). The ACN takes the outputs from CD's of all hidden B-cells and detects the maximum signal. If the maximum signal is nonzero, then it infers that a conflict exists at some hidden B-cell, and consequently it reduces the size of the attentional zone.

C. Competition over a Local Neighborhood

The output B-cells have local competition. This is necessary because each output B-cell is connected to a group of hidden B-cells (purview of the output B-cell). The cells are arranged in such a way that two different groups of hidden B-cells may or may not have overlap between them. If there is an overlap, confusion would arise regarding the location of objects. For example, let the hidden layer consist of two neighboring groups G_1 and G_2. In that case, some B-cells from G_1 and G_2 can be activated due to the presentation of the feature set corresponding to a single object. The output layer will collect activation values from the groups G_1 and G_2 (say output B-cells O_1 and O_2) and, as a result, indicate that the object is present at two neighboring locations. If there were a group (say G_3) having overlap with both G_1 and G_2 such that most of the activated hidden B-cells fall into G_3, then the activation of the output B-cell (say O_3) collecting activations from G_3 would be higher than those of O_1 and O_2. If there exists competition between O_1, O_2, and O_3, then O_3 would indicate the actual position of the object and there would be no confusion. Moreover, if an object consists of a large number of features, then the number of hidden cells required to represent the feature-object associations would be large. If there were no overlapping groups, then output cells would be widely separated and imprecision may arise.

The local competition can take place in several ways. Two examples are presented in Fig. 8. In the first case, each group has overlap with eight other groups (the figure shows only four spatially separated groups). Each output cell competes with eight neighboring output nodes in its local neighborhood. In the second example, each group of hidden nodes has overlap

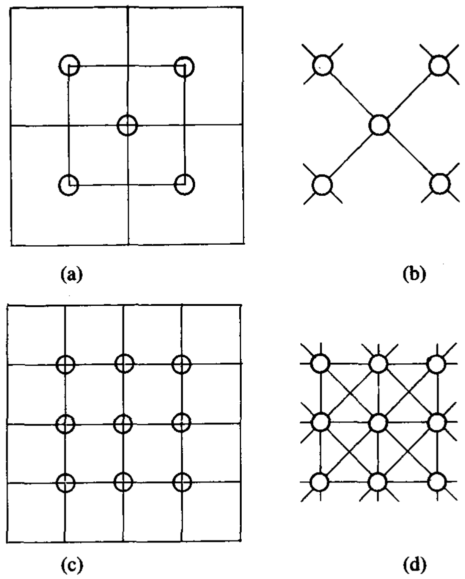


Fig. 8. (a) Overlap of the purview of one output node with those of its eight neighbors in (X, Y, θ) space (only four are shown here). (b) Connection of an output node with its eight neighbors in (X, Y, θ) space (only four are shown here). (c) Overlap of the purview of one output node with those of its 26 neighbors in (X, Y, θ) space (only eight are shown here). (d) Connection of an output node with its 26 neighbors in (X, Y, θ) space (only eight are shown here).

with 26 neighboring groups (only eight spatially separated groups are shown in the figure). In this case, the output cell competes with the neighboring 26 nodes. The overlapping between the groups of hidden B-cells can even be larger at the cost of larger number of output cells.

D. Dynamic Behavior of the System

Let us now describe the dynamic behavior of the system. In the initialization process, each activated coalition of input A- and B-cells activates a number of A- and B-cells in the hidden layer. Note that, the activation values of A-cells in any layer represent presence or absence of the corresponding entities, and therefore, the activation levels of A-cells are either one or zero. On the other hand, the activation values of the B-cells represent the confidence about the presence of the corresponding entities. In the consequent discussion, we will be considering the updating of the activation values of the B-cells only. The output layer B-cells, in the initialization process, receive activation from the hidden B-cells over the corresponding purview. In the settling process, the output B-cells send their activation values back to the hidden layer B-cells. After receiving the top-down feedback, the hidden layer B-cells connected to the same input B-cell selectively compete with each other. After competition, for each input B-cell there exists one winner-take-all (WTA) hidden B-cell. Each WTA hidden B-cell computes the difference of the activation signal coming from the input layer and the top-down feedback received from the output layer. The difference of the activation values (differential support) is propagated to the output B-cell. Each output B-cell has negative self-feedback and updates its state depending on the differential activation value received from the hidden layer and the self-feedback. Before going to mathematical formulation of network dynamics, let us briefly discuss about the properties of the cells.

1) Properties of the Cells: The input–output function of a hidden or output A-cell is formulated as

$$v = f\left(\max_i u_i - \varepsilon\right)$$

where v is the output of an A-cell, u_i is the ith input, ε is a small threshold, and $f(\cdot)$ is a step transfer function. Each input A-cell has a transfer function which produce maximum output for certain range of input. Mathematically

$$f(x) = \begin{cases} 1 & \text{for } x_1 < x < x_2 \\ 0 & \text{otherwise.} \end{cases}$$

The actual range of values (x_1, x_2) will be discussed in Section VI.

Each input B-cell has two parts. One of them holds the activation value and has a linear transfer function. The other part has a radial basis function [38] which produces maximum output for certain orientations. Each hidden layer B-cell has four parts. One of them detects if there exists any collision. One of them competes within a local neighborhood to represent the proper position of the hidden node activated by an input node as mentioned in Section IV-A. One part of each B-cell holds the activation value propagated from the input layer. The other part selectively competes with the other B-cells to determine which particular cell would support the output cells in the settling process. The response of collision detector part $(v1)$ of the B-cells is already discussed. The portions which enable hidden B-cells to compete within local neighborhoods in the initialization process, have simple linear gain, i.e.,

$$v2 = u_i$$

where u_i is input received from the input coalition to which the hidden node is biased. The portions which compete selectively with other B-cells have also similar linear gain, i.e.,

$$v3 = fb$$

where fb is the feedback received from output layer. The portion which holds the activation received from input layer has exactly the same gain as $v2$.In the output layer, each B-cell possesses two different parts, one of them represents the confidence about the presence of an entity at the corresponding location, which can be mathematically expressed as

$$vol = g\left(\sum_i u_i\right)$$

where $g(\cdot)$ is an S-function [34] (in the output layer linear gain is not used). u is the input received from hidden layer over a purview. The second portion of each output B-cell locally competes with other cells which has exactly the same gain as vol.

The notations used here to represent output of different cells will not be used any further. In the subsequent discussion, we will be considering the updating of the activation values of that portions of output B-cells only which represent the confidence about the presence of an entity (i.e., $vo1$). The notations used to represent the activation values, however, would be different and clarified in due context.

2) STM Equations: Let us now mathematically describe the dynamical behavior of the network. Before going into the details of the dynamics, let us clarify some symbols used here. As mentioned before, $W1$ and $W2$ represent the weights of the type 1 and type 2 links between input and hidden layer. $w1$ and $w2$ denote the weights of the type 1 and type 2 bottom-up links, respectively, and $z1$ and $z2$ denote the weights of the type 1 and type 2 top-down links, respectively, between hidden and output layer. The hidden layer A-cells are denoted by an ordered pair (i, k) where i and k denote the input and output A-cells to which it is connected. Let the $(i,\ k)$th hidden A-cell and (j, l)th hidden B-cell be conjunctively connected to the kth output A-cell and lth output B-cell, i.e., $(j,\ l)$th hidden B-cell is within the purview of the lth output B-cell and connected to jth input B-cell. The updating of the activation level of an output B-cell can be written as

$$\frac{dubo_l}{dt} = \sum w1_{ik} \cdot w2_{jl} \cdot e_{ijkl} - w_s \cdot (vb_l)^2 \tag{4}$$

where ubo total activation received by an output B-cell. The summation is taken over the purview of the output cell (l) and only for those links which are selectively stimulated. vb_l is the output of lth output B-cell (note that this is the same as $vo1$) which is given as

$$vb_l = g(ubo_l).$$

The differential support e_{ijkl} can be written as

$$e_{ijkl} = \begin{cases} (hb_{ijkl} - fb_{ijkl}) \\ \qquad \text{if } fb_{ijkl} > fb_{ijk'l'} \quad \text{for all } k' \neq k \wedge l' \neq l \\ 0 \qquad \text{otherwise} \end{cases} \tag{5}$$

where fb is the feedback support given as

$$fb_{ijkl} = z1_{ki} \cdot z2_{lj} \cdot vb_l \cdot va_k \tag{6}$$

hb_{ijkl} represents the activation value received by the hidden B-cell from the input layer in the transformed space due to the activation of ith input A-cell and jth input B-cell (2). Note that (5) shows that differential support is computed only at that node which receives maximum feedback for a given i, j. The convergence of the network dynamics can be proved in a similar way as presented in [34] and [35].

V. Learning Process

The weights of the type 1 links between input and hidden layer and the type 1 bottom-up and top-down links between hidden and output layer are learned under supervised mode. (Note that the weights of type 2 links are fixed and are not updated.) The updating of the weights takes place at two different levels. The weights from the input to the hidden layer represent the transformational offsets from the feature reference frame to the object reference frame, and the weights from the hidden layer to the output layer represent the likeliness of appearances of particular feature-object combinations. The two stages of learning are being discussed below.

The learning methodology can be structured as follows:

Step 1: Present the features (i.e., activate A-cells) and their locations (i.e., activate B-cells) at the input layer. Present the corresponding object with its location at the output layer. It is to be noted here that during the learning process only one object can be present at a time.

Step 2: Check if the required transformation values from the feature reference frame to the object reference frame already exist in the links from the input layer to the hidden layer. If the transformation value for a feature-object pair exists then the corresponding hidden B-cell would be activated within the purview of the output B-cell. Otherwise there will be no such activated hidden B-cell for that pair.

If any such activated hidden B-cell does not exist then:

A) Check if there exists any bottom-up and top-down link between that particular feature-object association (corresponding activated hidden A-cell) and the object (the desired activated output A-cell).

If it exists then:

i) update the weights of the bottom-up and top-down type 1 links (i.e., update $w1$ and $z1$) considering that the feature-object association is absent at that instant, i.e., decrease the corresponding weights.

ii) activate a hidden B-cell within the purview of the output B-cell such that hidden B-cell is nearest to the center of the purview.

iii) learn transformation values in the links from the input to the hidden layer.

If the corresponding bottom-up and top-down links do not exist (i.e., either the object is a new one or the corresponding feature did not appear in the object in the previous trials) then:

i) allocate a hidden A-cell for that feature-object association and activate a hidden B-cell within the purview of the output node such that the hidden B-cell is nearest to the center of the purview.

ii) create bottom-up and top-down type 1 links between the hidden A-cell and the output A-cell.

iii) initialize the weight of the bottom-up type 1 link to a certain small value and the weight of the top-down type 1 link to unity.

If the transformation exists then:

A) Check if the activated hidden B-cell is at the center of the purview of the output B-cell, and adjust the weights of the type 1 links from input layer to the hidden layer (i.e., the transformational values) depending on the desired position of the hidden B-cell (i.e., center of the purview) and the actual position of the hidden B-cell.

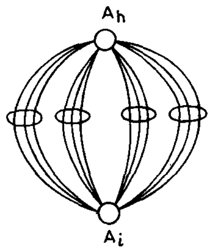


Fig. 9. Diagram of gating channels of the node Ah. A number of links are connected between the nodes Ai and Ah. The links going to the same channel only have equal weights. The links within a loop represent the links going to the same channel.

B) Update the weights of the bottom-up and top-down type 1 links between the hidden layer and the output layer.

Step 3: Copy the weights of the bottom-up and top-down type 1 links between the other type 1 links connected with the same hidden and output A-cells.

Step 4: Copy the weights of the type 1 links from input layer to the hidden layer between other type 1 links within the same group with same gating channel (gating channel is being discussed subsequently).

A. Weights from Input to Hidden Layer

The average transformational offset from the feature reference frame to the object reference frame are adaptively captured by learning the weights from input to the hidden layer. Note that, the type 2 links store informations about physical offsets (ϵ_r, ϵ_α, ϵ_θ) as described in (1). An object may have multiple instances of the same feature located at different places. In other words, there should be provision to store multiple instances of transformational offsets between an input A-cell and a hidden A-cell. To encounter this problem, each hidden A-cell has more than one gating channel (Fig. 9) and to each gating channel a group of type 1 links is connected. During learning, the transformational weights ($W1$) are copied between the type 1 links only if they are in the same group. The weights of the links in two different groups are not copied although they connect the same input and hidden A-cells. As a result, even if multiple instances of the same feature exist in a particular object, the different transformational offsets are stored in the different groups of the type 1 links and they would not affect each other.

Let ith A-cell and jth B-cell in the input layer are activated, and correspondingly kth A-cell and lth B-cell should desirably be activated in the output layer. In that case, $W1$ is updated in such a way that (i, k)th hidden A-cell and a B-cell at the same position of lth output B-cell get activated. Let us denote the hidden B-cell with the same notation, i.e., l. In that case, updating of the weight of type 1 link from input to hidden layer is given as

$$\frac{dW1^*_{ik}}{dt} = \alpha_{ik}(\epsilon_* - W1^*_{ik}) \qquad (7)$$

where * stands for r, θ, or ϕ. α is agility factor whose value decreases with the number of presentations [34]. ϵ_* represents the physical offset between the position of input B-cell and the desired position of output B-cell. With this learning rule (7), an iterative averaging of the offsets between the locations of features and objects under different presentations is performed. In such kind of learning, however, question may arise regarding how to measure ϵ_* locally. In the supervised process of training, the corresponding offsets between the features and objects can be supplied to the network.

Another technique may be used to learn $W1$ values using gradient descent technique. The activation value received by a hidden layer B-cell is given by (2)

$$ubh_{jl} = (W1^r_{ik}\cdot W2^r_{jl} + W1^\theta_{ik}\cdot W2^\theta_{jl} + W1^\phi_{ik}\cdot W2^\phi_{jl})\cdot xa_i\cdot xb_j$$

and the output of hidden B-cell is same as the input received by it, i.e., $hb_{jl} = ubh_{jl}$. The error at the lth cell is given by

$$E_l = \tfrac{1}{2}(t_{kl} - hb_{jl})^2.$$

The change in $W1$ can be given as

$$\Delta W1^*_{ik} = \eta \frac{\partial E_l}{\partial W1^*_{ik}}. \qquad (8)$$

After algebraic computation this becomes

$$\Delta W1^*_{ik} = \eta(t_{kl} - hb_{jl})W2^*_{jl}xa_i xb_j \qquad (9)$$

where ubh is the total input to the hidden layer B-cell and * stands for r, θ, or ϕ, and η is the rate of learning.

We have used the first technique, i.e., iterative averaging of the transformational offsets in our implementation. Once the transformation values are learned for a feature-object combination, they are copied over the other type 1 links (corresponding to the same feature-object combination) which are conjunctively connected to other type 2 links. This kind of copying the weights to other links has been introduced in [39].

B. Weights from Hidden to Output Layer

In the second part of training, the feature importance values with respect to the objects are learned. In this process, some asymptotic measures are considered, and they are used for the learning. The weights of the top-down links ($z1$) are asymptotically equated to the likeliness of appearance of a feature with respect to the objects. The weights of the bottom-up links ($w1$) are asymptotically equated to a normalized value of the product of appearances of the features with respect to the objects and objects with respect to the features. The details of the reasons for choosing such measures are provided in [34] and [35]. For any arbitrary kth object and ith feature, the asymptotic measures can be written as

$$z1_{ki} = p(f_i|o_k) \qquad (10)$$

and

$$w1_{ik} = \frac{p(f_i|o_k)p(o_k|f_i)}{\gamma + \sum_i p(f_i|o_k)p(o_k|f_i)}. \qquad (11)$$

The constant γ is used to incorporate Weber's law [40] which is necessary due to the following reason. If two objects are such that the feature set of one object is a proper subset of another and transformational values are the same then Weber's law prevents the larger object to get more activation than the smaller one when the feature set of the smaller one is presented to the network. The normalization is performed to ensure the total input received by an output B-cell is less than or equal to unity. The updating of $w1$ can be formulated as

$$\frac{dw1_{ik}}{dt} = \left(\alpha_i \delta_{kl} z1_{ki} + \alpha_k^o \left(\frac{w1_{ik}}{z1_{ki}}\right)\right) xa_i xb_j t_{kl} - (\alpha_i xa_i xb_j + \alpha_k^o t_{kl}) w1_{ik}. \qquad (12)$$

In this equation it is considered that the $(i,\, k)$th hidden A-cell and $(j,\, l)$th hidden B-cell are conjunctively connected to the kth A-cell and lth B-cell in the output layer. The activation values originally received by the hidden B-cells are considered to be exactly the same as the input B-cells in the transformed space. Note that, in (12), no subscript j and l have been used in the left side which indicates that the weights are copied over all positions. t_{kl} is the desired output of the kth object at the lth location in the B-cells. δ_{kl} is given as

$$\delta_{kl} = \frac{t_{kl} - va_k vb_l}{\gamma g'(ubo_l)}. \qquad (13)$$

Here the activation of the A-cell is not considered in the denominator since the activation of the A-cell is unity if the object is present. In the learning process, if it is found that the A-cell corresponding to kth object is absent then it is allocated and necessary connections are made. Similarly, in the hidden layer, corresponding to the new feature-object pairs, A-cells are allocated. The weights from the hidden layer to the output layer are set to a small value. Therefore, after the first part of learning, hidden B-cells within the purview of the kth object would receive some activations and therefore it will be always active.

The updating of the top-down links are given as

$$\frac{dz1_{ki}}{dt} = \alpha_k^o t_{kl}(xa_i xb_j - z1_{ki}). \qquad (14)$$

α is the agility factor of the nodes which provides an approximate measure of how long the node has been active during the training phase. The updating of $\alpha_i$is given as

$$\frac{d\alpha_i}{dt} = -(\alpha_i)^2 xa_i xb_j. \qquad (15)$$

Here, it is to be noted that agility factors are attributes of the A-cells only and it is updated by the activation of the corresponding B-cells. The B-cell activations are received by the A-cells through stimulated type 3 links. The updating of α_k^o is given as

$$\frac{d\alpha_k^o}{dt} = -(\alpha_k^o)^2 t_{kl}. \qquad (16)$$

The details of the derivation of the learning rules have been discussed in [34], [35]. In the second stage of learning also, the weights of the type 1 links are copied over other type 1 links representing the same feature-object combinations.

VI. Implementation of the Network

The effectiveness of the network has been demonstrated in learning and simultaneous recognition of multiple flat industrial objects (possibly occluded). The objects are presented with their identity, position and orientation. The position is specified by the coordinates of the centroid of the object. The orientation means the orientation of the of the principal axis of the object.

A. Representation of the Features

Different characteristic features/primitives like lines, edges, corners, holes, etc. can be used for the description of objects. In the present investigation, only corners are considered as the characteristic features. The significant interrelations among the corners can also be taken into consideration (Since the network learns the transformations from the feature reference frame to the object reference frame, the interrelations are not used in this work).

The silhouette images (512 × 512) of the objects are considered here. The image has been segmented using graylevel thresholding. The threshold was selected to be 90 (maximum gray value in the image was 255). The image has been smoothed by applying growing and shrinking operations on it. The boundaries of the objects are detected by checking the 4-neighborhood. The corner or break points on the boundary have been detected by using the divide and conquer strategy as developed by Han *et. al.* [41] (note that, the corners could have been detected by any other suitable algorithm).

Whenever a corner is detected, it is supposed to have a certain curvature or cornerity value and a direction. The curvature value depends on the angle of the corner. Depending on the cornerity value a corner is encoded into a particular feature. The cornerity value at a certain break point is measured as the angle between the two lines joining the two neighboring break points on either sides along the boundary. For example, let X be a break point where the cornerity value is to be measured. Let Y and Z be the breakpoints first encountered when the boundary is scanned clockwise and anticlockwise, respectively, starting from X. In that case the cornerity value at X is the angle between the lines XY and XZ, and the direction of the corner is the direction of the bisector of the angle YXZ.

B. Encoding of the Features

The input image (512×512) is spatially divided into 64×64 grids so that each grid contains 8 × 8 pixels. In each grid location at most a single feature is allowed to be present. The actual grid size depends on the nature of the image. In each grid location a number of input nodes is present. Each input node represents a particular encoded feature at that particular location. First, let us consider the way of encoding the features. The entire range of cornerity values is divided into a number of slots. Each slot is considered to be a separate feature. The corners are divided into slots of nine degrees so that there will be 40 different input features. In the present scheme all the corners within a slot will be treated as the same feature. The slots of corners are

$$(0-9)(9-18)\cdots(342-351)(351-360)$$

i.e., a feature can be written as

$$c_i = \begin{cases} 1 & \text{if } i = \lfloor angle/9 \rfloor \\ 0 & \text{otherwise} \end{cases}$$

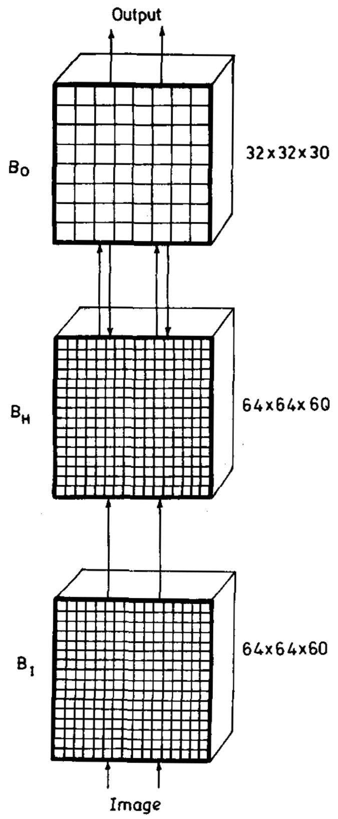


Fig. 10. The actual grid size for B-cells of the network. B_I, B_H, B_O are the input, hidden, and output layers, respectively.

where *angle* denotes the angle of the corner. c_i represents if the corresponding feature is present. If c_i is unity then xa_i (activation level of ith input A-cell) is unity (we consider that the ith feature is mapped to ith input A-cell).

C. Representation of Locations

The 512 × 512 image is divided into 64 × 64 × 60 grids and each grid is mapped to an input B-cell. Thus each input B-cell has a tolerance of 8 × 8 pixels and six degrees in orientation. The hidden layer also has 64 × 64 × 60 grids to represent the input-output association. The output layer is divided into 16 × 16 × 15 grids. Each grid location contains one output node, and in the junction of six neighboring output nodes another node is placed. As a result, each output node has a tolerance of 32 × 32 pixels spatially and 24 degrees in orientation. Each output node is connected to a group of 64 hidden nodes. This indicates that the network is able to accommodate at most 64 feature-object associations for any output node. In other words, the network is able to learn and recognize those objects which have less than 64 features. The input, hidden, and output grids of B-cells have been presented in Fig. 10.

D. Training and Testing

Four different objects (as shown in Figs. 11–15) have been considered. During the training phase of the system, each object has been presented to the network in different positions and orientations. In the present case four different instances of each object have been presented. The value of γ has been taken as 0.15. The time step for each learning trial has been selected as 0.05. After every 200 iterations the nodes have been flushed, i.e., the agility factors (Section V-B) of all the nodes were set to unity after every 200 trials. This causes the network to revive the learning capability for new situations. After 3000 trials the weights were found to change their value by less than 0.0005, and the training phase was terminated. After 3000 trials the network was found to consist of 288 type 1 links from the hidden layer to the output layer.

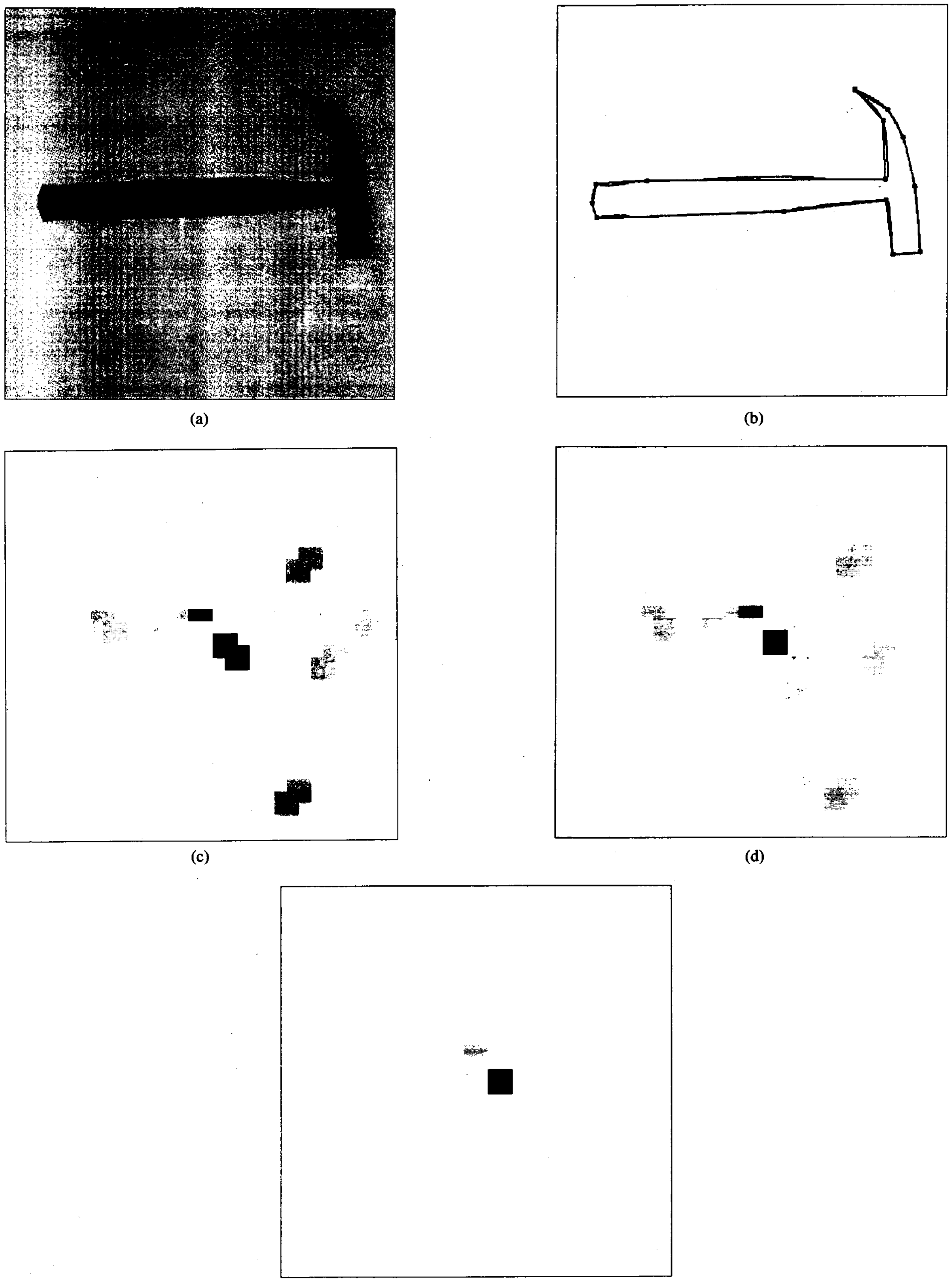
TABLE I
OUTPUTS OF THE SYSTEM AFTER 300 ITERATIONS

case	object	output	X	Y	θ
1 (Fig.11)	1	1.0	256.0	288.0	72.0
	2	0.0	-	-	-
	3	0.0	-	-	-
	4	0.0	-	-	-
2 (Fig.12)	1	0.0	-	-	-
	2	1.0	256.0	256.0	168.0
	3	0.0	-	-	-
	4	0.0	-	-	-
3 (Fig.13)	1	1.0	288.0	224.0	264.0
	2	1.0	208.0	240.0	60.0
	3	0.0	-	-	-
	4	0.0	-	-	-
4 (Fig.14)	1	1.0	240.0	176.0	36.0
	2	0.0	-	-	-
	3	1.0	256.0	352.0	24.0
	4	0.0	-	-	-
5 (Fig.15)	1	0.0	-	-	-
	2	1.0	224.0	192.0	336.0
	3	0.0	-	-	-
	4	1.0	208.0	272.0	12.0

In the recognition process the time step was considered as 0.1. The self-feedback was 0.05. Figs. 11–15 show the results for single and multiple objects for a different number of iterations. Table I shows the results of the five different instances of the objects (both single and mixed or overlapped) after 300 iterations. Case 1 and 2 correspond to the single instances of objects presented in Figs. 11 and 12. Cases 3, 4, and 5 correspond to the overlapped instances presented in Figs. 13–15. In Table I, (X, Y) represent the position, and θ represents the orientation of the objects.

E. Number of Nodes

In the input layer, 40 A-cells are used to represent the 40 slots of corners (features). The output layer has four A-cells to represent the output objects. The number of A-cells in the hidden layer depends on the type of objects presented. If there exists sufficient overlap between the patterns of the objects then the number of hidden nodes will decrease. It was found [42] that the number of hidden nodes is approximately proportional to $m + n$ where m is the number of objects and n is the number of features, for a sufficiently large number of objects. The input layer contains a B-cell in each grid location, i.e., 64 × 64 × 60 B-cells or approximarely 2^{18} B-cells. The hidden layer also contains a B-cell in each grid



(a)

(b)

(c)

(d)

(e)

Fig. 11. (a) Image of a hammer. (b) Feature map of the image. (c) Activations in the output layer (B-cells) after initialization process. (d) Activations in the output layer (B-cells) after 50 iterations. (e) Activations in the output layer (B-cells) after 300 iterations.

and the total number of B-cells in hidden layer is also 2^{18} approximately. The output layer uses the structure shown in Fig. 8(a). The number of B-cells in the output layer is therefore $16 \times 16 \times 15 + 15 \times 15 \times 14$ i.e., 2^{13} approximately. The total

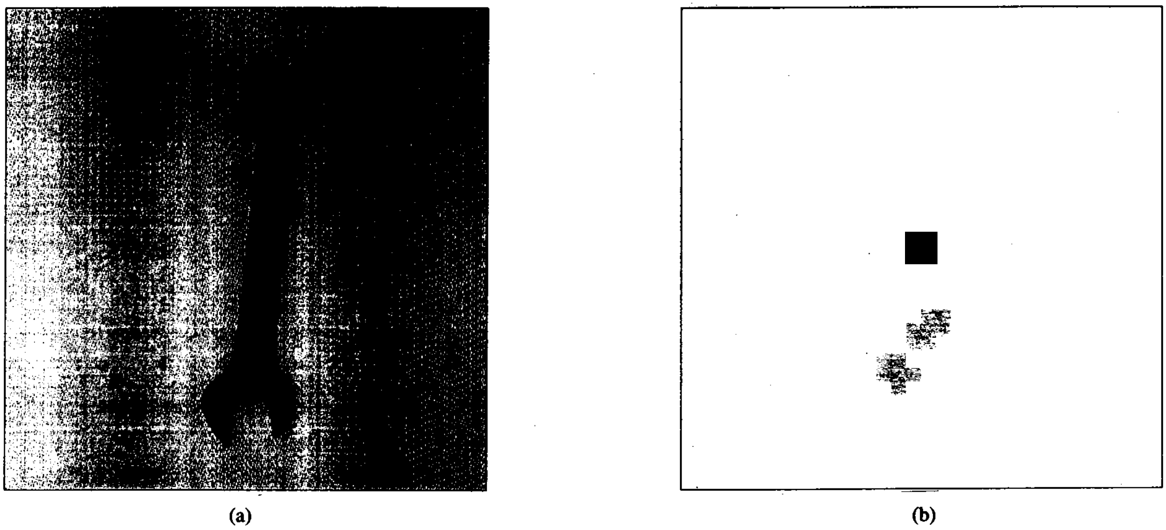


Fig. 12. (a) Image of a spanner. (b) Activations in the output layer (B-cells) after 300 iterations.

number of cells is therefore 2^{19} approximately, i.e., 5×10^5 approximately.

VII. Robustness of the Scheme

A. Noise Tolerance and Stability

The connectionist system presented here can take care of the noise present in the image level. Due to the presence of noise, a corner can change its curvature value from one slot to another, and as a result, the output of the corresponding object may degrade. It is intuitive that if a feature exhibits its variation during the training phase, then the network should be able to capture the variations without much affecting the output in the recognition process.

To mathematically model the noise redundancy, a feature is considered to have a distribution around its mean value. A feature (at any instant) can be represented as $\delta(c - c_i)$ in the analog domain where c_i is the mean value of the ith feature. The function $\delta(\cdot)$ is the Dirac-delta function (note that the noise redundancy should ideally be treated in discrete domain, but for the sake of simplicity, we have considered it in the analog domain).

Here, the value of the feature should not be confused with the confidence value of the feature. The feature value indicates which particular slot has been fired. The features (e.g., corners in the present model) are encoded in such a way that if the angle of the corner changes the feature value will change. The shift is dependent on the amount of change in angle.

The learning algorithms are designed in such a way [34] that the weights of the top-down links pick up the distribution of the features. The weights of the top-down link for the ith feature[2] can be represented as

$$z1_i(c, x, y, \xi) = p_i(c). \tag{17}$$

The second subscript in $z1$ is omitted to represent any arbitrary object. Since according to the learning rules, the weights of the bottom-up links are proportional to that of the top-down links (11), the distribution of the weights of the bottom-up links can be represented as

$$w1_i(c) = w_i p_i(c). \tag{18}$$

The output under stability is given as

$$w_s o = \sum_i \int_c w1_i(T_i C_i - z1_i o)\, dc \tag{19}$$

where o is used as the output (instead of the symbol vb). vb is the actual output where each output B-cell has a nonlinear transfer function. For the sake of simplicity we have assumed it to be linear, and a different notation is used (output of the corresponding A-cell is considered as unity). Here we consider that the activated hidden nodes are within the purview of the output cell (i.e., $w2$ and $z2$ are unity for all features). The transformational matrix is represented as T_i (obtained with the help of $W1$). C_i represents the ith feature along with its coordinates (x, y, ξ).[3] Since we are not considering the effect of positional and orientational variance, $T_i C_i$ can be represented as (or replaced by)

$$T_i C_i = \delta(c - c_i).$$

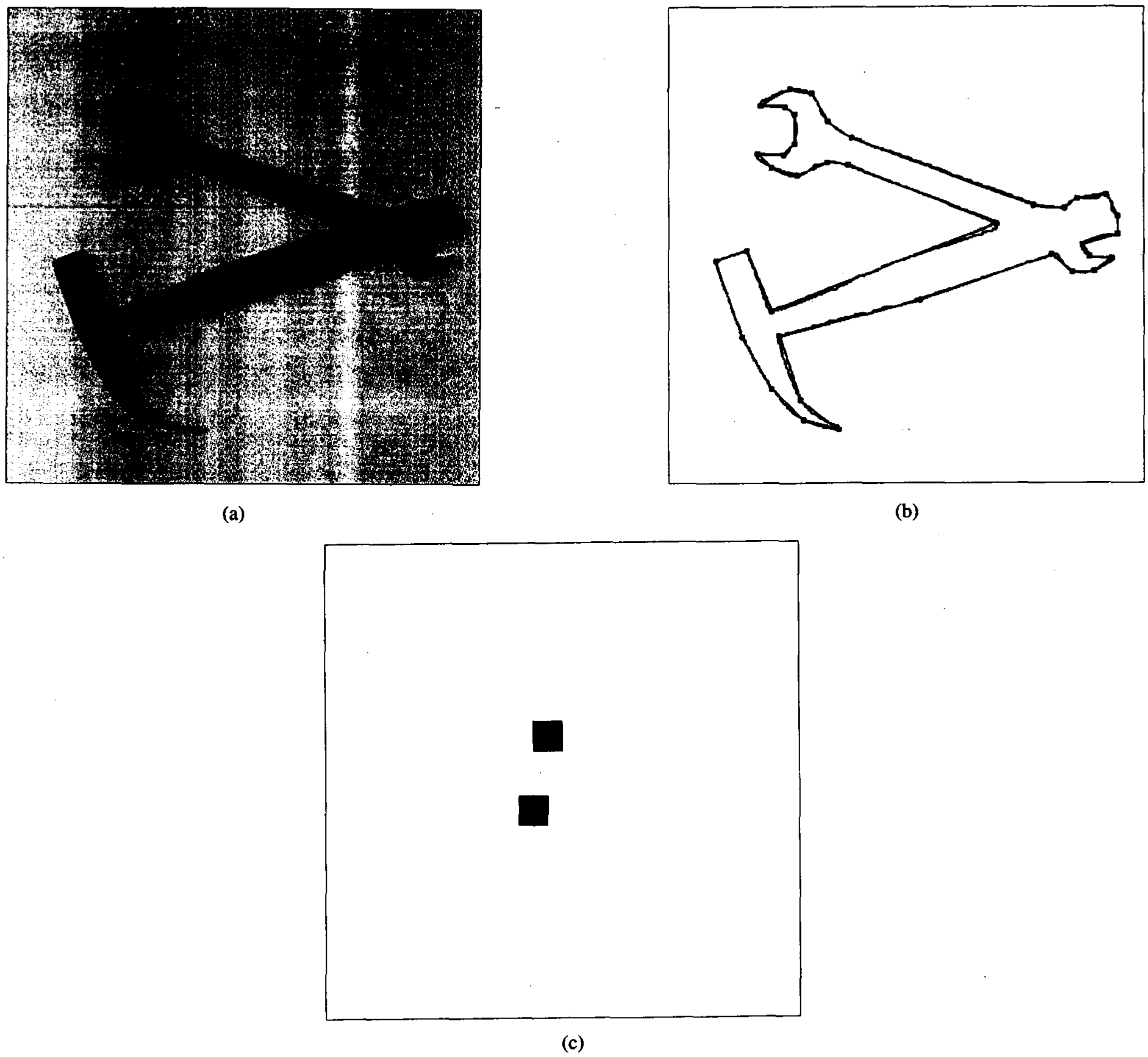
By algebraic manipulation, the output can be written as

$$o = \frac{\sum_i w_i \int_c \delta(c - c_i) p_i(c)\, dc}{w_s + \sum_i w_i \int_c p_i^2(c)\, dc}. \tag{20}$$

Considering the nonoverlapping distribution for each feature, a Gaussian distribution for the weights of the top-down

[2] $z1$ values pick up the feature value c, position (x, y), and orientation ξ.

[3] The integration is performed to represent that the feature values are distributed in analog domain.


(a)

(b)

(c)

Fig. 13. (a) Image of a hammer and a spanner overlapping each other. (b) Feature map of the image. (c) Activations in the output layer (B-cells) after 300 iterations. Note that the activation values have been thresholded. The threshold is selected as 0.3.

links corresponding to each feature, $p_i(c)$ can be written as

$$p_i(c) = \frac{1}{\sqrt{2\pi}\sigma_c} \exp\left(-\frac{(c - c_i)^2}{2\sigma_c^2}\right) \tag{21}$$

where σ_c is the variance of the feature value around the mean c_i for the ith feature under noisy environment.

Under a noiseless, ideal situation the feature values are expected to be the same as their mean values, i.e., the ith feature for a particular object will have the feature value c_i. Therefore, under noiseless condition the output of the desired object would be

$$o = \frac{\dfrac{\sum\limits_i w_i}{\sqrt{2\pi}\sigma_c}}{w_s + \dfrac{\sum\limits_i w_i}{2\sqrt{\pi}\sigma_c}}. \tag{22}$$

If the value of w_s is small enough compared to the total weights of the bottom-up links coming to a particular object then the output becomes

$$o = \sqrt{2}.$$

Since the transfer function of each neuron is such that the output saturates and cannot go beyond unity, the output under noiseless condition will ideally saturate to unity.

Let the object be such that the features do not coincide with the mean values, and let the ith feature in the transformed space be given as

$$T_i C_i = \delta(c - c_i - \Delta c). \tag{23}$$

In that case, from (20) the output of the desired object would be

$$o = \sqrt{2}\left(1 - \frac{w_i}{\sum\limits_i w_i}\left(1 - \exp\left(-\frac{\Delta c^2}{2\sigma_c^2}\right)\right)\right). \tag{24}$$

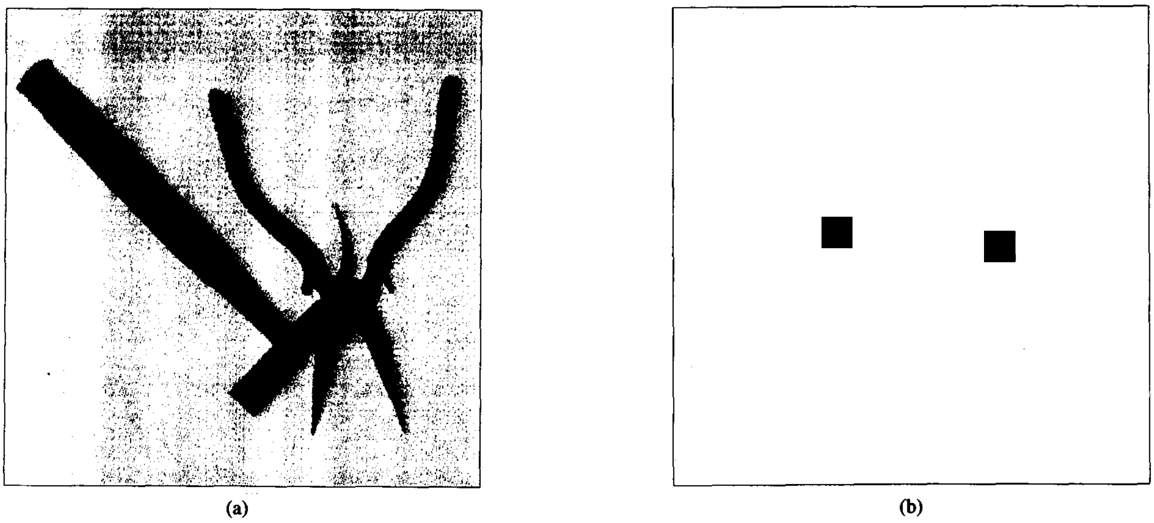


Fig. 14. (a) Image of a hammer and a plier overlapping each other. (b) Activations in the output layer (B-cells) after 300 iterations. Note that the activation values have been thresholded. The threshold is selected as 0.3.

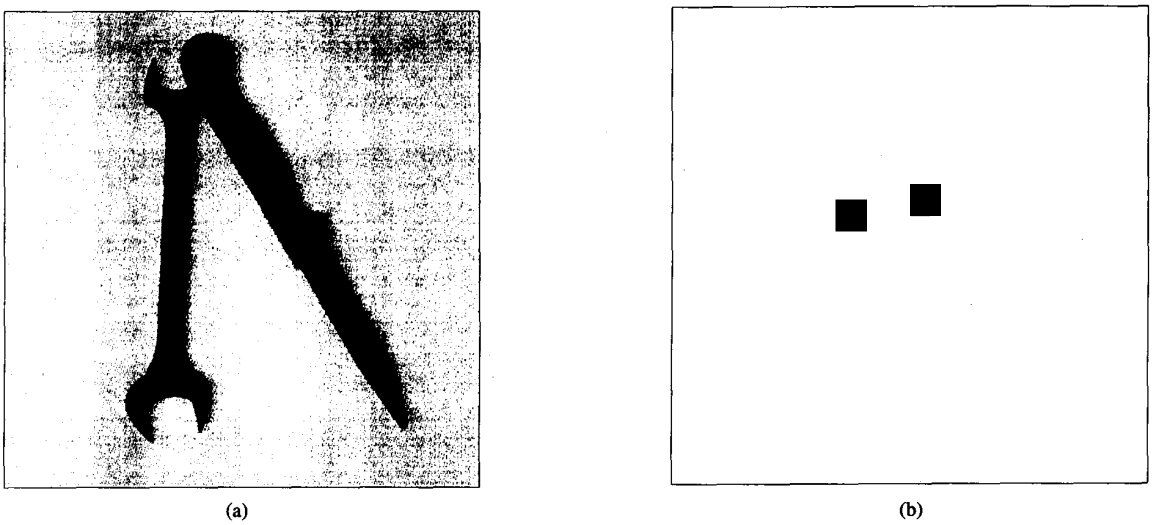


Fig. 15. (a) Image of a spanner and a knife overlapping each other. (b) Activations in the output layer (B-cells) after 300 iterations. Note that the activation values have been thresholded. The threshold is selected as 0.3.

Here, the effect of w_s has been neglected. Since the output saturates at unity, the effect of the shift will be perceived at the output depending on the shift in c. In this mathematical treatment, however, the confidence of a feature at any point has been modeled using Dirac-delta function which is not true in real life. Therefore, the noise degradation may not be so smooth as presented here.

The mathematical treatment basically reveals the fact that the noise degradation depends on the distribution of the feature during the learning process. If the feature suffers wide variation in the learning process (i.e., large σ_c), then the variation of that feature in the recognition process (i.e., Δc) does not cause much degradation in the output. In other words, no single instance of the feature is given great importance in the recognition process. On the other hand, if the feature does not suffer much deviation (i.e., small σ_c) in the learning trials, then in the recognition process if the feature suffers deviation (i.e., large Δc) then the output will be deteriorated.

B. Crosstalk and False Alarming

It was mentioned before that the links are stimulated by the modifiers attached to them. The modifiers will stimulate or deactivate certain links only when the signal coming through the modifier link is sufficiently high i.e., greater than certain threshold (ε). Let a particular combination of A- and B-cells be active in the input layer (say Ai1 and Bi1) and the type 12 link from the coalition of Ai1 and Bi1 stimulate the links connected to the coalition of Ah1, Bh1 in the hidden layer. Since the signals carried by the modifier links are sufficiently high over a neighborhood centered around Bh1, the links emanating from the coalitions of Ah1 and the neighborhood cells of Bh1 would also be stimulated. Therefore, even if the competition takes place over a neighborhood of Bh1, and only Bh1 wins, the links from the neighborhood of Bh1 remain stimulated. If the feature is such that it instantiates another feature-object pair in the neighborhood of Bh1 then a false coalition may be formed.

To prevent such crosstalk, the links are provided with a special attenuable LTM (ALTM). Initially all the links are not capable of carrying activations. If the links are modified by some stimulating modifier and the cells connected to the links are active then only the links retain their signal carrying capability. If the links are not modified or do not have any active cells connected to them then they lose their capability of carrying the activations. Mathematically, the decaying property is given as

$$\frac{dw}{dt} = -\frac{w}{\tau_w}. \tag{25}$$

The time constant τ_w can be selected depending on the rate of change of the zone of attention controlled by the ACN.

VIII. Conclusions and Discussion

A scheme for polygonal object recognition using the principles of connectionist computation has been presented here. Several concepts motivated by the psychological findings have been used here. The main contribution of the work is the introduction of the concept of separating the networks for identification and pose estimation (locating) of the objects. This very fact helps in reducing the total number of neurons to a great extent as compared to the other models [28], [29]. Nonetheless, there is not much gain as far as the number of links is concerned.

The model also uses the theory of selective attention for the initial hypotheses formation. The attentional mechanism used here is early selection process. This pseudoparallel mechanism, used in the initial hypotheses formation, basically helps in the simultaneous recognition of multiple objects (both single and overlap instances).

The learning rules are able to quantify the relative importance values of the features with respect to objects. The system is also designed to learn the transformation values from the features to the objects and is proved to be tolerant of variations in feature values.

The analysis of robustness of the model shows a graceful degradation in its performance when a feature suffers noise in its value, position or orientation. The experimental results also support the fact that even under occlusion or variation of the features in their position, orientation or value, the system is able to recognize the objects present in the scene.

The system has been tested on polygonal objects using only corner features. The general 2-D shapes are not considered here. The system can possibly be extended to accept arbitrary shapes if suitable feature representation scheme can be devised.

The scale invariance can be incorporated in this model. With each feature a scale value can be associated depending on the distances of the other features from that particular feature. Whenever a feature activates a hidden node, the scale value propagates through the links and helps in activating the proper hidden node. The scale invariance can also be efficiently achieved if a hierarchical model is used where the input features instantiate subparts and the subparts activate the objects in turn.

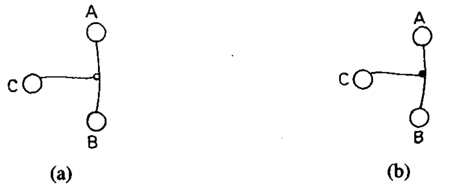


Fig. 16. (a) A stimulating modifier of the link connecting the nodes A and B, (b) A deactivating or inhibitory modifier of the link connecting the nodes A and B.

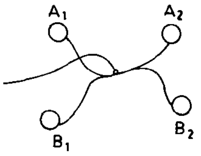


Fig. 17. The sigma-pi connection between two links emanating from the nodes A1 and B1. The link branches out to two other nodes A2 and B2.

Appendix

Two special type of connections are used to build the structured connectionist model. These are:

A: modifier of the links, and

B: sigma-pi connections of the links.

The concept of modifier of the links has been presented in [31]. Two types of modifiers, namely, stimulating and inhibitory modifiers, have been used in the present network [as shown in Fig. 16(a)]. The link emanating from node C stimulates [in Fig. 16(a)] or deactivates [in Fig. 16(b)] the link connecting the nodes A and B. In the first case, A is able to send a signal to B (or otherwise, i.e., B sends to A) only when the connecting link is stimulated by some positive signal from C through the modifier. In the second case, A is not able to send any signal to B (and also B is not able to send to A) so long as the connecting link is deactivated or inhibited by some positive signal from C through the inhibitory modifier, even if the connecting link is stimulated by some other modifier.

The concept of pi-connection was introduced in [39]. In this case, a node receives the product of more than one signal. For example, a node is connected to a group of links having weights $(w_{11}, \cdots, w_{1m}), \cdots, (w_{n1}, \cdots, w_{nm})$ such that every n-tuple of links have pi-connection. The links are connected to nodes which hold activation values given as $(x_{11}, \cdots, x_{1m}), \cdots, (x_{n1}, \cdots, x_{nm})$. In that case the total input received by the node (u) will be

$$u = \sum_{i=1}^{n} \prod_{j=1}^{m} w_{ij} x_{ij}.$$

Fig. 17 shows a typical case that we have used in the proposed model. Here, the links emanating from two nodes A1 and B1 get conjunctively or pi-connected and then branch out to two other nodes A2 and B2. The conjunctive connection also gets stimulated by another signal through the modifier.

The nodes A2 and B2 receives the weighted product of the outputs of A1 and B1 when the conjunctive connection is stimulated. The sigma-pi and the modifier connections play important roles in separating the two channels to represent the "what it is" and "where it is."

Acknowledgment

The authors gratefully acknowledge S. Chakraborty for drawing the diagrams, KBCS center for providing the computing facility, and Drs. S. Chaudhury and C. A. Murthy for their helpful discussion. Suggestions from Dr. B. Chanda and D. K. Banerjee are also acknowledged.

References

[1] P. Suetens, P. Fua, and A. Hanson, "Computational strategies for object recognition," *ACM Comput. Surveys*, vol. 24, pp. 5–61, 1992.

[2] A. Wallace, "A comparison of approaches to high-level image interpretation," *Pattern Recognition*, vol. 21, pp. 241–259, 1988.

[3] N. Ayache and O. Faugeras, "Hyper: A new approach for the recognition and positioning of two-dimensional objects," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-8, pp. 44–54, 1986.

[4] J. L. Turney, T. Mudge, and R. A. Volz, "Recognizing partially occluded parts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-7, pp. 410–421, 1985.

[5] B. Bhanu and J. Ming, "Clustering based recognition of occluded objects," in *Proc. 8th Int. Conf. Pattern Recognition*, Paris, 1986, pp. 732–734.

[6] T. Henderson and A. Samal, "Multiconstraint shape analysis," *Image Vision Comput.*, vol. 4, pp. 84–96, 1986.

[7] R. C. Bolles and R. A. Cain, "Recognizing and locating partially visible objects," in *Robot Vision*, A. Pugh, Ed. Berlin: Springer, 1983.

[8] M. Eshera and K. Fu, "A similarity measure between attributed relational graphs for image analysis," in *Proc. 7th Int. Conf. Pattern Recognition.*, 1984, pp. 75–77.

[9] D. Ballard, "Parameter nets," *Artificial Intell.*, vol. 22, pp. 235–267, 1984.

[10] ——, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, pp. 111–122, 1981.

[11] S. M. Kosslyn, "Information representation in visual images," *Cognitive Psychology*, vol. 7, pp. 341–370, 1975.

[12] S. Kosslyn, J. Holtzman, M. Farah, and M. Gazzaniga, "A computational analysis of mental image generation: Evidence from functional dissociations in split-brain patients," *J. Experimental Psych.: General*, vol. 114, pp. 311–341, 1985.

[13] L. Ungerleider and M. Mishkin, "Two cortical visual systems," in *Analysis of Visual Behavior*, D. Ingle, M. Goodale, and R. Mansfield, Eds. Cambridge, MA: MIT Press, 1982.

[14] J. Duncan and G. Humphreys, "Visual search and stimulus similarity," *Psych. Rev.*, vol. 96, pp. 433–458, 1989.

[15] D. LaBerge and V. Brown, "Theory of attentional operations in shape identification," *Psych. Rev.*, vol. 96, pp. 101–124, 1989.

[16] R. Phaf, A. Heijden, and P. Hudson, "Slam: A connectionist model for attention in visual selection tasks," *Cognitive Psych.*, vol. 22, pp. 273–341, 1990.

[17] R. Shiffrin and W. Schneider, "Controlled and automatic human information processing: ii, perceptual learning, automatic attending, and a general theory," *Psych. Rev.*, vol. 84, pp. 127–190, 1977.

[18] C. Bundsen, "A theory of visual attention," *Psych. Rev.*, vol. 97, pp. 523–547, 1990.

[19] K. Fukushima, "Cognitron: A self-organizing multilayered neural network," *Biol. Cybern.*, vol. 20, pp. 121–136, 1975.

[20] K. Fukushima and S. Miyake, "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position," *Pattern Recognition*, vol. 15, pp. 455–469, 1982.

[21] K. Fukushima, "Neural network model for selective attention in visual pattern recognition and associative recall," *Appl. Opt.*, vol. 26, pp. 4985–4992, 1987.

[22] K. Fukushima and T. Imagawa, "Recognition and segmentation of connected characters with selective attention," *Neural Networks*, vol. 6, pp. 33–41, 1993.

[23] G. Himes and R. Iñigo, "Automatic target recognition using a neocognitron," *IEEE Trans. Knowledge Data Eng.*, vol. 4, pp. 167–172, 1992.

[24] G. Hinton, "Shape representation in parallel systems," in *Proc. IJCAI-81, Int. Joint Committee Artificial Intell.*, 1981.

[25] ——, "A parallel computation that assigns cannnonical object-based frames of reference," in *Proc. IJCAI-81, Int. Joint Committee Artificial Intell.*, 1981.

[26] M. Mozer, *The Perception of Multiple Objects: A Connectionist Approach.* Cambridge, MA: MIT Press, 1991.

[27] M. Mozer and M. Behrmann, "On the interaction of selective attention and lexical knowledge: A connectionist account of neglect dyslexia," Dep. Comput. Sci., Univ. Colorado, Boulder, Tech. Rep. CU-CS-441-89, 1989.

[28] R. S. Zemel, M. C. Mozer, and G. E. Hinton, "Traffic: A model of object recognition based on transformation of feature instances," Univ. Toronto, Canada, Tech. Tep. CRG-TR-7, 1988.

[29] R. Zemel, "Traffic: A connectionist model of object recognition," Dep. Comput. Sci., Univ. Toronto, Canada, Tech. Rep. CRG-TR-89-2, 1989.

[30] J. A. Feldman, "Dynamic connections in neural networks," *Biol. Cybern.*, vol. 46, pp. 27–39, 1982.

[31] J. A. Feldmann and D. H. Ballard, "Connectionist models and their properties," *Cognitive Sci.*, vol. 6, pp. 205–254, 1982.

[32] J. Feldman, "Four frames suffice: A provisional model of vision and space," *Behavioral Brain Sci.*, vol. 8, pp. 265–289, 1985.

[33] D. Sabbah, "Computing with connections in visual recognition of origami objects," *Cognitive Sci.*, vol. 9, pp. 25–50, 1985.

[34] J. Basak, C. A. Murthy, S. Chaudhury, and D. Dutta Majumder, "A connectionist model category perceptron: Theory and implementation," *IEEE Trans. Neural Networks*, vol. 4, pp. 257–269, 1993.

[35] J. Basak, S. K. Pal, "X-tron: An incremental connectionist model for mixed category perception," *IEEE Trans. Neural Networks*, vol. 6, no. 5, 1995.

[36] S. Grossberg, "Competitive learning: From interactive activation to adaptive resonance," *Cognitive Sci.*, vol. 11, pp. 23–63, 1987.

[37] C. Stanfill and D. Waltz, "Toward memory-based reasoning," *Commun. ACM*, vol. 29, 1986, pp. 1213–1228.

[38] J. Hertz, A. Krogh, and R. G. Palmer, *Introduction to the Theory of Neural Computation.* Redwood City, CA: Addison-Wesley, 1991.

[39] D. E. Rumelhart and J. L. McClelland, *Parallel Distributed Processing: Explorations in Microstructures of Cognition (Ed.)*, vol I. Cambridge MA: Bradford Books/MIT Press, 1986.

[40] G. A. Carpenter and S. Grossberg, "A massively parallel architecture for a self-organizing neural pattern recognition machine," *Comput. Vision, Graphics, Image Process.*, vol. 37, pp. 34–115, 1987.

[41] M. H. Han, D. Jang, and J. Foster, "Identification of cornerpoints of two-dimensional images using a line search method," *Pattern Recognition*, vol. 22, pp. 13–20, 1989.

[42] J. Basak, C. A. Murthy, and S. K. Pal, "A self-organizing network for mixed category perception," *Neurocomput.*, accepted.

Jayanta Basak, for a photograph and biography, see p. 1108 of the September 1995 issue of this Transactions.

Sankar K. Pal, for a photograph and biography, see p. 63 of the January 1995 issue of this Transactions.