OBJECT RECOGNITION USING MULTI-LAYER HOPFIELD NEURAL NETWORK

Susan S. Young, Peter D. Scott and Nasser M. Nasrabadi
Department of Electrical & Computer Engineering
State University of New York at Buffalo
Amherst, New York 14260

To appear, Proc. Int. Conf. on Comp. Vision and Pat. Rec. (CVPR '94)

Abstract

An object recognition approach based on concurrent coarse-and-fine matching using a multi-layer Hopfield neural network is presented. The proposed network consists of several cascaded single layer Hopfield networks, each encoding object features at a distinct resolution, with bidirectional interconnections linking adjacent layers. The interconnection weights between nodes associating adjacent layers are structured to favor node pairs for which model translation and rotation, when viewed at the two corresponding resolutions, are consistent. This inter-layer feedback feature of the algorithm reinforces the usual intra-layer matching process in conventional single layer Hopfield nets in order to compute the model-object match which is most consistent across several resolution levels. The performance of the algorithm is demonstrated in cases of images containing single and multiple occluded objects. These results are compared with recognition results obtained using a single layer Hopfield network.

1 Introduction

Object recognition has emerged as a subject of wide research interest during the last decade[1]. Two common themes characterizing much of the recent work have been the use of a priori information in the form of models and constraints, and the incorporation of the most current image processing tools to enhance recognition performance. In this spirit, the objective of the present study is to explore the use of multi-resolution (pyramidal) image representation in the context of recently reported neural network implementation technology, with the goal of faster and more robust automated object recognition performance.

It is natural to seek object recognition cues concurrently at several resolution levels. Multi-resolution image representation and processing is a well known image analysis methodology. A multi-resolution image representation can be viewed as an image pyramid. Important classes of image pyramids include the Gaussian pyramid, Laplacian pyramid, and subband pyramid [2].

The most immediate utility of a multi-resolution pyramid representation is that it can reduce the computational cost of various image search operations. A major problem associated with this hierarchical strategy is that if a mistake occurs at an early stage, the low resolution error will propagate into each subsequent higher resolution level and finally a mismatch would occur. This mismatch could not be corrected by using the information at any level because the information flows top-down in a feed-forward manner and there is no feedback from higher resolution levels. To address this problem, a technique called "coarse-and-fine" matching is proposed in this paper, where top-down and bottom-up matching are concurrently performed for each pair of levels of the image pyramid in order to find the best matched features at each level pair simultaneously.

The proposed coarse-and-fine strategy is implemented by utilizing a multi-layer Hopfield neural network. The single layer Hopfield neural network from which it derives has been used in a wide range of applications, such as vision tasks. Vision tasks can be formulated as an optimization problem where an energy function is minimized. The search for its global minimum can be implemented through a Hopfield neural network with interconnection weights generating an equivalent energy function. Unfortunately, there typically exist multiple local minima in such energy functions due to its non-convexity and its argument high dimensionality, and a gradient descent procedure is vulnerable to early termination. The Hopfield network can get trapped in any of these local minima depending on the initial states of the network and the way it selects the sequence by which the states of the neurons are updated.

In this paper, a concurrent (coarse-and-fine) multi-resolution model-based object recognition technique is proposed using a multi-layer Hopfield neural network to alleviate some of these problems which arise when a single layer Hopfield network is utilized. The network is structured as a cascade of several single layer Hopfield networks with interconnections between adjacent layers. It uses matching results in each layer to rein-

[1]This work was partially supported by Amherst Systems, Inc., 30 Wilson Road, Buffalo NY 14221-7082, under Office of Naval Research contract number N00014-91-C-0257.

force the matching process for adjacent layers through inter-layer interconnection weights. The values of the weights between nodes on distinct layers depend on the intrinsic characteristics of the multi-resolution representation, that is, the relationship among the multi-resolution features belonging to adjacent levels of the image pyramid. Each layer of the network implements a matching process between the scene and the model features which are extracted at the corresponding resolution level of the image pyramid. However, each layer of the proposed network communicates with adjacent layers permitting inter-layer feedback during matching. Thus good matches at multiple levels reinforce one another, and matches at one level which are not corroborated at adjacent resolution levels do not propagate as strongly. Moreover, the equivalent energy function for the multi-layer Hopfield network is shown to be smoothed relative to the single layer case, mitigating the local minima problem, and the examples are shown to converge to local minima reasonably close to the global minimum.

This paper is organized as follows. In Section 2, scene and model pyramids are discussed. In Section 3, the multi-layer Hopfield neural network is introduced. In Section 4, the performance of this network is compared with that of a single layer Hopfield neural network for recognizing image scenes containing single objects and multiple occluded objects. Conclusions is given in Section 5.

2 Scene and Model Pyramid

In order to implement the matching based on a multi-layered Hopfield network at multiple resolution levels of images, first an image pyramid is constructed for each model. A QMF filter [2] is employed in this paper to build the subband pyramids for each model and the input scene. The feature primitives that are utilized in this paper are the high curvature points (corners). Therefore, a polygon approximation algorithm [3] is used on the boundaries of objects to obtain the corners (vertices) at each level. The numerical features quantifying a vertex are the angle between the two polylines that form the vertex and the location of the vertex. A set of graphs are generated for each 2-D prototype object, where each graph consists of a set of corners with their corresponding angle features and distance features at a particular level of the pyramid, we call this representation the *model graph pyramid*. All the model graph pyramids are then integrated into a single model-database, which is called a *global model graph pyramid*. Similarly, a graph pyramid can be generated for an input scene which is called a *scene graph pyramid*. During recognition, the scene graph pyramid is matched against the global model graph pyramid by a multi-layer Hopfield neural network to identify and locate the instances of the models in the test scene for each level of the pyramid.

3 Multi-layer Hopfield Network

3.1 Construction

A multi-layer Hopfield network consisting of several single layer Hopfield networks cascaded together is shown in Fig. 1. Inputs to each layer are the features extracted from the corresponding level of the model and scene pyramids. The nodes within each layer are fully connected. The adjacent layers of the multi-layered network are connected by a set of interconnection weights. For the remainder of this paper, we will restrict ourselves to a two-layered Hopfield neural network where the fine features are matched in the first layer L_1 and the coarse features are matched in the second layer L_2, with appropriate interconnection weights.

3.2 Energy Function

To consider the behaviour of a two-layered Hopfield network, let the state of the network in layer L_1 be represented by a binary state vector $\underline{A}_1$, the state of the second layer L_2 be the state vector $\underline{A}_2$ and the state vector for the entire two-layered network be denoted by

$$\underline{A} = [\underline{A}_1, \underline{A}_2] \tag{1}$$

where the entire state vector is the concatenation of the state vectors of the two layers.

The overall energy function representing the collective behaviour of the two-layered network can be characterized by the following energy function

$$E(\underline{A}) = E_1(\underline{A}_1) + \alpha_1 E_{12}(\underline{A}) + E_2(\underline{A}_2) + \alpha_2 E_{21}(\underline{A}) \tag{2}$$

where $E_1(\underline{A}_1)$ is the energy due to the current state of the layer L_1, $E_2(\underline{A}_2)$ is the energy due to the current state of the layer L_2, $E_{12}(\underline{A})$ and $E_{21}(\underline{A})$ are the inter-energy between the state of the layer L_1 and the state of the layer L_2 and vice versa, α_1 is a parameter that weights the inter-energy $E_{12}(\underline{A})$ relative to the energy $E_1(\underline{A}_1)$, and α_2 is a parameter that weights the inter-energy $E_{21}(\underline{A})$ relative to the energy $E_2(\underline{A}_2)$. α_1 and α_2 can also be considered as Lagrange multipliers.

The behaviour of the network at layer L_1 can be represented by an energy function given by

$$\psi_1(\underline{A}) = E_1(\underline{A}_1) + \alpha_1 E_{12}(\underline{A}) \tag{3}$$

Similarly, the energy function for layer L_2 can be written as

$$\psi_2(\underline{A}) = E_2(\underline{A}_2) + \alpha_2 E_{21}(\underline{A}) \tag{4}$$

Our multi-resolution matching process is defined as minimizing the overall energy function $E(\underline{A})$ over the entire domain of state vector $\underline{A}$ in order to obtain the best matches for each layer. To minimize the overall energy function $E(\underline{A})$, a node in layer L_1 is randomly picked and its state is updated, then a node in layer L_2 is randomly chosen and its state is updated. This process is repeated recursively by choosing a node from layer L_1 followed by a node from layer L_2.

3.3 Connection Weights in Each Layer

The matching process in each layer can be formulated as minimizing an energy function [1]. For example, considering the matching process at layer L_1, and assuming no interaction with layer L_2, an energy function that includes all the constraints of the matching

process can be written as:

$$E_1 = -\frac{1}{2}\sum_{i_1=1}^{M_1}\sum_{k_1=1}^{N_1}\sum_{j_1=1}^{M_1}\sum_{l_1=1}^{N_1} C_{i_1k_1j_1l_1}\, V_{i_1k_1}\, V_{j_1l_1}$$

$$+\sum_{i_1=1}^{M_1}\left(1-\sum_{k_1=1}^{N_1} V_{i_1k_1}\right)^2 + \sum_{k_1=1}^{N_1}\left(1-\sum_{i_1=1}^{M_1} V_{i_1k_1}\right)^2 \tag{5}$$

where $V_{i_1k_1}$ represents the degree of match between two features i_1 and k_1. It takes a value of 1 when the i_1th feature point in the input image matches the k_1th feature point in the model, otherwise, it takes a value of 0. N_1 is the total number of features in the model graph which is the sum of all the features in the models. M_1 is the corresponding number of feature points for the input image in layer L_1. The first term in the energy function represents the compatibility constraint. The second and the third terms represent the uniqueness constraint, i.e., for each feature point there can only be one match. The compatibility measure $C_{i_1k_1j_1l_1}$ can be expressed as follows:

$$C_{i_1k_1j_1l_1} = \begin{cases} W_1F(f_{i_1}, f_{k_1})+ \\ W_2F(f_{j_1}, f_{l_1})+ \\ W_3F(d_{i_1j_1}, d_{k_1l_1}), & \text{for } k_1 \text{ and } l_1 \in \text{q} \\ -1, & \text{for } k_1 \text{ and } l_1 \notin \text{q} \end{cases} \tag{6}$$

where W_i are the weighting factors and sum up to 1 (i.e.,$\sum_{i=1}^{3} W_i = 1$). q is qth model. The function $F(x, y)$ is a discrete non-linear compatibility function which was defined in [1], such that if x and y are compatible then $F(x, y)$ has a value 1; otherwise -1.

It has been shown in [1] that the energy function E_1 in (5) above is equivalent to the Hopfield-style energy function [4] given by

$$E_1 = -\frac{1}{2}\sum_{i_1=1}^{M_1}\sum_{k_1=1}^{N_1}\sum_{j_1=1}^{M_1}\sum_{l_1=1}^{N_1} T_{i_1k_1j_1l_1}\, V_{i_1k_1}\, V_{j_1l_1}$$

$$-\sum_{i_1=1}^{M_1}\sum_{k_1=1}^{N_1} I_{i_1k_1}\, V_{i_1k_1} \tag{7}$$

where

$$T_{i_1k_1j_1l_1} = C_{i_1k_1j_1l_1} - 2\delta_{i_1j_1} - 2\delta_{k_1l_1}, \tag{8}$$

$$I_{i_1k_1} = 4, \tag{9}$$

and $\delta_{i_1j_1} = 1$ if $i_1 = j_1$ and 0 otherwise, and similarly $\delta_{k_1l_1} = 1$ if $k_1 = l_1$ and 0 otherwise. $T_{i_1k_1j_1l_1}$ represents the connection weight between a node at (i_1, k_1) and a node at (j_1, l_1) within layer L_1.

3.4 Interconnection Weights between the Layers

At each layer, the matched features for each model can be used to find the mapping (translation and rotation parameters) between the model and the corresponding object in the input scene. The interconnections between adjacent layers of the network are based on the relationship between the mapping parameters of the models obtained at each layer. The relationship can be summarized as follows: consistent translation parameters of a model at layer L_1 are twice the translation parameters of the same model at layer L_2, and consistent rotation parameters at layer L_1 and L_2 must be the same. Therefore, the relationship between the mappings is

$$\begin{cases} \underline{tx}_{L_1} = 2 \times \underline{tx}_{L_2} \\ \underline{ty}_{L_1} = 2 \times \underline{ty}_{L_2} \\ \underline{\theta}_{L_1} = \underline{\theta}_{L_2} \end{cases} \tag{10}$$

where $(\underline{tx}_{L_1}, \underline{ty}_{L_1})$ and $(\underline{tx}_{L_2}, \underline{ty}_{L_2})$ are the translation parameters of models at layer L_1 and L_2, respectively, and $\underline{\theta}_{L_1}$ and $\underline{\theta}_{L_2}$ are the rotation parameters of models at layer L_1 and L_2, respectively. The calculation of $\underline{tx}_{L_1}, \underline{ty}_{L_1}$ and $\underline{\theta}_{L_1}$ are obtained as the average of the translation and rotation of each matched pair nodes in [5].

Using this consistency constraint between the two layers, we can define the interconnection weights between the two layers of the network. The interconnection weight $B_{i_1k_1i_2k_2}$, which is the connection between a node (i_1, k_1) in layer L_1 and a node (i_2, k_2) in layer L_2, is defined as

$$B_{i_1k_1i_2k_2} = \begin{cases} 1, & \text{if} \begin{cases} k_1, k_2 \in \text{model q} \\ [|\underline{tx}_{L_1} - 2\underline{tx}_{L_2}| \\ +|\underline{ty}_{L_1} - 2\underline{ty}_{L_2}|] \le \varepsilon_1 \\ |\underline{\theta}_{L_1} - \underline{\theta}_{L_2}| \le \varepsilon_2 \end{cases} \\ -1, & \text{otherwise} \end{cases} \tag{11}$$

where ε_1 and ε_2 are pre-specified thresholds. The interconnection weights between layer L_1 and L_2 are symmetrical. Fig. 2 shows the interconnection weights between the two layers for two nodes belong to the same model or to different models.

The inter-energy E_{12} and E_{21} are equal. They can be noted as E_c which stands for the coupling energy between the two layers.

$$E_c = -\frac{1}{2}\sum_{i_1=1}^{M_1}\sum_{k_1=1}^{N_1}\sum_{i_2=1}^{M_2}\sum_{k_2=1}^{N_2} B_{i_1k_1i_2k_2} V_{i_1k_1} V_{i_2k_2}. \tag{12}$$

Since the inter-energy E_{12} and E_{21} are equal, we will use α for both α_1 and α_2 in the remainder of this paper. Hence, the overall energy for the network is

$$E = E_1 + E_2 + 2\alpha E_c. \tag{13}$$

It should be pointed out that the values for the interconnection weights are changing as the network is updated, because as the states of the nodes are changed and as more correct matches are obtained the calculated values for the translation and rotation will also be changed.

3.5 Rate of Change in Energy

Consider changing the state $V_{i_1k_1}$ of a neuron (i_1, k_1) in layer L_1, i.e., $V''_{i_1k_1}$ is changed to $V^{n+1}_{i_1k_1}$, the

change in the total energy is

$$\Delta E = \Delta E_1 + \Delta E_2 + 2\alpha \Delta E_c. \tag{14}$$

Here, $\Delta E_2 = 0$, since the states of the neurons in layer L_2 are not changed. The change of the energy E_1 is given by [1]

$$\Delta E_1 = -(U_{i_1k_1} - 2\Delta V_{i_1k_1})\Delta V_{i_1k_1} \tag{15}$$

where

$$U_{i_1k_1} = \sum_{j_1=1}^{M_1}\sum_{l_1=1}^{N_1}(C_{i_1k_1j_1l_1} - 2\delta_{i_1j_1} - 2\delta_{k_1l_1})V_{j_1l_1} + I_{i_1k_1}, \tag{16}$$

and

$$\Delta V_{i_1k_1} = V_{i_1k_1}^{n+1} - V_{i_1k_1}^{n}. \tag{17}$$

The change in coupling energy E_c can be written as

$$\Delta E_c = -\frac{1}{2}Z_{i_1k_1i_2k_2} + \frac{1}{2}W_{i_2k_2}\Delta V_{i_1k_1} \tag{18}$$

where

$$Z_{i_1k_1i_2k_2} = \sum_{i_2=1}^{M_2}\sum_{k_2}^{N_q^{L_2}} \quad \sum_{k_1,k_2 \in \text{model q}} V_{i_2k_2}$$

$$\times \sum_{i_1=1}^{M_1}\sum_{k_1}^{N_q^{L_1}} \quad \sum_{k_1,k_2 \in \text{model q}} (B_{i_1k_1i_2k_2}^{n+1}V_{i_1k_1}^{n+1} - B_{i_1k_1i_2k_2}^{n}V_{i_1k_1}^{n}) \tag{19}$$

$$W_{i_2k_2} = \sum_{i_2}^{M_2}\sum_{k_2}^{N_2 - N_q^{L_2}} \quad \sum_{k_1,k_2 \notin \text{model q}} V_{i_2k_2}, \tag{20}$$

$$N_1 = \sum_{q=1}^{Q} N_q^{L_1}, \quad N_2 = \sum_{q=1}^{Q} N_q^{L_2}, \tag{21}$$

Q is the total number of models, and

$$B_{i_1k_1i_2k_2}^{m} = \begin{cases} 1, & \text{if } \begin{cases} k_1, k_2 \in \text{the same model} \\ [|\underline{tx}_{L_1} - 2\underline{tx}_{L_2}| \\ +|\underline{ty}_{L_1} - 2\underline{ty}_{L_2}|]|V_{i_1k_1}^{m} \le \varepsilon_1 \\ |\underline{\theta}_{L_1} - \underline{\theta}_{L_2}|V_{i_1k_1}^{m} \le \varepsilon_2 \end{cases} \\ -1, & \text{otherwise} \end{cases} \tag{22}$$

$B_{i_1k_1i_2k_2}^{n}$ and $B_{i_1k_1i_2k_2}^{n+1}$ represent the interconnections when state of the neuron (i_1, k_1) is $V_{i_1k_1}^{n}$ or $V_{i_1k_1}^{n+1}$, respectively. Therefore, the change of total energy is

$$\Delta E = -(U_{i_1k_1} - \alpha W_{i_2k_2})\Delta V_{i_1k_1} - \alpha Z_{i_1k_1i_2k_2} + 2. \tag{23}$$

Similarly if the state $V_{i_2k_2}$ of a neuron (i_2, k_2) in layer L_2 is changed, the change in the total energy is

$$\Delta E = -(U_{i_2k_2} - \alpha W_{i_1k_1})\Delta V_{i_2k_2} - \alpha Z_{i_2k_2i_1k_1} + 2 \tag{24}$$

where $U_{i_2k_2}$ is similar as in Eq. (16), $Z_{i_2k_2i_1k_1}$ and $W_{i_1k_1}$ are similar as in Eq. (19)-(20) except that the subscript 1 is changed to the subscript 2.

3.6 Summary of the Updating Algorithm

The updating algorithm is summarized in the following steps.

Step 1) Set the initial state of neurons for layers L_1 and L_2:

$$V_{i_mk_m} = \begin{cases} 1, & \text{if } |f_{i_m} - f_{k_m}| < \vartheta; \\ 0, & \text{otherwise} \end{cases} \tag{25}$$

where ϑ is a threshold to determine if feature f_{i_m} and f_{k_m} are compatible, $m = 1, 2$.

Step 2) Randomly select (i_1, k_1) in layer L_1.

Step 3) Update the state $V_{i_1k_1}$.

$$V_{i_1k_1}^{n+1} \longrightarrow \begin{cases} 1, & \text{if } U_{i_1k_1} - \alpha W_{i_2k_2} + \alpha Z_{i_1k_1i_2k_2} > 2; \\ 0, & \text{if } U_{i_1k_1} - \alpha W_{i_2k_2} - \alpha Z_{i_1k_1i_2k_2} < -2. \end{cases} \tag{26}$$

Step 4) Randomly select (i_2, k_2) in layer L_2.

Step 5) Update the state $V_{i_2k_2}$.

$$V_{i_2k_2}^{n+1} \longrightarrow \begin{cases} 1 & \text{if } U_{i_2k_2} - \alpha W_{i_1k_1} + \alpha Z_{i_2k_2i_1k_1} > 2; \\ 0, & \text{if } U_{i_2k_2} - \alpha W_{i_1k_1} - \alpha Z_{i_2k_2i_1k_1} < -2. \end{cases} \tag{27}$$

Step 6) Check for the termination condition. If it is satisfied, go to step 7) otherwise go to step 2).

Step 7) Output the final states of neurons $V_{i_1k_1}$ and $V_{i_2k_2}$ which will be the final matches between the model features and the input features in level L_1 and level L_2, respectively.

It is well known that the optimal solution is not always attained for non-convex gradient searches. Two termination strategies are used in this algorithm. One terminates at a local minimum, i.e., whenever the outputs of all the neurons in the network are converged to a local minimum in the sense of unity Hamming distance. This guarantees that (within Hamming distance unity of the output) there is no other state of lower energy. The other is that when the outputs of neurons are unchanged after a fixed number of iterations, the algorithm is terminated.

4 Results

In this section, the merits of the proposed algorithm are examined using several test objects which are images of different door keys. Each image is processed by a QMF filter [2] with 24-tabs in order to generate the multi-resolution images for the test object. In the next preprocessing step, high curvature points (corners) of the test object are extracted at each level of the image pyramid separately.

We investigated two sets of test objects to explore the performance of our proposed multi layer network. The first set is composed of image scenes with single object (one key). In the other set, we processed image scenes that contained multiple occluded objects (overlapping keys). The scene graph pyramid containing two feature graphs at two resolution levels for multiple occluded objects is shown in Fig. 3.

4.1 Single and Occluded Object Results

In this investigation, we formed our test objects by translating and rotating the key models. The model-database contained three keys. The single layer and

the two-layered Hopfield networks were both simulated and their recognition performance for a single and occluded object in the input scene were studied.

The state trajectory and final state at termination depend on the initial state vector and the particular realization of the random updating sequence by which the candidate state for updating is selected. Each network was tested with 17 different random updating sequence realization. Three test objects are used in our experiment, thus a total of 51 runs were performed for both single and occluded object. In the case of single object, for the single layer Hopfield net, the recognition success was 41%; for the two-layered Hopfield net, the rate of recognition was increased to 82%. In the case of input scenes with occluded objects, for the single layer Hopfield net, the rate of recognition was only 6%; for the two-layered Hopfield net, a recognition rate was 31%. The recognition of occluded object is more difficult than that of single object since extra matching ambiguities are introduced.

Our experiment also showed that the computing time for the two-layered Hopfield net takes 1 to 3 times or 1 to 1.5 times more than that of the one layer Hopfield net in the single and occluded object, respectively. This is because the number of nodes in the two-layered Hopfield network is larger than that of the single layer Hopfield network. These findings show that the two-layered Hopfield network is more powerful than the single layer Hopfield network.

4.2 Energy Function Behaviour

To analyze the differences between the one layer Hopfield net and the two-layered Hopfield net, we investigated the behaviour of the energy functions for both networks. Because the energy function is high dimensional and non linear, it is difficult to plot the shape of the energy function over the states of the network. Consider the energy function vs. the iteration numbers as the network seeks a stable state. This presents grounds for comparison between the behaviour of the energy functions of two networks. Fig. 4 shows plots of the energy functions vs. the number of iterations for the single layer Hopfield net as well as the two-layered Hopfield net when the input is a single object.

4.3 Effect of interconnection Parameter α

The interconnection parameter α scales the inter-layer energy function E_e relative to the intra-layer energy functions E_1 and E_2, respectively. It is useful to consider the effect of this parameter α on the behaviour of the two-layered Hopfield network. We examined the effect of this interconnection parameter on the network performance for α in the range of $0-1$. In Fig. 5 plots (b) and (c) show the energy functions, for α close to zero, trapped in local minima which are all far from the global minimum. For α close to zero, the two-layered network behaves as two independent single layer networks without interconnection. For $\alpha > 1$, the inter-energy function dominates the energy function for layers L_1 or L_2, the energy functions E_1 or E_2 never converge to a stable state as shown in plot (a) for E_1. When α is between 0 and 1, the energy function E_1 can converge to the global minimum or a local minimum which is close to the global minimum. Our results show that $0.3 \leq \alpha \leq 0.5$ is a good compromise for both single and two layer nets.

5 Conclusions

In this paper we have presented a multi-layered Hopfield neural network for object recognition. A single layer Hopfield neural network has significant limitations. For example, it could get trapped in one of many local minima of the energy function. Although it is difficult to fully analyze the dynamics of the energy function for a multi-layered Hopfield neural network due to high dimensionality of the energy function, experimental results indicate that the energy function of a multi-layered Hopfield neural network converges to a local minimum which is often equal to or very close to the global minimum.

The matching process proposed in this paper is a concurrent coarse-and-fine strategy. This choice efficiently utilizes the multi-layered Hopfield network by reinforcing the adjacent layers of the multi-layered network. The interconnections between the adjacent layers of multi-layered Hopfield neural network are defined on the basis of the compatibility characteristics of the multi-resolution image pyramid. This is one approach to define interconnections between the layers of the multi-layered Hopfield neural network to make the adjacent layers reinforce each other. Other kind of interconnection compatibility conditions are currently under investigation.

References

[1] N. M. Nasrabadi and W. Li, "Object recognition by a Hopfield neural network," *IEEE Trans. System, Man and Cybernetics*, vol. 21, no. 6, pp. 1523 - 1535, 1991.

[2] J.D. Johnston, "A Filter Family Designed for Use in Quadrature Mirror Filter Banks," *Proceedings of ICASSP*, pp. 291 - 294, April 1980.

[3] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *CGIP*, vol. 1, pp. 244-256, 1972

[4] J. J. Hopfield and D. W. Tank, "Neural computation of decisions in optimization problems," *Biol. Cybern.*, vol. 52, pp. 141 - 152, 1985.

[5] S. Young, "Multiresolution image analysis and pattern recognition," Ph. D. dissertation, 1994.

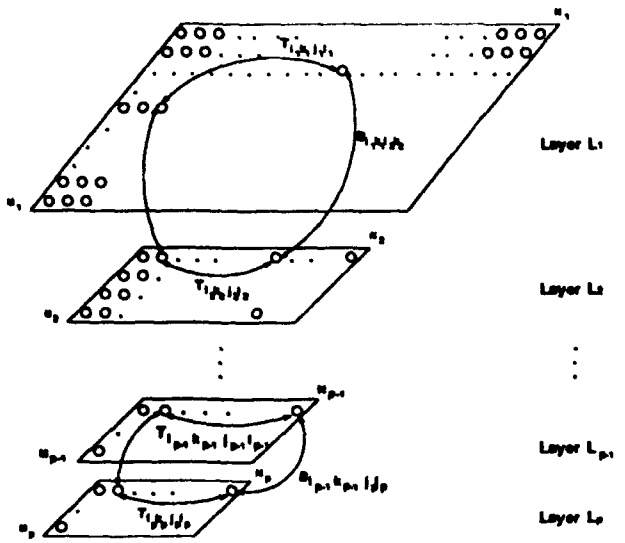

Figure 1.Architecture of multi-layered Hopfield network.

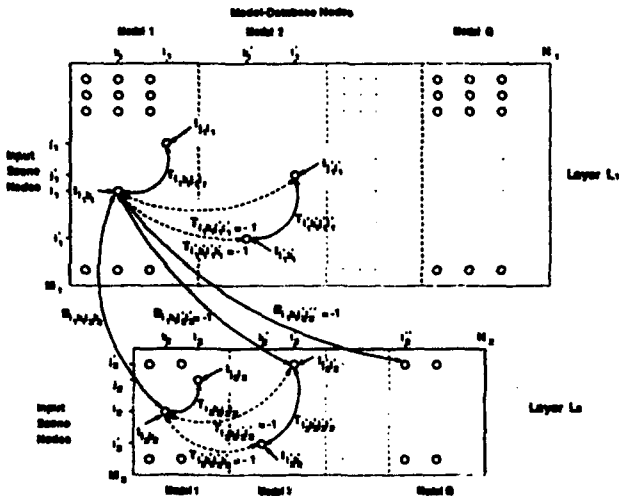

Figure 2. Inter-connections of neurons for two-layered Hopfield network.

Figure 3. A scene graph pyramid containing two feature graphs at two resolution levels for multiple occluded objects.

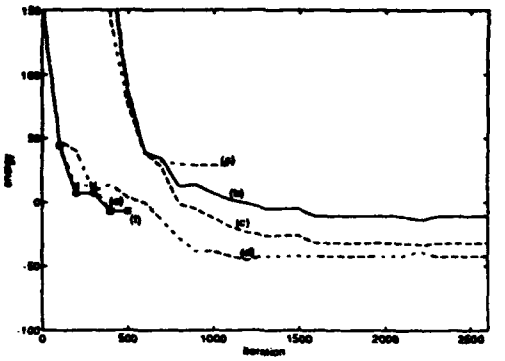

Figure 4. Energy behaviour. (a) E_1', the energy function for the layer L_1 when using single layer network, which stalled at a local minimum; (b) E_1, the energy function for the layer L_1 when using two-layered network, which converged to the global minimum; (c) ψ_1, the total energy at layer L_1, $\psi_1 = E_1 + \alpha \times E_c$ ($\alpha = 0.5$); (d) E_c, the inter-energy. (e) E_2', the energy function for the layer L_2 when using single layer network; (f) E_2, the energy function for the layer L_2 when using two-layered network.

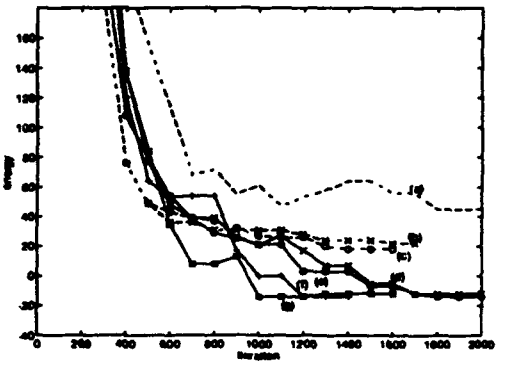

Figure 5. Energy function of the layer L_1 using two-layered net for different values of the interconnection parameter α. (a) $\alpha = 1$; (b) $\alpha = 0$; (c) $\alpha = 0.1$; (d) $\alpha = 0.3$; (e) $\alpha = 0.4$; (f) $\alpha = 0.5$; (g) $\alpha = 0.75$.