# A Hierarchical Neural Network Approach to Three-Dimensional Object Recognition

**Yongsheng Zhang**
Department of Photogrammetry & Remote Sensing
Zhengzhou Institute of Surveying & Mapping, PR China

**KEY WORDS: Three-dimensional Object Recognition, Neural Network, Image Matching**

**ABSTRACT-This paper proposes a hierarchical approach to solving the surface and vertex correspondence problems in multipe-view-based three-dimensional object recognition systems. The proposed scheme is a coarse-to-fine search process and a Hopfield network is employed at each stage. Compared with conventional object matching schemes, the proposed technique provides a more general and compact formulation of the problem and a solution more suitable for parallel implementation. At the coarse search stage, the surface matching scores between the input image and each object model in the database are computed through a Hopfield ntework and are used to select the candidates for further consideration. At the fine search stage, the object models selected from the previous stage are fed into another Hopfield network for vertex matching. The object model that has the best surface and vertex correspondences with the input image is finally singled out as the best matched model.**

## I. INTRODUCTION

THREE-DIMENSIONAL (3-D) object recognition is the process of matching an object to a scene description to determine the object's identity and/or its pose (position and orientation) in space [1]-[3]. Any system capable of recognizing its input image must in some sense be model-based. The problem of object recognition can be separated into two closely related subproblems-that of model building and that of recognition. There are different approaches to both these subproblems, and the procedure used for recognition will have a strong impact on the kind of model that will be required and vice versa.

The multiple-view approach to 3-D object recognition [4]-[6] models an object by collecting all its topologically different 2-D projections from various viewing angles. In the model database, each 2-D projection is topologically different from the others and is referred to as a *characteristic view* (CV) [4], [5]. In [7], we have proposed a computer system to automatically construct multiple-view model database for polyhedral objects. The database is organized as a graph in which a node represents a characteristic view and an arc represents the transformation between two characteristic views. It is also referred to as a CV library (or aspect graph [6]).

Although the redundancy of the model database has been reduced to the largest extent in the CV library generation process, the size of the library is still large if the target object is complex in shape. This makes the subsequent recognition process very time-consuming if a traditional sequential matching scheme is adopted. Generally, the bottleneck of the recognition process is to establish the correspondence relationships between the contents of the image and the object model.

In this paper, we propose a coarse-to-fine strategy to solve the correspondence problem in 3-D object recognition based on Hopfield networks [8], [9]. Compared with the conventional object matching schemes, the proposed technique provides a more general and compact formulation of the problem and a solution more suitable for parallel implementation.

## II. HOPFIELD NETWORKS FOR IMAGE MATCHING

A Hopfield net is built from a single layer of neurons, with feedback connections from each unit to every other unit (although not to itself). The weights on these connections are constrained to be symmetrical. Generally, a problem to be solved by a Hopfield net can be characterized by an energy function E. Through minimizing the energy function, an optimal (or near optimal) solution is ultimately reflected in the outputs of the neurons in the network. The applications of the Hopfield net are multifarious. In [10], object recognition is based on subgraph matching. The graph matching technique is formulated as an optimization problem where an energy function is minimized. The optimization problem is then solved by a discrete Hopfield network. In [11], a Hopfield network realizes a constraint satisfaction process to match visible surfaces of 3-D objects. In [12], the object recognition problem is casted as an inexact graph matching problem and then formulated in terms of constrained optimization. In [13], the problem of constraint satisfaction in computer vision is mapped to a network where the nodes are the hypotheses and the links are the constraints. The network is then employed to select the optimal subset of hypotheses which satisfy the given constraints.

In this paper, the Hopfield net for image matching is in the form of a two-dimensional array. The rows of the array represent the features of an input image, and the columns represent the features of an object model. The output of a neuron reflects the degree of similarity between two nodes, one from the image and the other from the object model. The matching process can be characterized as minimizing the following energy function [10]:

$$E = -\sum_i \sum_k \sum_j \sum_l C_{ikjl} V_{ik} V_{jl} + \sum_i \Big(1 - \sum_k V_{ik}\Big)^2 + \sum_k \Big(1 - \sum_i V_{ik}\Big)^2 \quad (1)$$

where $V_{ik}$ is an output variable which converges to 1.0 if the $i$th node in the input image matches the $k$th node in the object model; otherwise, it converges to 0. The first term in (1) is a compatibility constraint. The second and third terms are included for enforcing the uniqueness constraint so that each node in the object model eventually matches only one node in the input image and the summation of the outputs of the neurons in each row or column is equal to 1. The major component of the compatibility measure $C_{ikjl}$ (or strength of interconnection) between a neuron in row $i$ column $k$ and a neuron in row $j$ column $l$ is expressed in terms of a function $F$ defined as follows:

$$F(x,y) = \begin{cases} 1, & if\ |x-y| < \theta \\ -1, & \text{otherwise} \end{cases} \quad (2)$$

where $\theta$ is a threshold value and $x$ and $y$ are features pertaining to row and column nodes, respectively. In this paper, two different sets of features and relations will be used for surface and vertex matching, and they will be detailedly described in Sections Ⅲ and Ⅳ, respectively. In general, $C_{ikjl}$ can be expressed as

$$C_{ikjl} = \sum_n W_n \times F(x_n, y_n) \quad (3)$$

where $x_n$ is the $n$th feature of the node in row $i$ column $k$, $y_n$ is the $n$th feature of the node in row $j$ colume $l$, and the summation of the weighting functions $W_n$'s is equal to 1(i.e., $\sum_n W_n = 1$). Equation (1) can be simplified and fit into the energy function form of a Hopfield network [8] as follows:

$$E = -\frac{1}{2}\sum_i \sum_k \sum_j \sum_l T_{ikjl} V_{ik} V_{jl} - \sum_i \sum_k I_{ik} V_{ik} \quad (4)$$

where $I_{ik} = 2$, and

$$T_{ikjl} = C_{ikjl} - \delta_{ij} - \delta_{kl} \quad (5)$$

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

Matching can be considered as a constraint satisfaction process. The global information extracted from the image provides positive or negative supports for local feature matching. According to Hopfield and Tank [10], the strength of the connection between each neuron pair can be derived from the energy function. Based on these connections, the equation of motion for state $u_{ik}$ of a neuron at position $(i,k)$ can be derived as follows:

$$du_{ik}/dt = \sum_j \sum_i C_{ikjl} V_{jl} - \sum_l V_{il} - \sum_j V_{jk} - u_{ik}/\tau + I_{ik} \quad (6)$$

where

$$V_{ik} = g(u_{ik}) = [1 + exp(-2u_{ik}/u_0)]^{-1} \quad (7)$$

Since the sum of $V_{ik}$ on all the neurons at initialization is constrained to be equal to the number of the final desired output, i.e.,

$$\sum_i \sum_k V_{ik} = N \quad (8)$$

where $N$ is either the number of rows or the number of columns of the array depenting on which one is smaller, we can derive the initial condition for $u_{ik}$ from (7) and (8) as follows:

$$u_{init} = -\frac{u_0}{2}\ln(N-1) \quad (9)$$

In order to prevent the system from being trapped in an unstable equilibrium in which the voltage of each neuron is equal, a certain amount of noise must be added to this initial value. We can rewrite the initial conditions as follows:

$$u^0_{ik} = u_{init} + \delta \quad (10)$$

and

$$V^0_{ik} = g(u^0_{ik}) \quad (11)$$

where $\delta$ is a random number uniformly distributed between $-0.1u_{init}$ and $+0.1u_{init}$.

The algorithm for matching, based on the continuous Hopfield network model, is summarized as follows.

**Algorithm**

**Input**: A set of neurons arranged in a two-dimensional array with initial values $V^0_{ik}$, where $0 \leqslant i \leqslant row\_A\_max - 1$, $0 \leqslant k \leqslant column\_max - 1$, and $row\_max$ and $column\_max$ are the numbers of rows and columns in the array, respectively.

**Output**: A set of stabilized neurons with output values $V_{ik}$, where $0.0 \leqslant V_{ik} \leqslant 1$ for $0 \leqslant i \leqslant row\_max$ and $0 \leqslant k \leqslant column\_max$.

**Method**:

1)Set the initial conditions using (10) and (11).

2)Set index=1 and limit=$n$.

3)Randomly pick up a node $(i,k)$.

4)Update the value of $u_{ik}$.

5)Calculate the new output of neuron $(i,k)$ as follows:

$$V_{ik} = g(u_{ik})$$

6)Increment index by 1.

7)If index $< n$, then go to (3), else stop and output the final values of all the neurons based on the following rule:

$$V_{ik} = \begin{cases} 1, & if\ V_{ik} > \theta 1 \\ 0, & otherwise \end{cases} \qquad (12)$$

where θl is a threshold value.

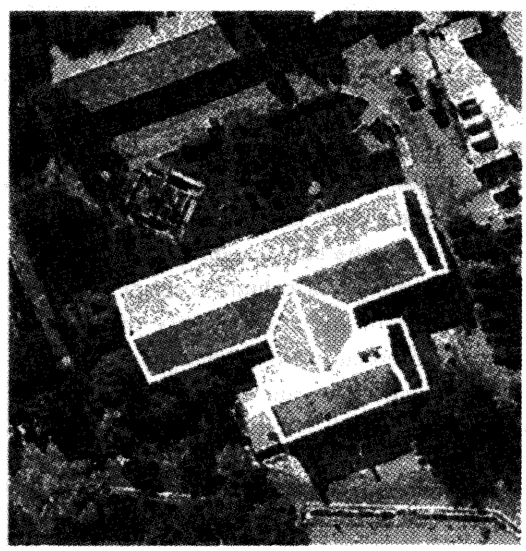
In the actual implementation, $N$ is replaced by a value $N_+$ which is greater than $N$. This is to adjust the neutral positions of neurons. The coefficient $\tau$ is 1 and $u_0$ is 0.002. The instability problem of the Hopfield neural network has been extensively studied [17]. It is well known that the optimal solution is not always found. In the algorithm, two termination strategies are used to handle different situations. The freeze strategy is adopted whenever the outputs of all the neurons in the network are convergent. When a small number of neurons are not convergent after a long period of time, the time-out strategy is adopted to force the system to stop.

## Ⅲ. HOPFIELD NETWORKS FOR SURFACE CORRESPONDENCE ESTABLISHMENT

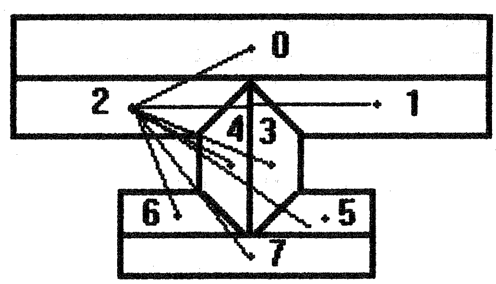
In this section, a Hopfield net is designed to establish the surface correspondences between an unknown object and an object model in the database. It is assumed that the unknown object is in the form of line drawings which are obtained by segmenting the original image and each object model is a 2-D projection of a 3-D object whose identity and pose are to be determined. The features for surface matching are firstly described. This is followed by the introduction of row-column assignment. Then, the strength of interconnection $C_{ikjl}$ is defined. A method for quantitatively evaluating the degree of match between the input image and the object model is presented. Finally, the characteristics of the networks are discussed. Since regions (in 2-D) are the projections of surfaces (in 3-D), we use the two terms interchangeably in this paper.

### *A. Feature Selection and Row-Column Assignments*

Before establishing surface correspondence, each object model or image has to be preprocessed. We first label all the regions in the image or model in order from left to right and top to bottom. This labeling scheme provides the basis for subsequent row-column assignment process. An example of labeled image is shown in Fig. 1(b). During the labeling process for each region, the area is calculated and the boundary traced for locating high curvature (or corner) points [15]. The original image is then converted to a set of polygons with vertices numbered in a certain order. The centroid of each polygon is then computed. For each polygon, two feature are extracted for surface matching. One is a local feature, which is the area of the polygon. The other is a relational feature, which is defined as a set of distances originated from its centroid to all the centroids in other polygons of the image. An example demonstrating both features of a polygon is shown in Fig. 1. Since these two features are not scale invariant, a normalization process is performed to compensate this effect. This process starts with selecting the longest distance from the set of intercentroid distances in the input image. Then, based on the ratio between the longest, intercentroid distance in the input image and the longest intercentroid distance in the object model, all the distances in the input image are divided by this ratio for normalization. The area of each polygon in the input image is normalized by dividing it by the square of this ratio.



(a)




(b)

Fig. 1 (a)A aerial image of a building; (b)Its polygons labeled and the feature set of polygon 2, local feature: area of polygon 2, relational features: $\{d_{20}, d_{21}, d_{23}, d_{24}, d_{25}, d_{26}, d_{27}\}$.

The labels of the polygons in an input image or an object model is derived during the labeling process. In order to perform matching, each polygon in the input image is assigned a row index and each polygon in the object model is assigned a column index. An example is shown in Fig. 2.

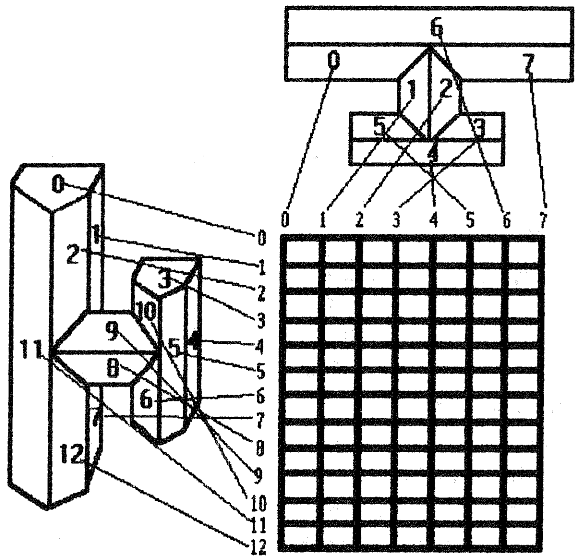



Fig. 2. Row-column assignment for surface correspondence establishment

*B. $C_{ikjl}$ for Surface Matching*

At the stage of surface correspondence establishment. $C_{ikjl}$ is expressed as follows:

$$C_{ikjl} = W_1 \times F(I_i, M_k) + W_2 \times F(I_j, M_l) + W_3 \times F(d_{I_i I_j}, d_{M_k, M_l}) \qquad (13)$$

where $I_x$ represents the (normalized) area of the $x$th polygon in the input image, $M_y$ the area of the $y$th polygon in the object model, $d_{I_i I_j}$ the (normalized) distance between the centroids of the $i$th and the $j$th polygons in the input image, and $d_{M_k M_l}$ the distance between the centroids of the $k$th and the $l$th polygons in the object model. The values of $W_i^{\prime}s$ reflect the importance of each term. They can be adjusted as long as the sum equals to 1. For the symmetric terms, the associated weights should be set equal (e. g. , $W_1 = W_2$). The weight of the relational feature ($W_3$) in more important than the other two and is thus set with higher value.

*C. Similarity Measure from Surface Matching*

After the states of the Hopfield net for surface matching are stabilized, we can count the number of active neurons in the network and use it to measure the degree of match (or similarity) between the object model and the input image. The procedure consists of the following four steps.

*Step* 1: Initialize both *row_match* and *column_match* to be 0.

*Step* 2: Count the number of 1's in each row. If there is no1 in a row, skip to the next row and leave *row_match* unchanged. If there is only one 1 in a row, add 1 to *row_match*. If there are $n$ 1s ($n>1$) in a row, then add $1/n$ to *row_match*. Repeat this for all the rows.

*Step* 3: Do the same ealculation for all the columns and update *column_match*.

*Step* 4: Pick up the larger one from *row_match* and *column_match*, divide it by the number of rows, and take the result as the similarity measure.

Given an input image and a large number of object models (which is usually the case in a multiple-view approach), the degree of match between the input image and different object models can be derived by comparing the input image and each of the object models in the final state of the Hopfield net. Ideally, there should be at most one active neuron in each row or column. However, due to the influence of the first term in the right-hand side of (1), it is possible to have more than one candidate in the same row or column in the final state of the network. As far as matching is concerned, this situation should be considered as unfavorable and hence decreases the degree of match. This is the reason why $1/n$ is added to the degree of match (row or column) instead of 1 when there are $n$ 1s simultaneously existing in the same row or column. When there is no 1 in a row (or column), it means a surface in the input image (or object model) does not have a corresponding surface in the object model (or input image). This does not contribute to the degree of match between the input image and the object model and thus the degree of match is left unchanged. For a model-based 3-D object recognition system using multiple-view approach, a set of 2-D object models are in the model database. To derive their degrees of match with the input image in the Hopfield net, we associate the input image with row indexes and object models column indexes. This arrangement allows us to compare all the object models simultaneously with the input image, provided that the dimension of the neuron array is large enough. This also explains why the number of rows is used as the denominator in the derivation of similarity measure.

*D. Discussion*

The proposed Hopfield net for surface matching has a flexible structure and is able to solve the surface correspondence problem even if the numbers of polygons in the input image and the object model are different. in other words, the two-dimensional neuron array may have different numbers of rows and columns and an inexact matching [18] can be performed in this net. Furthermore, based on the outputs of the neurons in the network, a similarity measue between any object model and the input image can be derived even if they contain a different number of surfaces.

This similarity measure is used to reduce the search space. Based on the proposed simple features, a set of object models most similar to the input image is selected from the database. This process can be considered as a coarse search step because "good" candidates as well as some "bad" candidates are selected due to roughness of the feature set. However, this proeess discards a large number of object models in the database and significantly reduces the search space.

## IV. HOPFIED NETWORKS FOR VERTEX CORRESPONDENCE ESTABLISHMENT

After the surface correspondences between the unknown object and the object model are confirmed, the next step is to apply Hopfield networks to establish the vertex correspondences. In this section, the features for vertex correspondence establishment are first described. This is followed by the introduction of row-column assignment. Next, the strength of interconnection $C_{ikjl}$ is defined. Then, a technique for systematically deriving the best vertex correspondence is presented. Finally, the characteristics of the networks are discussed.

### *A. Feature Selection and Row-Column Assignments*

Before we start establishing the vertex correspondences, the order of all the vertices in each polygon must be determined. This is usually achieved at the preprocessing stage by selecting the vertex of a polygon with minimum $x$ coordinate as the starting vertex ($o$th vertex). If two vertices happen to possess the same minimum $x$ coordinate, the one with smaller $y$ coordinate is selected as the starting vertex. The subsequent vertices are numbered sequentially in a clockwise direction. The reason for making this ordering is to facilitate the subsequent row-column assignment process. A useful feature for vertex correspondence establishment is the *shape number* proposed in [16]. This feature is invariant to rotation, translation, and scaling in 3-D space. The method of deriving the shape number for each detected vertex is as follows. Consider a polygon in Fig. 3 which has a clockwise edge sequence of $(E_iE_{i+1}E_{i+2}\cdots)$ and a vertex sequence eof $(N_iN_{i+1}N_{i+2}\cdots)$. Point $I_i$ is the intersection of vector $\overrightarrow{N_iN_{i+2}}$ with vector $\overrightarrow{N_{i+1}N_{i+3}}$. Then, the ratio (distance from $N_i$ to $I_i$)/(distance from $N_i$ to $N_{i+2}$) which is assigned to vertex $N_i$ of this polygon remains constant for any positioning of the surface in 3-D space.

For the general case, consider a polygon of $n$ edges with a clockwise edge sequence of $(E_1E_2\cdots E_{n-1}E_n)$. Edge $E_i$ has a clockwise orientation of $N_i$ to $N_{(i+1)_n}$ where

$$(i+k)_n = \begin{cases} i+k, & \text{if } i+k \leqslant n \\ i+k-n, & \text{otherwise} \end{cases} \quad (14)$$

and

$$i+k \leqslant 2n$$

Let [A B] be the distance from point $A$ to point $B$. Then, the shape number for vertex $N_i$ can be expressed as

$$Sh_i = \frac{[N_i \quad I_i]}{[N_i \quad N_{(i+2)_n}]} \times 100, \quad 1 \leqslant i \leqslant n \quad (15)$$

where $I_i$ is the intersection of vector $\overrightarrow{N_iN_{(i+2)_n}}$ with vector $\overrightarrow{N_{(i+1)_n}N_{(i+3)_n}}$. For a nonconvex polygon, the intersection of the two vectors may be outside the contour or may not occur at all. This means that some $Sh_i$s may be greater than 100 and some may not exist. An upper limit of 100 is placed on the calculation for those shape numbers which are over 100 and 0s are assigned to those $Sh_i$s which do not exist.

To establish vertex correspondences, we associate a local feature and a relational feature with each vertex. The local feature is its corresponding angle value and the relational feature is its shape number. In order to perform vertex matching, we select a reliable and matched polygon pair obtained from the surface matching process. Each vertex of the matched polygon in the input image is assigned a row index identical to its label. Similarly, each vertex of the matched polygon in the object model is assigned a column index identical to its label. The attributes of each row or column include the local and relational features of its corresponding vertex. An example of this assignment is shown in Fig. 4.

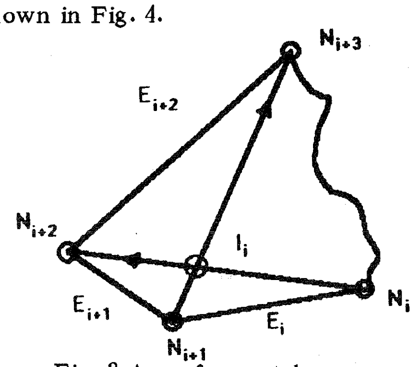


Fig. 3 A surface patch.

### *B. $C_{ikjl}$ for Vertex Correspondence Establishment*

For vertex correspondence establishment, $C_{ikjl}$ can be expressed by an equation as follows:

$$C_{ikjl} = w_1 \times F(I_i, M_k) + w_2 \times F(I_j, M_l) + w_3 F(II_i, MM_k) + w_4 \times F(II_j, MM_l) \quad (16)$$

where $I_x$ represents the shape number of the $x$th vertex of a polygon in the input image, $M_y$ the shape number of the $y$th vertex of the corresponding polygon in the object model, $II_x$ the angle of the $x$th ver-

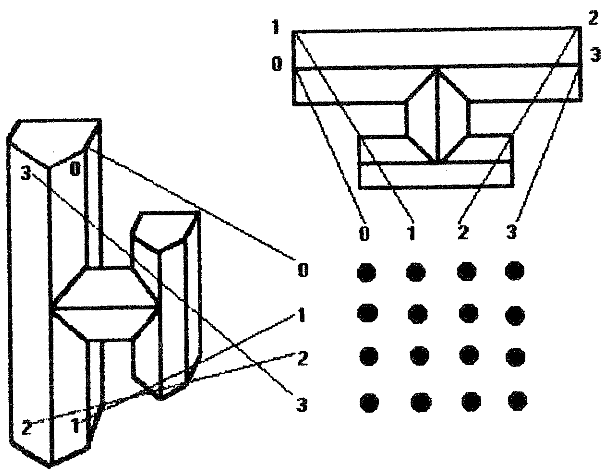



Fig. 4. Row-column assignment for vertex correspondence establishment.

tex in a polygon of the input image, and $MM_y$ the angle of the $y$th vertex in the corresponding polygon of the object model. The two selected polygons. one from the image and the other from the object model, have been matched at the previous stage. As to the weights ($w_i's$) on the right-hand side of (16), the following restrictions must be satisfied, i. e., $w_1 = w_2$, $w_3 = w_4$, and $\sum_{i=1}^{4} w_i = 1$. In general, the weights assigned to the relational features ($w_3$ and $w_4$) are higher than those of the local features ($w_1$ and $w_2$).

*C. Deriving the Best Vertex Correspondences*

Based on the row-column assignment mentioned in Section Ⅳ-A, the vertex labels of a matched polygon in the input image are arranged as the row indexes and the vertex labels of its corresponding polygon in the object model are arranged as the column indexes. Because of this particular assignment, the vertex correspondence problem can be analyzed in a systematic manner.

Before we proceed, we will define some terminologics which wil lbe frequently used in the sequel. Let $P$ represent an $n$-sided polygon whose vertices are sequenced clockwise as $(p_0 p_1 \cdots p_{n-1})$. If polygons $P$ is rotated clockwise by $m$-vertex (m < n) into a new polygon $P'$, then the vertex sequence is updated from $(p_0 p_1 \cdots p_{n-1})$ to $(p_m p_{(m+1)n} p_{(m+2)n} \cdots p_{n-1} p_0 \cdots p_{m-2} p_{m-1})$. Let $A$ and $B$ be two $n$-sided polygons with clockwise vertex sequences $(a_0 a_1 \cdots a_{n-1})$ and $(b_0 b_1 \cdots b_{n-1})$. respectively. Suppose the original vertex correspondences between $A$ and $B$ is $a_0 \rightarrow b_0, a_1 \rightarrow b_1, \cdots, a_{n-1} \rightarrow b_{n-1}$. If polygon $A$ is rotated clockwise by $m$-vertex distance into $A'$, then a new vertex correspondences between $A'$ and $B$ is $a_m \rightarrow b_0$, $a_{(m+1)n} \rightarrow b_1$, $a_{(m+2)n} \rightarrow b_2, \cdots, a_{m-2} \rightarrow b_{n-2}, a_{m-1} \rightarrow b_{n-1}$.

The simplest way to determine the correspondences between the vertices of two polygons is to fix one of them. rotate the other in 2-D space each time by 1-vertex (either clockwise or counterclockwise) and calculate the total error by accumulating the differences between all the corresponding node pairs. The comparison continues until the rotated polygon is brought back to the original position. For two n-sided polygons, there are $n$ sequential comparisons to be performed. By using a Hopfield network, the comparisons can be executed concurrently and the results shown explicitly in the network.

Given two n-sided polygons with vertex labels prearranged as described before, we can construct a Hopfield networks in the form of an $n \times n$ neuron matrix in which the row and column indexes correspond to the vertex labels of the two polygons, respectively. The two polygons will be referred to as row polygon and column polygon, respectively. Let $neuron(i, j)$ represent a neuron at position $(i, j)$. Because the vertex orders of a polygon is preserved under rotation in 2-D space, the degree of match between a fixed polygon and each of the $n$ rotated instances of the other polygon can be analyzed systematically as follows. Suppose we rotate the row polygon clockwise $k$-vertex ($k \neq 0$) and fix the column polygon. Then, the degree of match between the column polygon and the rotated row polygon can be computed from the following set of neurons:

$$\{neuron(i, j) \mid \text{where } i = (j-k) \bmod n, \quad 0 \leqslant i, j < n, \quad 1 \leqslant k < n\} \tag{17}$$

where

$$(j-k) \bmod n = \begin{cases} j-k, & \text{if } j \geqslant k \\ j-k+n, & \text{otherwise} \end{cases} \tag{18}$$

The degree of match between the fixed column polygon and the rotated row polygon can be determined by counting the number of active neurons (after the network stabilizes) in the neuron set represented in (17). Generally, the set of ncurons in (17) with different $k$ values can be represcnted by the union of neurons in two diagonals parallel to the main diagonal of the matrix. Using the main diagonal as basis, the upper-right diagonal starts from $neuron(0, k)$ and ends at $neuron(n-k-1, n-1)$. The lower-left diagonal starts from $neuron(n-k, 0)$ and ends at $neuron(n-1, k-1)$. When $k=0$, only one diagonal starting from $neuron(0,0)$ and ending at $neuron(n-1, n-1)$ exists. This happens if neither row nor column polygon is rotated. Based on this arrangement, the degree of match between the fixed column polygon and each of the $n$ instances of the rotated row polygon can be determined concurrently in the network.

Let $a_{ij}$ represent the output of the neuron at position $(i, j)$, where $0 \leqslant i, j < n$. The best match out of the $n$ comparisons can be determined by the following procedures.

*Step* 1: Calculate the number of active neurons from each of the $n$ comparisons by the following equation:

$$\mathrm{Match}(k) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \delta_{(i+k) \bmod n, j} a_{ij}, \quad 0 \leqslant k < n. \quad (19)$$

*Step* 2: Determine the best match by

$$Max\ Match(m) = max\ [Match(k)], \quad 0 \leqslant k < n.$$

*Step* 3: If the best match measure $Max\ match(m)$ is larger than a threshold $\theta$, then the original vertex correspondenee matrix as follows:

$$a_{ij} = \begin{cases} 1, & \text{if } i = (j - m) mod\ n \\ 0, & \text{otherwise} \qquad (20) \\ & 0 \leqslant i, j < n \text{ and } 0 \leqslant m < n. \end{cases}$$

In the first step, (19) is used to calculate the number of active neurons in each of the $n$ comparisons. In the second step, the results derived from Step 1 are compared and the one that contains the largest number of active neurons is selected. In the third step, it is inteded to generate the best vertex correspondence matrix by removing all ambiguities. All the neurons corresponding to the best match are activated by applying (17). Those neurons irrelevant to the activated neuron set are then inactivated.

*D. Discussion*

Being designed as a constraint satisfaction network, the Hopfield net may encounter problems such as multiple active neurons in a row or column after it is stabilized. This is because the given set of constraints is not adequate for singling out the optimal solution. However, a procedure which takes advantage of the fact that the vertex ordering of a polygon is preserved under any rotation in 2-D space is designed to determine the best vertex correspondences. This procedure is valid even if some intermediate vertices are not matched due to distortion caused by changing viewpoints.

We have mentioned that, at the surface correspondence stage, a set of object models are selected from the model database due to higher surface matching measures. The vertex correspondence establishment can be considered as a fine search process. At this stage, we eliminate those object models unable to estblish any consistent vertex correspondence with the input image. The discarded object models may include the folliwing: 1) those which are actually the projections of different objects but were accidentally picked up; and 2) those which are projections of the same object but are quite different from the input image. They were mistakenly selected due to the roughness of the features for surface matching. Among the object models whose vertices of the kernel region match those of the input image, the one with the highest surface matching score and the largest number of vertex correspondences is finally selectde as the largest number of vertex correspondcnces is finally selected as the best matched model. Since the pose of the viewpoint where the best matched object model is visualized is predetermined in the modeling phase, the pose of the unknown object can be obtained by computing a 2-D rotation which brings the vertices in the input image to align with thcir corresponding vertices in the best matehed object model.

## V. CONCLUSIONS

In this paper, we use Hopfield networks to solve both the surface and vertex correspondence problems for 3-D object recognition. The proposed scheme can be considered as a coarse-to-fine search process. In a 3-D object recognition system adopting multiple-view approach, the database is usually a set of 2-D projections which are topologically different. By calculating the surface matching score between the input image and each object model in the model database, a set of 2-D models with higher surface matching score is selected. This phase can be considered as a coarse search process. The object models selected from the first stage are then fed into the Hopfield net for establishing the vertex correspondences between the input image and each of these models. This phase is the fine search process. The object model that has the best vertex and surface correspondences with the input image is finally selected. Once an object model is identified, we can use the model coordinate frame as the reference frame to derive the pose of the unknown object.

In comparison with conventional methods, there are several advantages in using Hopfield nets for image matching. Image matching can be regarded as a process of finding homomorphisms between two relational structures and is basically an NP-complete problem. In the worst case, its time complexity is expected to be exponential. In order to speed up the performance, several methods adopting look-ahead or relaxation [19], [20] schemes have been proposed. However, in general, it takes more effort to manage a look-ahead table or devise a relaxation algorithm. A Hopfield network takes advantage of its massively parallel structure to deal with matching problems in an elegant and systematic manner and quantitatively reflects the degree of similarity in the final states of the neurons. The formulation of the network structure is simple and the matching process is easy to implement.

## REFERENCES


[1]P. J. Besl and R. C. Jain, "Three-dimensional object recognition", *ACM Comput. Surveys*, vol. 17, no. 1, pp. 75-145, Mar. 1985.

[2]M. Oshima and Y. Shirai, "Object recognition using three-dimensional information", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-5, no. 4, pp. 353-361, July 1983.

[3]R. T. Chin and C. R. Dyer, "Model-based recognition in robot vision", *ACM Comput. Surveys*, vol. 18, no. 1 pp. 67-108, Mar. 1986.

[4]I. Chakravarty and H. Freeman, "Characteristic views as a basis for three-dimensional object recognition", in *Proc. Soc. Photo-Opt. Instr. Eng*, vol. 336, pp. 37-45, 1982.

[5]H. Freeman and I. Chakravarty, "The use of characteristic views in the recognition of three-dimensional objects", E. Gelsema and L. Kanal, Eds., *Pattern Recognition in Practice*. Amsterdam: The Netherlands, North-Holland, pp. 277-288, 1980.

[6]L. Stark, D. Eggert, and K. Bowyer, "Aspect graphs and nonlinear optimization in 3-D object recognition", *IEEE 2nd Int. Conf Computer Vision*, Tampa, FL. Dec. 5-8, 1988, pp. 501-507.

[7]Y. Zhang, "An automated procedure to construct minimum aspect graphs for 3-D object recognition", Journal of Pattern Recognition, Vol. 10, pp. 132-141, 1994.

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

[9]—, "Computing with neural circuits: A model", *Science*, vol. 233, pp. 625-633, 1986.

[10]W. Li and M. Nasrabadi, "Object recognition based on graph matching implemented by a hopfield-style neural network", *Int. J. Conf. Neural Networks, II*. Washington, DC, 1989.

[11]B. Parvin and G. Medioni, "A constraint satisfaction network for matching 3D objects", *Int. J. Conf. Neural Networks, II*. Washington, DC, 1989.

[12]E. Mjolsness, G. Gindi, and P. Anandan, "Optimization in model matching and perceptual organization", *Neural Computation*, vol. 1, pp. 218-229, 1989.

[13]R. Mohan, "Application of neural constraint satisfaction networks to vision", *Int. J. Conf. Neural Networks, II*. Washington, DC, 1989.

[14]P. J. Besl and R. C. Jain, "Segmentation through variable-order surface fitting", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-10, no. 2, pp. 167-192, Mar, 1988.

[15]S. Y. Chen and M. Y. Chern, "Flexible corner detection based on a single-parameter control", *Proc. of SPIE*, vol. 848, pp. 108-114, 1987.

[16]S. A. Underwood and C. L. Coates, Jr., "Visual learning from multiple views", *IEEE Trans. Comput.*, vol C-24, no. 6, pp. 651-661, 1975.

[17]G. W. Davis, "Sensitivity analysis in ncural net solutions", *IEEE Trans. Syst., Man, Cybernet.*, vol. 19. no. 5, pp. 1078-1082, Sept./Oct. 1989.

[18]L. G. Shapiro and R. M. Haralick, "Structural description and inexact matching", *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-3, pp. 504-515, 1981.

[19]L. G. Shapiro, J. D. Moriarty, and R. M. Haralick, "Matching three-dimensional objects using a relational paradigm", *Pattern Recognition*, vol. 17, no. 4, pp. 385-405, 1984.

[20]R. M. Haralick and J. S. Kartus, "Arrangements, homomorphisms, and discrete relaxation", *IEEE Trans Syst., Man, Cybern.*, vol. SMC-8, pp. 600-612, Aug. 1978