Journal of Computer Science and Technology, Vol. 10, No. 5, pp. 385-402, 1995.
Science & Allerton Press

A Parallel Voting Scheme For Aspect Recovery

Ke CHEN[1]
National Laboratory of Machine Perception and The Center for Information Science
Peking University, Beijing 100871, China

Masumi ISHIKAWA
Department of Control Engineering and Science, Kyushu Institute of Technology, Japan

Abstract

Recently, a *qualitative* approach was proposed for 3-D shape recovery based on a hybrid object representation. In this approach, *aspect recovery* is the most important stage which binds regions in the image into meaningful aspects to support 3-D primitive recovery. There is no known polynomial time algorithm to solve this problem. The previous approach dealt with this problem by using a heuristic method based on the conditional probability. Unlike the previous method, in this paper, we present a novel parallel voting scheme to conquer the problem for efficiency. For this purpose, the previous global aspect representation is replaced with a distributed representation of aspects. Based on this representation, we propose a three-layer parallel voting network to complete aspect recovery. For evaluating likelihood, a continuous Hopfield net is employed so that we can enumerate all aspect coverings in decreasing order of likelihood. We describe this method in detail and demonstrate its usefulness with simulation.

Keywords: Computer vision, 3-D shape recovery, parallel computing, voting scheme, Hopfield net.

1 Introduction

Recently, a hybrid object representation was presented[1] in which objects are composed of a set of chosen 3-D object-centered volumetric primitives; the primitives, in turn, are mapped to a set of 2-D viewer-centered aspects. Based on this novel object representation, for *unexpected* object recognition, one of the authors also presented a bottom-up approach to recovery of qualitative 3-D volumetric primitives from a 2-D image[2]. Using this approach, the primitive recovery was formulated as a heuristically guided search through the various groupings of image regions into aspects, each representing a view of a volumetric part. Moreover, an attention mechanism was also developed to extend this approach to top-down *expected* object recognition[3], by using prior knowledge of the target object to focus the various search procedure inherent in the previous unexpected object recognition paradigm.

A major limitation of the aforementioned approach was its dependency on a complete and consistent covering of the image regions in terms of a set of aspects. As a result, an aspect recovery is necessary for the current framework of object recognition. However, the problem can be formulated as *partition into isomorphic subgraphs* problem[1]; there is no known polynomial time algorithm to solve it[4]. For this problem, in the previous approach, a heuristic mechanism was employed with the conditional probability matrices which provide constraints so as to the problem trackable. In addition, some efforts were also made for developing a connectionist approach to conquer this problem[5]. Unfortunately, both approaches cannot still avoid suffering from quite high computational burden. Voting techniques have been developed for handling object recognition problems with high computational complexity. Its efficiency lies in its parallel working mechanism. The previous techniques, like Hough transform[6] and geometric hashing[7], usually employ elaborate transforms and invariance for mapping data from one space to the other space so as to accumulate evidence in the new space

[1]Corresponding author. E-mail: chen@cis.pku.edu.cn

parallely. Unlike the previous methods, in this paper, we propose a novel parallel voting scheme for aspect recovery so that an approximated solution can be achieved simply using local constraints. Instead of using a transform, we employ a distributed representation of aspects to accumulate evidence. Moreover, a network structure is also presented for parallel implementation of the proposed voting scheme. In addition, some connectionist techniques are also employed to rank likelihood for ordering aspect coverings.

This paper is organized as follows. First we review the hybrid object representation and describe the 3-D shape recovery framework based on the voting scheme. After that, a distributed representation of aspects is proposed, followed by the parallel implantation techniques of this scheme. Moreover, a neural computing technique is described to rank likelihood. A line drawing is used to justify our method in instance. Some problems and possible solutions are given in the section of discussion. In the final, conclusions are drawn.

2 The Hybrid Object Representation

It is well known that there are two object representations, namely, object-centered representation and viewer-centered representation. Both of them have been used in object recognition extensively. For object-centered representation, moreover, a set of 3-D volumetric primitives can be used to construct the object models instead of 3-D objects themselves. For the set of volumetric modeling primitives chosen, they may be mapped to a set of viewer-centered aspects.

In the current hybrid object representation, a set of ten primitives is employed according to Biederman's RBC theory[8]. As a subset of Biederman's geons, basically, the set of primitives can include three properties of the geons, namely, cross-section shape, axis shape and cross-section size variation[8].To construct objects, the primitives are attached to one another with the restriction that any junction of two primitives involves exactly one distinct surface from each primitive.

Unlike traditional aspect graph representation of 3-D objects model, the current method differs in that we use aspect to represent a (typically small) set of volumetric primitives is constructed, rather than representing an entire object directly. Consequently, our goal is to use aspects to recover the 3-D primitives that make up the object in order to carry out a recognition-by-parts procedure, rather than attempting to use aspects to recognize entire objects. The advantage of this approach is that since the number of qualitatively different primitives is generally small, the number of possible aspects is limited and, more important, *independent* of the numbers of objects in the database. The disadvantage is that if a primitive is occluded from a given 3-D viewpoint, its projected aspect in the image will also be occluded. Thus we must accommodate the matching of occluded aspects, which we accomplish by use of a hierarchical representation called the *aspect hierarchy*.

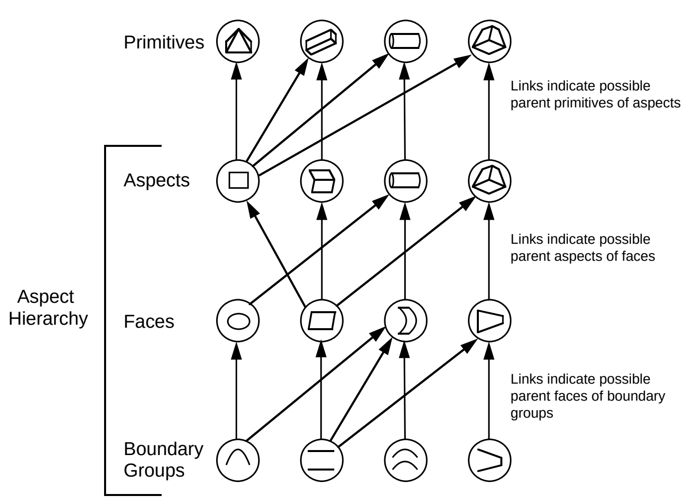


Fig 1. The Augmented Aspect Hierarchy.

The aspect hierarchy includes three levels, consisting of the set of *aspects* that model the chosen primitives, the set of component *faces* of the aspects, and the set of *boundary groups* representing all subsets of contours bounding the faces. The ambiguous mappings between the levels of the aspect hierarchy were originally captured in a set of upward conditional probabilities[1][2], mapping boundary groups to faces, faces to aspects, and aspects to primitives. Fig. 1 illustrates a portion of the augmented aspect hierarchy.

3 3-D Shape Recovery Based On The Hybird Representation

In the context of the aspect hierarchy, a framework of 3-D shape recovery has been proposed. Working with the bottom-up style, the framework consists of three stages, viz. face recovery, aspect recovery and primitive recovery.

rom an analysis of the conditional probabilities[1], we can conclude that the best mapping to the aspect is from faces rather than from the boundary groups. This suggests that faces are an appropriate starting point in the 3-D shape recovery process so as to extract a set of contours from the image. Once a set of contours has been extracted from the image, the next step is to partition the contours at significant curvature discontinuities. The segmented contours are captured in a contour graph in which nodes represent junctions, and arcs are the actual bounding face contours. Given the contour graph representation of an input image, we can also its corresponding face graph in which nodes represent faces and arcs represent face adjacencies. For the faces extracted, a classification process follows to label faces. The classification of a face consists of comparing its graph to those graphs representing the faces in the aspect hierarchy. If there is an exact match, then we immediately generate a *face hypothesis* for that image face, identifying the label of the face. If, due to occlusion, there is no match, we must descend to the boundary group level of the aspect hierarchy. We then compare *subgraphs* of the graph representing the image face to those graphs at the boundary group level of the aspect hierarchy. For each subgraph that matches we generate a face hypothesis with a probability determined by the appropriate entry in the conditional probability matrix mapping boundary groups to faces.

Once the face graph(FG) has been extracted, the following work is to generate possible aspect coverings for aspect recovery. We can formulate the problem of extracting aspects as follows: Given a region topology graph and a set of face hypotheses(labels) at each region, finding an *aspect covering* of FG using aspects in the aspect hierarchy, such that no region is left uncovered and each region is covered by only one aspect. There is no known polynomial time algorithm to solve this problem[1][2]. In this paper, we will present a parallel voting scheme for conquering the problem. Using this scheme, we may recover all aspect instantiations(AINs). We will describe this scheme in detail in the following section.

From an *aspect covering* of faces in the image, a set of primitive labels and their corresponding probabilities is inferred (using the aspect hierarchy) from each AIN. Primitive recovery is formulated as a search through the space of primitive labelings of the aspects in the aspect covering, guided by a heuristic based on the probabilities of the primitive labels. Each solution, or *primitive covering*, found by the search is a valid primitive interpretation of the input image. The connections in the resulting primitive covering are then hypothesized. Each primitive covering is a graph in which nodes represent qualitatively defined primitives and arcs specify primitive connectivity. Encoded in each primitive is the aspect in which it is viewed; the aspect, in turn, encodes the faces that were used in instantiating the aspect, while each face specifies those contours in the image used to instantiate the face.

In this framework, aspect recovery is the most difficult in the aforementioned three stages. So we will focus our attention on this problem in the rest of the paper.

4 The Distributed Representation of Aspects

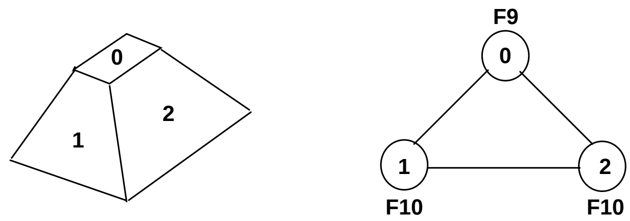
As described in section 2, an aspect in the aspect hierarchy is represented as a model which consists of an labeled graph; nodes represents faces, arcs represent face adjacencies and arc labels indicate those contours shared by adjacent faces. Using such a representation, during aspect recovery, we have to encounter the problem called *partition into isomorphic subgraphs* which is a NP problem. After investigation, we found that such a centralized representation is hard to be employed to develop a parallel method of aspect recovery since the global information is integrated in the single model and components depend on each other. To develop a parallel voting scheme, we are going to present a new distributed representation of aspects called the submodel representation for encoding voting knowledge.

For the proposed voting scheme, we expect that it can work parallelly by accumulating evidence of each face in a face graph and voting for a specified aspect simultaneously. After investigation, we think it is necessary and sufficient for a face to accumulate *local* evidence from face types of it and its immediate neighbors as well as connections between the face and its immediate neighbors. There are at least two facts to support this opinion. contributions in Psychology have shown that the proximity is the necessary condition for grouping features into a meaningful structure or *perceptual grouping.* On the other hand, there is always one face connecting with other face(s) in every aspect model except aspect 34 (numbered in the previous representation[1]), which can capture most of global information about an aspect. Based on two facts, we can develop a new representation of aspects to support the parallel voting scheme.

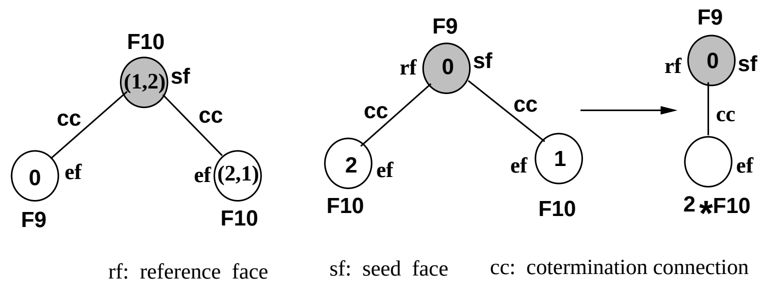
In order to develop the submodel representation of aspects, we define new labels for faces and arcs in an aspect at first. For each face in an aspect, we focus on it once a time and prune the graph representing the aspect so that only the focused face and its immediate neighbors be left in a two-layer tree. For such a tree, we call root node or focused face *seed* face and call its leaf node(s) or neighboring face(s) *evidence* face(s). As a result, we assign the label **sf** to the seed face and the label **ef** to an evidence face. As for a connection between faces in an aspect, we also define a connection type called *co-termination connection* labeled with **cc** which means a contour shared by two faces in an aspect must be co-terminated if the contour is a line segment. That is, instead of a part of the line segment, the entire one must be a component of both neighboring faces in an aspect. After adding some labels in an aspect model, we can give a algorithm to produce submodels of an aspect. Assume that aspect A_p consists of K faces $f_1, ..., f_K$.

Algorithm 1: (Produce submodels of an aspect)

1. Initialization: $i \leftarrow 1$.
2. Assign f_i to a *feed* face and prune the graph representing A_p so that only f_i and its immediate neighbors, say $f_{i_1}, ..., f_{i_s}$, are left as a tree T_{f_i}, are left as a tree.
3. For the tree T_{f_i}, label f_i with **sf** and $f_{i_1}, ..., f_{i_s}$ with **ef**.
4. According to the definition of connection, assign the corresponding label (e.g. **cc**) to the arc between f_i and f_{i_j} $(j = 1, ..., s)$.
5. $i \leftarrow i + 1$, if $i \leq K$ then go to 1.
6. Check the sequence $T_{A_p} = \{T_{f_1}, ..., T_{f_K}\}$ and remove some tree(s) from T_{A_p} so that all trees in T_{A_p} is different each other in configuration including structure or attributes(face types, labels of faces and connections). So a new sequence is created, say $T'_{A_p} = \{T_{f_1}, ..., T_{f_{K'}}\}$ $(K' \leq K)$.
7. For each tree in T'_{A_p}, merge leaf nodes labelled **ef** with the same configuration, viz. the same face type and the same type of connection with the root node. Relabel the final node labelled **ef** with the number of merged nodes and face type instead of the previous face type.



(a) The previous definition of Aspect 31.



(b) The current definition using local features

Fig. 2 An example used to state the encoding mechanism.

After running with this algorithm, we can achieve the distributed representation of aspect A_p. So we call each tree in T'_{A_p} a submodel of aspect A_p. Fig. 2 illustrates a process that the algorithm works on aspect 31 for instance. Fig. 2(a) shows the previous model and Fig. 2(b) gives a process of producing submodels of aspect 31. In the example, two submodels are generated as the distributed representation of aspect 31.

It should be pointed out that for each aspect, we need to define a *reference* face and label it with **rf** within its submodel(s) for locating the aspect instantiations of the aspect during voting. As a reference face, one face must be the seed face of a submodel. Within all seed faces of an aspect, we assign one of them as the reference face according to the following criteria; (1) The face must connect all other component faces appearing in the previous aspect model.[2] (2) The face type of the face should be different from ones of other component faces in an aspect as possible as it can. (3) Only one submodel's seed face is defined as in all submodels of its even though there are more than one face satisfying constraints described in (1) and (2).[3] Using these criteria, in Fig. 2, we may assign the seed face with $F9$ to the reference face and label it with **rf**. Moreover, we have developed the distributed representation of all 40 aspects[8] with the aforementioned method[9].

5 A Parallel Voting Network For Aspect Recovery

5.1 The Extracting Input Array from Face Graph

After face recovery, a face graph(FG) can be achieved in which each face is labelled with either a face type or multiple ones. Based on the aspect hierarchy and the contour graph, we can extract easily a vector for each face type of a face as follows:

$$(\mathbf{ft_i},\ \mathbf{CS_i},\ \mathbf{P_i}) \tag{1}$$

where ft_i is the face type, CS_i is a set of all *seed* contour subsets of the face which can be used for generating ft_i and P_i is the set of corresponding probabilities of ft_i generated by each subset in CS_i. When a face in FG has more than one face type, it may own a set of aforementioned vectors

[2]The selection of reference face for aspect 34 is an exception since we cannot find such a face in its previous definition. Instead we will deal with it as a special case in the section of discussions.

[3]When faces in an aspect is situated in *rotation symmetry*, each face is able to become a reference face. As a result, we simply choose one of them as the reference.

henceforth called *input-array*. In general, if face *k* in FG has *t* face types, its input-array is represented as follows:

$$\begin{pmatrix} F_k(1) & (CS^{(1)}_{F_k(1)}, \ ..., \ CS^{(k_1)}_{F_k(1)}) & (P^{(1)}_{F_k(1)}, \ ..., \ P^{(k_1)}_{F_k(1)}) \\ F_k(2) & (CS^{(1)}_{F_k(2)}, \ ..., \ CS^{(k_2)}_{F_k(1)}) & (P^{(1)}_{F_k(2)}, \ ..., \ P^{(k_2)}_{F_k(2)}) \\ ...... & ...... & ...... \\ F_k(t) & (CS^{(1)}_{F_k(t)}, \ ..., \ CS^{(k_t)}_{F_k(1)}) & (P^{(1)}_{F_k(t)}, \ ..., \ P^{(k_t)}_{F_k(t)}) \end{pmatrix} \qquad (2)$$

where $t \geq 1$ and $k_i \geq 1$, $i = 1, ..., t$. All input-array of a face graph will constitute the unique information source of voting.

5.2 The Structure of the Voting Network

In the aspect hierarchy, all 40 aspects can be classified into two types, viz. aspects composed of at most 2 faces and aspects composed of at least 3 faces. For the former, we can deal with this case as follows: (1) When an aspect consists of only one face, we can easily determine a face is an AIN of this aspect if it has a face type consistent with the one required in the aspect. (2) When an aspect consists of two faces, what we need to do is only to locate all faces consistent with the face labelled with **rf** in the submodel and to enumerate all AINs of this aspect when such a face has neighbor(s)(as evidence face(s)) satisfying constraints defined in the submodel. Obviously, all tasks in this case can be completed parallelly. For the latter, it will be insufficient if we simply use this method and we shall discuss this case in the rest of this section.

For aspects composed of at least 3 faces, we introduce a three layers network to it. Since each aspect hypothesis is independent each other, we can deal with them parallelly by mapping them onto several independent subnetworks called *clusters*. A cluster is a mechanism used for finding all AINs of the aspect from an input image. Fig. 3 illustrates a general structure of such a cluster corresponding to aspect A_p.

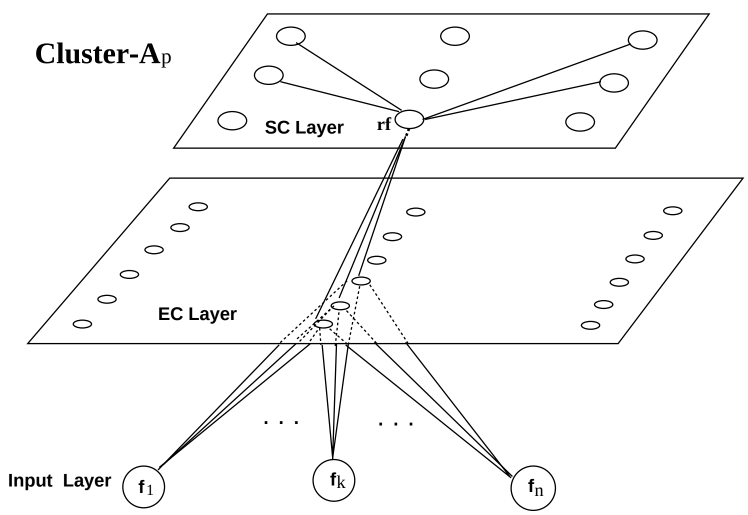


Fig. 3 The general structure of a cluster corresponding to aspect A_p.

Suppose that aspect A_p consists of k submodels. For the purpose of parallel computation, we establish a correspondence between each face in a face graph(FG) and each seed face in submodels of the aspect base on the fact that each face in FG is a possible candidate of each component face of the aspect before processing. In addition, if we focus one of faces in FG graph on the seed face, other faces may be evidence faces of the face. Therefore, we can design this cluster naturally. In *input* layer, each cell maps a node in FG in which there is a buffer storing its input-array with the form of (2). If there are n nodes in FG, n cells are required in input layer. The middle layer is usually a 2-D array[4] used to accumulate evidence, called *EC layer*. In this array, the rows correspond to all

[4]It may be a 1-D vector of cells if an aspect consists of only one submodel in which there is only one evidence node.

evidence faces in submodels of A_p, and the columns correspond to all n faces in FG. The number of cells in EC layer is $n \times N$ ($N = \sum_{t=1}^{k} s_t$), if there are s_1, s_2, ..., s_k evidence faces in k submodels of A_p, respectly. Each *EC* cell may connect all *n* cells in input layer for collecting evidence parallelly and connects only one cell in the higher layer as an evidence cell. Since evidence faces are independent each other in the submodel representation, there is no lateral connection among them in *EC* layer. The output layer is usually a 2-D array to evaluate scores of accumulated evidence for tolling the final vote, henceforth called *SC* layer. In this layer, the rows correspond to *seed* faces of all submodels of A_p, and the columns also correspond to *n* faces in FG. For aspect A_p, hence, there are $n \times k$ cells in SC layer. According to the submodel representation, Cell (i, j) in *SC* layer only connects *evidence* cells belonging to it, viz. cell $(i, T+1)$, ..., cell $(i, T+s_j)$ ($T = \sum_{t=1}^{j-1} s_t$) in EC layer. In addition, for the purpose of collecting voting results, there may be lateral connections in SC layer. If a SC cell is also labelled **rf**, it may connect all cells which are not located in both the same row and column with itself.[5] In addition, the general structures of EC cell (connecting n input cells) and SC cell (connecting s EC cells) are illustrated in (a) and (b) of Fig. 4.

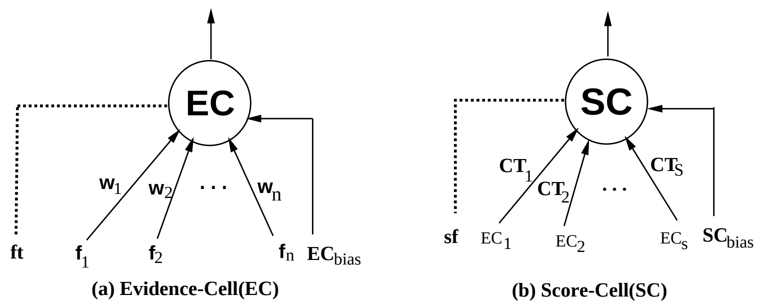


Fig. 4 The structures of EC cell and SC cell.

5.3 The Computational Mechanism of A Cluster

In a cluster, the link between a cell in EC layer and a cell in input layer is used to reflect whether the face in input layer can become an evidence face consistent with the requirements of the evidence face to supply necessary evidence for the vote of a seed face. For the purpose of computation, we define the the link between cell x in input layer and cell (i, j) in EC layer as follows:

$$link[(i,j),x] = \begin{cases} 1, & \text{if face } x \text{ satisfies with the determinant condition.} \\ 0, & \text{Otherwise.} \end{cases} \tag{3}$$

where the determinant condition will be defined according to the definition of aspects. Intuitively, for the submodel of a specified aspect, face x ($x \neq i$) can become an evidence node corresponding to j to support face i to vote for the aspect only if face x has both a specified face type and a connecting relation with face i which are predefined in the submodel. Let $EC_{bias}(j)$ denote the number of nodes with the same attributes($face\ type,\ connecting\ type$) in the submodel. Obviously, all links between input and EC layer can be computed parallelly to accumulate evidence as followings:

$$V_{EC}(i,j) = \frac{P[FT(sf)|f_i]}{EC_{bias}(j)} \sum_{\substack{x=1 \\ x \neq i}}^{n} link[(i,j),x]P(ft|f_x) \tag{4}$$

where $P(ft|f_x)$ is the probability required as the evidence node corresponding to evidence face j in the submodel, deriving from the probability matrix related to the aspect hierarchy. $P[FT(sf)|f_i]$ is the probability hat face i can become the seed face of the submodel, deriving from the corresponding input-array.

In the SC cell, cell (i, k) implies that face i corresponds to the seed node of submodel k in a specified aspect. It is used to compute the score that face i is a component of a specified aspect

[5]There is an exception that a SC cell with **rf** may connect with all SC cells expect itself when an aspect consists of only one submodel.

according to evidence accumulated in EC layer. Suppose the seed node connects s_k evidence nodes in submodel k, the score can be computed as follows:

$$V_{SC}(i,k) \; = \; h[\frac{1}{s_k}\sum_{j=1}^{s_k} V_{EC}(i,\sum_{t=1}^{k-1} s_t + j) - SC_{bias}(k)] \quad s_k, k \geq 1. \tag{5}$$

$$h(x) \; = \; \begin{cases} 1, & x > 0 \\ -1, & x \leq 0 \end{cases}$$

where $SC_{bias}(k)$ is a voting threshold depending upon the number of evidence nodes in submodel k and prior knowledge about the image in the case of expected recognition.

6 Finding All Aspect Instantiations Using The Parallel Network

In the previous section, we described a parallel voting network for aspect recovery. In this section, we shall describe how to use this network to extract all aspect instantiation(AINs) from FG.

6.1 Generating A Minimal Network for The Face Graph

For a face graph(FG), it usually may not include all *40* aspects. As we have known, the aspect hierarchy is able to provide a powerful constraint for removing impossible aspects and generate a *minimal* network. This process is also called *filtering*. According to the face-to-aspect mapping in the aspect hierarchy, using face type $F_k(i)$ of face F_k in FG ($i = 1, ..., t;\; k = 1, ..., n$), we can generate a set of aspect hypotheses $AH_{F_k(i)}$ as follows,

$$AH_{F_k(i)} \; = \; \{AH^{(1)}_{F_k(i)}, \; ..., \; AH^{(k_i)}_{F_k(i)}\} \tag{6}$$

For FG, the set of aspects hypotheses corresponding a minimal network is a *minimal* set which only includes possible aspect hypotheses generated by all faces in FG. For generating such a set, first, we use the aforementioned mapping to achieve all possible aspect hypotheses of each face in FG. Then we can achieve a set including the minimal set, AH'_{FG}, as follows:

$$AH'_{FG} \; = \; \bigcup_{k=1}^{n}\bigcup_{i=1}^{t_i} AH_{F_k(i)} \tag{7}$$

Using the information of face types, moreover, we can further remove all false aspect hypotheses from AH'_{FG}. Using input-array of faces in FG, we can obtain a set of face types $F_{type}(FG)$. On the other hand, for $AH_i \in AH'_{FG}$ ($i = 1, 2, ..., t_i$), we can get all face types of its component faces, say $F_{type}(AH_i)$. When $F_{type}(AH_i) \subseteq F_{type}(FG)$, AH_i remains as a true hypothesis; otherwise, it is removed from AH'_{FG} as a false one. Using this method, we can achieve the minimal set of aspect hypotheses, say AH_{FG}, used to generate the minimal network corresponding to FG.

6.2 Determining Links in A Cluster

For using the network, it is important to determine the link values between input layer and EC layer using the input-array and submodels of the aspect corresponding to the current cluster. Moreover, we also need to determine the lateral connection in SC layer.[6]

According to (3), we can determine the link values between a cell of input layer and a cell of EC layer. Concerning the connection between any two faces in a face graph, we can describe them as four possible cases. For face x and y, suppose face x has a face type $F_x(i)$ and face y has a face type $F_y(j)$, contours $CS_{F_x(i)}$ and $CS_{F_y(j)}$ are seed contour sets used to generating $F_x(i)$ and $F_y(j)$ in contour sets of two faces, respectively. Let M is *adjacency* matrix of the face graph and we describe three cases as follows:

Case 1: Face x is not adjacent to face y at all in the face graph, namely $M(x, y) = 0$. This case is also called *no connection*.

[6] Links between EC layer and SC layer in a cluster are fixed in advance since we establish EC layer and SC layer simply by duplicating all sumbodels of the aspect corresponding the cluster for each face in face graph.

Case 2: Face x is adjacent to face y illegally viz. $M(x, y) \neq 0, CS_{F_x(i)} \bigcap CS_{F_y(j)} = \phi$. This case is also called *illegal connection*.

Case 3: Face x is adjacent to face y viz. $M(x, y) \neq 0$, $CS_{F_x(i)} \bigcap CS_{F_y(j)} \neq \phi$. More intuitively, contour(s) shared by two faces is / are seed contour(s) of both faces for generating $F_x(i)$ and $F_y(j)$. This case is called *basic connection*.

Case 4: Face x is adjacent to face y viz. $M(x, y) \neq 0$ and both of them have line-segment contours, $CS_{F_x(i)} \bigcap CS_{F_y(j)} \neq \phi$. Moreover, suppose

$$CS_{F_x(i)} \bigcap CS_{F_y(j)} = cs \subseteq CS^{(p)}_{F_x(i)}, \;\; CS_{F_y(j)} \bigcap CS_{F_x(i)} = cs \subseteq CS^{(q)}_{F_y(j)}, \;\; cs \neq \phi.$$

where $CS^{(p)}_{F_x(i)}$ and $CS^{(q)}_{F_y(j)}$ are the entire collinear contour sets including cs in $CS_{F_x(i)}$ and $CS_{F_y(j)}$ so that

$$T \leq \frac{length(cs)}{length(CS^{(p)}_{F_x(i)})} \leq T^{-1} \; and \; T \leq \frac{length(cs)}{length(CS^{(q)}_{F_y(j)})} \leq T^{-1} \quad 0.5 \ll T < 1.0$$

More intuitively, the shared contour(s) between faces are complete collinear segments for both faces instead of a part of them; that is, endpoints of line segments belonging to different faces should coincide closely. This case is also called *co-termination connection*.

Obviously, the last case described here exists only in two adjacent faces bound with line-segment contours. Thus the determinant condition can be defined so that link values between input layer and EC layer are determined easily by checking connection between the focussed face and all other faces in face graph parallelly as follows: (1) If the case is *no connection* or *illegal connection* then the link value is assigned to 0.0. (2) If the case is *basic connection*, then the link value is assigned to 1.0 only when two adjacent faces are bound with curve contours; otherwise, it is assigned to 0.0. (3) If the case is *co-termination connection* for line-segment contours, then the link value is assigned to 1.0.

In SC layer, a cell with **rf** may connect all other SC cells without **rf** except those aspects consisting of only one submodel. In fact, there is a relation between lateral link values in SC layer and link values between input layer and EC layer so that we can determine them. by means of link values between input layer and EC layer achieved before. Suppose there are cell (i, x) with label **rf** and cell (j, y) without label **rf** $(i \neq j,\ x \neq y)$ in SC layer[7] and $link_{SC}[(i, x), (j, y)]$ denotes the lateral link value between the two cells. In the current network structure, we always arrange cells with **rf** on row 1 of SC array, viz. $x = 1$ so that we can describe the relation as follow:

$$link_{SC}[(i, 1), (j, y)] \; = \; link[(i, y), j] \tag{8}$$

Using this relation, we can determine the lateral link values in SC layer.

6.3 Generating Aspect Instantiations

Once all links values are determined, we can use this network to compute scores parallelly to vote for all possible aspects appearing in AH_{FG} based on the computational mechanism described previously. when we have voting results, we will encounter another problem how to collect these voting results and how to locate an AIN of this aspect in the face graph. Since we have the reference face and lateral links in SC layer, we can design a collecting algorithm for generating all AINs of this aspect. For an aspect with $s + 1$ submodels, we have the algorithm as follows:

Algorithm 2: (Collecting Algorithm for Generating AINs)

1. Initialization: **AIN** $\leftarrow \Phi$.
2. In SC layer, **RF** $\leftarrow \{Cell(1, x_1), ..., Cell(m, x_m)\}$, only if $Cell(1, x_i) > 0$; $(i = 1, ..., m)$ $m \leq n$. Intuitively, collecting all activated **rf** cells in SC layer and put them into a set called **RF**.

[7] For the case of *rotation symmetry*, SC layer consists of 1-D vector of cells instead of 2-D array of cells, viz. $i \neq j,\ x = y$.

3. For $Cell(1, x_i)$, $(i = 1, ..., m)$ and row k $(k = 2, ..., s+1)$, $VS(x_i, k) \leftarrow \{Cell(k, y_1), ..., Cell(k, y_p)\}$, only if $Cell(k, y_j) > 0$ and $link[(1, x_i), (k, y_j)] > 0$; $(j = 1, ..., p)$, $p \leq n$. Intuitively, based on a **rf** cell, collecting all activated cells without **rf** which connect with the **rf** cell in each row of SC layer and put them into corresponding sets, respectively.
4. Using the submodel including **rf** node, we can find each number label in its **ef** nodes, say $n_1, ..., n_t$. For $Cell(1, x_i) \in \mathbf{RF}$ $(i = 1, ..., m)$, generating an AIN called **ain** by selecting n_k cells from $VS(x_i, k)$ $(k = 1, ..., t)$. When $\mathbf{ain} \notin \mathbf{AIN}$, $\mathbf{AIN} \leftarrow \mathbf{AIN} \bigcup \mathbf{ain}$.

7 Generate Aspect Coverings of The Face Graph

Once we achieved all aspect instantiations(AINs), we must be able to enumerate, in decreasing order of likelihood, all aspect coverings since we cannot guarantee that a given aspect covering represents a correct interpretation of the scene. Concerning likelihood, an objective is defined as getting a covering by using minimum number of AINs and the highest prior probabilities of each component faces in each AIN. Using likelihood we can rank AINs, which is a constraint-satisfaction problem. If we employ a neuron to represent an AIN, the problem can be tractable by minimizing the energy function in the standard *continuous* Hopfield net[10] as follows:

$$E = -\frac{1}{2}\sum_i \sum_{\substack{j \\ j \neq i}} T_{ij} V_i V_j - \sum_i I_i V_i \tag{9}$$

where the first term in this equation is a global consistent constraint of using minimum number of aspects, and the second term reflects the contribution of all AINs to the whole objective mentioned above. For AIN_i, We may represent it with a quadruplet (i, AT_i, S_i, PP_i), where S_i is the set of component nodes in AIN_i, AT_i is *aspect type* of AIN_i and PP_i is the set of prior probabilities of nodes in AIN_i. Therefore, T_{ij} is defined as follows: there is a mutual stimulation between two AINs if they share no vertex in FG; otherwise, there is a mutual inhibition between them. I_i can be measured by the number of faces in AIN_i and prior probabilities of which component faces in AIN_i can be instantiated as AT_i.

$$T_{ij} = \begin{cases} 1, & \text{if } S_i \bigcap S_j = \phi \\ -1 & \text{if } S_i \bigcap S_j \neq \phi \end{cases} \tag{10}$$

$$I_i = w_1 \frac{N(S_i)}{5} + w_2 \frac{\sum_{k=1}^{N(S_i)} P_k}{N(S_i)}; \quad \sum_i w_i = 1, \ P_k \in PP_i. \tag{11}$$

The global minimum is achieved through *mean field annealing*. In the steady state, the output of each neuron expresses the optimal likelihood of the corresponding AIN. As a result, the *enumeration* of aspect coverings in decreasing order of likelihood can be easily based on likelihood of each AIN.

8 Experiment

In this section, we use a line drawing to justify our voting scheme. For a line drawing, its face graph is easily achieved. In the current example, its contour graph and face graph are illustrated in Fig. 5.

Using face-to-aspect mapping in the aspect hierarchy, we can generate the possible aspect hypotheses(AHs) for each face type appearing in the face graph as follows:

$AH_{F9} = \{A4, A19, A20, A21, A26, A27, A28, A29, A30, A31, A33, A35, A36, A37, A38, A39, A40\}$

$AH_{F10} = \{A5, A21, A22, A31, A35, A40\}$

As a result, all possible AHs for the face graph is included in AH'_{FG},

$AH'_{FG} = AH_{F9} \bigcup AH_{F10} = \{A4, A5, A19, A20, A21, A26, A27, A28, A29, A30, A31, A33, A35, A36, A37, A38, A39, A40\}$

Furthermore, we can use *filtering* processing in section 6 to remove false hypotheses from AH'_{FG}. As a result, the minimal set of aspect hypotheses is achieved as follows:

$AH'_{FG} = \{A4, A5, A20, A21, A22, A29, A30, A31, A35, A40\}$

From both face graph and contour graph, we can generate an input-array for each face in the face graph as follows:

$f0: [F9\ \ (0\ 1\ 9\ 8)\ \ (1.00)]$

$f1: [F9\ \ (9\ (2\ 3)\ 4\ 10)\ \ (1.00)]$

$f2: [F9\ \ (5\ (6\ 7)\ 8\ 10)\ \ (1.00)]$

$$f3: \begin{bmatrix} F9 & ((11\ 12\ 13)\ (12\ 13\ 14)\ \ (13\ 14\ 15)) & (0.86\ 0.86\ 0.86) \\ F10 & ((11\ 12\ 13)\ (12\ 13\ 14)\ \ (13\ 14\ 15)) & (0.14\ 0.14\ 0.14) \end{bmatrix}$$

$f4: [F9\ \ (13\ 16\ (17\ 18\ 19\ 20\ 21\ 22)\ \ (1.00)]$

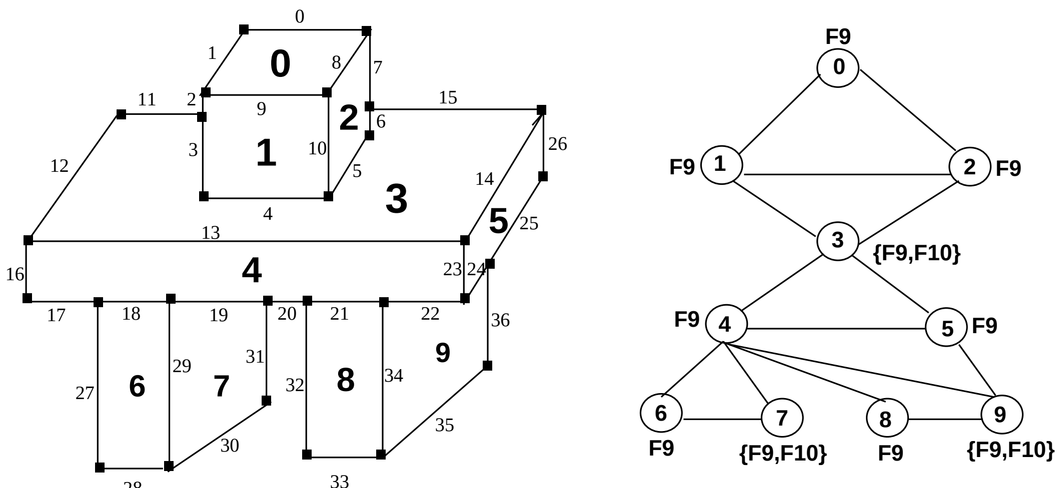
$f5: [F9\ \ (14\ 23\ (24\ 25)\ 26)\ \ (1.00)]$

$f6: [F9\ \ (18\ 27\ 28\ 29)\ \ (1.00)]$

$$f7: \begin{bmatrix} F9 & (29\ 30\ 31) & (0.86) \\ F10 & (29\ 30\ 31) & (0.14) \end{bmatrix}$$

$f8: [F9\ \ (21\ 32\ 33\ 34)\ \ (1.00)]$

$$f9: \begin{bmatrix} F9 & (34\ 35\ 36) & (0.86) \\ F10 & (34\ 35\ 36) & (0.14) \end{bmatrix}$$



(1) Contour Graph (2) Face Graph

Fig. 5 The contour graph and face graph of a line drawing.

As pointed out previously, the voting network only serves for aspects composed of at least 3 faces. For such aspect hypotheses in the current minimal set, we can use the proposed network to complete the voting parallelly. Results of accumulating evidence in EC layer and voting in SC layer are illustrated in Fig 6.

In addition, we can enumerate all aspect instantiations(AINs) directly for aspects composed of at most 2 faces as follows:

$AIN(A_4) = \{f0, f1, f2, f3, f4, f5, f6, f7, f8, f9\}$ $\quad AIN(A_5) = \{f3, f7, f9\}$

$AIN(A_{20}) = \{(f0, f1), (f0, f2), (f1, f2), (f3, f4), (f3, f5), (f4, f5), (f6, f7), (f8, f9)\}$

$AIN(A_{21}) = \{(f3, f4), (f3, f5), (f6, f7), (f8, f9)\}$ $\quad AIN(A_{22}) = \{\ \}$

Moreover, we can use algorithm 2 and *physical* features of aspects to locate AINs of aspects composed of at least 3 faces according to lateral connections in SC layer illustrated in Fig. 7.

$AIN(A_{29}) = \{(f0, f1, f2), (f3, f4, f5)\}$ $AIN(A_{30}) = \{\ \}$ $AIN(A_{31}) = \{\ \}$ $AIN(A_{35}) = \{\ \}$ $AIN(A_{40}) = \{\ \}$

Using a continuous Hopfield net, we can achieve likelihood of All AINs by using $w_1 = 0.70$. Therefore, we can order all AINs in decreasing order of likelihood and put them into **H**.

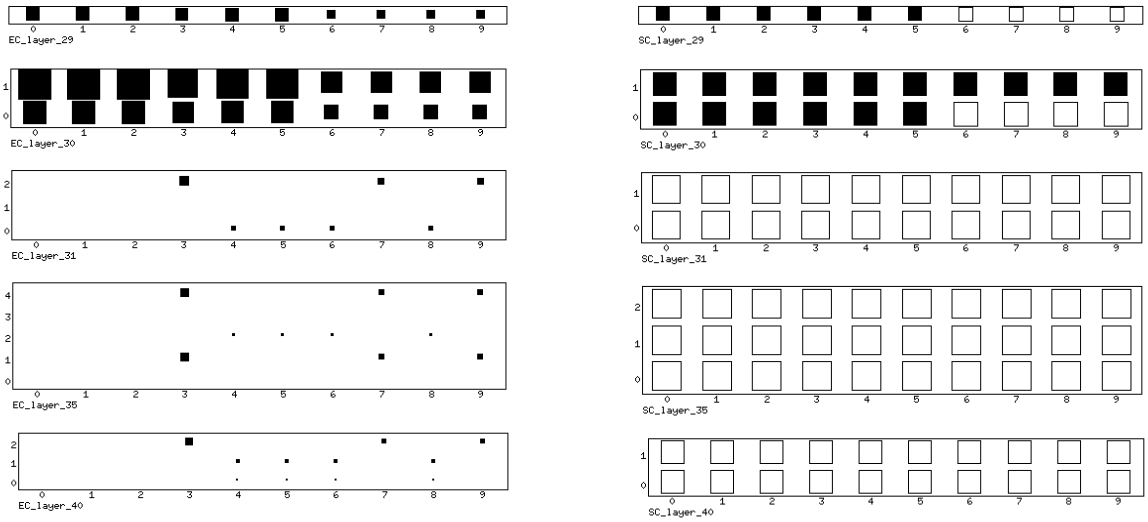
$$\mathbf{H} = \{[A29, (f0, f1, f2)],\ [A29, (f3, f4, f5)],\ [A20, (f0, f1)],\ [A20, (f0, f2)],\ [A20, (f1, f2)],\ [A20, (f4, f5)],\ [A20, (f6, f7)],\ [A20, (f8, f9)],\ [A20, (f3, f4)],\ [A20, (f3, f5)],\ [A4, f0],\ [A4, f1],\ [A4, f2],\ [A4, f4],\ [A4, f5],\ [A4, f6],\ [A4, f8],\ [A4, f7],\ [A4, f9],\ [A4, f3],\ [A21, (f6, f7)],\ [A21, (f8, f9)],\ [A21, (f3, f4)],\ [A21, (f3, f5)],\ [A5, f7],\ [A5, f9],\ [A5, f3]\}$$

From **H**, we can enumerate the first three coverings with the highest likelihood.

$Covering(1) = \{[A29, (f0, f1, f2)],\ [A29, (f3, f4, f5),\ [A20, (f6, f7)],\ [A20, (f8, f9)]\}$

$Covering(2) = \{[A29, (f0, f1, f2)],\ [A29, (f3, f4, f5),\ [A20, (f6, f7)],\ [A4, f8],\ [A4, f9]\}$

$Covering(3) = \{[A29, (f0, f1, f2)],\ [A29, (f3, f4, f5),\ [A20, (f6, f7)],\ [A21, (f8, f9)]\}$



(1) (2)

Fig. 6 Results of voting process. (1) The result of accumulating evidence in EC layer (2) The result of voting in SC layer.

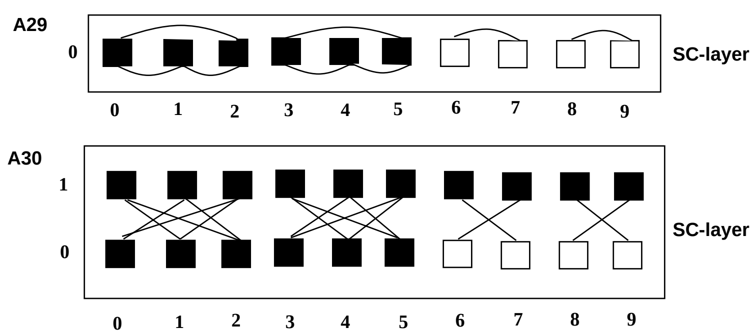


Fig. 7 The lateral connection of cluster A29 and A30 in SC layer.

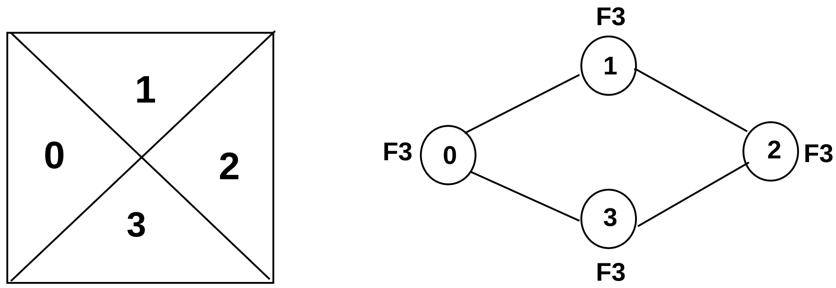
9 Discussion

In the section, we also pay our attention to the distributed representation of aspects since it is the basis of the proposed method.

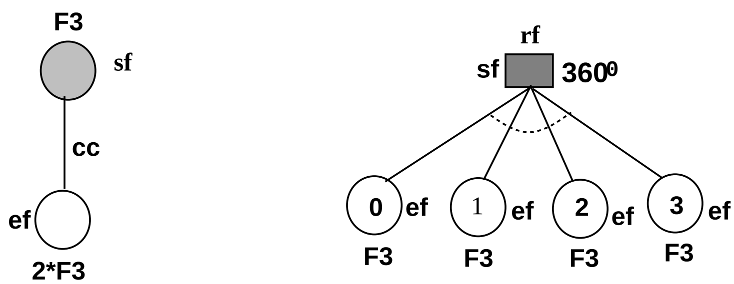
In section 4, we mentioned Aspect 34 is an exception of producing submodels using algorithm 1. The problem lies in that we cannot find a face in the previous model of Aspect 34 which satisfies condition (1) defined in criteria of choosing a reference face. That is, there is no face connecting all other component faces in Aspect 34 so that we cannot capture any global information. Fig. 8(a) shows its previous model. In the current method, we modify the original aspect by adding a *virtual* face as **rf** illustrated in in Fig. 9b). The original model is encoded as only one submodel according to algorithm 1 and the adding submodel is defined in Fig. 8(b) in which **rf** is a virtual face corresponding to an endpoint shared by four faces and the sum of four angles around the shared endpoint in four faces must be within $360^0 \pm \theta$. Since there are three endpoints in a component face with face type $F3$ in Aspect 34, as an exception, each face will correspond to three cells with label **rf** in SC layer. Moreover, we compute the score of such a cell as follows:

$$V_{SC}^{34}(1,(i,j)) \;=\; \begin{cases} 1, & \text{if endpoint } j \text{ in face } i \text{ is a converging point of at least 4 line segment.} \\ 0, & \text{Otherwise.} \end{cases} \tag{12}$$

where $i = 1, 2, ..., n;\;\, j = 1, 2, 3.$



(a) the original definition of aspect 34.



(b) the submodels of aspect 34.

Fig. 8 The submodels of aspect 34 in the encoding mechanism.

As pointed out previously, the generation of an AIN is a process of packing component faces recovered from the face graph and locating it in the face graph. However, there are some overlappings among distributed representation of aspects if we only consider their subgraph representation. For example, Aspect 30 is just a pure subset of Aspect 29. It results in a new problem that a *false* AIN of Aspect 30 is generated when we actually find an AIN of Aspect 29 from an image. Fortunately, we can conquer this problem by using some *physical* features of aspects. In the current example, we may use the location relation of contours shared by two faces, viz. parallel vs intersection, to determine further whether an AIN of Aspect 29 or Aspect 30 is generated when the score is high enough[7]. In fact, the enforced representation has been used in the previous experiment.

10 Conclusions

We have proposed a parallel voting scheme for aspect recovery in the context of the hybrid object recognition[1]. Unlike the previous approach, the parallel voting network simply uses *local* constraints for extracting aspect instantiations from the face graph of an image. Moreover, neural computation techniques are also employed to rank likelihood such that we can enumerate all aspect coverings in a decreasing order of likelihood. As a consequence, the efficiency of this method has been demonstrated with our simulations.

Acknowlegments

The authors are grateful to S.J. Dickinson for constructive discussions. This work was basically motivated by his ideas on the use of voting techniques for aspect recovery. The work reported in this paper was partially supported by Japan Society for The Promotion of Science under the grant of Postdoctoral Researches for Foreign Researchers and Fund of Monbusho Grant-in-Aid FY 1993.

References

1. S.J. Dickinson, A.P. Pentland and A. Rosenfeld, *"3-D Shape Recovery Using Distributed Aspect Matching*, IEEE Transaction on PAMI, Vol. 14, No. 2, Feb. 1992, 174-190.
2. S.J. Dickinson, A.P. Pentland and A. Rosenfeld, *"From Volumes to Views: An Approach to 3-D object Recognition,"* Computer Vision, Graphics, and Image Processing: Image Understanding, Vol. 55, No. 2, March, 1992, 130-154.
3. S.J. Dickinson, H.I. Christensen and J. Tsotsos, *"Active Object Recognition Integrating Attention and Viewpoint Control,"* Proceedings of ECCV'94, Stockholm, Sweden, May 1994.
4. M.R. Garey and D.S. Johnson, *Computers And Intractability*, W.H. freeman and Company, 1979.
5. K. Chen and M. Ishikawa, *"3-D Shape Recovery by Incorporating Context — A Connectionist Approach"*, Proceedings of IJCNN'93, Nagoya, Oct. 1993, 1180-1183.
6. J. Illingworth and J. Kittler, "A Survey of The Hough Transform" CVGIP 44, 1988, 87-116.
7. Y. Lamdan and H.J. Wolfson, "Geometric Hashing: A General And Efficient Model-Based Recognition Scheme," Proc. ICCV'88, 1988, 238-249.
8. I. Biederman, *" Human Image Understanding: Recent Research And A Theory,"* Computer Vision, Graphics and Image Processing, Vol. 32, 1985, 29-73.
9. K. Chen, S. Dickinson and M. Ishikawa, *"3-D Shape Recovery Based on A Parallel Voting Network,"* Research Memo, Kyutech Institute of Technology, Oct., 1993.
10. J.J. Hopfield and D.W. Tank, "*'Neural' Computation of Decisions in O ptimization Problems*," Biol. Cybern. 52, 1985, pp. 47-58.

Ke Chen received the B.S. and M.S. degrees in Computer Science from Nanking University in 1984 and 1987 and Ph.D. degree from Department of Computer Science and Engineering, Harbin Institute of Technology, in 1990. He is currently an associate professor of Center for Information Science, Peking University. From 1990 to 1992, he was a postdoctoral fellow in Department of Automation at Tsinghua University. During 1992-1993 he was awarded a postdoctoral fellowship by Japan Society for Promotion of Science to work in Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Japan. His current research interest lies in neural computing and its applications to machine perception including computer vision and the biologically plausible auditory model.

Masumi Ishikawa received the B.E.E., M.E.E. and D. Eng. degree in Electrical Engineering from the University of Tokyo, Japan, in 1969, 1971, and 1974, respectively. He is currently a professor of Department of Control Engineering and Science, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Japan. During 1974-1979 he was a research associate at Electrotechnical Laboratory, MITI, Japan. During 1979-1990 he was a senior scientist at ETL, MITI. He received Prize Paper Award from the institute of Electrical Engineers of Japan in 1981. During 1986-1987 he was awarded a Fulbright Research Grant to work at Univ. of California, San Diego. Since 1990 he has been at the present position. His current research interests on neural networks include a learning algorithm for discovery of regularities and extraction of rules, and its applications to human interfaces, cognitive models, and time series prediction. Dr. Ishikawa is a member of IEEE, Japanese Neural Network Society, Japanese Cognitive Science Society, and the Institute of Electronics, Information and Communication Engineers, Japan.