Correspondence

Isolated 3-D Object Recognition through Next View Planning

Sumantra Dutta Roy, Santanu Chaudhury, and Subhashis Banerjee

***Abstract*—In many cases, a single view of an object may not contain sufficient features to recognize it unambiguously. This paper presents a new on-line recognition scheme based on next view planning for the identification of an isolated three-dimensional (3-D) object using simple features. The scheme uses a probabilistic reasoning framework for recognition and planning. Our knowledge representation scheme encodes feature based information about objects as well as the uncertainty in the recognition process. This is used both in the probability calculations as well as in planning the next view. Results clearly demonstrate the effectiveness of our strategy for a reasonably complex experimental set.**

***Index Terms*—Active vision, reactive planning, 3-D object recognition.**

I. Introduction

In this paper, we present a new on-line scheme for the recognition of an isolated three-dimensional (3-D) object using reactive next view planning. A hierarchical knowledge representation scheme facilitates recognition and the planning process. The planning process utilizes the current observation and past history for identifying a sequence of moves to disambiguate between similar objects.

Most model-based object recognition systems consider the problem of recognizing objects from the image of a single view [1]–[4]. However, a single view may not contain sufficient features to recognize the object unambiguously. In fact, two objects may have all views in common with respect to a given feature set, and may be distinguished only through a sequence of views. Further, in recognizing 3-D objects from a single view, recognition systems often use complex feature sets [2]. In many cases, it may be possible to achieve the same, incurring less error and smaller processing cost using a simpler feature set and suitably planning multiple observations. A simple feature set is applicable for a larger class of objects than a model base specific complex feature set. Model base-specific complex features such as 3-D invariants have been proposed only for special cases so far (e.g., [3]). The purpose of this paper is to investigate the use of suitably planned multiple views and two-dimensional (2-D) invariants for 3-D object recognition.

A. Relation with Other Work

With an active sensor, object recognition involves identification of a view of an object and if necessary, planning further views. Tarabanis *et al.* [5] survey the field of sensor planning for vision tasks. We can compare various active 3-D object recognition systems on the basis of the following four issues.

1) *Nature of the Next View Planning Strategy:* The system should plan moves with maximum ability to discriminate between views common to more than one object in the model base. The cost incurred in this process should also be minimal. The system should, preferably be on-line and reactive—the past and present inputs should guide the planning mechanism at each stage.

   While the scheme of Maver and Bajcsy [6] is on-line, that of Gremban and Ikeuchi [7] is not. Due to the combinatorial nature of the problem, an off-line approach may not always be feasible.
2) *Uncertainty Handling Capability of the Hypothesis Generation Mechanism:* The occlusion-based next view planning approach of Maver and Bajcsy [6], as well as that of Gremban and Ikeuchi [7] are essentially deterministic. A probabilistic strategy can make the system more robust and resistant to errors compared to a deterministic one. Dickinson *et al.* [8] use Bayesian methods to handle uncertainty, while Hutchinson and Kak [9] use the Dempster–Shafer theory.
3) *Efficient Representation of Domain Knowledge:* The knowledge representation scheme should support an efficient mechanism to generate hypotheses on the basis of the evidence received. It should also play a role in optimally planning the next view.

   Dickinson *et al.* [8] use a hierarchical representation scheme based on volumetric primitives, which are associated with a high feature extraction cost. Due to the non-hierarchical nature of Hutchinson and Kak's system [9], many redundant hypotheses are proposed, which have to be later removed through consistency checks.
4) *Speed and Efficiency of Algorithms for Both Hypothesis Generation and Next View Planning:* It is desirable to have algorithms with low order polynomial-time complexity to generate hypotheses accurately and fast. The next view planning strategy acts on the basis of these hypotheses.

   In Hutchinson and Kak's system [9], although the polynomial-time formulation overcomes the exponential time complexity associated with assigning beliefs to all possible hypotheses, their system still has the overhead of intersection computation in creating common frames of discernment. Consistency checks have to be used to remove the many redundant hypotheses produced earlier. Though Dickinson *et al.* [8] use Bayes nets for hypothesis generation, their system incurs the overhead of tracking the region of interest through successive frames.

The next view planning strategy that this paper presents is reactive and on-line—the evidence obtained from each view is used in the hypothesis generation and the planning process. Our probabilistic hypothesis generation mechanism can handle cases of feature detection errors. We use a hierarchical knowledge representation scheme which not only ensures a low-order polynomial-time complexity of the hypothesis generation process, but also plays an important role in planning the next view. The hierarchy itself enforces different constraints to prune the set of possible hypotheses. The scheme is independent of the type of features used, unlike that of [8]. We present results of over 100 experiments with our recognition scheme on two sets of models. Extensive experimentation shows the effectiveness of our proposed strategy of using simple features and multiple views for recognizing complex 3-D shapes.

The organization of the rest of the paper is as follows: Section II presents our knowledge representation scheme. We discuss hypothesis generation for class and object recognition in Section III. Section IV

Manuscript received October 23, 1997; revised May 5, 1998.

S. Dutta Roy and S. Banerjee are with the Department of Computer Science and Engineering, Indian Institute of Technology, New Delhi-110 016, India (e-mail: sumantra@ee.iitd.ernet.in; suban@cse.iitd.ernet.in).

S. Chaudhury is with the Department of Electrical Engineering, Indian Institute of Technology, New Delhi 110 016, India.

Publisher Item Identifier S 1083-4427(00)01177-2.

describes our algorithm for planning the next view. In Section V we demonstrate the working of our system on two sets of objects. We summarize the salient features of our scheme and discuss areas for further work in Section VI.

II. The Knowledge Representation Scheme

A view of a 3-D object is characterized by a set of features. With respect to a particular feature set and over a particular range of viewing angles, a view of a 3-D object is independent of the viewpoint. Koenderink and van Doorn [10] define aspects as topologically equivalent classes of object appearances. Ikeuchi *et al.* generalize this definition: object appearances may be grouped into equivalence classes with respect to a feature set. These equivalence classes are aspects [11]. In this context, we define the following terms:

Class A: Class (or, aspect-class) is a set of aspects, equivalent with respect to a feature set.

Feature-Class: A feature-class is a set of equivalent aspects defined for *one* particular feature.

Fig. 1 shows a simple example of an object with its associated aspects and classes. The locus of view-directions is one-dimensional (1-D) and we assume orthographic projection. The basis of the different classes is the number of horizontal lines (h) and vertical lines (v) in a particular view of the object. Thus, a class may be represented as $\langle hv \rangle$. There are six aspects of the object shown, belonging to three classes. In this example, for simplicity we assume only one feature detector so that each feature-class is also a class.

We propose a new knowledge representation scheme encoding domain knowledge about the object, relations between different aspects, and the correspondence of these aspects with feature detectors. Fig. 2 illustrates an example of this scheme. We use this knowledge representation scheme both in belief updating as well as in next view planning. Sections III and IV discuss these topics, respectively. The representation scheme consists of two parts.

1) *The Feature-Dependence Subnet:* In the feature-dependence subnet
   - **F** represents the complete set of features $\{F_j\}$ used for characterizing views.
   - A feature node F_j is associated with feature-classes f_{jk}.

   Factors such as noise and nonadaptive thresholds can introduce errors in the feature detection process. Let p_{jlk} represent the probability that the feature-class present is f_{jl}, given that the detector for feature F_j detects it to be f_{jk}. We define p_{jlk} as the ratio of the number of times the detector for feature F_j interprets feature-class f_{jl} as f_{jk}, and the number of times the feature detector reports the feature-class as f_{jk}. The F_j node stores a table of these values for its corresponding feature detector.
   - A class node C_i stores its *a priori* probability, $P(C_i)$. A link between class C_i and feature-class f_{jk} indicates that f_{jk} forms a subset of features observed in C_i. This accounts for a *PART-OF* relation between the two. Thus, a class represents an n-vector $[f_{1j_1} \;\; f_{2j_2} \; \cdots \; f_{nj_n}]$. Since a class cannot be independent of any feature, each class has n input edges corresponding to the n features.

2 *The Class-Aspect Subnet:* The class-aspect subnet encodes the relationships between classes, aspects, and objects.
   - **O** represents the set of all objects $\{O_i\}$
   - An object node O_i stores its probability, $P(O_i)$.
   - An aspect node a_{ij} stores its angular extent θ_{ij} (in degrees), its probability $P(a_{ij})$, its parent class C_j, and its neighboring aspects.
   - Aspect a_{ij} has a *PART-OF* relationship with its parent object O_i. Thus, 3-tuple $\langle O_i, C_j, \theta_{ik} \rangle$ represents an aspect. Aspect node a_{ij} has exactly one link to any object (O_i) and exactly one link to its parent class C_j.

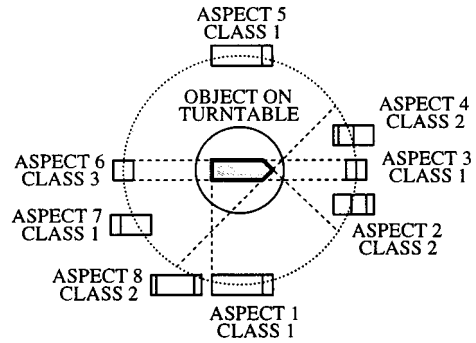

Fig. 1. Aspects and classes of an object.

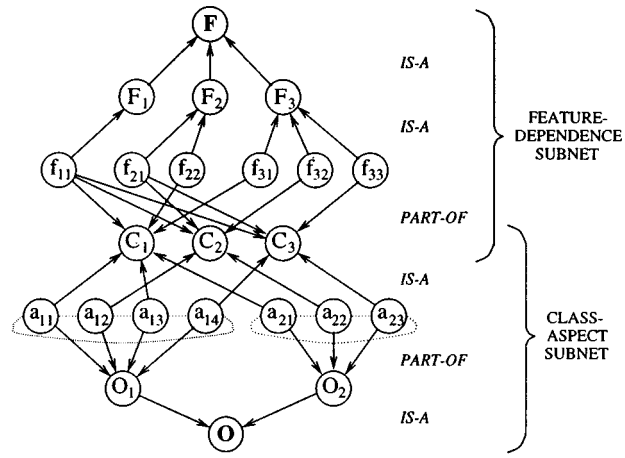

Fig. 2. Example of the knowledge representation scheme.

III. Hypothesis Generation

The recognition system takes any arbitrary view of an object as input. Using a set of features (the feature-classes), it generates hypotheses about the likely identity of the class. This is, in turn used for generating hypotheses about the object's identity. The interaction of the hypothesis generation part with the rest of the system is shown in Fig. 3. Hypothesis generation consists of two steps namely, class identification, and object identification.

A. Class Identification, Accounting for Uncertainty

Our algorithm suitably schedules feature detectors to perform probabilistic class identification. In what follows, we discuss its various aspects. Fig. 4 presents the overall algorithm.

1) Ordering of Feature Detectors: A proper ordering of feature detectors speeds up the class recognition process. At any stage, we choose the hitherto unused feature detector for which the feature-class corresponding to the most probable class has the least number of outgoing arcs, i.e., the least out-degree. This is done in order to obtain that feature-class which has the largest discriminatory power in terms of the number of classes it could correspond to. For example, in Fig. 2 if all feature detectors are unused and C_2 has the highest *a priori* probability, F_3 will be tried first, followed by F_2 and F_1, if required.

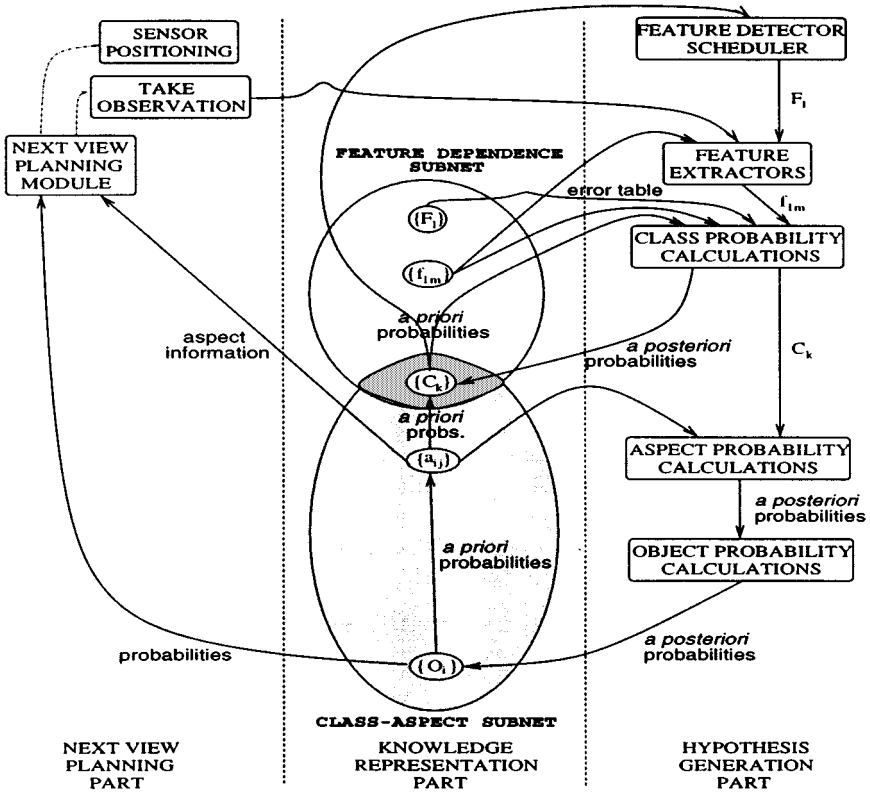

Fig. 3. Flow diagram depicting the flow of information and control in our system.

ALGORITHM identify_class

```
1. compute_a_priori_class_probabilities();
        (* Eq. 1; Section III-A.2 Part 1 *)
2. fd := identify_feature_detector_to_use();
        (* Section III-A.1 *)
3. fcl := get_feature_class(image,fd);
        (* Use fd on the image, identify feature class *)
4. compute_a_posteriori_class_probabilities(fcl);
        (* Eqs. 2,3; Section III-A.2 Part 2 *)
5. IF the probability of some class is above a
   predetermined threshold THEN
        pass this class as evidence to the
        object recognition phase, EXIT
6. IF all feature detectors have been used
   AND the probability of no class is above
   the threshold THEN EXIT
7. GO TO Step 2
```

Fig. 4. Class recognition algorithm.

2) Class Probability Calculations Using the Knowledge Representation Scheme: We obtain the *a priori* probability of class C_i as

$$P(C_i) = \sum_p \left[P(O_p) \cdot \sum_q P(a_{pq}|O_p) \right]. \tag{1}$$

Here, aspects a_{pq} belong to class C_i. Let N_{F_j}, N_C, and N_a denote the number of feature-classes associated with feature detector F_j, the number of classes, and the number of aspects, respectively. $P(a_{pq}|O_p)$ is $\theta_{pq}/360$. We can compute $P(C_i)$ from our knowledge representation scheme by considering each aspect node belonging to an object and testing if it has a link to node C_i; this takes $O(N_C + N_a)$ time. (The N_C term is for the initialization of class probabilities to 0.)

Let the detector for feature F_j report the feature-class obtained to be f_{jk}. Given this evidence, we obtain the probability of class C_i from the Bayes rule

$$P(C_i|f_{jk}) = \frac{P(C_i) \cdot P(f_{jk}|C_i)}{\sum_m [P(C_m) \cdot P(f_{jk}|C_m)]} \tag{2}$$

$P(f_{jk}|C_i)$ is 1 for those classes which have a link from feature-class f_{jk}. It is 0 for the rest. The computation of (2) takes $O(N_C)$ time—this is done for each feature-class. Hence, the computation of $P(f_{jk}|C_i)$ for all feature-classes f_{jk} for feature detector F_j takes time $O(N_{F_j} \cdot N_C)$.

For an error-free situation, $P(C_i|f_{jk})$ is $P'(C_i)$, the *a posteriori* probability of class C_i. However, due to errors possible in the feature detection process, a degree of uncertainty is associated with the evidence. The value of $P'(C_i)$ is, then

$$P'(C_i) = \sum_l P(C_i|f_{jl}) \cdot p_{jlk} \tag{3}$$

where f_{jl}'s are feature-classes associated with feature F_j. According to our knowledge representation scheme, only one feature-class under feature F_j, say f_{jr} has a link to class C_i. The summation reduces to one term, $P(C_i|f_{jr}) \cdot p_{jrk}$. Thus, our knowledge representation scheme also enable recovery from feature detection errors.

B. Object Identification

Based on the outcome of the class recognition scheme, we estimate the object probabilities as follows. Initially, we calculate the *a priori* probability of each aspect as

$$P(a_{j_p k_p}) = P(O_{j_p}) \cdot P(a_{j_p k_p}|O_{j_p}). \tag{4}$$

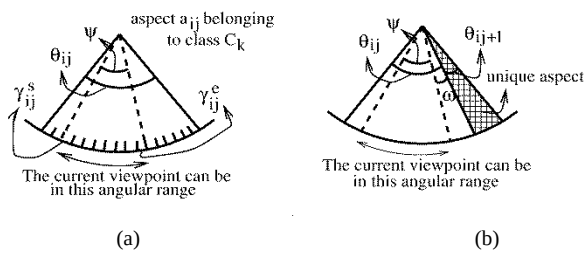

Fig. 5. (a) The notation used (Section IV) and (b) a case when our algorithm is not guaranteed to succeed (Section IV-A).

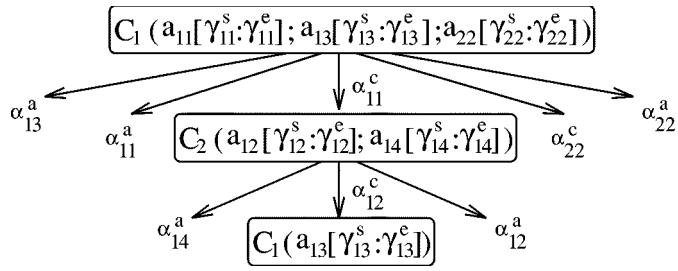

Fig. 6. Partially constructed search tree.

If there are N objects in the model base, we initialize $P(O_{j_p})$ to $1/N$ before the first observation. For the first observation, $P(a_{j_pk_p}|Oj_p)$ is $\theta_{j_pk_p}/360$. *A priori* aspect probability calculations take $O(N_a)$ time.

For any subsequent observation, we have *to account for the movement in the probability calculations.* For example, a particular movement may preclude the occurrence of some aspects for a given class observed. The value of $P(a_{j_pk_p}|O_{j_p})$ is given by

$$P(a_{j_pk_p}|O_{j_p}) = \phi_{j_pk_p}/360 \tag{5}$$

where $\phi_{j_pk_p}$ ($\phi_{j_pk_p} \in [0, \theta_{j_pk_p}]$) represents the angular range possible within aspect $a_{j_pk_p}$ for the move(s) taken to reach this position. Due to the movement made, we could have observed only m ($0 \leq m \leq r$) aspects out of a total of r aspects belonging to class C_i.

Experiments with Model Base I

Let the class recognition phase report the observed class to be C_i. Let us assume that C_i could have come from aspects $a_{j_1k_1}$, $a_{j_2k_2}$, $\cdots$, $a_{j_mk_m}$, where all $j_1, j_2, \cdots, j_m$ are not necessarily different. We obtain the *a posteriori* probability of aspect $a_{j_lk_l}$ given this evidence using the Bayes rule

$$P(a_{j_lk_l}|C_i) = \frac{P(a_{j_lk_l}) \cdot P(C_i|a_{j_lk_l})}{\sum_{p=1}^{m}[P(a_{j_pk_p}) \cdot P(C_i|a_{j_pk_p})]} \tag{6}$$

$P(C_i|a_{j_lk_l})$ is 1 for aspects with a link to class C_i, 0 otherwise. Finally, we obtain the *a posteriori* probability

$$P(O_{j_p}) = \sum_l P(a_{j_pk_l}|C_i) \tag{7}$$

where aspects $a_{j_pk_l}$ belong to class C_i.

If the probability of some object is above a predetermined threshold (experimentally determined, e.g., 0.87 for Model Base I), the algorithm reports a success, and stops. If not, it means that the view of the object is not sufficient to identify the object unambiguously. We have to take the next view.

In our hierarchical scheme, the link conditional probabilities (representing relations between nodes) themselves enforce consistency checks at each level of evidence. The feature evidence is progressively refined as it passes through different levels in the hierarchy, leading to simpler evidence propagation and less computational cost. This is an advantage of our scheme over that proposed in [9].

IV. Next View Planning

The class observed in the class recognition phase could have come from many aspects in the model base, each with its own range of positions within the aspect. Due to this ambiguity, one has to search for the best move to discern between these competing aspects subject to memory and processing limitations, if any. The parameters described above characterize the state of the system. The planning process aims to determine a move from the current step, which would uniquely identify the given object. We pose the planning problem as that of a forward search in the state space which takes us to a state in which the aspect list corresponding to the class observed has exactly one node. We use a search tree for this purpose. A search tree node represents the following information: [Fig. 5(a)] the unique class observed for the angular movement made so far, the aspects possible for this angle-class pair, and for each aspect, the range of positions possible within it ($\gamma^s_{ij} - \gamma^e_{ij}$). γ^s_{ij} and γ^e_{ij} denote the two positions within aspect a_{ij} where the current viewpoint can be, as a result of the movement made thus far. Here, $\gamma^s_{ij} \leq \gamma^e_{ij}$; and $\gamma^s_{ij}, \gamma^e_{ij} \in [0, \theta_{ij}]$, where θ_{ij} is the angular extent of aspect a_{ij}. A leaf node is one which has either one aspect associated with it or corresponds to a total angular movement of 360° or more from the root node.

Fig. 6 shows an example of a partially constructed search tree. From a view point, we categorize possible moves as follows.

ALGORITHM identify_object

```
(* ------ FIRST PHASE ------ *)
1.  initialize_object_probabilities();
            (* Initialize to 1/N *)
2.  image:=get_image_of_object();
3.  class:=identify_class(image);
            (* Section III-A *)
    IF class=UNKNOWN THEN exit;
4.  search_tree_root:=construct_search_tree_node(class,0);
5.  compute_object_probabilities(search_tree_root);
            (* Eqs. 6,7 *)
6.  IF the probability of some object is above a
    predetermined thresh.  THEN exit & declare success;
7.  expand_search_tree_node(search_tree_root,0,class);
            (* Section IV *)
    best_leaf:=get_best_leaf_node(search_tree_root);
(* ------ SECOND PHASE ------ *)
        previous:=search_tree_root;
        expected:=best_leaf;
8.  angle:=compute_angle_to_move_by(expected,previous);
    make_movement(angle);
    image:=get_image_of_object();
9.  class:=identify_class(image);
    IF class=UNKNOWN THEN exit;
10. new_node:=construct_search_tree_node(class,angle);
11. compute_object_probabilities(new_root);
12. IF the probability of some object is above a
    predetermined thresh.  THEN exit & declare success;
13. expand_search_tree_node(new_node);
    best_leaf:=get_best_leaf_node(new_node);
        previous:=new_node;
        expected:=best_leaf;
14. GO TO step 8
```

Fig. 7. Object recognition algorithm.

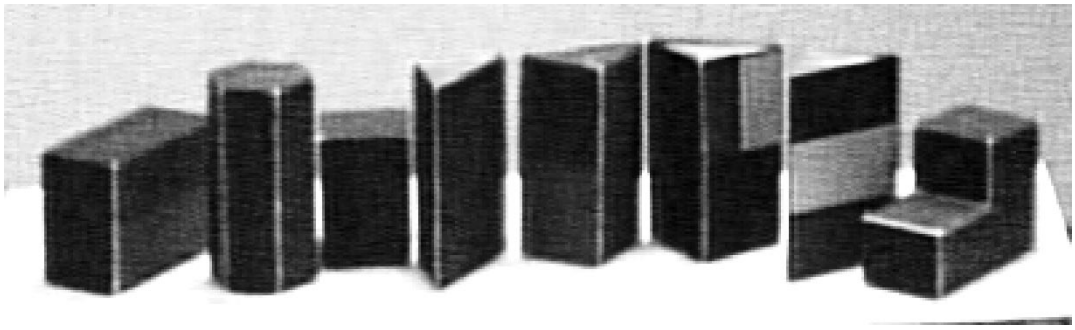

Fig. 8. Model Base I: The objects (from left) are O_1, O_2, O_3, O_4, O_5, O_6, O_7, and O_8, respectively.

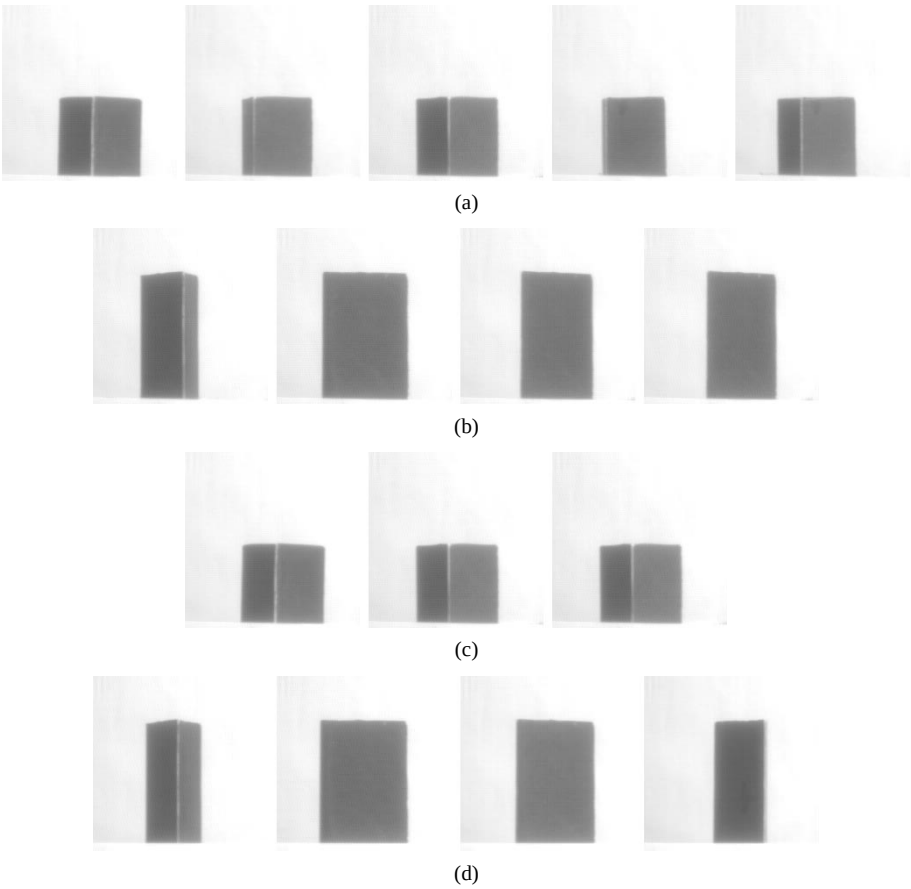

Fig. 9. Some experiments with Model Base I: initial class $\langle 232 \rangle$. The objects are O_3[(a) and (c)], and O_4[(b) and (d)], respectively. (a) $\langle 232 \rangle \overset{-40}{\rightarrow} \langle 231(221) \rangle \overset{-12}{\rightarrow} \langle 232 \rangle \overset{-34}{\rightarrow} \langle 221 \rangle \overset{-11}{\rightarrow} \langle 232 \rangle$. (b) $\langle 232 \rangle \overset{-40}{\rightarrow} \langle 221 \rangle \overset{-12}{\rightarrow} \langle 221 \rangle \overset{-2}{\rightarrow} \langle 221 \rangle$. (c) $\langle 232 \rangle \overset{-40}{\rightarrow} \langle 232 \rangle \overset{-77}{\rightarrow} \langle 221 \rangle$. (d) $\langle 232 \rangle \overset{-40}{\rightarrow} \langle 221 \rangle \overset{-12}{\rightarrow} \langle 221 \rangle \overset{-33}{\rightarrow} \langle 221 \rangle$. The numbers above the arrows denote the number of turntable steps. A negative sign indicates a clockwise movement. (The figure in parentheses shows an example of recovery from feature detection errors.)

Primary Move: A primary move represents a move from an aspect by α, the minimum angle needed to move out of it.

Auxiliary Move: An auxiliary move represents a move from an aspect by an angle corresponding to the primary move of another competing aspect.

Let α_{ij}^c and α_{ij}^a represent the minimum angles necessary to move out of the current assumed aspect in the clockwise and counterclockwise directions, respectively. Three cases are possible.

1) *Type I Move*: α_{ij}^c and α_{ij}^a both take us out of the current aspect to a single aspect in each of the two directions—a_{ip} and a_{iq}, respectively. We construct search tree nodes corresponding to both moves.
2) *Type II Move*: Exactly one out of α_{ij}^c and α_{ij}^a takes us to a single aspect a_{ip}. For the other direction, the aspect we would reach depends upon the initial position $(\in \gamma_{ij}^s,\ \gamma_{ij}^e])$ in the current aspect. We construct a search tree node corresponding to the former move.
3) *Type III Move:* Whether we move in the clockwise or the counterclockwise direction, the aspect reached depends on the initial position in the current aspect. We choose the move which leads

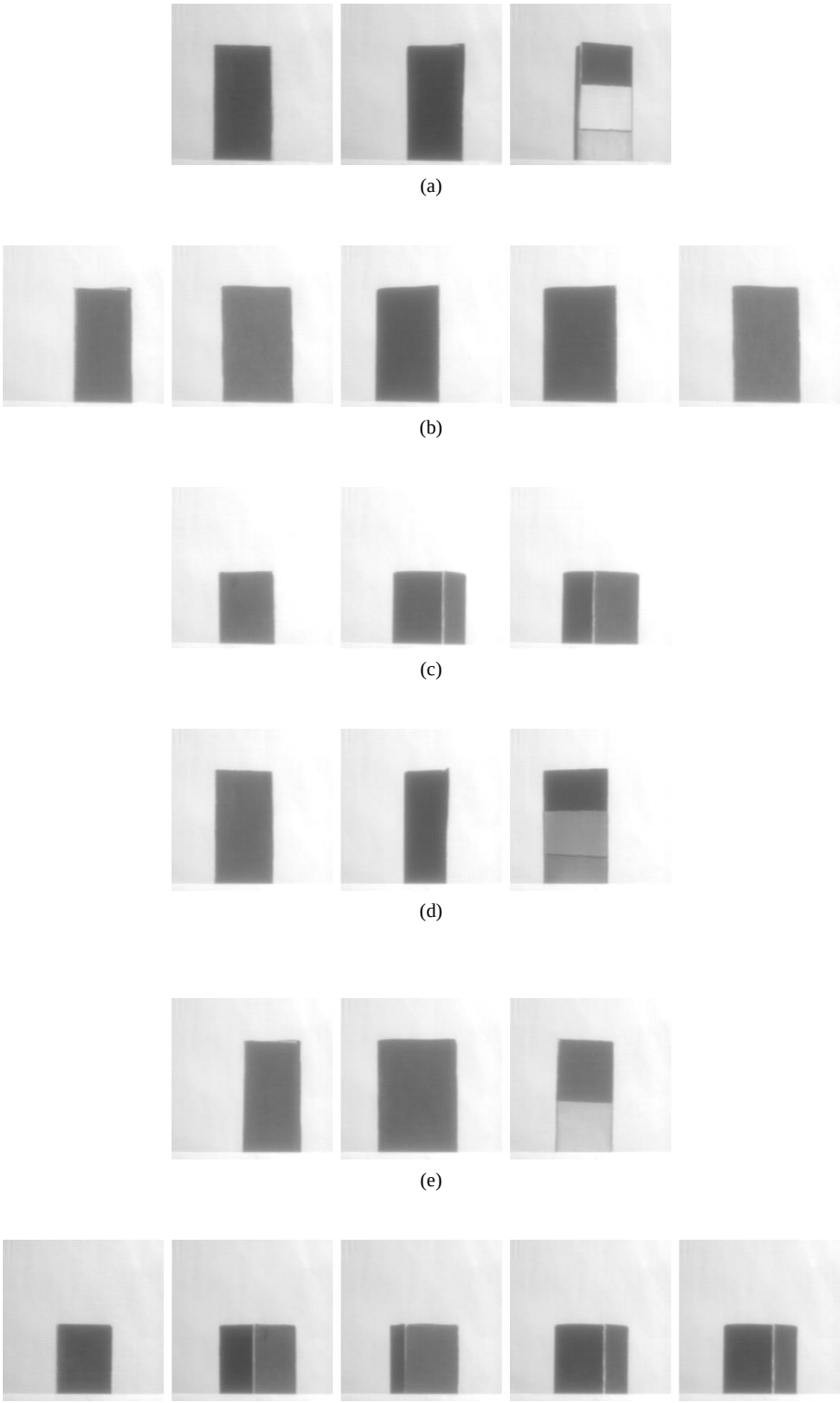

Fig. 10. Some experiments with Model Base I: initial class $\langle 221 \rangle$. Primary moves alone (a) $O_7 : \langle 221 \rangle \overset{-61}{\rightarrow} \langle 221 \rangle \overset{-35}{\rightarrow} \langle 423 \rangle$. (b) $O_5 : \langle 221 \rangle \overset{-61}{\rightarrow} \langle 221 \rangle \overset{-35}{\rightarrow} \langle 221 \rangle \overset{-8}{\rightarrow} \langle 221 \rangle \overset{-30}{\rightarrow} \langle 221 \rangle$. (c) $O_3 : \langle 221 \rangle \overset{-61}{\rightarrow} \langle 232 \rangle \overset{-16}{\rightarrow} \langle 232 \rangle$. Primary and auxiliary moves (d) $O_7 : \langle 221 \rangle \overset{-77}{\rightarrow} \langle 221 \rangle \overset{-72}{\rightarrow} \langle 423 \rangle$. (e) $O_5 : \langle 221 \rangle \overset{-77}{\rightarrow} \langle 221 \rangle \overset{-72}{\rightarrow} \langle 322 \rangle$. (f) $O_3 : \langle 221 \rangle \overset{-77}{\rightarrow} \langle 232 \rangle \overset{-138}{\rightarrow} \langle 232 \rangle \overset{-11}{\rightarrow} \langle 221 \rangle \overset{-13}{\rightarrow} \langle 232 \rangle$. The numbers above the arrows denote the number of turntable steps. A negative sign indicates a clockwise movement.

us to the side with the largest angular range possible in any reachable aspect.

We expand a nonleaf node by generating child nodes corresponding to primary moves for all competing aspects in its aspect list. We can also generate additional child nodes by considering auxiliary moves. We assign a code to each move, a higher code to a less preferred move. We assign a code 0 to Types I and II primary moves and 1 to Type II auxiliary moves. Type III primary moves get a code of 2, and Type III

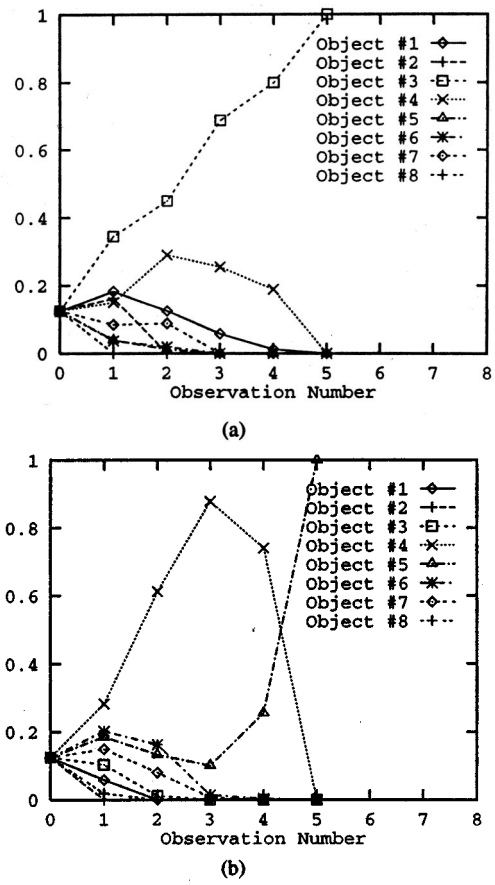

Fig. 11. Variation of object probabilities: two examples (see text).

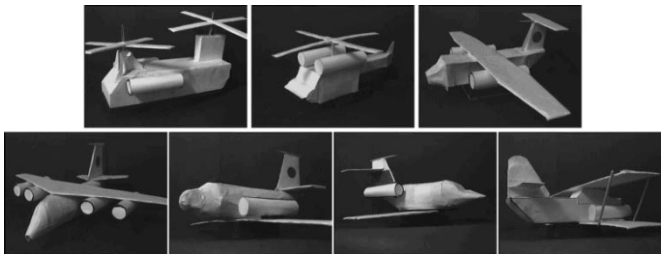

Fig. 12. Model Base II: The objects (in row major order) are heli_1, heli_2, plane_1, plane_2, plane_3, plane_4, and biplane.

TABLE I
THE AVERAGE NUMBER OF MOVES FOR A GIVEN NUMBER OF COMPETING ASPECTS

Model Base I: Polyhedral Objects		
Number of Competing Aspects	*Average number of observations*	
	Primary Moves	*Pri. & Aux. Moves*
5	2.00	2.50
17	3.09	3.07
18	4.00	3.38
Model Base II: Aircraft Models		
Number of Competing Aspects	*Average number of observations*	
	Primary Moves	*Pri. & Aux. Moves*
4	2.00	2.00
5	2.00	2.09
7	2.00	2.00
9	2.00	2.00
10	2.67	2.67

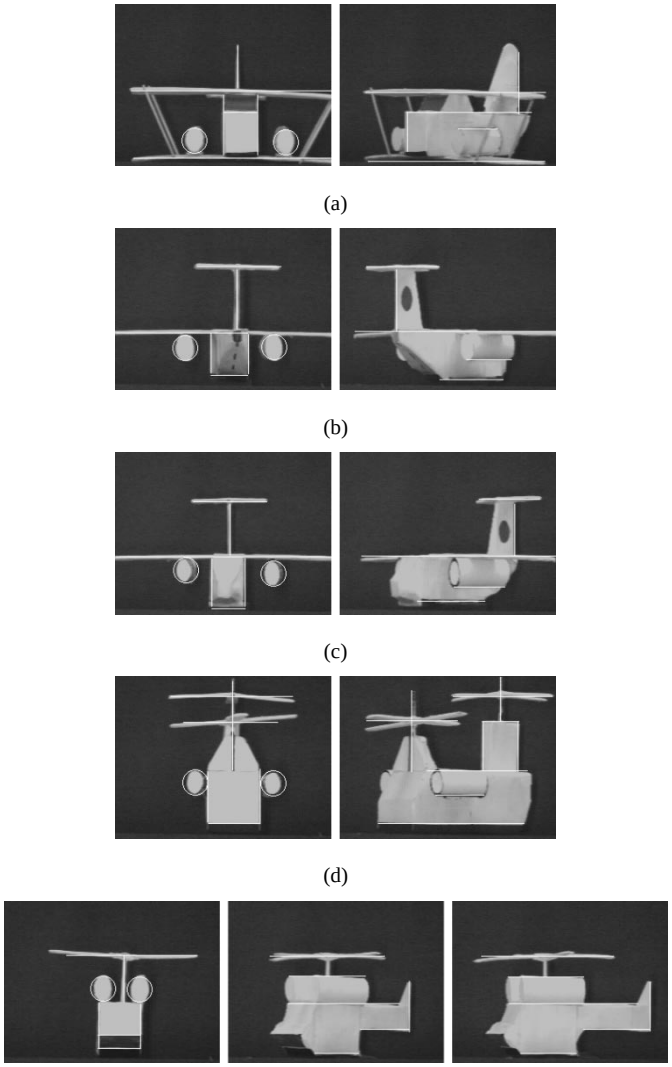

Fig. 13. Experiments with the initial class as $\langle 332 \rangle$. (a) biplane: $\langle 332 \rangle \xrightarrow{-26} \langle 420 \rangle$. (b) plane_1: $\langle 342(332) \rangle \xrightarrow{-26} \langle 410 \rangle$. (c) plane_1: $\langle 332 \rangle \xrightarrow{-26} \langle 410 \rangle$. (d) heli_1: $\langle 332 \rangle \xrightarrow{-26} \langle 540 \rangle$. (e) heli_2: $\langle 332 \rangle \xrightarrow{-26} \langle 510 \rangle \xrightarrow{-12} \langle 510 \rangle$. (The figure in parentheses shows an example of recovery from feature detection errors.) In each of these cases, the results for planning with primary moves alone, and those for both primary and auxiliary moves are identical.

auxiliary moves, 3. The weight associated with a node is $4^i \cdot Code$, where i is the depth of the node in the search tree. We use three levels of filtering to determine the best leaf node. First, we consider those on a path from the most probable aspect(s) corresponding to the previously observed node. Among these, we consider those having paths of least weight. From these, we finally select one with the minimum total movement.

A. The Planning Process and Object Recognition

In our object identification algorithm, aspect and object probabilities are initialized to their *a priori* values. We use our class identification algorithm (Section III-A) to identify the class corresponding to this view of the object. It then calculates the *a posteriori* object probabilities. If the probability of some object is above a predetermined threshold, then the algorithm declares that object as being present and exits. Else, the algorithm initiates the search process to get the best distinguishing move to resolve the ambiguity associated with this view. It then decides on the best move and takes the next view. All the above steps starting at

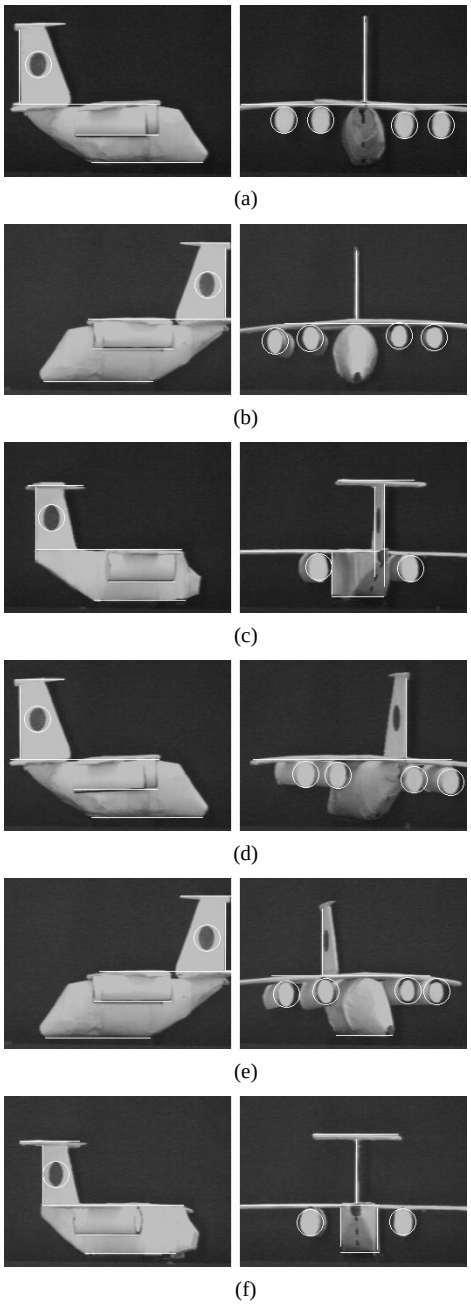

Fig. 14. Experiments with the initial class as $\langle 411 \rangle$. Primary moves alone (a) plane_2: $\langle 411 \rangle \xrightarrow{-48} \langle 114 \rangle$. (b) plane_2: $\langle 411 \rangle \xrightarrow{-48} \langle 114 \rangle$. (c) plane_1: $\langle 411 \rangle \xrightarrow{-48} \langle 332 \rangle$. Primary and auxiliary moves (d) plane_2: $\langle 411 \rangle \xrightarrow{-61} \langle 114 \rangle$. (e) plane_2: $\langle 411 \rangle \xrightarrow{-61} \langle 215(214) \rangle$. (f) plane_1: $\langle 411 \rangle \xrightarrow{-61} \langle 332 \rangle$. (The figure in parentheses shows an example of recovery from feature detection errors.)

the class identification phase are repeated. Fig. 7 presents our overall object identification algorithm in detail. Fig. 3 shows the interaction of the next view planning part with the rest of the system.

Search tree node expansion is always finite due to the following reasons: the number of aspects is finite, and no aspect is repeated along a search tree path. Further, even if competing objects have the same aspects, search tree expansion stops when the total movement along a path is 360°. Primary moves eliminate redundant image processing operations, while auxiliary moves enable better aspect resolution. Our planning scheme is global—its reactive nature incorporates all previous movements and observations both in the probability calculations (Section III-B) as well as in the planning process. Our robust class recognition algorithm can recover from many feature detection errors at the class recognition phase itself (Section III-A-2). If the view indeed corresponds to the most probable aspect at a particular stage, then our search process using primary and auxiliary moves is guaranteed to perform aspect resolution and uniquely identify the object in the following step, assuming no feature detection errors. Even if the view does not correspond to the most probable aspect, the list of possible aspects a view could correspond to is refined at each observation stage. The planning process is initiated with the new aspect list. This illustrates the reactive nature of our planning strategy.

Assuming no feature detection errors, our algorithm is guaranteed to succeed except in three cases. The first is for objects with the same aspect structure (i.e., the layout of classes in the aspect graph) but different aspect angles. Further, our strategy does not handle the case when the aspect angles are greater than or equal to 180°. Fig. 5(b) shows an example of the third case. Let us suppose that we have to move counterclockwise. Let ψ denote the angular extent of the smallest aspect observed so far. The current viewpoint lies in this angular range. Let a_{ij+1} be a unique aspect for the assumed object. The counterclockwise movement will be by an angle $\psi + \omega$. If $\psi + \omega > \theta_{ij+1}$, we may miss this unique aspect altogether.

B. Bounds on the Number of Observations

It is instructive to consider bounds on $T_{avg}(n)$, the number of observations required to disambiguate between a set of n aspects (corresponding to the initially observed class). For a simple case to serve as a benchmark, let us assume the number of aspects reachable from any aspect as 1, and no movement or image processing errors. We also assume no errors in either movement or image processing. We choose a move that partitions the initial aspect set into more than one equivalence class. If the size of the aspect list in one such equivalence class is j, the expected additional number of observations is $T_{avg}(j)$, where $j \in [1, n)$. We have $T_{avg}(n) = 1 + (\sum_{j=1}^{n-1} T_{avg}(j))/(n-1)$, and $T_{avg}(1) = 1$. By induction, we can show that $T_{avg}(n) = O(\log_e n)$.

V. Results and Discussion

Our experimental setup has a camera connected to a MATROX image processing card and a stepper motor-controlled turntable. The turntable moves by 200 steps to complete a 360° movement. We use simple and robust features with low feature extraction cost, compared to systems using complex features (e.g., [8] uses volumetric primitives).

We have experimented extensively with two object sets as model bases. We have chosen such objects in our model base that most of them have more than one view in common. The list of possible aspects associated with one initial view is quite large. Our experiments have been with both strategies—to have primary moves alone, and both primary and auxiliary moves for expanding the search tree node corresponding to an observation.

1) Polyhedral Objects: We use as features, the number of horizontal and vertical lines ($\langle hv \rangle$), and the number of nonbackground segmented regions in an image ($\langle r \rangle$). We represent a class as $\langle hvr \rangle$. We use a Hough transform-based line detector [12]. For getting the number of regions in the image, we perform sequential labeling (connected components: pixel labeling) [12] on a thresholded gradient image. We have chosen this model base so that most objects have more than one view in common—the degree of ambiguity associated with a view is very large. Fig. 8 shows the objects in this model base. Figs. 9 and 10 show some experiments with the objects in the first model base. For

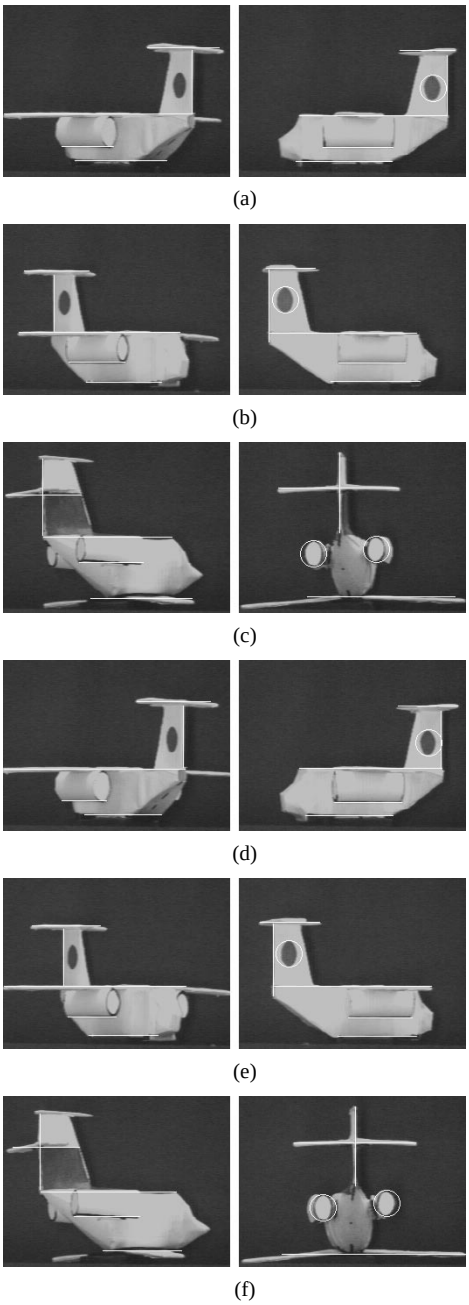

Fig. 15. Experiments with the initial class as $\langle 410 \rangle$. Primary moves alone (a) plane_1: $\langle 410 \rangle \xrightarrow{-28} \langle 411 \rangle$. (b) plane_1: $\langle 410 \rangle \xrightarrow{-28} \langle 411 \rangle$. (c) plane_4: $\langle 410 \rangle \xrightarrow{-28} \langle 212 \rangle$. Primary and auxiliary moves (d) plane_1: $\langle 410 \rangle \xrightarrow{-38} \langle 411 \rangle$. (e) plane_1: $\langle 410 \rangle \xrightarrow{-38} \langle 411 \rangle$. (f) plane_4: $\langle 410 \rangle \xrightarrow{-38} \langle 212 \rangle$.

Fig. 9, the initial class observed in each case is $\langle 232 \rangle$, while it is $\langle 221 \rangle$ in Fig. 10. We make the following observations.

2) Primary and Auxiliary Moves: In most cases, the number of image processing steps required is less in the latter case compared to the former. When memory and search times are limited, the planning process may use primary moves alone. An interesting case is observed in Fig. 10(c) and (f)—an opportunistic case when the number of steps with primary moves is less than the one with both primary and auxiliary moves. At step 2, the move planned was not for the aspect eventually observed in step 3. Due to the move, however, the sequence of moves turns out to be unique for object O_3.

3) Ordering of Feature Detectors: The third image in Fig. 9(a) shows advantage of our scheduling of feature detectors. The line detector reports the feature-class present to be $\langle 23 \rangle$. For the objects in our model base, this could correspond to classes $\langle 232 \rangle$ and $\langle 233 \rangle$. Our probability calculations account for the movement taken around the object. The probability of class $\langle 232 \rangle$ for the movement made so far exceeds the class probability threshold (0.87). Hence, the system does not need to use the other feature detector.

4) Recovering from Feature Detection Errors: The second image in Fig. 9(a) shows a situation where the system recovers from an error in the feature detection process. Due to the thresholds we use, the correct class is $\langle 221 \rangle$. The line detector, however, reports the probabilities of classes $\langle 221 \rangle$ and $\langle 231 \rangle$ as 0.004 and 0.856, respectively. The probability of no class is above the threshold. The other feature detector is now scheduled, which reports the number of regions to be 1. The probability calculations of (3) result in the probabilities of the two as 0.997 and 0.002, respectively.

5) Variation of Object Probabilities: Fig. 11 shows the variation in object probabilities with each observation. The two cases shown here are for the moves in Fig. 9(a) and Fig. 10(b). The latter shows an interesting case. Aspects belonging to class $\langle 221 \rangle$ occupy a large extent for object O_4. The sequence of moves until observation 3 could correspond to O_4, O_5, O_6, and O_7 with probabilities 0.877, 0.102, 0.014, and 0.007, respectively. The reactive nature of our strategy ensures a correct and progressively refined aspect list corresponding to each observation (sizes: 17, 8, 6, 4, and 1, respectively). The move leading to observation 4 reduces the number of competing aspects from 6 to 4. The aspects, the angular extents possible within the aspects and hence, their probabilities depend upon the sequence of moves from the initial viewpoint. The probabilities of O_4 and O_5 are 0.740 and 0.225, respectively. The sequence of moves leading to observation 5 is unique only for O_5, identifying it uniquely.

6) Some Sample Search Tree Details: We now consider some cases in detail. For each row in Fig. 9, the initial view could have come from 18 aspects belonging to objects in our model base and for Fig. 10, the corresponding number is 17. For the strategy involving primary moves alone, the total number of search tree nodes generated for Figs. 9(a) and (b), 10(a) and (b) are 53, 48, 34, and 48, respectively. For the strategy involving both primary and auxiliary moves [Fig. 9(c) and (d), Fig. 10(d) and (e)], the corresponding numbers are 324, 279, 127, and 127, respectively. Let us consider Fig. 10(e). The algorithm plans a move of 77 steps. The second observation reports the number of aspects possible as 6. The next move by 72 steps corresponds to a unique aspect.

7) Average Number of Observations for a Given Number of Competing Aspects: The upper part of Table I gives an idea of the average number of observations for a given number of competing aspects for the experiments with the first model base. The average is computed over 46 experiments.

A. Experiments with Model Base II

Aircraft Models: We use the number of horizontal and vertical lines ($\langle hv \rangle$), and the number of circles ($\langle c \rangle$) as features. We represent a class as $\langle hvc \rangle$. We use Hough transform-based line and circle detectors [12]. We have chosen this relatively feature-rich model base to demonstrate the effectiveness of our system using simple features and multiple views. Fig. 12 shows the objects in this model base.

For most of the 58 experiments (Figs. 13–15), the number of observations required with primary moves alone, is the same as that considering auxiliary moves also.

This can be attributed to the lower degree of uncertainty associated with a view for an object in this model base (a maximum of ten), compared to that for the first (18). The second images in Fig. 14(a), (b), and (d) show cases where the system does not need to use the second feature detector. In the first image in Fig. 13(b), due to the shadow of the wing on the fuselage of the aircraft, the feature detector detects

four vertical lines instead of three, the correct number. Our recovery mechanism (Section III-A-2) corrects this error. For the experiments shown in Fig. 14, the number of search tree nodes constructed for primary moves alone is 14, whereas the corresponding number for both primary and auxiliary moves is 125. The corresponding numbers for the experiments in Fig. 15 are 14 and 41, respectively.

VI. Conclusions

This paper presents an integrated approach for the recognition of an isolated 3-D object through on-line next view planning using probabilistic reasoning. Our knowledge representation scheme facilitates planning by exploiting the relationships between features, aspects, and object models. The recognition scheme has the ability to correctly identify objects even when they have a large number of similar views. If a feature set is not rich enough to identify an object from a single view, this strategy may be used to identify it from multiple views. We demonstrate that the proposed recognition strategy works correctly even under processing and memory constraints due to the incremental reactive planning strategy. No related work has addressed this problem.

While we use simple features for the purpose of illustration, one may use other features such as texture, color, specularities, and reflectance ratios. Over 100 experiments demonstrate the effectiveness of using simple features and multiple views even on a relatively complex class of objects with a high degree of ambiguity associated with a view of the object. Our experiments show that one may use simple features to recognize objects with complex 3-D shapes (as in Fig. 12).

Major areas for further work include multiple object recognition and searching for an object in a cluttered environment. This would require suitably incorporating occlusion handling techniques (e.g., those in [13]). An extension of this work would take movement errors into account.

References

[1] P. J. Besl and R. C. Jain, "Three-dimensional object recognition," *ACM Comput. Surv.*, vol. 17, pp. 76–145, Mar. 1985.

[2] R. T. Chin and C. R. Dyer, "Model based recognition in robot vision," *ACM Comput. Surv.*, vol. 18, pp. 67–108, Mar. 1986.

[3] A. Zisserman, D. Forsyth, J. Mundy, C. Rothwell, J. Liu, and N. Pillow, "3-D object recognition using invariance," *Artif. Intell.*, vol. 78, pp. 239–288, 1995.

[4] D. P. Mukherjee and D. Dutta Majumder, "On shape from symmetry," *Proc. Ind. Nat. Sci. Acad.*, vol. A-62, no. 5, pp. 415–428, 1996.

[5] K. A. Tarabanis, P. K. Allen, and R. Y. Tsai, "A survey of sensor planning in computer vision," *IEEE Trans. Robot. Automat.*, vol. 11, pp. 86–104, Feb. 1995.

[6] J. Maver and R. Bajcsy, "Occlusions as a guide for planning the next view," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 76–145, May 1993.

[7] K. D. Gremban and K. Ikeuchi, "Planning multiple observations for object recognition," *Int. J. Comput. Vis.*, vol. 12, pp. 137–172, Apr. 1994.

[8] S. J. Dickinson, H. I. Christensen, J. Tsotsos, and G. Olofsson, "Active object recognition integrating attention and view point control," *Comput. Vis. Image Understand.*, vol. 67, pp. 239–260, Sept. 1997.

[9] S. A. Hutchinson and A. C. Kak, "Planning sensing strategies in a robot work cell with multi-sensor capabilities," *IEEE Trans. Robot. Automat.*, vol. 5, pp. 765–783, Dec. 1989.

[10] J. J. Koenderink and A. J. van Doorn, "The internal representation of solid shape with respect to vision," *Biol. Cybern.*, vol. 32, pp. 211–216, 1979.

[11] K. D. Gremban and K. Ikeuchi, "Appearance-based vision and the automatic generation of object recognition programs," in *Three-Dimensional Object Recognition Systems*, A. K. Jain and P. J. Flynn, Eds. Amsterdam, The Netherlands: Elsevier, 1993, pp. 229–258.

[12] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*. Reading, MA: Addison-Wesley, 1992.

[13] D. Dutta Majumder and K. S. Ray, "Recognition and position determination of partially occluded object for a computer vision system," *J. IETE*, vol. 37, no. 5/6, pp. 419–442, 1991.

Fuzzy Critical Path Method Based on Signed Distance Ranking of Fuzzy Numbers

Jin-Shing Yao and Feng-Tse Lin

***Abstract*—In this paper, we apply a signed distance ranking method for fuzzy numbers to a critical path method for activity-on-edge (AOE) networks. We use signed distance ranking to define ordering simply, which means we can use both positive and negative values to define ordering. The primary result obtained in this paper is the use of signed distance ranking of fuzzy numbers obtaining Properties 3 and 4. We conclude that the fuzzy AOE network is an extension of the crisp AOE network, and thus the fuzzy critical path in a fuzzy AOE network, under some conditions, is the same as the crisp critical path in a crisp AOE network.**

***Index Terms*—Activity-on-edge (AOE) network, critical path method, fuzzy number, signed distance ranking.**

I. Introduction

Activity-on-edge (AOE) networks have proved very useful for performance evaluation of some types of projects. This evaluation includes determining certain aspects about the project, e.g., what is the least amount of time in which the project may be completed, and which individual activities should be speeded to reduce overall project length, etc. [2]. Since the activities in an AOE network can be carried out in parallel, the minimum time to complete the project is the length of the longest path from the start of project to its finish. The longest path is the critical path. To identify the critical path, three parameters for each of its activities are determined:

1) earliest event time;
2) latest event time;
3) slack time.

The critical path is the one from the start of project to the finish of project where the slack times are all zeros. The purpose of the critical path method (CPM) is to identify critical activities on the critical path so that resources may be concentrated on these activities in order to reduce project length time. Besides, CPM has proved very valuable in evaluating project performance and identifying bottlenecks. Thus, CPM is a vital tool for the planning and control of complex projects.

The successful implementation of CPM requires the availability of a clear determined time duration for each activity. However, in practical situations this requirement is usually hard to fulfill since many of activities will be executed for the first time. Hence, there is always uncertainty about the time duration of activities in the network planning, leading to the development of fuzzy critical path methods. In developing the fuzzy critical path approach, several approaches have been

Manuscript received February 6, 1999; revised July 24, 1999.

J.-S. Yao is with the Department of Mathematics, National Taiwan University, Taipei, Taiwan, R.O.C.

F.-T. Lin is with the Department of Applied Mathematics, Chinese Culture University, Taipei, Taiwan, R.O.C.

Publisher Item Identifier S 1083-4427(00)01180-2.

proposed over the past years. Gazdik [1] assumes that in a fuzzy network the duration of activities and some other input variables are imprecise and biased, and the imprecision is summarized to four classes. This work proposed a technique called FNET based on a combination of fuzzy sets and the theory of graphs. The membership function expressing the activity duration time for FNET depends on such diverse factors as expert opinions; the availability of means of production, materials, or staff; and personal experience. The task is then to simulate quasideterministic outcomes of a process based on those imprecise or subjectively biased input data. An extension of FNET was proposed by Nasution [4]. In this paper, it is shown that fuzzy numbers can be exploited further in the network. This is to be done by first introducing an interactive fuzzy subtraction in the backward calculations; then observing that if time were represented by fuzzy numbers only the nonnegative times should be taken into account, since the negative times have no physical meaning. Based on these two assumptions, it is possible to obtain the latest allowable event time and the slack of each event in the network. Therefore, with this approach CPM can be generalized by accepting imprecise, fuzzy data for the duration of the activities. Since FNET never uses any method for ranking fuzzy numbers, some assumptions are required for the fuzzy numbers to facilitate this fuzzy CPM.

In this paper, we propose a method for ranking fuzzy numbers without the need for any assumptions. We use signed distance ranking to define ordering simply, which means we can use both positive and negative values to define ordering. The signed distance we use here has some properties very similar to those signed distances introduced in real numbers. We use signed distance for ranking fuzzy numbers, and then applying it to CPM. Therefore, our work is quite different from the previous methods proposed in the literature [1], [4].

The paper is organized as follows. Section II outlines the preliminaries, in which we consider the definition of a signed distance ranking system for fuzzy numbers. In Section III, we list some definitions of the AOE network and give an example to explain crisp CPM. Then, we make the AOE fuzzy network by using fuzzy numbers and then eliminate its fuzziness by using signed distance ranking for fuzzy numbers to construct the AOE network in the fuzzy sense. The results are listed in Properties 3 and 4. Finally, the concluding remarks of the paper are stated in Section IV.

II. Preliminaries

For a fuzzy critical path method, all pertinent definitions of fuzzy sets are given below.

Definition 1—Fuzzy Point: Let $\tilde{b}_1$ be a fuzzy set on $R = (-\infty, \infty)$. It is called a fuzzy point if its membership function is

$$\mu_{\tilde{b}_1}(x) = \begin{cases} 1, & x = b \\ 0, & x \neq b. \end{cases} \tag{1}$$

Definition 2—Level λ Fuzzy Interval: Let $[a_\lambda, b_\lambda]$ be a fuzzy set on $R = (-\infty, \infty)$. It is called a level λ fuzzy interval, $0 \leq \lambda \leq 1$, if its membership function is

$$\mu_{[a_\lambda, b_\lambda]}(x) = \begin{cases} \lambda, & a \leq x \leq b \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Definition 3—Fuzzy Numbers: Let $\tilde{A}$ be a fuzzy set on $R = (-\infty, \infty)$. It is called a fuzzy number, if its membership function is

$$\mu_{\tilde{A}}(x) = \begin{cases} \dfrac{x-a}{b-a}, & a \leq x \leq b \\ \dfrac{c-x}{c-b}, & b \leq x \leq c \\ 0, & \text{otherwise,} \end{cases} \tag{3}$$

where $a < b < c$. In addition, we let the family of all fuzzy numbers be denoted by $F_N = \{(a,b,c) | \forall a < b < c, a, b, c \in R\}$. In particular, the fuzzy number (a,b,c) will be regarded as the degenerated case of the fuzzy point $(b,b,b) = \tilde{b}_1$, if $a = c = b$.

Before defining the ranking of fuzzy numbers on F_N as in Yao and Wu [5], we should first consider the definition of the signed distance on R.

Definition 4—The Signed Distance: We define $d^*(b, 0) = b$ when $b, 0 \in R$.

Remark 1: Geometrically, $0 < b$ means that b lies to the right of the origin 0 and the distance between b and 0 is denoted by $b = d^*(b, 0)$. Similarly, $b < 0$ means that b lies to the left of 0 and the distance between b and 0 is denoted by $-b = -d^*(b, 0)$. Therefore, $d^*(b, 0)$ denotes the signed distance of b, which is measured from 0.

Let $\tilde{A} = (a,b,c) \in F_N$. From (3) we know the α-cut of $\tilde{A}$ is $A(\alpha) = [A_L(\alpha), A_R(\alpha)]$, $0 \leq \alpha \leq 1$, where $A_L(\alpha) = a + (b-a)\alpha$ is the left endpoint of the α-cut and $A_R(\alpha) = c - (c-b)\alpha$ is the right endpoint of the α-cut.

Fig. 1 shows that P' and Q' are the signed distances measured from the origin 0, $0 \leq \alpha \leq 1$. From Definition 4, we find the signed distance of P' is $d^*(A_L(\alpha), 0) = A_L(\alpha)$ and also that of Q' is $d^*(A_R(\alpha), 0) = A_R(\alpha)$. Hence, the signed distance of interval $[A_L(\alpha), A_R(\alpha)]$, which is measured from the origin 0, is defined as

$$\begin{aligned} & d^*([A_L(\alpha), A_R(\alpha)], 0) \\ &= \tfrac{1}{2}[d^*(A_L(\alpha), 0) + d^*(A_R(\alpha), 0)] \\ &= \tfrac{1}{2}[a + c + (2b - a - c)\alpha] \end{aligned}$$

where $0 \leq \alpha \leq 1$. In addition, for each $\alpha \in [0, 1]$, because the intervals $[A_L(\alpha), A_R(\alpha)]$ and $[A_L(\alpha)_\alpha, A_R(\alpha)_\alpha]$ have a one-to-one mapping (Fig. 1), therefore, we can define the signed distance of $[A_L(\alpha)_\alpha, A_R(\alpha)_\alpha]$, which is measured from $\tilde{0}_1$ (y-axis), as

$$d([A_L(\alpha)_\alpha, A_R(\alpha)_\alpha], \tilde{0}_1) = \tfrac{1}{2}[a + c + (2b - a - c)\alpha].$$

Notice that the function α is continuous over the interval where $0 \leq \alpha \leq 1$. Consequently, we can use the method of integration as an explanatory tool for obtaining the mean value of the signed distance.

Definition 5—Signed Distance of $\tilde{A}$: Let $\tilde{A} = (a,b,c) \in F_N$. Then, $d(\tilde{A}, \tilde{0}_1)$ is the signed distance of $\tilde{A}$, which is measured from $\tilde{0}_1$ (y-axis) as defined by

$$d(\tilde{A}, \tilde{0}_1) = \int_0^1 d([A_L(\alpha)_\alpha, A_R(\alpha)_\alpha], \tilde{0}_1)\, d\alpha = \tfrac{1}{4}(2b + a + c).$$

Note that if $\tilde{A} = (a,a,a) = \tilde{a}_1$, then $d(\tilde{a}_1, \tilde{0}_1) = a$.

Definition 6—The Ranking: For $\tilde{A}$, $\tilde{B} \in F_N$, we define the following rankings on F_N

$$\begin{aligned} \tilde{A} &\prec \tilde{B} \text{ iff } d(\tilde{A}, \tilde{0}_1) < d(\tilde{B}, \tilde{0}_1) \\ \tilde{A} &\approx \tilde{B} \text{ iff } d(\tilde{A}, \tilde{0}_1) = d(\tilde{B}, \tilde{0}_1). \end{aligned}$$

From [3] and [6] we have the following properties of binary operations.

Property 1: For $\tilde{A} = (a,b,c)$ and $\tilde{B} = (p,q,r) \in F_N$, we have

1) $\tilde{A} \oplus \tilde{B} = (a+p, b+q, c+r)$
2) $\tilde{A} \ominus \tilde{B} = (a-r, b-q, c-p)$.

Property 2: For $\tilde{A} = (a,b,c)$ and $\tilde{B} = (p,q,r) \in F_N$, we have

1) $d(\tilde{A} \oplus \tilde{B}, \tilde{0}_1) = d(\tilde{A}, \tilde{0}_1) + d(\tilde{B}, \tilde{0}_1)$
2) $d(\tilde{A} \ominus \tilde{B}, \tilde{0}_1) = d(\tilde{A}, \tilde{0}_1) - d(\tilde{B}, \tilde{0}_1)$.

Proof: From Property 1 and Definition 5, we obtain Property 2.

III. Fuzzy Critical Path Method

In this section, we begin with some definitions of AOE networks and also with an example to explain the crisp CPM. Next, we present a procedure based on signed distance ranking of fuzzy numbers, to obtain the fuzzy CPM.

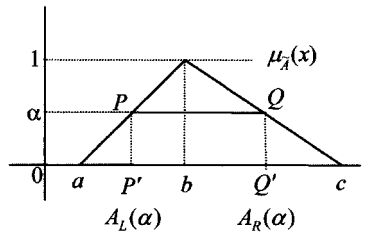

Fig. 1. The α-cut of fuzzy number $\bar{A}$.

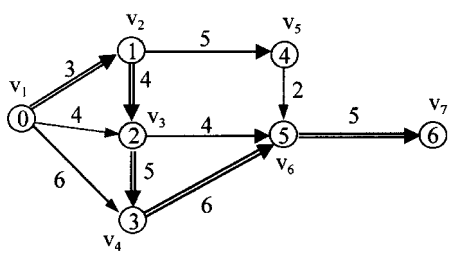

Fig. 2. Crisp AOE network $N = (V, A, T)$.

A. CPM in Crisp Case

An AOE network is a directed acyclic graph in which the vertices represent events and the edges represent project activities or tasks to be performed on a project [2]. Formally, an AOE network is represented by $N = (V, A, T)$. Let $V = \{v_1, v_2, \cdots, v_n\}$ be a set of vertices representing a set of events, where v_1 is the start of the project, v_n is its completion, and $A \subset V \times V$ is the set of directed edges connecting the vertices. The tasks to be performed on the project are represented by directed edges. For each activity $b \in A$, a magnitude $t_b \in T$ is defined, where t_b is the time required for the completion of activity b [1], [4].

Example 1: Finding a crisp critical path in an AOE network, $N = (V, A, T)$.

Fig. 2 is an example of an AOE network for a hypothetical project with seven events and ten tasks or activities. Let $V = \{v_j = j - 1 | j = 1, 2, \cdots, 7\}$ be the set of seven events; $A = \{(v_1, v_2), (v_1, v_3), (v_1, v_4), (v_2, v_3), (v_3, v_4), (v_2, v_5), (v_3, v_6), (v_4, v_6), (v_5, v_6), (v_6, v_7)\}$ be the set of ten activities; and $T = \{t_{v_1v_2}, t_{v_1v_3}, t_{v_1v_4}, t_{v_2v_3}, t_{v_3v_4}, t_{v_2v_5}, t_{v_3v_6}, t_{v_4v_6}, t_{v_5v_6}, t_{v_6v_7}\}$ be the number associated with each activity representing the time needed to perform that activity, where $t_{v_1v_2} = t_{01} = 3$, $t_{v_1v_3} = t_{02} = 4$, $t_{v_1v_4} = t_{03} = 6$, etc. Thus, the activity (v_1, v_2) requires three days, whereas (v_1, v_3) requires four days. Usually, these times are only estimates. The critical path of this network is (v_1, v_2), (v_2, v_3), (v_3, v_4), (v_4, v_6), (v_6, v_7).

In an AOE network, $N = (V, A, T)$, where $V = \{v_1, v_2, \cdots, v_n\}$. Let $t_{v_iv_j}$ be the processing time for each activity (v_i, v_j). We define the earliest event time for event v_i and the latest event time for event v_j as t_{Ev_i} and t_{Lv_j}, respectively. Assume that the values of $t_{v_iv_j}$, t_{Ev_i}, and t_{Lv_j} are already known. From Fig. 3 we see that t_{Ev_j} and t_{Lv_i}, representing the earliest event time for event v_j and the latest event time for event v_i, satisfy the following equations:

$$t_{Ev_j} = t_{Ev_i} + t_{v_iv_j} \tag{4}$$

$$t_{Lv_i} = t_{Lv_j} - t_{v_iv_j}. \tag{5}$$

Also, let $T_{v_iv_j}$ be the total available time for activity (v_i, v_j). We obtain

$$T_{v_iv_j} = t_{Lv_j} - t_{Ev_i}. \tag{6}$$

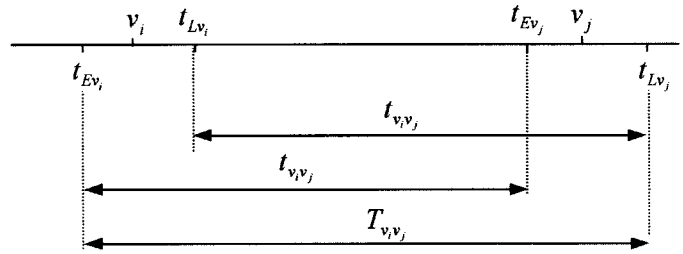

Fig. 3. Diagram of $t_{v_iv_j}, t_{Ev_i}, t_{Lv_i}, t_{Ev_j}, t_{Lv_j}$, and $T_{v_iv_j}$ in $N = (V, A, T)$.

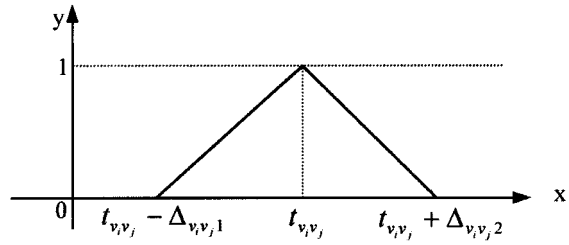

Fig. 4. A triangular fuzzy number $\bar{t}_{v_iv_j}$.

Let $D_j = \{v_i | v_i \in V$ and $(v_i, v_j) \in A\}$ be a set of events obtained from event $v_j \in V$ such that $(v_i, v_j) \in A$ and $v_i < v_j$. In Example 1, for instance, if $v_j = v_4 (= 3)$ then $D_4 = \{v_i | v_i \in V$ and $(v_i, v_4) \in A\} = \{v_1, v_3\} = \{0, 2\}$. Clearly, from Fig. 3, we can obtain t_{Ev_j} for event v_j by using the following equations:

$$t_{Ev_j} = \max_{v_i \in D_j} [t_{Ev_i} + t_{v_iv_j}] \quad \text{and} \quad t_{Ev_1} = t_{Lv_1} = 0. \tag{7}$$

Similarly, let $E_i = \{v_j | v_j \in V$ and $(v_i, v_j) \in A\}$ be a set of events obtained from event $v_i \in V$, such that $(v_i, v_j) \in A$ and $v_i < v_j$. For instance in Example 1, if $v_i = v_3 (= 2)$ then $E_3 = \{v_j | v_j \in V$ and $(v_3, v_j) \in A\} = \{v_4, v_6\} = \{3, 5\}$. Then, we obtain t_{Lv_i} for event v_i by using the following equations:

$$t_{Lv_i} = \min_{v_j \in E_i} [t_{Lv_j} - t_{v_iv_j}] \quad \text{and} \quad t_{Lv_n} = t_{Ev_n}. \tag{8}$$

Finally, when

$$T_{v_iv_j} = t_{v_iv_j}, \text{ i.e., } t_{Lv_j} = t_{Ev_i} + t_{v_iv_j} \tag{9}$$

we conclude that activity (v_i, v_j) is definitely on the critical path of the crisp network.

B. Fuzzy CPM Based on Signed Distance Ranking of Fuzzy Numbers

As noted earlier in this section, for each activity (v_i, v_j), we assume that the values of $t_{v_iv_j}$, t_{Ev_i}, and t_{Lv_j} are already known and t_{Ev_j}, t_{Lv_i}, and $T_{v_iv_j}$ can be obtained from (7), (8), and (6), respectively. However, this assumption may cause severe difficulties in practice. Therefore, here we consider $t_{v_iv_j}$ is only an estimate and is imprecise. Thus, we make $t_{v_iv_j}$ fuzzy by using the following triangular fuzzy number (Fig. 4)

$$\tilde{t}_{v_iv_j} = (t_{v_iv_j} - \Delta_{v_iv_j1}, t_{v_iv_j}, t_{v_iv_j} + \Delta_{v_iv_j2}),$$
$$0 < \Delta_{v_iv_j1} < t_{v_iv_j}, \quad 0 < \Delta_{v_iv_j2}. \tag{10}$$

By Definition 5, we have $d(\tilde{t}_{v_iv_j}, \tilde{0}_1) = t_{v_iv_j} + \frac{1}{4}(\Delta_{v_iv_j2} - \Delta_{v_iv_j1})$. It is the signed distance of $\tilde{t}_{v_iv_j}$ measured from $\tilde{0}_1$. Since $d(\tilde{t}_{v_iv_j}, \tilde{0}_1) = \frac{1}{4}(3t_{v_iv_j} + \Delta_{v_iv_j2}) + \frac{1}{4}(t_{v_iv_j} - \Delta_{v_iv_j}) > 0$, we conclude that $d(\tilde{t}_{v_iv_j}, \tilde{0}_1)$ is a positive distance between $\tilde{t}_{v_iv_j}$ and $\tilde{0}_1$. In other words, the processing time is measured from the origin 0. Thus, we define

$t^*_{v_iv_j}$ to be an estimate of the processing time for activity (v_i, v_j) in the fuzzy sense, i.e.

$$t^*_{v_iv_j} = d(\tilde{t}_{v_iv_j}, \tilde{0}_1) = t_{v_iv_j} + \tfrac{1}{4}(\Delta_{v_iv_j2} - \Delta_{v_iv_j1}). \tag{11}$$

Remark 2: If $\Delta_{v_iv_j1} = \Delta_{v_iv_j2}$, i.e., the triangle in Fig. 4 is an isosceles triangle, then we have $t^*_{v_iv_j} = t_{v_iv_j}$. In particular, if $\Delta_{v_iv_j1} = \Delta_{v_iv_j2} = 0$, then the fuzzy case will become crisp.

As we mentioned above, in the crisp AOE network, $N = (V, A, T)$, the earliest event time t_{Ev_i} for event v_i, the latest event time t_{Lv_j} for event v_j, and the total available time $T_{v_iv_j}$ for activity (v_i, v_j) are directly derived from (7), (8), and (6), respectively. However, in practice, the decision-maker should realize that these time variables are imprecise and consider them in a fuzzy sense. Thus, an estimate of the earliest event time for event v_i is in the interval $[t_{Ev_i} - \Delta E_{Ev_i1}, t_{Ev_i} + \Delta_{Ev_i2}]$, $0 < \Delta_{Ev_i1} < t_{Ev_i}$, $0 < \Delta_{Ev_i2}$, where t_{Ev_i} is a known number. In considering the accuracy, we find that when the earliest event time is exactly t_{Ev_i}, the error rate is definitely 0. Clearly, greater imprecision of time will produce larger error rates. When an estimate eventually approaches one of the two ends of the interval, i.e., $t_{Ev_i} - \Delta_{Ev_i1}$ or $t_{Ev_i} + \Delta_{Ev_i2}$, the error rate becomes largest. However, it is preferable to use the term "*confidence level*" rather than "*error rate.*" Thus, when the earliest event time is exactly t_{Ev_i}, we obtain the confidence level is 1. On the other hand, when an estimate approaches one of the two ends of the interval, the confidence level becomes the smallest. In this section, we will use the fuzzy number from (12) below and use the membership grade to represent the confidence level.

Similar to the above $t_{v_iv_j}$ in (10), which corresponds to the interval $[t_{Ev_i} - \Delta_{Ev_i1}, t_{Ev_i} + \Delta_{Ev_i2}]$, we then define the fuzzy number of t_{Ev_i} as

$$\tilde{t}_{Ev_i} = (t_{Ev_i} - \Delta_{Ev_i1}, t_{Ev_i} + \Delta_{Ev_i2}) \tag{12}$$

where $0 < \Delta_{Ev_i1} < t_{Ev_i}$, $0 < \Delta_{Ev_i2}$, and the parameters satisfy (15) and (23) below. We then define $t^*_{Ev_i}$ to be an estimate of the earliest event time for event v_i in the fuzzy sense, i.e.,

$$t^*_{Ev_i} = d(\tilde{t}_{Ev_i}, \tilde{0}_1) = t_{Ev_i} + \tfrac{1}{4}(\Delta_{Ev_i2} - \Delta_{Ev_i1})\,(>0). \tag{13}$$

From (4), i.e., $t_{Ev_j} = t_{Ev_i} + t_{v_iv_j}$, we fuzzify both sides of the equation to obtain $\tilde{t}_{Ev_j} \approx \tilde{t}_{Ev_i} \oplus \tilde{t}_{v_iv_j}$. Note that $\approx$ is the ranking for F_N, as defined in Definition 6. From Definitions 5 and 6, and also from Property 2, we have $d(\tilde{t}_{Ev_j}, \tilde{0}_1) = d(\tilde{t}_{Ev_i} \oplus \tilde{t}_{v_iv_j}, \tilde{0}_1) = d(\tilde{t}_{Ev_i}, \tilde{0}_1) + d(\tilde{t}_{v_iv_j}, \tilde{0}_1)$. From (11) and (13), we obtain the following equation:

$$t^*_{Ev_j} = t^*_{Ev_i} + t^*_{v_iv_j}. \tag{14}$$

Thus, we have

$$\begin{aligned} &t_{Ev_j} + \tfrac{1}{4}(\Delta_{Ev_j2} - \Delta_{Ev_j1}) \\ &\quad = t_{Ev_i} + t_{v_iv_j} + \tfrac{1}{4}[\Delta_{Ev_i2} - \Delta_{Ev_i1} + (\Delta_{v_iv_j2} - \Delta_{v_iv_j1})]. \end{aligned}$$

According to (4), we obtain the following condition for parameters:

$$\Delta_{Ev_j2} - \Delta_{Ev_j1} = (\Delta_{Ev_i2} - \Delta_{Ev_i1}) + (\Delta_{v_iv_j2} - \Delta_{v_iv_j1}). \tag{15}$$

Similarly, the fuzzy number of t_{Lv_j} is defined as

$$\tilde{t}_{Lv_j} = (t_{Lv_j} - \Delta_{Lv_j1}, t_{Lv_j}, t_{Lv_j} + \Delta_{Lv_j2}) \tag{16}$$

where $0 < \Delta_{Lv_j1} < t_{Lv_j}$ and $0 < \Delta_{Lv_j2}$, and the parameters must satisfy (19) and (23) below. Then, we define $t^*_{Lv_j}$ to be an estimate of the latest event time for event v_j in the fuzzy sense, i.e.

$$t^*_{Lv_j} = d(\tilde{t}_{Lv_j}, \tilde{0}_1) = t_{Lv_j} + \tfrac{1}{4}(\Delta_{Lv_j2} - \Delta_{Lv_j1}). \tag{17}$$

Fuzzifying both sides of (5), $t_{Lv_i} = t_{Lv_j} - t_{v_iv_j}$, yields $\tilde{t}_{Lv_i} \approx \tilde{t}_{Lv_j} \ominus \tilde{t}_{v_iv_j}$. From Definitions 5 and 6, and from Property 2, we obtain $d(\tilde{t}_{Lv_i}, \tilde{0}_1) = d(\tilde{t}_{Lv_j}, \tilde{0}_1) - d(\tilde{t}_{v_iv_j}, \tilde{0}_1)$. From (11) and (17) we obtain the following equation:

$$t^*_{Lv_i} = t^*_{Lv_j} - t^*_{v_iv_j}. \tag{18}$$

Then, we have

$$\begin{aligned} &t_{Lv_i} + \tfrac{1}{4}(\Delta_{Lv_i2} - \Delta_{Lv_i1}) \\ &\quad = t_{Lv_j} - t_{v_iv_j} + \tfrac{1}{4}[(\Delta_{Lv_j2} - \Delta_{Lv_j1}) \\ &\qquad - (\Delta_{v_iv_j2} - \Delta_{v_iv_j1})]. \end{aligned}$$

According to (5), we obtain the following condition for parameters

$$\Delta_{Lv_i2} - \Delta_{Lv_i1} = (\Delta_{Lv_j2} - \Delta_{Lv_j1}) - (\Delta_{v_iv_j2} - \Delta_{v_iv_j1}). \tag{19}$$

The fuzzy number of the total available time $t_{v_iv_j}$ is defined as

$$\tilde{T}_{v_iv_j} = (T_{v_iv_j} - \varpi_{v_iv_j1}, T_{v_iv_j}, T_{v_iv_j} + \varpi_{v_iv_j2}) \tag{20}$$

where $0 < \varpi_{v_iv_j1} < T_{v_iv_j}$ and $0 < \varpi_{v_iv_j2}$, and the parameters must satisfy (23) and (24) below. Let $T^*_{v_iv_j}$ be an estimate of the total available time for activity (v_i, v_j) in the fuzzy sense, i.e.

$$T^*_{v_iv_j} = d(\tilde{T}_{v_iv_j}, \tilde{0}_1) = T_{v_iv_j} + \tfrac{1}{4}(\varpi_{v_iv_j2} - \varpi_{v_iv_j1}). \tag{21}$$

Fuzzifying both sides of (6), $T_{v_iv_j} = t_{Lv_j} - t_{Ev_i}$, yields $\tilde{T}_{v_iv_j} \approx \tilde{t}_{Lv_j} \ominus \tilde{t}_{Ev_i}$. From Definitions 5 and 6, and from Property 2, we have $d(\tilde{T}_{v_iv_j}, \tilde{0}_1) = d(\tilde{t}_{Lv_j}, \tilde{0}_1) - d(\tilde{t}_{Ev_i}, \tilde{0}_1)$. From (13), (17), and (21) we obtain the following equation

$$T^*_{v_iv_j} = t^*_{Lv_j} - t^*_{Ev_i}. \tag{22}$$

Therefore we have

$$\begin{aligned} &T_{v_iv_j} + \tfrac{1}{4}(\varpi_{v_iv_j2} - \varpi_{v_iv_j1}) \\ &\quad = t_{Lv_j} - t_{Ev_i} + \tfrac{1}{4}[\Delta_{Lv_j2} - \Delta_{Lv_j1} + (\Delta_{Ev_i2} - \Delta_{Ev_i1})]. \end{aligned}$$

From (6), we obtain the following condition for parameters:

$$\varpi_{v_iv_j2} - \varpi_{v_iv_j1} = (\Delta_{Lv_j2} - \Delta_{Lv_j1}) - (\Delta_{Ev_i2} - \Delta_{Ev_i1}). \tag{23}$$

Finally, fuzzifying both sides of (9), $T_{v_iv_j} = t_{v_iv_j}$, yields $\tilde{T}_{v_iv_j} \approx \tilde{t}_{v_iv_j}$. Similarly, we obtain $T^*_{v_iv_j} = t^*_{v_iv_j}$ and also have

$$T_{v_iv_j} + \tfrac{1}{4}(\varpi_{v_iv_j2} - \varpi_{v_iv_j1}) = t_{v_iv_j} + \tfrac{1}{4}(\Delta_{v_iv_j2} - \Delta_{v_iv_j1}).$$

Since $T_{v_iv_j} = t_{v_iv_j}$, we obtain the condition for parameters

$$\varpi_{v_iv_j2} - \varpi_{v_iv_j1} = \Delta_{v_iv_j2} - \Delta_{v_iv_j1}. \tag{24}$$

Furthermore, from (15), (19), (23), and (24) we obtain the following conditions for parameters,

$$\begin{aligned} \Delta_{Lv_j2} - \Delta_{Lv_j1} &= \Delta_{Ev_j2} - \Delta_{Ev_j1} \\ \Delta_{Lv_i2} - \Delta_{Lv_i1} &= \Delta_{Ev_i2} - \Delta_{Ev_i1} \end{aligned}$$

and

$$\begin{aligned} &(\Delta_{Lv_j2} - \Delta_{Lv_j1}) - (\Delta_{Ev_i2} - \Delta_{Ev_i1}) \\ &\quad = \Delta_{v_iv_j2} - \Delta_{v_iv_j1}. \end{aligned} \tag{25}$$

Remark 3: Here we compare the crisp AOE network for the efforts made by a decision-maker with the fuzzy AOE network. Consider each activity (v_i, v_j) in the crisp AOE network. The values of $t_{v_iv_j}$, t_{Ev_i}, and t_{Lv_j} are already known, however, the values of t_{Ev_j}, t_{Lv_i}, and $T_{v_iv_j}$ are determined according to (6)–(8). On the other hand, for each activity (v_i, v_j) in the fuzzy AOE network, the fuzzy number $\tilde{t}_{v_iv_j}$ in (10) must satisfy $0 < \Delta_{v_iv_j1} < t_{v_iv_j}$ and $0 < \Delta_{v_iv_j2}$. The other fuzzy numbers $\tilde{t}_{Ev_i}$, $\tilde{t}_{Lv_j}$, and $\tilde{T}_{v_iv_j}$, as defined in (12), (16), and (20), respectively, are determined as follows. Since the values of t_{Ev_i}, t_{Ev_j}, t_{Lv_i}, t_{Lv_j}, and $T_{v_iv_j}$, as well as the values of $t_{v_iv_j}$, $\Delta_{v_iv_j1}$, and $\Delta_{v_iv_j2}$ in (10) are already known; the decision-maker can choose appropriate values for Δ_{Ev_iq}, Δ_{Ev_jq}, Δ_{Lv_iq}, Δ_{Lv_jq}, and $\varpi_{v_iv_jq}$, where $q = 1, 2$ to satisfy (15), (19), (23), and (24), respectively. (See Example 2 and Remark 4.)

While $T^*_{v_iv_j} = t^*_{v_iv_j}$, i.e., $t^*_{Lv_j} = t^*_{Ev_i} + t^*_{v_iv_j}$, we conclude that the activity (v_i, v_j) is on the fuzzy critical path. Next, the processing

time $t_b(\in T)$ is considered for each activity $b(\in A)$ in the crisp AOE network, $N = (V, A, T)$. We fuzzify t_b as $\tilde{t}_b$, and then obtaining an estimate of the processing time for activity b in the fuzzy sense, $t_b^* = d(\tilde{t}_b, \tilde{0}_1)$. Let $T^* = \{t_b^* | \forall t_b \in T, b \in A, t_b^* = d(\tilde{t}_b, \tilde{0}_1\}$. Hence, we construct a fuzzy AOE network, $N^* = (V, A, T^*)$. From the previous discussions we summarize the following property.

Property 3: Consider the fuzzy AOE network, $N^* = (V, A, T^*)$. The fuzzy numbers of $t_{v_i v_j}$, t_{Ev_i}, t_{Lv_j}, and $T_{v_i v_j}$, are $\tilde{t}_{v_i v_j}$, $\tilde{t}_{Ev_i}$, $\tilde{t}_{Lv_j}$, and $\tilde{T}_{v_i v_j}$, respectively. When those fuzzy numbers satisfy (15), (19), (23), and (24), we obtain the following significant results.

1) An estimate of the processing time of activity (v_i, v_j) in the fuzzy sense is
$$t_{v_i v_j}^* = t_{v_i v_j} + \tfrac{1}{4}(\Delta_{v_i v_j 2} - \Delta_{v_i v_j 1}).$$
2) An estimate of the earliest event time of event v_i in the fuzzy sense is
$$t_{Ev_i}^* = t_{Ev_i} + \tfrac{1}{4}(\Delta_{Ev_i 2} - \Delta_{Ev_i 1}).$$
3) An estimate of the latest event time of event v_j in the fuzzy sense is
$$t_{Lv_j}^* = t_{Lv_j} + \tfrac{1}{4}(\Delta_{Lv_j 2} - \Delta_{Lv_j 1}).$$
4) An estimate of the total available time of activity (v_i, v_j) in the fuzzy sense is
$$T_{v_i v_j}^* = T_{v_i v_j} + \tfrac{1}{4}(\varpi_{v_i v_j 2} - \varpi_{v_i v_j 1}).$$
5) $t_{Ev_j}^* = t_{Ev_i}^* + t_{v_i v_j}^*, t_{Lv_i}^* = t_{Lv_j}^* - t_{v_i v_j}^*$, and $T_{v_i v_j}^* = t_{Lv_j}^* - t_{Ev_i}^*$.
6) When $T_{v_i v_j}^* = t_{v_i v_j}^*$, i.e., $t_{Lv_j}^* = t_{Ev_i}^* + t_{v_i v_j}^*$, the activity (v_i, v_j) is on the fuzzy critical path.

Now, from (7) we know $t_{Ev_i} + t_{v_i v_j} \le t_{Ev_j}, \forall v_i \in D_j$, and also know there is at least one equal sign which holds. When both sides of the above equation are fuzzified, we obtain $\tilde{t}_{Ev_i} \oplus \tilde{t}_{v_i v_j} \lesssim \tilde{t}_{Ev_j}$, $\forall v_i \in D_j$, and also know there is at least one $\approx$ which holds. The symbols $\prec$ are $\approx$ the ranking on F_N (see Definition 6). From Definitions 5 and 6 and also from Property 2, we obtain the following equations $d(\tilde{t}_{Ev_i}, \tilde{0}_1) + d(\tilde{t}_{v_i v_j}, \tilde{0}_1) \le d(\tilde{t}_{Ev_j}, \tilde{0}_1), \forall v_i \in D_j$, and also know there is at least one equal sign which holds. Furthermore, from (11) and (13) we know $t_{Ev_i}^* + t_{v_i v_j}^* \le t_{Ev_j}^*, \forall v_i \in D_j$, and also know at least one equal sign holds there. Hence, we obtain the following equation:
$$t_{Ev_j}^* = \max_{v_i \in D_j}(t_{Ev_i}^* + t_{v_i v_j}^*). \tag{26}$$
Since $t_{Ev_i}^* + t_{v_i v_j}^* \le t_{Ev_j}^*, \forall v_i \in D_j$, we obtain
$$\begin{aligned} t_{Ev_i} + t_{v_i v_j} + \tfrac{1}{4}\{(\Delta_{Ev_i 2} - \Delta_{Ev_i 1}) + (\Delta_{v_i v_j 2} - \Delta_{v_i v_j 1}\} \\ \le t_{Ev_j} + \tfrac{1}{4}(\Delta_{Ev_j 2} - \Delta_{Ev_j 1}), \quad \forall v_i \in D_j. \end{aligned}$$
In fact, the above equation can be derived directly from (15) and $t_{Ev_i} + t_{v_i v_j} \le t_{Ev_j}, \forall v_i \in D_j$. Therefore, no additional conditions for parameters are needed in (26).

Next, consider the equation $t_{Ev_i} = t_{Lv_i} = 0$. After fuzzifying both sides of the equation, we obtain $\tilde{t}_{Ev_1} \approx \tilde{t}_{Lv_1} \approx \tilde{0}_1$ and $d(\tilde{t}_{Ev_1}, \tilde{0}_1) = d(\tilde{t}_{Lv_1}, \tilde{0}_1) = d(\tilde{0}_1, \tilde{0}_1)$. Hence, we obtain $t_{Ev_1}^* = t_{Lv_1}^* = 0$, but it requires an additional condition, $\Delta_{Ev_1 2} = \Delta_{Ev_1 1} = \Delta_{Lv_1 2} = \Delta_{Lv_1 1} = 0$.

Similarly, from (8) we have $t_{Lv_i} \le t_{Lv_j} - t_{v_i v_j}, \forall v_j \in E_i$, and also have at least one equal sign which holds. We use the same procedure to fuzzify both sides of the equation, obtaining $\tilde{t}_{Lv_i} \lesssim \tilde{t}_{Lv_j} \ominus \tilde{t}_{v_i v_j}$, $\forall v_j \in E_i$ and also determine that at least one $\approx$ holds. From Definitions 5 and 6, and also from Property 2, we derive the equation, $d(\tilde{t}_{Lv_i}, \tilde{0}_1) \le d(\tilde{t}_{Lv_j}, \tilde{0}_1) - d(\tilde{t}_{v_i v_j}, \tilde{0}_1), \forall v_j \in E_i$, where at least one equal sign holds. Furthermore, from (11) and (17) we obtain $t_{Lv_i}^* \le t_{Lv_j}^* - t_{v_i v_j}^*, \forall v_j \in E_i$, where at least one equal sign holds. Thus we derive the following equation:
$$t_{Lv_i}^* = \max_{v_j \in E_i}(t_{Lv_j}^* - t_{v_i v_j}^*). \tag{27}$$
Similar to (26), the above equation can be derived from (19) and $t_{Lv_i} \le t_{Lv_j} - t_{v_i v_j}, \forall v_j \in E_i$. Therefore, no additional conditions for parameters are needed in (27). Finally, we consider the equation $t_{Lv_n} = t_{Ev_n}$. After fuzzifying the equation, we obtain $\tilde{t}_{Lv_n} \approx \tilde{t}_{Ev_n}$ and $d(\tilde{t}_{Lv_n}, \tilde{0}_1) = d(\tilde{t}_{Ev_n}, \tilde{0}_1)$. Then, by (13) and (17), we obtain $t_{Lv_n}^* = t_{Ev_n}^*$ and an additional condition for parameters:
$$\Delta_{Lv_n 2} - \Delta_{Lv_n 1} = \Delta_{Ev_n 2} - \Delta_{Ev_n 1}. \tag{28}$$
If $t_{Ev_j}^* = t_{Lv_j}^*$, then an estimate of the earliest event time for event v_j in the fuzzy sense is equal to an estimate of the latest event time for event v_j in the fuzzy sense. Obviously, this indicates there is no slack time. In conclusion, we summarize the above statements in the following property.

Property 4: Consider the fuzzy AOE network, $N^* = (V, A, T^*)$. If $\Delta_{Ev_1 2} = \Delta_{Ev_1 1} = \Delta_{Lv_1 2} = \Delta_{Lv_1 1} = 0$, and $\Delta_{Lv_n 2} - \Delta_{Lv_n 1} = \Delta_{Ev_n 2} - \Delta_{Ev_n 1}$, as well as the conditions in Property 3 hold, we summarize the following results.

1) An estimate of the earliest event time for event v_j in the fuzzy sense is $t_{Ev_j}^*$, which can be derived from $t_{Ev_j}^* = \max_{v_i \in D_j}(t_{Ev_i}^* + t_{v_i v_j}^*)$ and $t_{Ev_1}^* = t_{Lv_1}^* = 0$.
2) An estimate of the latest event time for event v_i in the fuzzy sense is $t_{Lv_i}^*$, which can be derived from $t_{Lv_i}^* = \min_{v_j \in E_i}(t_{Lv_j}^* - t_{v_i v_j}^*)$ and $t_{Lv_n}^* = t_{Ev_n}^*$.
3) The activity (v_i, v_j) will on the critical path if $t_{Ev_i}^* = t_{Lv_i}^*$ and $t_{Ev_j}^* = t_{Lv_j}^*$.

Proof: Equations (1) and (2) can be proved directly from (26) and (27). Since $t_{Ev_i}^* = t_{Lv_i}^*$ and $t_{Ev_j}^* = t_{Lv_j}^*$, from (5) of the Property 3, we obtain $t_{v_i v_j}^* = t_{Ev_j}^* - t_{Ev_i}^* = t_{Lv_j}^* - t_{Ev_i}^* = T_{v_i v_j}^*$. By (6) of Property 3, we have proved (3).

Example 2: Construct a fuzzy AOE network, $N^* = (V, A, T^*)$, from the crisp AOE network, $N = (V, A, T)$, of Example 1. Let $\tilde{t}_{v_1 v_2}$ = (2.1, 3, 3.8), $\tilde{t}_{v_1 v_3}$ = (3.5, 4, 5), $\tilde{t}_{v_1 v_4}$ = (5, 6, 7.2), $\tilde{t}_{v_2 v_3}$ = (3.2, 4, 4.8), $\tilde{t}_{v_3 v_4}$ = (4, 5, 6.3), $\tilde{t}_{v_2 v_5}$ = (4.1, 5, 6.1), $\tilde{t}_{v_3 v_6}$ = (2.8, 4, 5), $\tilde{t}_{v_4 v_6}$ = (4.9, 6, 7.2), $\tilde{t}_{v_5 v_6}$ = (1.5, 2, 2.7), $\tilde{t}_{v_6 v_7}$ = (3.8, 5, 6). After calculating by (1) of Property 3, we obtain the following estimates of processing time in the fuzzy sense, i.e. $t_{v_1 v_2}^* = 2.975$, $t_{v_1 v_3}^* = 4.125$, $t_{v_1 v_4}^* = 6.05$, $t_{v_2 v_3}^* = 4$, $t_{v_3 v_4}^* = 5.075$, $t_{v_2 v_5}^* = 5.05$, $t_{v_3 v_6}^* = 3.95$, $t_{v_4 v_6}^* = 6.025$, $t_{v_5 v_6}^* = 2.05$, and $t_{v_6 v_7}^* = 4.95$. We let $T^* = \{t_{v_1 v_2}^*, t_{v_1 v_3}^*, t_{v_1 v_4}^*, t_{v_2 v_3}^*, t_{v_3 v_4}^*, t_{v_2 v_5}^*, t_{v_3 v_6}^*, t_{v_4 v_6}^*, t_{v_5 v_6}^*, t_{v_6 v_7}^*\}$. Hence, we construct a fuzzy AOE network, $N^* = (V, A, T^*)$, as shown in Fig. 5.

Next, we should choose appropriate values for the parameters, i.e., $\Delta_{Ev_j 1}$, $\Delta_{Ev_j 2}$, $\Delta_{Ev_i 1}$, $\Delta_{Ev_i 2}$, $\Delta_{Lv_j 1}$, $\Delta_{Lv_j 2}$, $\Delta_{Lv_i 1}$, $\Delta_{Lv_i 2}$, $\varpi_{v_i v_j 1}$, and $\varpi_{v_i v_j 2}$, to satisfy Properties 3 and 4. Then we calculate t_{Ei}^* and t_{Ej}^* by using (1) and (2) of Property 4 in the following table (see Remark 4).

In Table I, for example, we calculate t_{E3}^* by using (1) of Property 4 as follows. Since $j = 3$, we have $D_j = \{i | (i, j) \in A\} = \{i = 0, i = 2\}$ (see Fig. 5). From Table I, we find that $t_{E0}^* = 0$, $t_{E2}^* = 6.975$, $t_{03}^* = 6.05$, and $t_{23}^* = 5.075$. Thus, we obtain
$$\begin{aligned} t_{E3}^* &= \max\{t_{E0}^* + t_{03}^*, t_{E2}^* + t_{23}^*\} \\ &= \max\{0 + 6.05, 6.975 + 5.075\} = 12.05. \end{aligned}$$
In Table II, for example, we calculate t_{L2}^* by using (2) of Property 4 as follows. Since $i = 2$, we have $E_i = \{j | (i, j) \in A\} = \{j = 3, j = 5\}$ (see Fig. 5). From Table II, we find that $t_{L3}^* = 12.05, t_{L5}^* = 18.075$, $t_{23}^* = 5.075$, and $t_{25}^* = 3.95$. Thus, we obtain $t_{L2}^* = \min\{t_{L5}^* - t_{25}^*, t_{L3}^* - t_{23}^*\} = \min\{18.075 - 3.95, 12.05 - 5.075\} = 6.975$.

From Tables I and II, we obtain Table III. Table III shows the process of finding a fuzzy critical path in N^*. According to the rule of (3) in

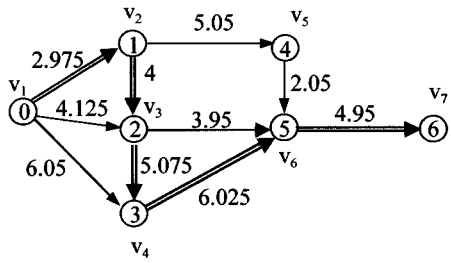

Fig. 5. Fuzzy AOE network $N^* = (V, A, T^*)$.

TABLE I
COMPUTATION OF t^*_{Ei}, THE EARLIEST EVENT TIME FOR EVENT i IN THE FUZZY SENSE

t^*_{Lj}									t^*_{Ei}
j	0	1	2	3	4	5	6	i	
		2.975	4.125	6.05 (t^*_{03})				0	0 (t^*_{E0})
			4		5.05			1	2.975
				5.075 (t^*_{23})		3.95		2	6.975 (t^*_{E2})
						6.025		3	12.050 (t^*_{E3})
						2.05		4	8.025
							4.95	5	18.075
								6	23.025

Note: Event i in Table I corresponds to node v_{i+1} in Fig. 5, i.e. $t^*_{Ei} = t^*_{Ev_{i+1}}$.

TABLE II
COMPUTATION OF t^*_{Lj}, THE EARLIEST EVENT TIME FOR EVENT j IN THE FUZZY SENSE

t^*_{Lj}	0	2.975	6.975 (t^*_{12})	12.05 (t^*_{13})	16.025	18.075 (t^*_{15})	23.025		t^*_{Ei}
j	0	1	2	3	4	5	6	i	
		2.975	4.125	6.05				0	0
			4		5.05			1	2.975
				5.075 (t^*_{23})		3.95 (t^*_{25})		2	6.975
						6.025		3	12.050
						2.05		4	8.025
							4.95	5	18.075
								6	23.025

Note: Event j in Table II corresponds to node v_{j+1} in Fig. 5, i.e. $t^*_{Lj} = t^*_{Lv_{j+1}}$.

TABLE III
PROCESSES OF FINDING A FUZZY CRITICAL PATH IN $N^* = (V, A, T^*)$

j	0	1	2	3	4	5	6
t^*_{Ej}	0	2.975	6.975	12.05	8.025	18.075	23.025
t^*_{Lj}	0	2.975	6.975	12.05	16.025	18.075	23.025
critical path	0	1	2	3		5	6

Property 4, we find a critical path $0 \to 1 \to 2 \to 3 \to 5 \to 6$ (i.e., $v_1 \to v_2 \to v_3 \to v_4 \to v_6 \to v_7$). The total time of the path is $t^* = 23.025$. However, in the crisp case of Example 1, the critical path of $N = (V, A, T)$ is also $0 \to 1 \to 2 \to 3 \to 5 \to 6$, and the total time of the path is $t = 3 + 4 + 5 + 6 + 5 = 23$ (see Table IV).

Remark 4: An example to explain our approach for determining the appropriate values for those parameters to satisfy the conditions of Properties 3 and 4. Here we only consider the activity (v_5, v_6). Since $\tilde{t}_{v_5v_6} = (1.5, 2, 2.7)$, we obtain $\Delta_{v_5v_6 1} = 0.5$ and $\Delta_{v_5v_6 2} = 0.7$ by simple subtraction. Property 3 states that the fuzzy numbers

TABLE IV
COMPUTATION OF THE TOTAL TIME OF A CRITICAL PATH IN $N = (V, A, T)$

j	0	1	2	3	4	5	6
t_{Ej}	0	3	7	12	8	18	23
t_{Lj}	0	3	7	12	16	18	23

should satisfy (15), (19), (23), and (24). Hence, $\Delta_{Ev_6 2} - \Delta_{Ev_6 1} = (\Delta_{Ev_5 2} - \Delta_{Ev_5 1}) + 0.2$, $\Delta_{Lv_5 2} - \Delta_{Lv_5 1} = (\Delta_{Lv_6 2} - \Delta_{Lv_6 1}) - 0.2$, $\varpi_{v_5v_6 2} - \varpi_{v_5v_6 1} = (\Delta_{Lv_6 2} - \Delta_{Lv_6 1}) - (\Delta_{Ev_5 2} - \Delta_{Ev_5 1})$, $\varpi_{v_5v_6 2} - \varpi_{v_5v_6 1} = 0.2$. Since $v_j = j - 1$, from Table IV we have $t_{Ev_5} = t_{E4} = 8$, $t_{Lv_5} = t_{L4} = 16$ (etc.), $t_{Ev_6} = 18$, $t_{Lv_6} = 18$, $T_{v_5v_6} = t_{Lv_6} - t_{Ev_5} = 10$, $t_{Ev_1} = t_{Lv_1} = 0$, and $t_{Ev_7} = t_{Lv_7} = 23$. Subsequently, we take the following equations $\Delta_{Ev_6 2} - \Delta_{Ev_6 1} = \Delta_{Lv_6 2} - \Delta_{Lv_6 1} = 0.3$, $\Delta_{Ev_5 2} - \Delta_{Ev_5 1} = \Delta_{Lv_5 2} - \Delta_{Lv_5 1} = 0.1$, and $\varpi_{v_5v_6 2} - \varpi_{v_5v_6 1} = 0.2$, where $\Delta_{Ev_5 q}$, $\Delta_{Ev_6 q}$, $\Delta_{Lv_5 q}$, $\Delta_{Lv_6 q}$, $\varpi_{v_5v_6 q}$, $q = 1, 2$, to satisfy the following criteria, $0 < \Delta_{Ev_5 1} < t_{Ev_5} = 8$, $0 < \Delta_{Ev_5 2}$, $0 < \Delta_{Ev_6 1} < 18$, $0 < \Delta_{Ev_6 2}$, $0 < \Delta_{Lv_5 1} < 16$, $0 < \Delta_{Lv_5 2}$, $0 < \Delta_{Lv_6 1} < 18$, $0 < \Delta_{Lv_6 2}$, $0 < \varpi_{v_5v_6 1} < T_{v_5v_6} = 10$, and $0 < \varpi_{v_5v_6 2}$. Finally, we could choose appropriate values for the above parameters to satisfy Property 3. For example, we choose $\Delta_{Ev_5 2} = 0.2$, $\Delta_{Ev_5 1} = 0.1$, $\Delta_{Ev_6 2} = 0.5$, $\Delta_{Ev_6 1} = 0.2$, $\Delta_{Lv_5 2} = 0.3$, $\Delta_{Lv_5 1} = 0.2$, $\Delta_{Lv_6 2} = 0.4$, $\Delta_{Lv_6 1} = 0.1$, $\varpi_{v_5v_6 2} = 0.3$, and $\varpi_{v_5v_6 1} = 0.1$. Then, we obtain $\tilde{t}_{Ev_5} = (8 - 0.1, 8, 8 + 0.2)$, $\tilde{t}_{Ev_6} = (17.8, 18, 18.5)$, $\tilde{t}_{Lv_5} = (15.8, 16, 16.3)$, $\tilde{t}_{Lv_6} = (17.9, 18, 18.4)$, and $\tilde{T}_{v_5v_6} = (9.9, 10, 10.3)$. From (13), (17), and (21) we obtain $t^*_{Ev_5} = t^*_{E4} = 8.025$, $t^*_{Lv_5} = t^*_{L4} = 16.025$ (etc.), $t^*_{Ev_6} = 18.075$, $t^*_{Lv_6} = 18.075$, and $T^*_{v_5v_6} = 10.05$ $(= t^*_{Lv_6} - t^*_{Ev_5})$. Notice that we have the same results here as in Table III.

IV. CONCLUDING REMARKS

This paper has presented a ranking method for fuzzy numbers in a CPM of AOE networks. The focus of the paper was to introduce the signed distance ranking of fuzzy numbers, and use them to obtain fuzzy critical paths. In Section III, we discussed fuzzy CPM based on a signed distance ranking of fuzzy numbers. When the processing time for activity (v_i, v_j) in the crisp case is $t_{v_iv_j}$ and the fuzzy number of $t_{v_iv_j}$ is $\tilde{t}_{v_iv_j} = (t_{v_iv_j} - \Delta_{v_iv_j 1}, t_{v_iv_j}, t_{v_iv_j} + \Delta_{v_iv_j 2})$, where $0 < \Delta_{v_iv_j 1} < t_{v_iv_j}$, and $0 < \Delta_{v_iv_j 2}$, then an estimate of the processing time in the fuzzy sense is given by $t^*_{v_iv_j} = t_{v_iv_j} + \frac{1}{4}(\Delta_{v_iv_j 2} - \Delta_{v_iv_j 1})$. Interpretation of the result is as follows, viz. Fig. 4. When $\Delta_{v_iv_j 1} < \Delta_{v_iv_j 2}$, the triangle goes to the right side (larger $t_{v_iv_j}$), and we obtain $t_{v_iv_j} < t^*_{v_iv_j}$. Conversely, when $\Delta_{v_iv_j 2} < \Delta_{v_iv_j 1}$, the triangle goes to the left side (smaller $t_{v_iv_j}$), and we obtain $t^*_{v_iv_j} < t_{v_iv_j}$. However, if $\Delta_{v_iv_j 2} = \Delta_{v_iv_j 1}$, then it is an isosceles triangle, so we have $t_{v_iv_j} = t^*_{v_iv_j}$. We conclude that if the following criteria a)–d) hold, the fuzzy AOE network, $N^* = (V, A, T^*)$, which is defined in Properties 3 and 4, becomes the crisp AOE network, $N = (V, A, T)$, which is defined in Section III-A.

1) For each activity (v_i, v_j), if $\Delta_{v_iv_j 2} = \Delta_{v_iv_j 1}$, which is defined in (11), then
$$t^*_{v_iv_j} = t_{v_iv_j}.$$
2) For each event v_j, if $\Delta_{Ev_j 2} = \Delta_{Ev_j 1}$, as defined in (13), then $t^*_{Ev_j} = t_{Ev_j}$.
3) For each event v_j, if $\Delta_{Lv_j 2} = \Delta_{Lv_j 1}$, as defined in (17), then $t^*_{Lv_j} = t_{Lv_j}$.
4) For each activity (v_i, v_j), if $\varpi_{v_iv_j 2} = \varpi_{v_iv_j 1}$, as defined in (21), then $T^*_{v_iv_j} = T_{v_iv_j}$.

Consequently, we conclude that the fuzzy AOE network, $N^* = (V, A, T^*)$, is an extension of the crisp AOE network, $N = (V, A, T)$.

In addition, we let $(v_{j_0}, v_{j_1}), (v_{j_1}, v_{j_2}), \cdots, (v_{j_{m-1}}, v_{j_m})$, where $0 < m < n$, $j_0 = v_1$, and $v_m = v_n$, be the critical path in the network $N = (V, A, T)$. The processing times for each activity on the critical path are $t_{v_{j_0}v_{j_1}}$, $t_{v_{j_1}v_{j_2}}, \cdots, t_{v_{j_{m-1}}v_{j_m}}$ and they satisfy $T_{v_{j_{p-1}}v_{j_p}} = t_{v_{j_{p-1}}v_{j_p}}$, $p = 1, 2, \cdots, m$, in (9). We make both sides of $T_{v_{j_{p-1}}v_{j_p}} = t_{v_{j_{p-1}}v_{j_p}}$ fuzzy, thus obtaining $\tilde{T}_{v_{j_{p-1}}v_{j_p}} \approx \tilde{T}_{v_{j_{p-1}}v_{j_p}}$. By Definition 6 we have $d(\tilde{T}_{v_{j_{p-1}}v_{j_p}}, \tilde{0}_1) = d(\tilde{t}_{v_{j_{p-1}}v_{j_p}}, \tilde{0}_1)$. Then, from (11) and (21) we obtain

$$T^*_{v_{j_{p-1}}v_{j_p}} = t^*_{v_{j_{p-1}}v_{j_p}}, \qquad p = 1, 2, \cdots, m. \tag{29}$$

Therefore,

$$\begin{aligned} &T_{v_{j_{p-1}}v_{j_p}} + \tfrac{1}{4}(\varpi_{v_{j_{p-1}}v_{j_p}2} - \varpi_{v_{j_{p-1}}v_{j_p}1}) \\ &\quad = t_{v_{j_{p-1}}v_{j_p}} + \tfrac{1}{4}(\Delta_{v_{j_{p-1}}v_{j_p}2} - \Delta_{v_{j_{p-1}}v_{j_p}1}). \end{aligned}$$

Since $T_{v_{j_{p-1}}v_{j_p}} = t_{v_{j_{p-1}}v_{j_p}}$, we obtain the following condition for parameters

$$\begin{aligned} &\varpi_{v_{j_{p-1}}v_{j_p}2} - \varpi_{v_{j_{p-1}}v_{j_p}1} \\ &\quad = \Delta_{v_{j_{p-1}}v_{j_p}2} - \Delta_{v_{j_{p-1}}v_{j_p}1}, \qquad p = 1, 2, \cdots, m. \end{aligned} \tag{30}$$

In summary, we conclude that if we hold (29) and (30), then by (6) of Property 3 and (3) of Property 4, the fuzzy critical path in a fuzzy AOE network is the same as the critical path in a crisp AOE network.

REFERENCES

[1] I. Gazdik, "Fuzzy-network planning-FNET," *IEEE Trans. Rel.*, vol. R-32, no. 3, pp. 304–313, 1983.

[2] E. Horowitz, S. Sahni, and D. Mehta, *Fundamental of Data Structures in C++*, New York: Freeman, 1995.

[3] A. Kaufmann and M. M. Gupta, *Introduction to Fuzzy Arithmetic, Theory and Applications*, New York: Van Nostrand Reinhold, 1991.

[4] S. H. Nasution, "Fuzzy critical path method," *IEEE Trans. Syst., Man., Cybern.*, vol. 24, pp. 48–57, Jan. 1994.

[5] J. S. Yao and K. M. Wu, "Ranking Fuzzy numbers based on decomposition principle and signed distance," *Fuzzy Sets Syst.*, to be published.

[6] H.-J. Zimmermann, *Fuzzy Set Theory and Its Applications*, 2nd ed. Boston, MA: Kluwer, 1991.