Isolated 3D Object Recognition through Next View Planning

Sumantra Dutta Roy, Santanu Chaudhury, Subhashis Banerjee

S. Dutta Roy and S. Banerjee are with the Department of Computer Sc. & Engg., Indian Institute of Technology, Hauz Khas, New Delhi - 110 016, INDIA. E-mail: {sumantra,suban}@cse.iitd.ernet.in

S. Chaudhury is with the Department of Electrical Engg., Indian Institute of Technology, Hauz Khas, New Delhi - 110 016, INDIA. E-mail: santanuc@{cse,ee}.iitd.ernet.in

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 3D 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.

Keywords

Active Vision, Reactive Planning, 3D Object Recognition

I. Introduction

In this paper, we present a new on-line scheme for the recognition of an isolated 3D 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], [2], [3], [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 3D 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 3D invariants have been proposed only for special cases so far (eg. [3]). The purpose of this paper is to investigate the use of suitably planned multiple views and 2D invariants for 3D 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, Allen and Tsai [5] survey the field of sensor planning for vision tasks. We can compare various active 3D 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 3D 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 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 the concluding section.

II. The Knowledge Representation Scheme

A view of a 3D 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 3D object is independent of the viewpoint. Koenderink and van Doorn [10] define Aspects as topologically equivalent classes of object appearances. Ikeuchi and co-workers 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.

Figure 1 shows a simple example of an object with its associated aspects and classes. The locus of view-directions is one-dimensional 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.

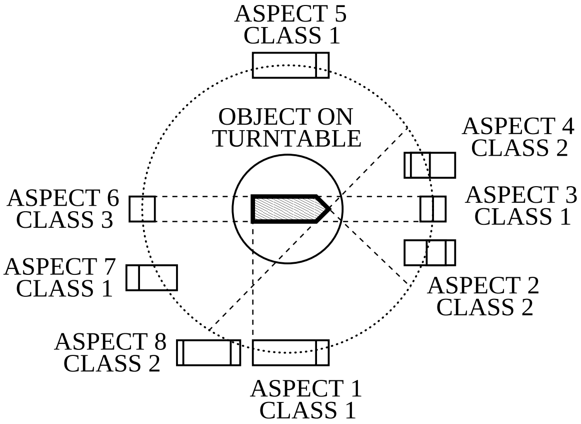


Fig. 1. Aspects and Classes of an object

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. Figure 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

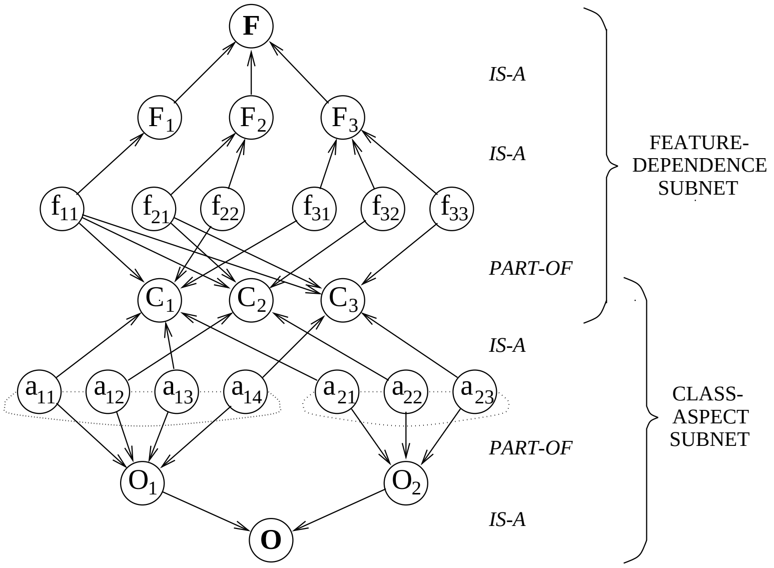


Fig. 2. The knowledge representation scheme: an example

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 non-adaptive 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 $[f_{1j_1}\ f_{2j_2}\ \ldots\ f_{nj_n}]$. 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.

- $\mathbf{O}$ 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 neighbouring 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.

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 Figure 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. Figure 4 presents the overall

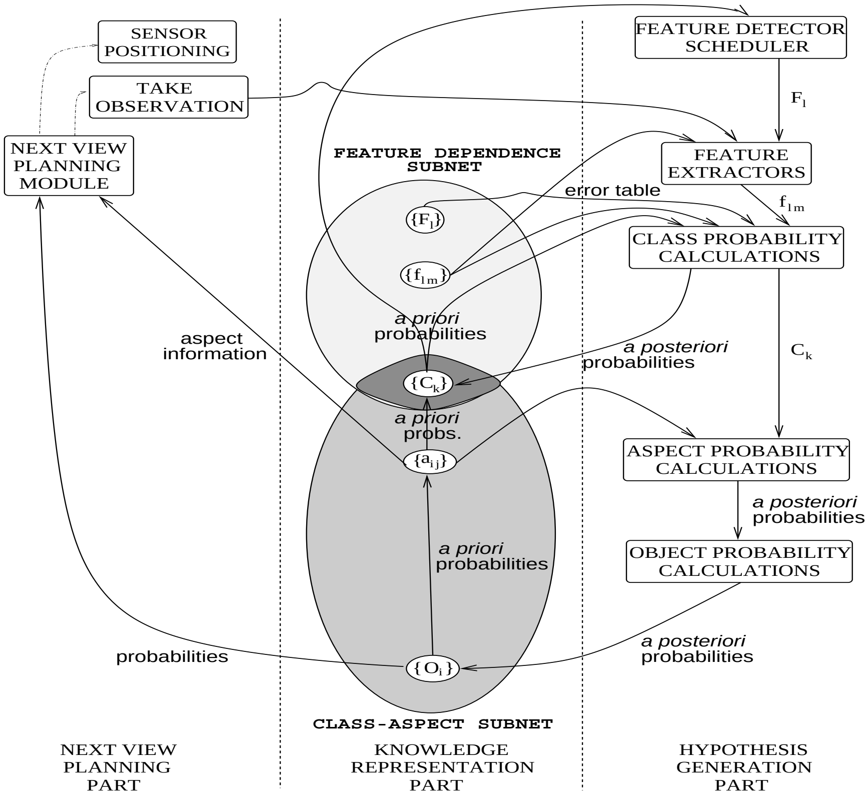

SENSOR POSITIONING
TAKE OBSERVATION
NEXT VIEW PLANNING MODULE
FEATURE DETECTOR SCHEDULER
F_l
FEATURE EXTRACTORS
f_{lm}
FEATURE DEPENDENCE SUBNET
error table
CLASS PROBABILITY CALCULATIONS
$\{F_l\}$
$\{f_{lm}\}$
a priori probabilities
aspect information
a posteriori probabilities
C_k
$\{C_k\}$
a priori probs.
$\{a_{ij}\}$
ASPECT PROBABILITY CALCULATIONS
a posteriori probabilities
a priori probabilities
OBJECT PROBABILITY CALCULATIONS
probabilities
$\{O_i\}$
a posteriori probabilities
CLASS-ASPECT SUBNET
NEXT VIEW PLANNING PART
KNOWLEDGE REPRESENTATION PART
HYPOTHESIS GENERATION PART

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

algorithm.

A.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 Figure 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.

A.2 Class Probability Calculations Using the Knowledge Representation Scheme

We obtain the *a priori* probability of class C_i as:

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

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 Equation 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.

```
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. The Class Recognition Algorithm

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:

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

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, $P(a_{j_p k_p}|Oj_p)$ is $\theta_{j_p k_p}/360$. *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 Equation 5 below:

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

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.

Let the class recognition phase report the observed class to be C_i. Let us assume that C_i could have come from aspects $a_{j_1k_1}$, $a_{j_2k_2}$, $\ldots a_{j_mk_m}$, where all j_1, j_2, $\ldots j_m$ 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, eg. 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 disambiguate 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: (Figure 5(a)) the unique class observed for the angular movement made so far, the aspects

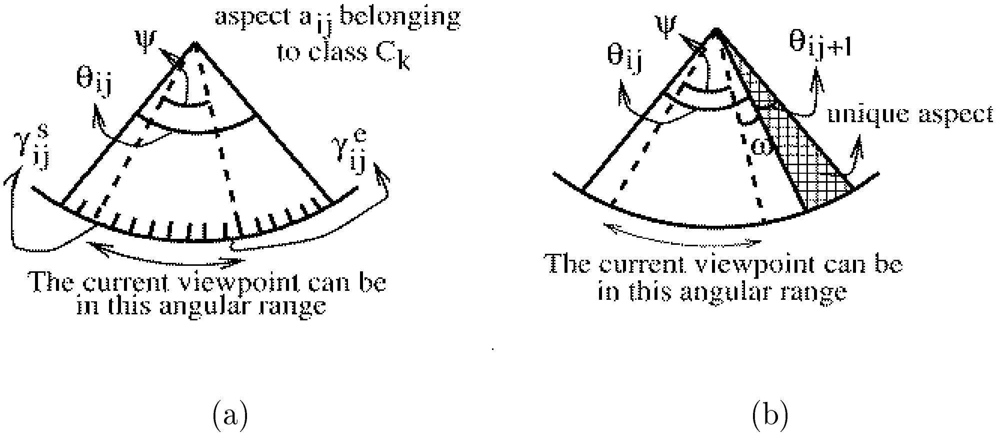

aspect a_{ij} belonging to class C_k
ψ
θ_{ij}
γ_{ij}^{s}
γ_{ij}^{e}
The current viewpoint can be in this angular range
ψ
θ_{ij}
θ_{ij+1}
unique aspect
ω
The current viewpoint can be in this angular range
(a)
(b)

Fig. 5. (a) The notation used (Section IV) (b) A case when our algorithm is not guaranteed to succeed (Section IV-A) possible for this angle-class pair, and for each aspect, the range of positions possible within it $(\gamma_{ij}^s - \gamma_{ij}^e)$. γ_{ij}^s and γ_{ij}^e denote the two positions within aspect a_{ij} where the current viewpoint can be, as a result of the movement made thus far. Here, $\gamma_{ij}^s \leq \gamma_{ij}^e$; and γ_{ij}^s, $\gamma_{ij}^e \in [0, \theta_{ij}]$, 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 degrees or more from the root node.

Figure 6 shows an example of a partially constructed search tree. From a viewpoint, we categorize possible moves as follows.

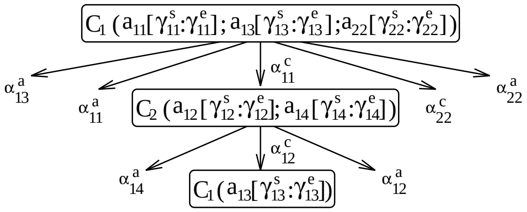

$C_1\ (a_{11}[\gamma_{11}^s{:}\gamma_{11}^e];\ a_{13}[\gamma_{13}^s{:}\gamma_{13}^e];a_{22}[\gamma_{22}^s{:}\gamma_{22}^e])$
α_{13}^a
α_{11}^a
α_{11}^c
α_{22}^c
α_{22}^a
$C_2\ (a_{12}[\gamma_{12}^s{:}\gamma_{12}^e];\ a_{14}[\gamma_{14}^s{:}\gamma_{14}^e])$
α_{14}^a
α_{12}^c
α_{12}^a
$C_1(a_{13}[\gamma_{13}^s{:}\gamma_{13}^e])$

Fig. 6. A Partially Constructed Search Tree

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 anti-clockwise 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 anti-clockwise direction, the aspect reached depends on the initial position in the current aspect. We choose the move which leads us to the side with the largest angular range possible in any reachable aspect.

We expand a non-leaf 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 Type I and II primary moves and 1 to Type II auxiliary moves. Type III primary moves moves get a code of 2, and Type III 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 the class identification phase are repeated. Figure 7 presents our overall object identification algorithm in detail. Figure 3 shows the interaction of the next view planning part with the rest of the system.

```
                 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. The Object Recognition Algorithm

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^o. 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 degrees. Figure 5(b) shows an example of the third case. Let us suppose that we have to move anti-clockwise. 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 anti-clockwise 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 $T_{avg}(n) = 1 + \frac{\sum_{j=1}^{n-1} T_{avg}(j)}{n-1}$, 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 degree movement. We use simple and robust features with low feature extraction cost, compared to systems using complex features (eg. [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. **Experiments with Model Base I**

(Polyhedral Objects): We use as features, the number of horizontal and vertical lines

($\langle hv \rangle$), and the number of non-background 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. Figure 8 shows the objects in this model base. Figures 9 and 10 show some experiments with the objects in the first model base. For Figure 9, the initial class observed in each case is $\langle 232 \rangle$, while it is $\langle 221 \rangle$ in Figure 10. We make the following observations:

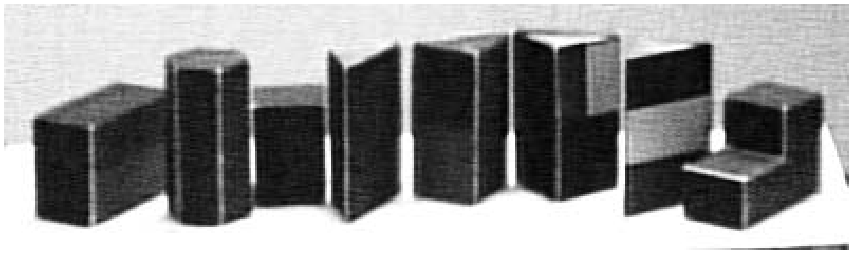


Fig. 8. Model Base I: The objects (from left) are O_1, O_2, O_3, O_4, O_5, O_6, O_7 and O_8, respectively.

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 time are limited, the planning process may use primary moves alone. An interesting case is observed in Figures 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.

Ordering of feature detectors: The third image in Figure 9(a) shows an 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.

Recovering from feature detection errors: The second image in Figure 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 Equation 3 result in the probabilities of the two as 0.997 and 0.002, respectively.

Variation of object probabilities: Figure 11 shows the variation in object probabilities with each observation. The two cases shown here are for the moves in Figure 9(a) and Figure 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 till 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.

Some sample search tree details: We now consider some cases in detail. For each row in Figure 9, the initial view could have come from 18 aspects belonging to objects in our model base and for Figure 10, the corresponding number is 17. For the strategy involving primary moves alone, the total number of search tree nodes generated for Figures 9(a), 9(b), 10(a) and 10(b) are 53, 48, 34 and 48, respectively. For the strategy involving both primary and auxiliary moves (Figures 9(c), 9(d), 10(d) and 10(e)), the corresponding numbers are 324, 279, 127 and 127, respectively. Let us consider Figure 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.

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.

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

2. 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. Figure 12 shows the objects in this model base.

For most of the 58 experiments (Figures 13, 14 and 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 10), compared to that for the first(18). The second images in Figures 14(a), (b) and (d) show cases where the system does not need to use the second feature detector. In the first image in Figure 13(b), due to the shadow of the wing on the fuselage of the aircraft, the feature detector detects 4 vertical lines instead of 3, the correct number. Our recovery mechanism (Section III-A.2) corrects this error. For the experiments shown in Figure 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 Figure 15 are 14 and 41, respectively.

VI. Conclusions

This paper presents an integrated approach for the recognition of an isolated 3D 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, colour, 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 3D shapes (as in Figure 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 (eg. 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 Computing Surveys*, vol. 17, no. 1, pp. 76–145, March 1985.

[2] R. T. Chin and C. R. Dyer, “Model Based Recognition in Robot Vision,” *ACM Computing Surveys*, vol. 18, no. 1, pp. 67–108, March 1986.

[3] A. Zisserman, D. Forsyth, J. Mundy, C. Rothwell, J. Liu, and N. Pillow, “3D object recognition using invariance,” *Artificial Intelligence*, vol. 78, pp. 239–288, 1995.

[4] D. P. Mukherjee and D. Dutta Majumder, “On Shape from Symmetry,” *Proc. Indian National Science Academy*, vol. 62, A, 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*

Transactions on Robotics and Automation, vol. 11, no. 1, pp. 86–104, February 1995.

[6] J. Maver and R. Bajcsy, “Occlusions as a Guide for Planning the Next View,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 5, pp. 76–145, May 1993.

[7] K. D. Gremban and K. Ikeuchi, “Planning Multiple Observations for Object Recognition,” *International Journal of Computer Vision*, vol. 12, no. 2/3, pp. 137–172, April 1994, Special Issue on Active Vision II.

[8] S. J. Dickinson, H. I. Christensen, J. Tsotsos, and G. Olofsson, “Active Object Recognition Integrating Attention and View Point Control,” *Computer Vision and Image Understanding*, vol. 67, no. 3, pp. 239–260, September 1997.

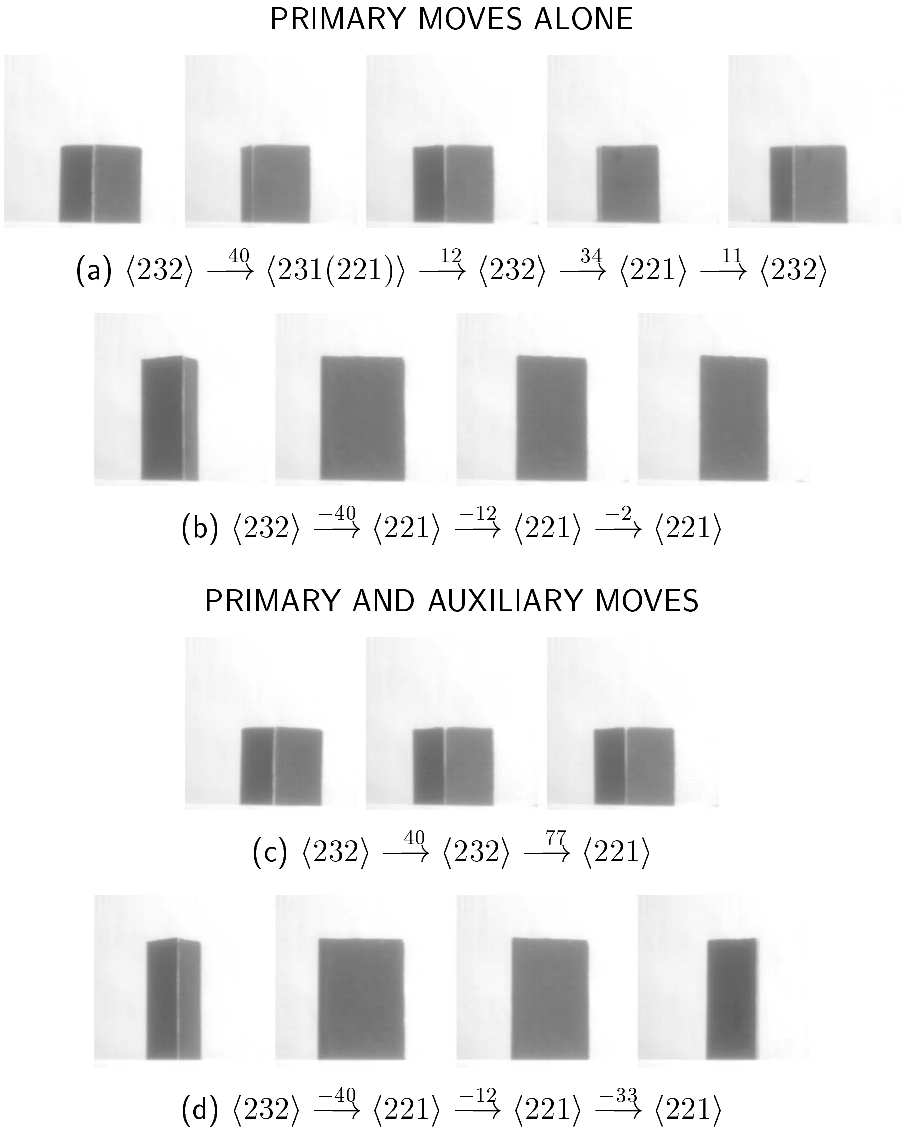
[9] S. A. Hutchinson and A. C. Kak, “Planning Sensing Strategies in a Robot Work Cell with Multi-Sensor Capabilities,” *IEEE Transactions on Robotics and Automation*, vol. 5, no. 6, pp. 765–783, December 1989.

[10] J. J. Koenderink and A. J. van Doorn, “The internal representation of solid shape with respect to vision,” *Biological Cybernetics*, 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 Sytems*, A. K. Jain and P. J. Flynn, Eds., pp. 229–258. Elsevier-Science Publishers, 1993.

[12] R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision*, vol. I, Addison-Wesley Publishing Company, Inc., 1992.

[13] D. Dutta Majumder and K. S. Ray, “Recongnition and Position Determination of Partially Occluded Object for a Computer Vision System,” *Journal of the IETE*, vol. 37, no. 5 & 6, pp. 419–442, 1991.



PRIMARY MOVES ALONE

(a) $\langle 232\rangle \xrightarrow{-40} \langle 231(221)\rangle \xrightarrow{-12} \langle 232\rangle \xrightarrow{-34} \langle 221\rangle \xrightarrow{-11} \langle 232\rangle$

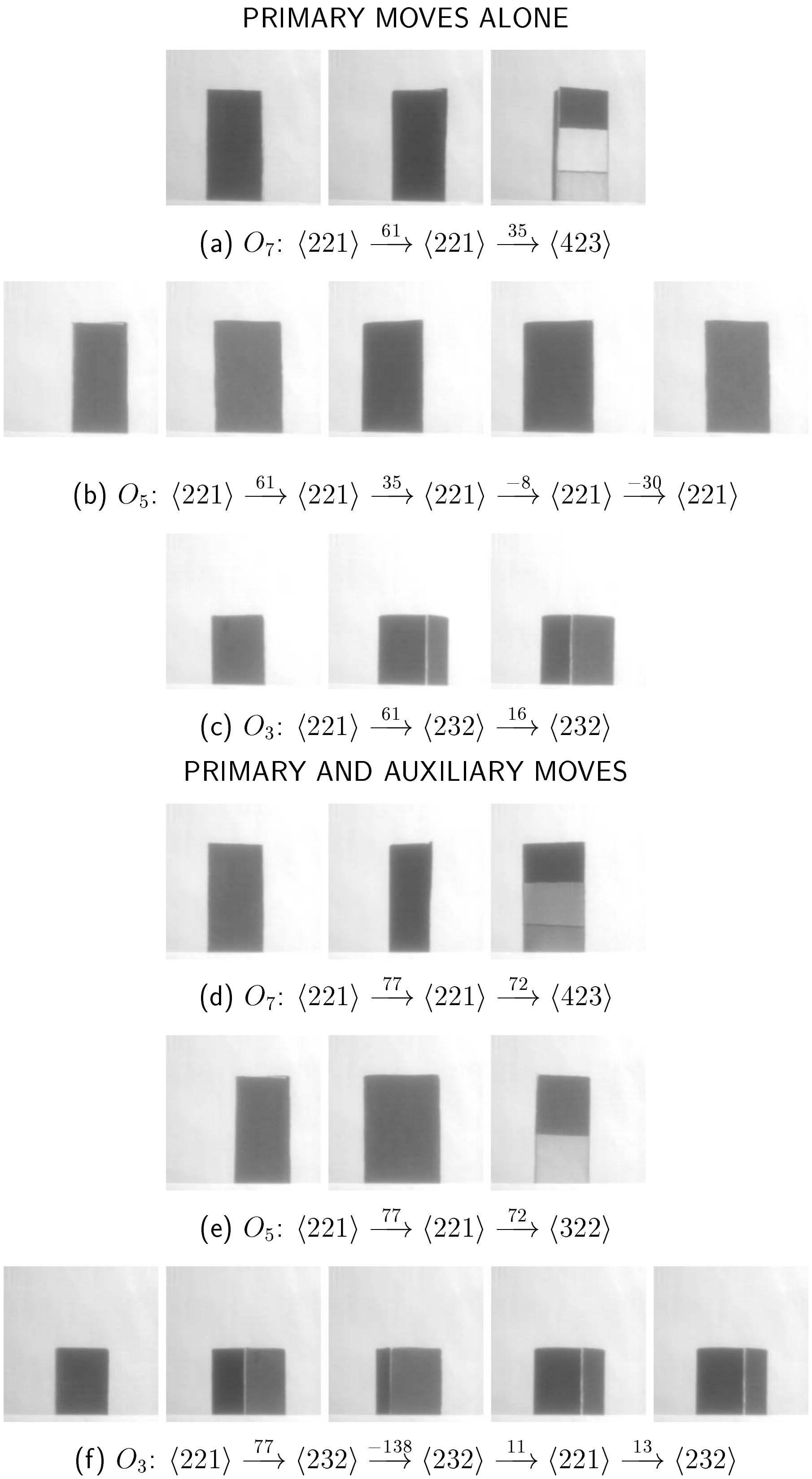
(b) $\langle 232\rangle \xrightarrow{-40} \langle 221\rangle \xrightarrow{-12} \langle 221\rangle \xrightarrow{-2} \langle 221\rangle$

PRIMARY AND AUXILIARY MOVES

(c) $\langle 232\rangle \xrightarrow{-40} \langle 232\rangle \xrightarrow{-77} \langle 221\rangle$

(d) $\langle 232\rangle \xrightarrow{-40} \langle 221\rangle \xrightarrow{-12} \langle 221\rangle \xrightarrow{-33} \langle 221\rangle$

Fig. 9. Some experiments with Model Base I: initial class $\langle 232\rangle$. The objects are O_3((a), (c)) and O_4((b), (d)), respectively. The numbers above the arrows denote the number of turntable steps. A negative sign indicates a clockwise movement. (The figure in parenthesis shows an example of recovery from feature detection errors)



PRIMARY MOVES ALONE

(a) O_7: $\langle 221\rangle \xrightarrow{61} \langle 221\rangle \xrightarrow{35} \langle 423\rangle$

(b) O_5: $\langle 221\rangle \xrightarrow{61} \langle 221\rangle \xrightarrow{35} \langle 221\rangle \xrightarrow{-8} \langle 221\rangle \xrightarrow{-30} \langle 221\rangle$

(c) O_3: $\langle 221\rangle \xrightarrow{61} \langle 232\rangle \xrightarrow{16} \langle 232\rangle$

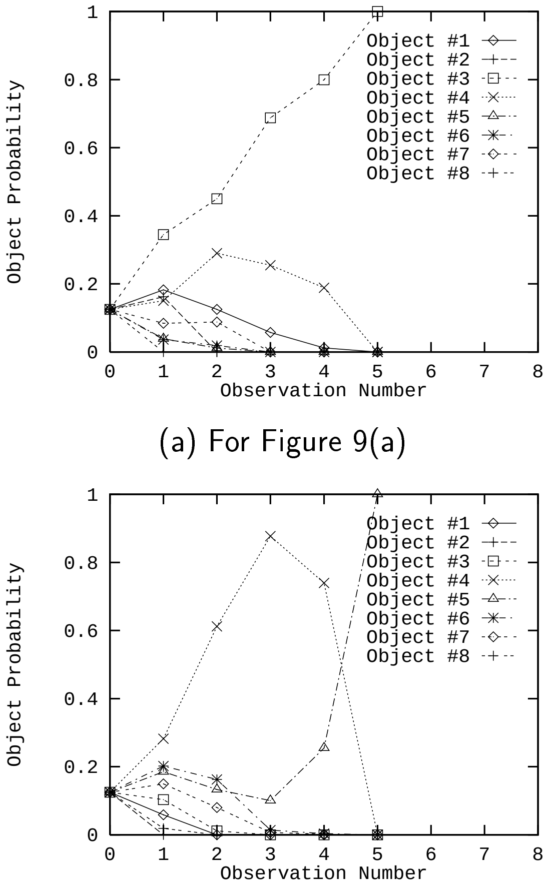
PRIMARY AND AUXILIARY MOVES

(d) O_7: $\langle 221\rangle \xrightarrow{77} \langle 221\rangle \xrightarrow{72} \langle 423\rangle$

(e) O_5: $\langle 221\rangle \xrightarrow{77} \langle 221\rangle \xrightarrow{72} \langle 322\rangle$

(f) O_3: $\langle 221\rangle \xrightarrow{77} \langle 232\rangle \xrightarrow{-138} \langle 232\rangle \xrightarrow{11} \langle 221\rangle \xrightarrow{13} \langle 232\rangle$

Fig. 10. Some experiments with Model Base I: initial class $\langle 221\rangle$. The numbers above the arrows denote the number of turntable steps. A negative sign indicates a clockwise movement



(a) For Figure 9(a)

(b) For Figure 10(b)

Fig. 11. Variation of object probabilities: two examples

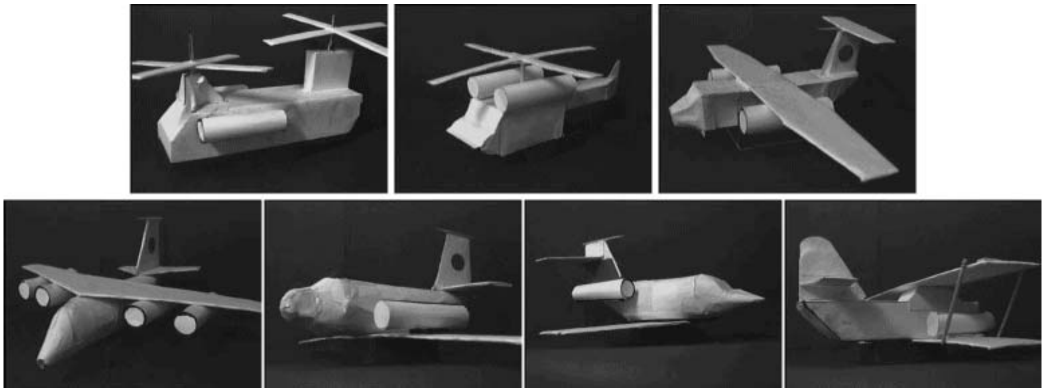
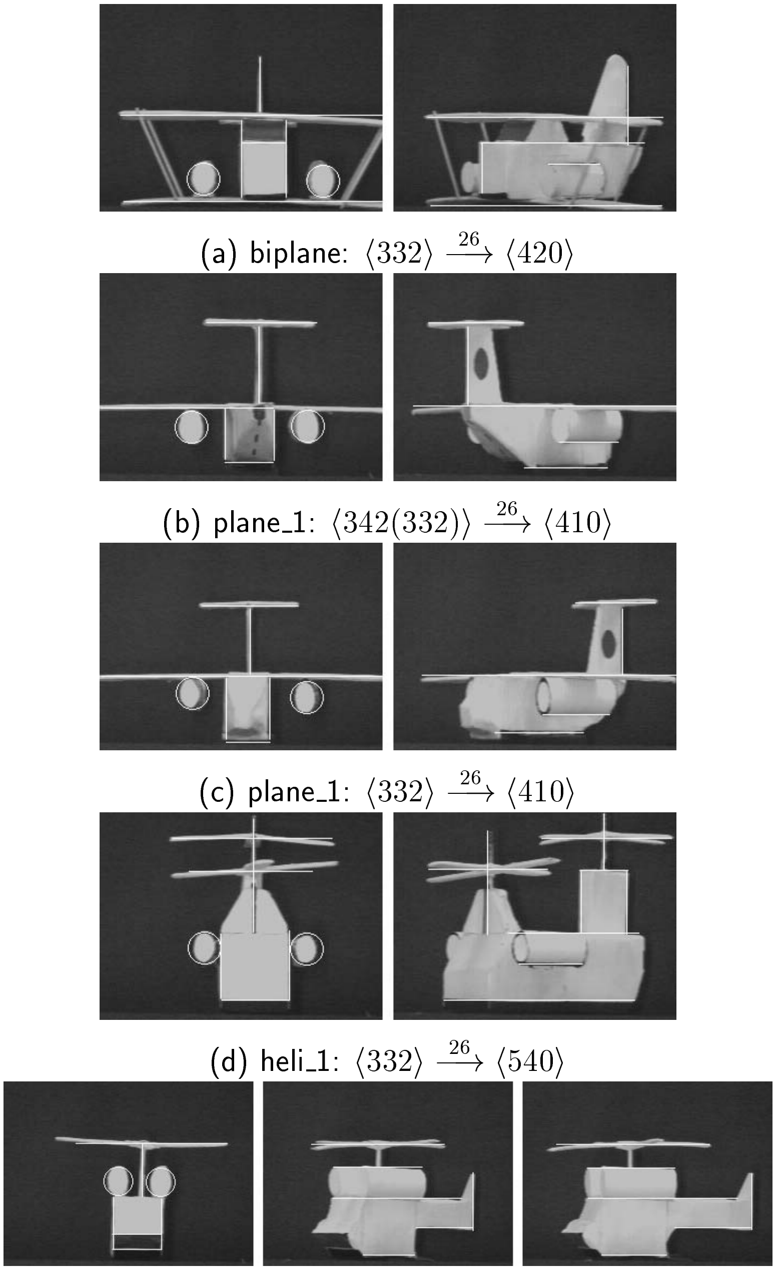


Fig. 12. Model Base II: The objects (in row major order) are heli_1, heli_2, plane_1, plane_2, plane_3, plane_4, and biplane.



(a) biplane: $\langle 332\rangle \xrightarrow{26} \langle 420\rangle$

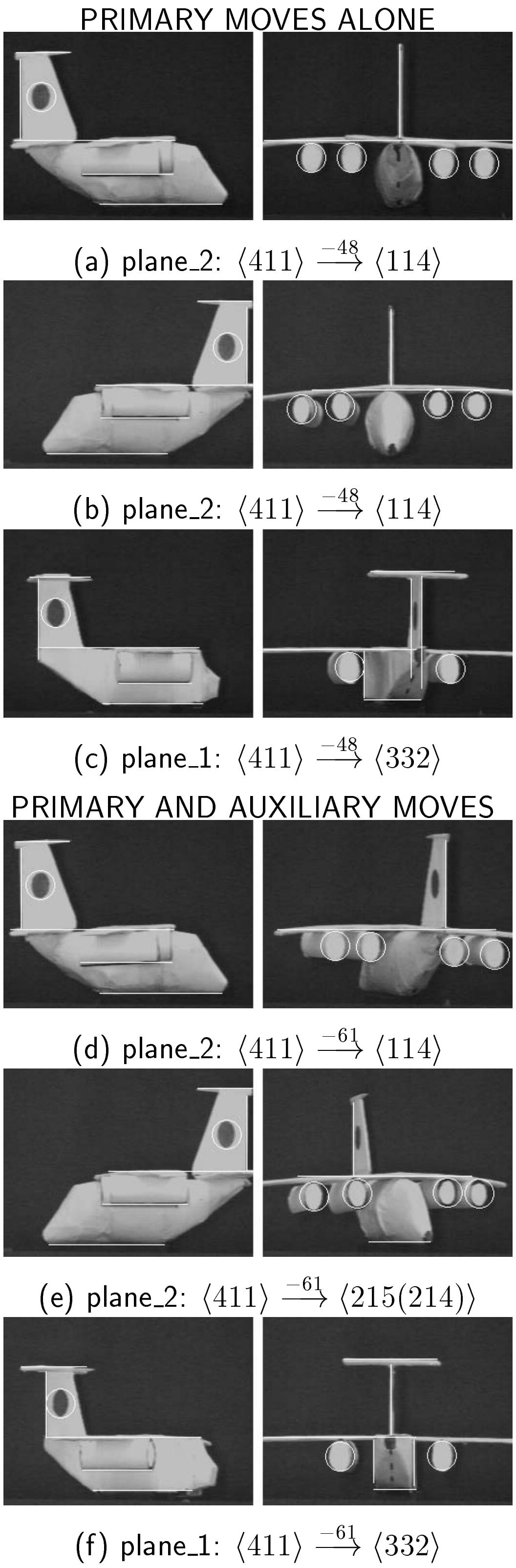
(b) plane_1: $\langle 342(332)\rangle \xrightarrow{26} \langle 410\rangle$

(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: $\langle 332\rangle \xrightarrow{26} \langle 510\rangle \xrightarrow{12} \langle 510\rangle$

Fig. 13. Experiments with the initial class as ⟨332⟩. (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



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: $\langle 411\rangle \xrightarrow{-48} \langle 332\rangle$

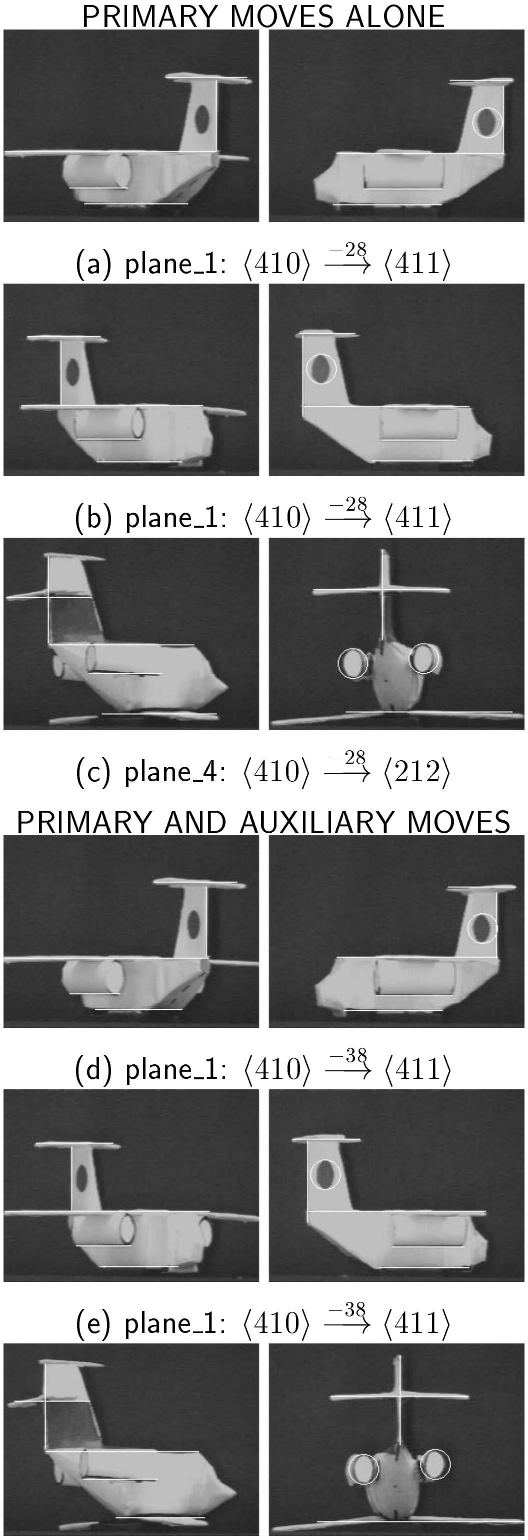
PRIMARY AND AUXILIARY MOVES

(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: $\langle 411\rangle \xrightarrow{-61} \langle 332\rangle$

Fig. 14. Experiments with the initial class as $\langle 411\rangle$. (The figure in parentheses shows an example of recovery from feature detection errors).



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. 15. Experiments with the initial class as $\langle 410\rangle$