Isolated 3D Object Recognition through Next View Planning

Sumantra Dutta Roy
Department of CSE,
I.I.T., Hauz Khas,
New Delhi-110016.
sumantra@cse.iitd.ernet.in

Santanu Chaudhury
Department of EE,
I.I.T., Hauz Khas,
New Delhi-110016.
santanuc@ee.iitd.ernet.in

Subhashis Banerjee
Department of CSE,
I.I.T., Hauz Khas,
New Delhi-110016.
suban@cse.iitd.ernet.in

Abstract

Most object recognition systems use information from a single image of an object. In many cases, a single view may not contain sufficient features to recognize the object unambiguously. Hence, more than one view is necessary. With an active sensor, the recognition process therefore involves identification of a view of an object and if necessary, planning the next view. 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. We present a knowledge representation scheme which encodes both feature-based information about objects in the model base as well as the uncertainty in the recognition process. This scheme is used both in the probability calculations as well as in planning the next view. The recognition scheme is on-line wherein past observations guide the planning process. Results clearly demonstrate its effectiveness for a reasonably complex experimental set.

1 Introduction

Most model-based object recognition systems consider the problem of recognizing objects from the image of a single view [1, 2]. However, a single view may not contain sufficient features to recognize the object unambiguously. In single-view object recognition, systems often need to 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 simple feature set and suitably planning multiple observations [4]. Besides, a simple feature set is more applicable for a larger class of objects.

In this paper, we present a new reactive and on-line recognition scheme based on next view planning for the identification of an isolated 3D object. Our algorithm plans the sequence of views that can provide reliable recognition incurring minimal image processing cost. We propose a probabilistic reasoning framework for recognition and planning. We also present a knowledge representation scheme which encodes both feature-based information about objects in the model base as well as the uncertainty in the recognition process. This scheme is used both in the probability calculations as well as in planning the next view. Due to the hierarchical nature of our scheme, we do not face a problem as in [5] namely, of many redundant hypotheses being generated and having to remove them later through consistency checks. Further, our system does not incur the computational overhead of [3] in tracking the region of interest over successive frames. We have experimented with object sets wherein a view could have come from a large number of poses of a number of objects. We present results demonstrating the effectiveness of our scheme.

2 The Knowledge Representation Scheme

A 3D object has different views, independent of the viewpoint over a particular range of viewing angles. A view is characterized by a set of features. Aspects have been defined as topologically equivalent classes of object appearances [6]. In this context, we define the following terms:

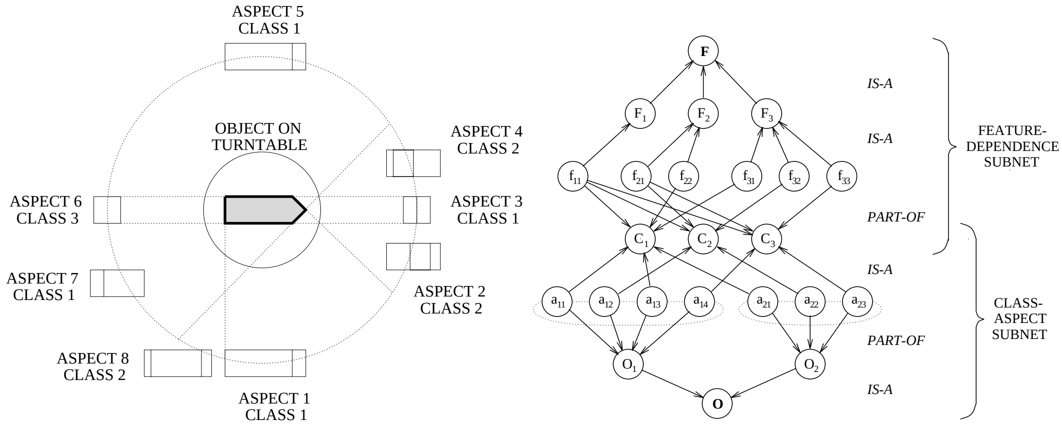
Class A Class (or, Aspect-Class) is a set of aspects, equivalent with respect to a feature set.

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

Figure 1 (a) 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. In this example, the basis of the different classes is the number of horizontal and vertical lines in a particular view of the 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 1(b) illustrates an example of this scheme. We use this knowledge representation scheme both in belief updating as well as in next view planning. The representation scheme logically consists of two parts:

1. **The Feature-Dependence Subnet**
In the feature-dependence subnet



(a) (b)

Figure 1: (a) Aspects and classes of an Object (b) The knowledge representation scheme: an example

- **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.

- **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, the 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.

3 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. Each hypothesis is associated with a probability. Hypothesis generation consists of two steps:

1. Class Identification
2. Object Identification

We present an algorithm to find the class and object probabilities given a feature class as evidence. The algorithm selects feature detectors according to suitability and need. All probability calculations can be performed in low-order polynomial (maximum quadratic) time.

3.1 Class Identification, Accounting for Uncertainity

3.1.1 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 outdegree. 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 1(b) 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.

3.1.2 Class Probability Calculations Using the Knowledge Representation Scheme

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

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

Here, aspects a_{pq} belong to class C_i. $P(a_{pq}|O_p)$ 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.

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:

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

$P(f_{jk}|C_i)$ is 1 for those classes which have a link from feature-class f_{jk}. It is 0 for the rest.

Class Recognition in the presence of feature detection errors

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 uncertainity is associated with the evidence. The value of $P'(C_i)$ is, then:

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

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 enables recovery from feature detection errors.

Figure 2 shows a flow diagram depicting the interaction of the hypothesis generation part with the rest of the system. 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. *a priori* class probabilities can be computed in time $O(N_C + N_A)$, and *a posteriori* values in time $O(N_{F_j} \cdot N_C)$. Figure 3 outlines our class recognition algorithm.

4 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 state of the recognition system consists of information such as the class observed, the aspects possible for the movement made thus far, and the range of positions possible within each aspect. The planning problem thus reduces to a search in this state space for the best move to distinguish between competing aspects at any state. We use a search tree for this purpose.

Our planning algorithm performs an efficient search in the state space. We assign weights to different state transitions possible within the current assumed aspect and the adjoining aspects during search tree traversal. The algorithm selects the sequence of state transitions with the highest discriminating power and lowest movement cost as the move from the current state, corresponding to the most probable aspect. As a benchmark, we prove that the average number of observations required to uniquely identify the given object is $O(log_e n)$, where n is the number of aspects the initially observed class could correspond to, for a simple deterministic case.

4.1 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:

$$P(a_{j_pk_p}) = P(O_{j_p}) \cdot P(a_{j_pk_p}|O_{j_p}) \quad (4)$$

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, $P(a_{j_pk_p}|Oj_p)$ is $\theta_{j_pk_p}/360$.

For any subsequent observation, we have *to account for the movement in the probability calculations.* For example, a particular movement may preclude the occurence 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:

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

where ϕj_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 ($0 \le m \le r$) 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:

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

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

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

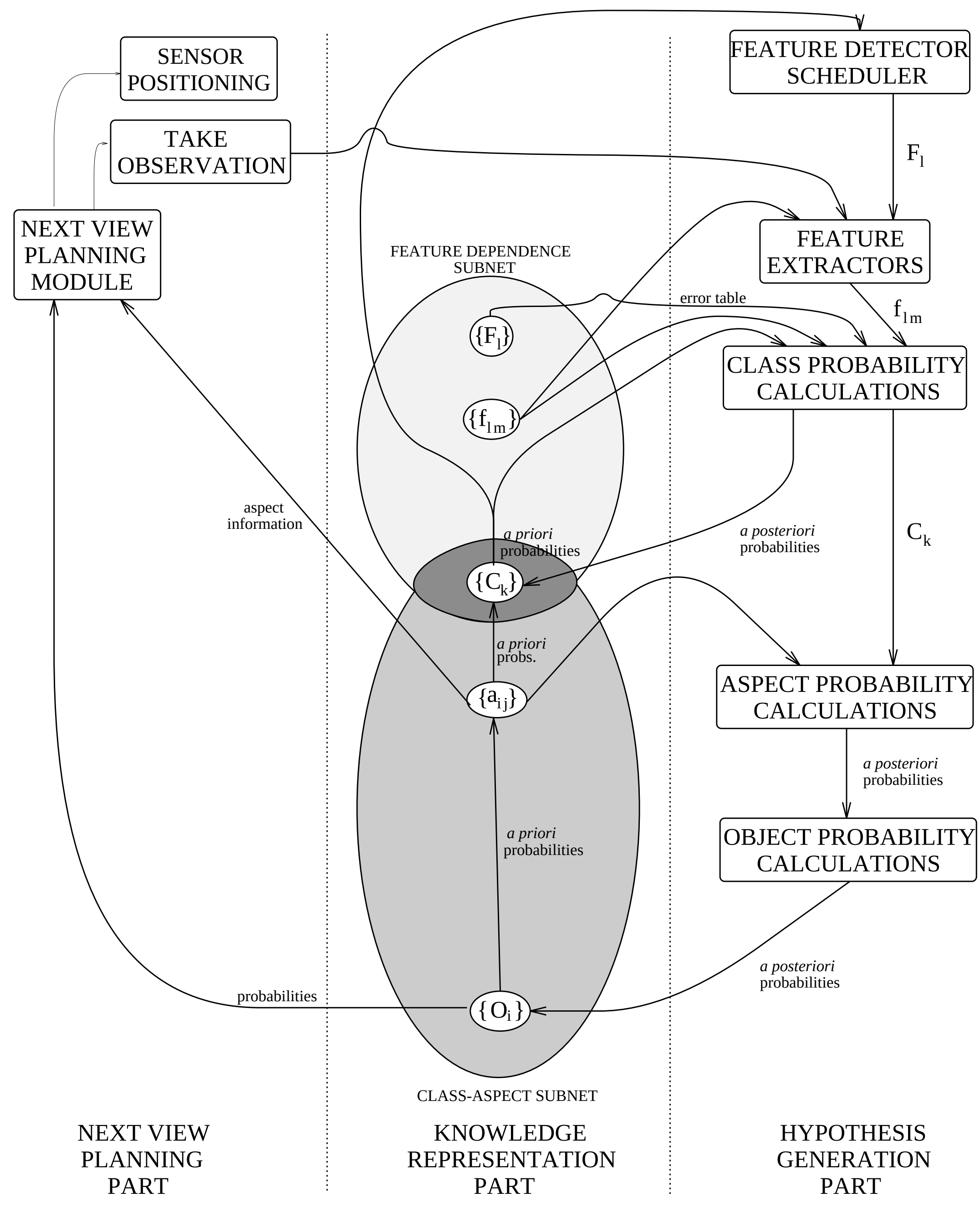


Figure 2: Flow diagram depicting the flow of information and control in our system

ALGORITHM identify_class

```
1. compute_a_priori_class_probabilities();          (* Eq. 1; Section 3.1.2 Part 1 *)
2. fd := identify_feature_detector_to_use();         (* Section 3.1.1 *)
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 3.1.2 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
```

Figure 3: The Class Recognition Algorithm

where aspects $a_{j_p k_l}$ belong to class C_i.

If the probability of some object is above a predetermined threshold, 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.

Figure 4 shows our object recognition algorithm. The time taken for all aspect and object probability calculations is linear in the number of aspects. 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.

5 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 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.

1. **Model Base I: 8 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 $\langle r \rangle$. (A class is represented as $\langle hvr \rangle$.) Figure 5 shows the objects in this model base.

2. **Model Base II: 7 Aircraft Models**
   We use the number of horizontal and vertical lines $\langle hv \rangle$, and the number of circles $\langle c \rangle$ as features. (A class is represented as $\langle hvc \rangle$.) Figure 6 shows the objects in this model base.

We use hough transform-based line and circle detectors. For getting the number of regions in a view, we use sequential labeling on a thresholded gradient image.

Experiments with Model Base I: Polyhedral Objects

Figure 7 shows some results of experimentation with the objects in the first model base. Figure 7(a) and (b) show the moves for two objects O_3 and O_4 with the same class initially observed, namely $\langle 232 \rangle$. The aspect list associated with the initial observation has 18 aspects from the 8 possible objects. For Figures 7(c) and (d), the initial class is $\langle 221 \rangle$, which could have come from 17 aspects. The moves are shown for objects O_7 and O_5, respectively.

To give an idea of the number of moves required by our system, we present some results of 46 observations on model base I. An aspect list of size 18 on the first view required an average of 3.4 moves. The corresponding numbers for aspect lists of sizes 17, 5 and 3 are 3.21, 2.33 and 3.00, respectively.

Experiments with Model Base II: Aircraft Models

Figure 8 shows some results of experimentation with the objects in the second model base. Figures 8(a), (b), (c), (d) and (e) show the moves for objects biplane, two aspects of plane-1, heli-1 and heli-2, respectively. The aspect list corresponding to the initial observation has 5 aspects.

The average number of moves for a total of 58 observations for aspect lists of sizes 10, 9, 7, 5 and 4 are 2.67, 2.00, 2.00, 2.05 and 2.00, respectively.

6 Conclusions

We present a scheme for the recognition of an isolated 3D object through on-line next view planning using probabilistic reasoning. The recognition scheme has the ability to correctly identify objects even when they

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 3.1 *)
    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 threshold THEN exit AND declare success;
7.  expand_search_tree_node(search_tree_root,0,class);    (* Section 4 *)
    best_leaf := get_best_leaf_node(search_tree_root);    (* Section 4 *)
                    (* ------ 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 threshold THEN exit AND 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
```

Figure 4: The Object Recognition Algorithm

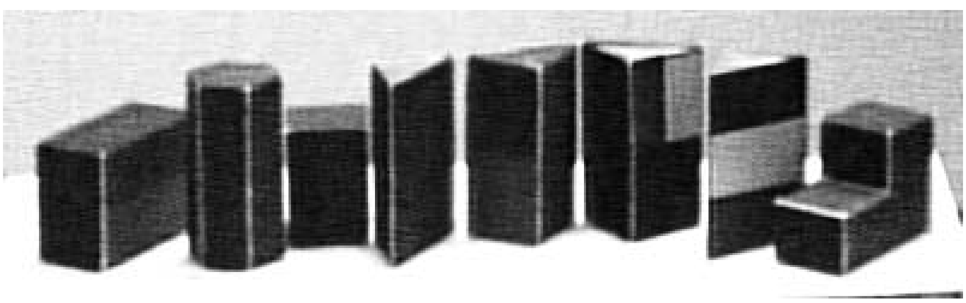


Figure 5: Model Base I: The objects (from left) are O_1, O_2, O_3, O_4, O_5, O_6, O_7 and O_8, respectively.

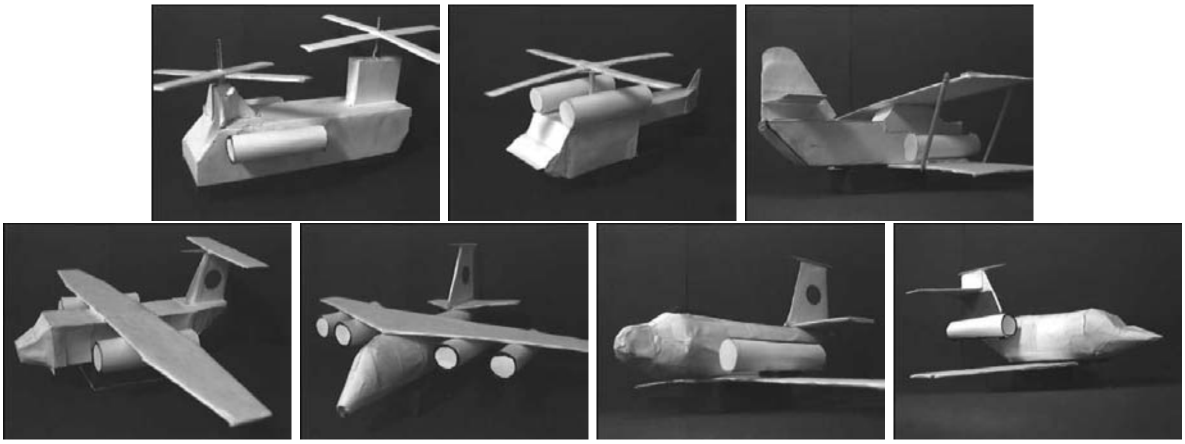
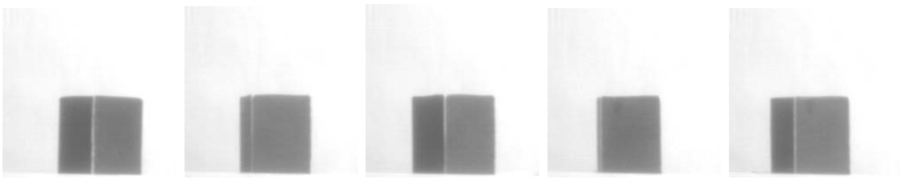
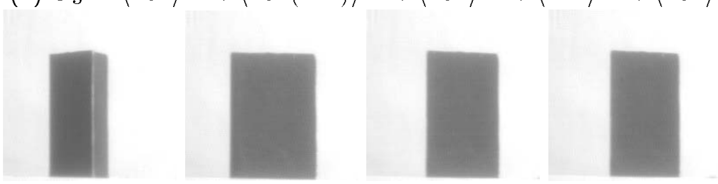


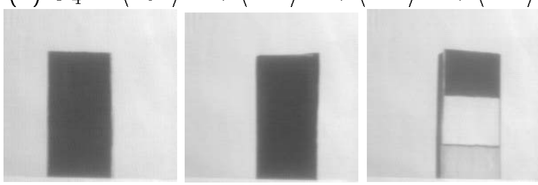
Figure 6: Model Base II: The objects (in row major order) are heli-1, heli-2, biplane, plane-1, plane-2, plane-3 and plane-4.



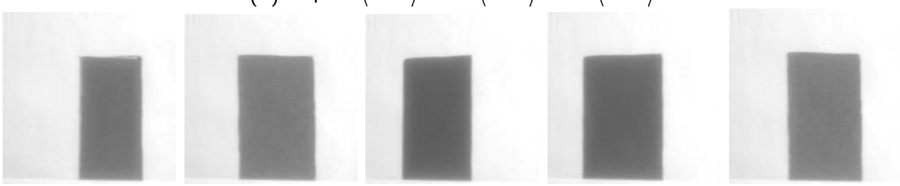
(a) O_3: $\langle 232 \rangle \xrightarrow{-40} \langle 231(221) \rangle \xrightarrow{-12} \langle 232 \rangle \xrightarrow{-34} \langle 221 \rangle \xrightarrow{-11} \langle 232 \rangle$



(b) O_4: $\langle 232 \rangle \xrightarrow{-40} \langle 221 \rangle \xrightarrow{-12} \langle 221 \rangle \xrightarrow{-2} \langle 221 \rangle$

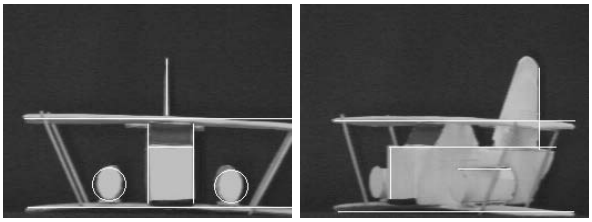


(c) O_7: $\langle 221 \rangle \xrightarrow{61} \langle 221 \rangle \xrightarrow{35} \langle 423 \rangle$

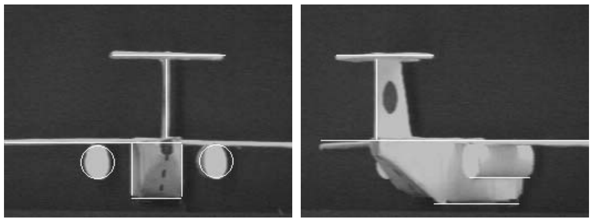


(d) O_5: $\langle 221 \rangle \xrightarrow{61} \langle 221 \rangle \xrightarrow{35} \langle 221 \rangle \xrightarrow{-8} \langle 221 \rangle \xrightarrow{-30} \langle 221 \rangle$

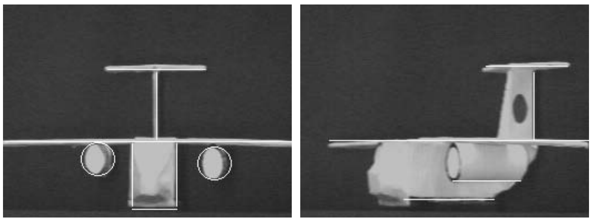
Figure 7: Some experiments with Model Base I: The initial classes are $\langle 232 \rangle$ and $\langle 222 \rangle$, respectively for each pair of rows. (The figure in parentheses shows an example of recovery from feature detection errors)



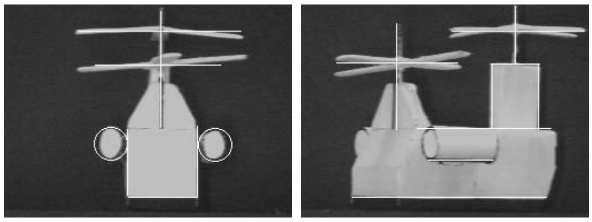
(a) biplane: $\langle 332 \rangle \xrightarrow{26} \langle 420 \rangle$



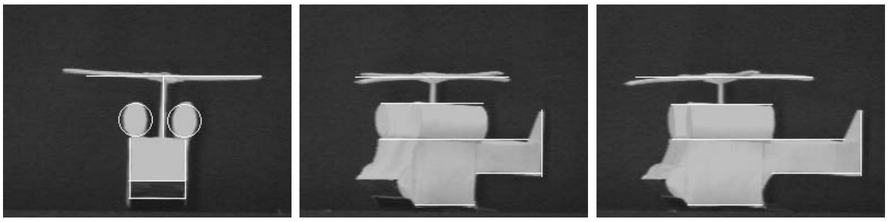
(b) plane-1(one pose): $\langle 342(332) \rangle \xrightarrow{26} \langle 410 \rangle$



(c) plane-1(another pose): $\langle 332 \rangle \xrightarrow{26} \langle 410 \rangle$



(d) heli-2: $\langle 332 \rangle \xrightarrow{26} \langle 540 \rangle$



(e) heli-1: $\langle 332 \rangle \xrightarrow{26} \langle 510 \rangle \xrightarrow{12} \langle 510 \rangle$

Figure 8: Some experiments with Model Base II: The initial class is $\langle 332 \rangle$. (The figure in parentheses shows an example of recovery from feature detection errors) For clarity, the lines and circles detected are shown superimposed on the original images.

have a large number of similar views. While we use simple features for the purpose of illustration, one can use other features such as texture, colour, specularities and reflectance ratios. Our knowledge representation scheme facilitates planning by exploiting the relationships between features, aspects and object models. If a feature set is not rich enough to identify an object from a single view, this strategy can be used to identify it from multiple views, considering simple features.

An extension of this work would take movement errors into account. Major areas for further work include multiple object recognition and searching for an object in a cluttered environment.

References

[1] P. J. Besl and R. C. Jain. Three-Dimensional Object Recognition. *ACM Computing Surveys*, 17(1):76–145, March 1985.

[2] R. T. Chin and C. R. Dyer. Model Based Recognition in Robot Vision. *ACM Computing Surveys*, 18(1):67–108, March 1986.

[3] S. J. Dickinson, H. I. Christensen, J. Tsotsos, and G. Olofsson. Active Object Recognition Integrating Attention and View Point Control. In *Proc. European Conference on Computer Vision*, pages 3–14. Springer-Verlag, 1994.

[4] K. D. Gremban and K. Ikeuchi. Planning Multiple Observations for Object Recognition. *International Journal of Computer Vision*, 12(2/3):137–172, April 1994. Special Issue on Active Vision II.

[5] 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*, 5(6):765–783, December 1989.

[6] J. J. Koenderink and A. J. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.