Part-based Isolated 3-D Object Recognition through Next View Planning using Inner Camera Invariants

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

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

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

Abstract

A single view of an object may not contain sufficient features to recognize it unambiguously. Further, the entire object may not fit in the camera's field of view. This paper presents a new on-line recognition scheme for the recognition and pose estimation of an isolated 3-D object. Our approach is independent of the internal parameters of the camera. The scheme uses a probabilistic reasoning framework for recognition and planning. Our knowledge representation scheme encodes part-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.

Keywords 3-D Object Recognition, Next View Planning, Pose Estimation, Inner Camera Invariants

1 Introduction

In this paper, we present a new on-line scheme for the recognition and pose estimation of an isolated 3-D object using reactive next view planning. We consider an uncalibrated projective camera, and consider the case when the internal parameters of the camera may vary.

Most model-based 3-D object recognition systems use information from a single view of an object. However, a single view of a 3-D object may not contain sufficient features to recognize it unambiguously. Further, 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. A further complication arises when in an image, we do not have a complete view of an object. Figure 1(a) shows such an example. Such a view could have come from any of the three models, different views of which are shown in Figure 1(b), (c) and (d), respectively. Further, even if the identity of the object were known, the same configuration of parts could occur at more than one place in the object – it is not possible to know the exact pose of the camera with respect to the object from one view alone. In these situations, multiple observation-based recognition strategies are needed.

In the context of planning the next view, one needs a sensor which can be positioned using vision-guided feedback. Such a sensor is called an active sensor, and recognition systems using such sensors are referred to as active recognition systems. Active recognition systems have been proposed which can work with different assumptions about the nature of the sensors and the environment, the degrees of freedom between the object and the sensor, and the object models themselves. While our earlier work on isolated 3-D object recognition through next view planning [1], [2] does not have the limitations associated with other systems such as [3], [4], [5], [6], like the others, it suffers from two important limitations. First, all these approaches assume that the object completely fits into the camera's field of view. The second is handling the case when internal parameters of the camera are allowed to vary, either unintentionally or on purpose.

2 Part-based Object Recognition

Part-based object recognition systems such as [7], [6], [8], [9] assume that the object to be identified is wholly composed of identifiable parts. While the first two use volumetric primitives (which are associated with a high feature extraction cost), the other two assume the view of the object to be partitioned into 'appearance-based parts' - Appearance-based methods have the constraining requirement of segmenting out the object from the background. While [7] does not consider multiple views, systems [6], [8], [9] additionally have the overhead of tracking the region of interest through successive views.

In this paper, we specifically consider situations where a complete view of a 3-D object is not available. We consider a very general definition of the word **'part'**. What may be observed are 2-D or 3-D *parts* of

[*] Author for Correspondence

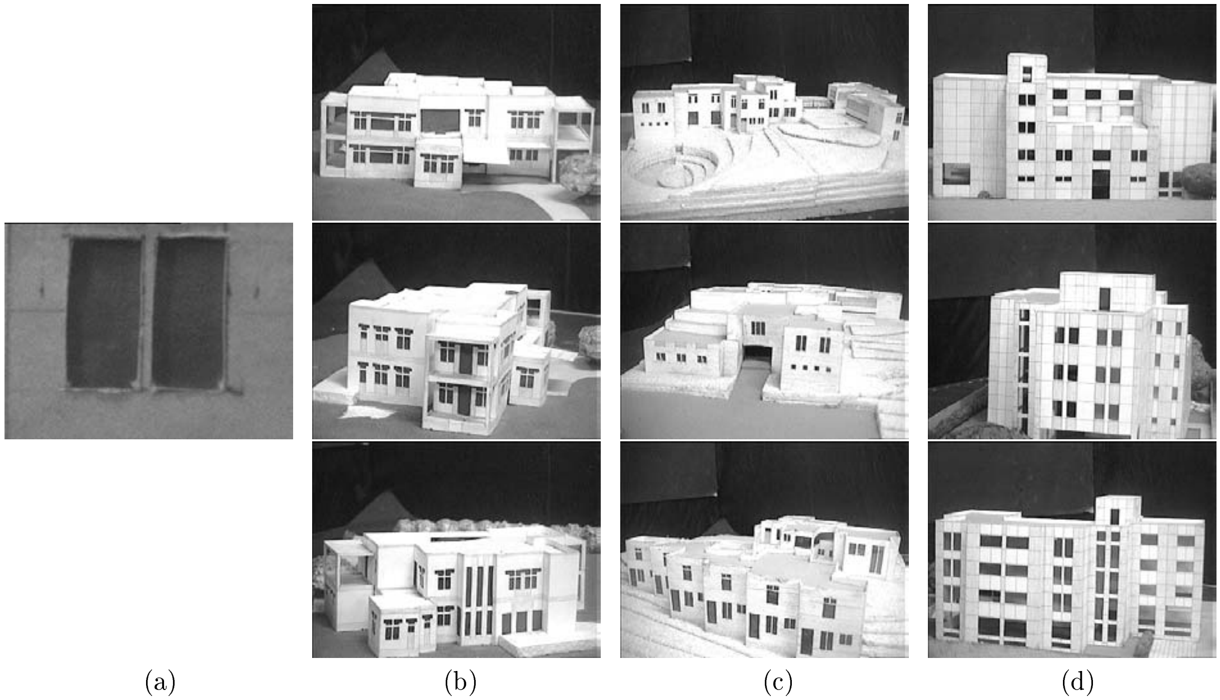


Figure 1: (a) The given view of an object: only a portion of it is visible. This could have come from any of the models, different views of which are shown in (b), (c) and (d), respectively

objects (which are detectable using 2-D or 3-D invariants, for example), and other 'blank' or 'featureless' regions which the given set of feature detectors cannot identify. Thus, an object is composed of parts, but is not partitioned into a collection of parts.

This paper presents a new on-line recognition scheme for 3-D objects when the complete 3-D object does not lie within the camera's field of view. The scheme uses a probabilistic reasoning framework for both recognition and planning. We propose a hierarchical knowledge representation scheme which encodes both domain knowledge, as well as the uncertainty in the recognition process. This is used both for probability calculations, as well as in planning the next view. An important feature of our scheme is the use of an *uncalibrated projective camera* to estimate the pose of various parts visible in a given view of the object.

3 Pose Estimation using Inner Camera Invariants

A multi-view 3-D object recognition system needs pose information for a given view, to generate different hypotheses corresponding to the information extracted from the view. The next view planning module uses this information to propose a move from the current position to disambiguate between the competing hypotheses.

We use the basic perspective projection model of a pin-hole camera [10] to derive new image-computable constraints which are invariant to the internal parameters of the camera. We refer to these constraints as **Inner Camera Invariants**. We show that these new constraints can be used for pose estimation – without going through the often cumbersome step of camera calibration. We have described Inner Camera Invariants in detail in an earlier work [11].

The following equation describes the imaging process [10]:

$$\lambda \mathbf{m} = \mathbf{P}\mathbf{M} = \mathbf{A}\,[\mathbf{R} \mid \mathbf{t}]\,\mathbf{M} \tag{1}$$

Here, $\mathbf{M} = (X, Y, Z, W)^T$ is a 3-D world point, and $\mathbf{m} = (x, y, 1)^T$ is the corresponding image point. $\mathbf{R}$ (3×3) and $\mathbf{t}$ (3×1) are the rotation and translation aligning the world coordinate system with the camera coordinate system (the external camera parameters), and $\mathbf{A}$ is the matrix of the internal parameters of the camera. $\mathbf{A}$ may be written as [10]:

$$\mathbf{A} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{2}$$

where f_x and f_y are the effective focal lengths in the x and y directions and (u_0, v_0) is the principal point. Suppose we know three 3-D points, $\mathbf{M}_p = (X_p, Y_p, Z_p, 1)^T$, $p \in \{i, j, k\}$, and their images on the image plane, $\mathbf{m}_p = (u_p, v_p, 1)^T$, $p \in \{i, j, k\}$. By eliminating the internals of the camera, we obtain

$$\left\{ \begin{array}{l} J_{ijk} = \frac{u_i - u_j}{u_i - u_k} = \frac{\frac{\mathbf{r_1 M}_i}{\mathbf{r_3 M}_i} - \frac{\mathbf{r_1 M}_j}{\mathbf{r_3 M}_j}}{\frac{\mathbf{r_1 M}_i}{\mathbf{r_3 M}_i} - \frac{\mathbf{r_1 M}_k}{\mathbf{r_3 M}_k}} \\ K_{ijk} = \frac{v_i - v_j}{v_i - v_k} = \frac{\frac{\mathbf{r_2 M}_i}{\mathbf{r_3 M}_i} - \frac{\mathbf{r_2 M}_j}{\mathbf{r_3 M}_j}}{\frac{\mathbf{r_2 M}_i}{\mathbf{r_3 M}_i} - \frac{\mathbf{r_2 M}_k}{\mathbf{r_3 M}_k}} \end{array} \right. , \quad (3)$$

where J_{ijk} and K_{ijk} are *image measurements* that are functions of $\mathbf{R}$ $(= [\mathbf{r_1}\ \mathbf{r_2}\ \mathbf{r_3}]^T)$, $\mathbf{t}$ and $\mathbf{M}_p$ $(p \in \{i, j, k\})$, and are independent of camera internals. We may write the above equations as:

$$\left\{ \begin{array}{l} J_{ijk} = f_{ijk}(\mathbf{R}, \mathbf{t}, \mathbf{M}_i, \mathbf{M}_j, \mathbf{M}_k) \\ K_{ijk} = g_{ijk}(\mathbf{R}, \mathbf{t}, \mathbf{M}_i, \mathbf{M}_j, \mathbf{M}_k) \end{array} \right. \quad (4)$$

J_{ijk} and K_{ijk} are **Inner Camera Invariants** – image measurements that are independent of the internals of the camera.

Suppose we know the Euclidean coordinates $(X_i, Y_i, Z_i, 1)^T$ of 5 points in the world coordinate system. *Six* independent (inner camera) invariant measurements give us six equations. For a 4-DOF case (*e.g.*, a setup with one rotational degree of freedom and all three translational degrees of freedom), *Four* independent (inner camera) invariant measurements result in four equations. These equations can be solved numerically, for pose estimation using an uncalibrated camera and known landmarks. The solutions in general, require non-linear optimization. In [11], we also show two special cases where it is possible to obtain closed-form linear solutions for pose estimation. Since these impose a special structure on the landmarks used for pose estimation, we consider the general pose estimation case in this paper.

4 The Knowledge Representation Scheme

We propose a hierarchical knowledge representation scheme that encodes domain knowledge about the objects in the model base. Each object O_i is composed of N_i parts. We represent the jth part of object O_i as $\rho_{i,j}$, $1 \leq j \leq N_i$. In this context, we define the following term:

Part-Class A Part-Class is a set of parts, equivalent with respect to a feature set. In other words, the set of parts is partitioned into different equivalence classes with respect to a given feature set. These equivalence classes are part-classes.

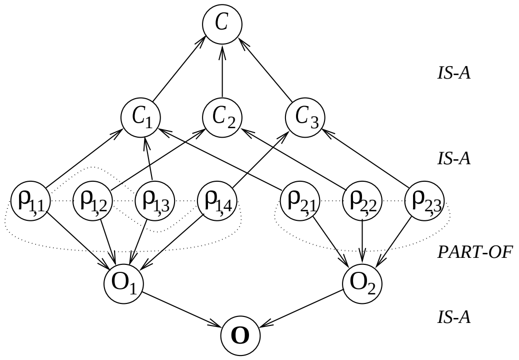


Figure 2: The knowledge representation scheme: an example

Figure 2 illustrates an example of our knowledge representation scheme.

- **O** represents the set of all objects $\{O_i\}$. An object node O_i stores its probability, $P(O_i)$
- A part $\rho_{i,j}$ has a *PART-OF* relationship with its parent object O_i. A part node stores the 3-D Euclidean structure of its n constituent vertices $[X_i, Y_i, Z_i]^T$, $1 \leq i \leq n$. (Section 3: $n \geq 5$ for the 6-DOF case, and $n \geq 4$ for the 4-DOF case.)
- A part node has links with its neighbouring parts. Each link represents the $\mathbf{R}$ and $\mathbf{t}$ values *i.e.*, the rotations and translations needed to go from one part to its neighbours. Figure 2 shows an example where the parts nodes form a complete graph.
- $\mathcal{C}$ represents the set of all part-classes $\{\mathcal{C}_1, \mathcal{C}_2, \ \ldots\ \mathcal{C}_k\}$ for all parts belonging to the objects in the model base.
- A part node $\rho_{i,j}$ has exactly one link with its corresponding part-class node $\mathcal{C}_k$, and the node for the object O_i, to which it belongs.

5 Object Recognition and Pose Identification through Next View Planning

We are given an arbitrary view of an object in our model base. Let this view contain m parts. Our aim is to identify the given object, and the viewer pose with respect to it. Our recognition scheme is divided into three parts:

1. Hypothesis generation
2. Probability calculations, and
3. Next view planning

In what follows, we discuss these three topics in detail. Figure 3 describes the main steps in our algorithm.

```
ALGORITHM identify_object_and_pose

(* ------ FIRST PHASE ------ *)
1.  initialize_object_probabilities();
         (* Initialize to 1/N *)
2.  image:=get_image_of_object();
3.  part_class_info:=identify_part_classes(image);
    IF NO part_class observed THEN
             make random movement; GOTO step 2;
4.  search_tree_root:=
    construct_search_tree_node(
             part_class_info,$[\mathbf{I}|\mathbf{0}]$);
5.  compute_hypothesis_probabilities(
             search_tree_root);  (* Eq. 6 *)
6.  IF the probability of some hypothesis
    is $\geq$ a pre-determined thresh THEN
             exit & call success;
7.  expand_search_tree_node(search_tree_root,
         MAX_LEVELS);  (* Section 5.3 *)
(* ------ SECOND PHASE ------ *)
      previous:=search_tree_root;
      expected:=get_best_leaf_node(
             search_tree_root);
8.  $\{[\mathbf{R}|\mathbf{t}]\}$:=compute_movements(expected,previous);
    make_movements($\{[\mathbf{R}|\mathbf{t}]\}$);
    image:=get_image_of_object();
9.  part_class_info:=identify_part_classes(image);
    IF NO part_class observed THEN
       (* — backtrack — *)
       undo_movements($\{[\mathbf{R}|\mathbf{t}]\}$);
       expected:=get_next_best_leaf_node(
             previous);
       GOTO step 8;
10. IF obs view does NOT correspond to expected
    THEN
      new_node:=construct_search_tree_node(
                 part_class_info,$\{[\mathbf{R}|\mathbf{t}]\}$);
    ELSE
      modify_search_tree_node_with_observation(
                  expected,part_class_info);
      new_node:=expected;
11. compute_hypothesis_probabilities(new_node);
12. IF the probability of some hypothesis
    is $\geq$ a pre-determined thresh THEN
             exit & call success;
13. expand_search_tree_node(new_node,MAX_LEVELS);
      expected:=get_best_leaf_node(previous);
      previous:=new_node;
14. GOTO step 8
```

Figure 3: The Object Recognition and Pose Identification Algorithm

5.1 Hypothesis Generation

The input to the system is a view of the given object. Let this given view contain m parts – $\rho_{i,j_1},\ \rho_{i,j_2},\ \ldots\ \rho_{i,j_m}$. From the image information, we can only identify the *part-classes* $\mathcal{C}_{k_1},\ \mathcal{C}_{k_2},\ \ldots\ \mathcal{C}_{k_m}$ (where $\mathcal{C}_{k_p}$ and $\mathcal{C}_{k_q}$ are not necessarily different) corresponding to each observed part, respectively ($PART-CLASS(\rho_{i,j_p}) \ = \ \mathcal{C}_{k_p}$). The part-classes may be identified by using 2-D or 3-D projective invariants, for example. *However, our scheme is independent of the particular technique to identify a part-class.* This configuration of visible parts could belong to any of the n objects in the model base. Further, this configuration could have come from many different positions within the same object O_i.

We wish to generate different hypotheses corresponding to the identity of the observed configuration of parts in the image. For the first part, we construct hypotheses corresponding to every part node $\rho_{i,j}$, which has an outward link to part-class node $\mathcal{C}_{k_1}$ *i.e.*, every part which belongs to part-class $\mathcal{C}_{k_1}$. For every such part, we associate the given image coordinates of the part to the 3-D Euclidean structure of the hypothesized part (having the same part-class), and compute its pose. Since we use general pose estimation (Section 2), we use non-linear optimization routines with rough bounds ($\pm\ 5^o$ and $\pm\ 20mm$) within which to look for suitable solutions. Any such part whose pose does not lie within these bounds is considered invalid, and is pruned from the list of hypotheses. We repeat this procedure for each observed part in the image – looking for parts in the model base which could give consistent hypotheses for the part being considered, with respect to the existing hypothesized configurations. *At each stage, we use the pose information to prune out invalid hypotheses.* At the end of this phase, we are left with a list of hypotheses of part configurations, which could have given rise to the observed configuration of parts in the given view.

5.2 Probability Calculations

The given view consists of m parts $\rho_{i,j_1},\ \rho_{i,j_2},\ \ldots\ \rho_{i,j_m}$. The hypothesis generation stage computes a list of valid hypotheses of part configurations, which could have given rise to the observed view. First, we compute *a priori* probabilities for each such hypothesis. For N objects in the model base, the *a priori* probability of each object before taking the first observation, is $1/N$. We need estimates of the *a priori* probabilities of different configurations of parts that may occur.

$$P(\rho_{i,j_1},\ \rho_{i,j_2},\ \ldots\ \rho_{i,j_m}) \ =$$

$$P(O_i) \cdot P(\rho_{i,j_1},\ \rho_{i,j_2},\ \ldots\ \rho_{i,j_m} \mid O_i) \qquad (5)$$

We may form estimates of $P(\rho_{i,j_1},\ \rho_{i,j_2},\ \ldots\ \rho_{i,j_m} \mid O_i)$ by taking a very large number of views of the given object from different positions, and different values of the internals of the camera (the focal length, for example on which the field of view of the camera depends) — this is done *off-line*, before taking the first observation.

For an observation, we compute the *a posteriori* probability of each hypothesized configuration using the Bayes rule:

$$P(\rho_{i,j_1},\ \rho_{i,j_2},\ \ldots\ \rho_{i,j_m} \mid \mathcal{C}_{k_1},\ \mathcal{C}_{k_2},\ \ldots\ \mathcal{C}_{k_m}) \\ = Numerator/Denominator \qquad (6)$$

where $Numerator$ is given by

$$P(\rho_{i,j_1},\ \rho_{i,j_2},\ \ldots\ \rho_{i,j_m}) \cdot \\ P(\mathcal{C}_{k_1},\ \mathcal{C}_{k_2},\ \ldots\ \mathcal{C}_{k_m} | \ \rho_{i,j_1},\ \rho_{i,j_2},\ \ldots\ \rho_{i,j_m})$$

and $Denominator$, by

$$\sum [\ P(\rho_{l,j_1},\ \rho_{l,j_2},\ \ldots\ \rho_{l,j_m}) \cdot \\ P(\mathcal{C}_{k_1},\ \mathcal{C}_{k_2},\ \ldots\ \mathcal{C}_{k_m} | \ \rho_{l,j_1},\ \rho_{l,j_2},\ \ldots\ \rho_{l,j_m})\ ]$$

The summation in $Denominator$ is for all objects O_l, and all possible configurations of parts within the object. Because of the $IS - A$ relation between a part and a part-class in our knowledge representation scheme (Section 4), each of the terms $P(\mathcal{C}_{k_1},\ \mathcal{C}_{k_2},\ \ldots\ \mathcal{C}_{k_m} | \ \rho_{l,j_1},\ \rho_{l,j_2},\ \ldots\ \rho_{l,j_m})$ is 1 for all parts belonging to a particular part-class and 0, otherwise.

We now compute the *a posteriori* probability of each object in the model base:

$$P(O_l) = \sum P(\rho_{l,j_1},\ \rho_{l,j_2},\ \ldots\ \rho_{l,j_m} | \mathcal{C}_{k_1},\ \mathcal{C}_{k_2},\ \ldots\ \mathcal{C}_{k_m}) \qquad (7)$$

The summation is for all configurations of parts $\rho_{l,j_1}, \rho_{l,j_2}, \ldots\ \rho_{l,j_m}$ belonging to object O_l, which could have given rise to the given view containing part-classes $\mathcal{C}_{k_1}, \mathcal{C}_{k_2}, \ldots\ \mathcal{C}_{k_m}$. Each object node in the knowledge representation scheme updates its probability with values from Equation 7.

5.3 Next View Planning

If the probability of no hypothesis is above a predetermined threshold, we have to take the next view to try to disambiguate between the competing hypotheses. The state of the system may be described in terms of the competing view interpretation hypotheses, and the set of **R** and **t** movements made thus far. The planning process aims to determine a move from the current step, which would uniquely correspond to exactly one part-configuration for one object. Our search process uses a search tree for this purpose. The aim is to get to a leaf node – one corresponding to a unique part-configuration. A search tree node is expanded for each part in a view interpretation hypothesis. The moves from a viewpoint are based on the pose information calculated using inner camera invariants (Sections 3 and 5.1). Here, we assume that the principal point of the camera is somewhere near the centre of the image. However, we do not assume that we know is value in any way, nor do we assume it to be fixed. The first move gets the expected part in the camera's line of view. The subsequent moves are from the current expected part to its neighbours, using the **R** and **t** relations between parts in the knowledge representation scheme (Section 4). Thus, the only significance of the above assumption is *to maximize the chance of the expected part to be present in the camera's field of view.* This offers some robustness to small movement errors. Further, a zoom-in/zoom out, or focusing operation may be performed. If the principal point is near the centre of the expected part, chances of having the expected part in the camera's field of view and hence detecting it, are higher than otherwise.

Due to the exponential space and time complexity corresponding to search tree expansion, one may expand the search tree only to a fixed depth (MAX_LEVELS in Figure 3). We now use three stages of filtering to get the best leaf node or pseudo-leaf node (a node which has no child nodes, but does not correspond to a unique part, but has not been expanded due to the fixed maximum search depth from a node). First, we consider those leaf/pseudo-leaf nodes which lie along a path from the most probable hypothesized view interpretation in the search tree node corresponding to the previous observation (the 'previously observed node'). For each node in the search tree, we assign the weight s^{level}, where s represents the number of hypothesized view interpretations corresponding to this node, and *level* is the search tree level (depth) the node lies on. The rationale behind this strategy is to favour nodes with low ambiguity among the different hypothesized view interpretations, and those corresponding to less movement cost. A leaf/pseudo-leaf node also stores the sum total of the weights of all nodes which lie on the path to it, from the previously observed search tree node. From among those leaf and pseudo-leaf nodes selected in the first stage, we select those with minimum total weight. The third stage of filtering concerns a setup limitation – our camera setup can achieve more accurate translational movement compared to a rotational one. From among the second stage selections, we choose a node having the least number of rotational movements.

The system makes the required movements $\{\langle R_x, R_y, R_z, t_x, t_y, t_z\rangle\}$, and takes an image at this position. We then find out the part-class information corresponding to this image. Similar to the process in Section 5.1, we generate different interpretation hypotheses corresponding to this view, for the particular sequence of $\mathbf{R}$ and $\mathbf{t}$ movements taken to reach this particular viewpoint. We now check if this observed configuration of parts corresponds to the best leaf/pseudo-leaf node. Since we do not make any assumption regarding the knowledge of the camera internal parameters or their constancy, we do not make any assumptions about the field of view of the camera. We simply check if the observed configuration of parts corresponds to the expected node. This is a simple way to make the system robust to slight movement errors, or intentional/unintentional changes in the focusing and field of view. Since we do not predict any view which might be observed, even if some parts in the vicinity of the expected part are not detected (due to feature detection errors), this does not affect the system in any way. *Another important consequence of this fact is the robustness of the system to the presence of clutter in a view.* If the current observation does correspond to the expected search tree node, we update the search tree node with the information from the current view (Step 12 in Figure 3). We compute the probabilities of each view interpretation hypothesis. If the probability of some hypothesis is above a the predetermined threshold, we declare success, and exit. The pose of the camera with respect to the object is the one corresponding to this hypothesis, and the parts corresponding to the view are the ones in this view interpretation.

If the current observation does not correspond to the expected search tree node, we search for the node corresponding to this observation among all leaf/pseudo-leaf nodes corresponding to the movements made from the previous viewpoint. If we find one, then we repeat the process described in the previous paragraph. If not, we undo the current movements, get the next best leaf node, and proceed (Figure 3, step 11).

If the probability of no hypothesis is above the threshold, this node needs to be expanded further. The system finds out the best leaf node again, and the entire process is repeated.

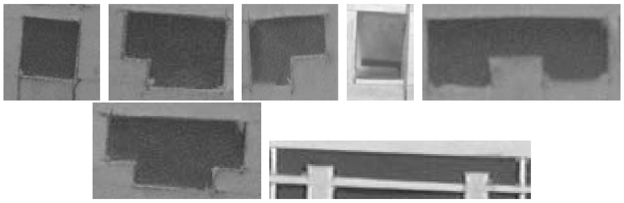


Figure 4: The 7 part-classes which the 459 parts belong to, for our model base: $DW4$, $DW6L$, $DW6R$, $OPEN$, $DW8HANDLE$, $DW8T$, and $DW12$, respectively in row-major order.

6 Experimental Results & Discussion

Our experimental setup has a camera system with 4 degrees of freedom - translations along the **X**-, **Y**- and **Z**- axes, and rotation about the **Y**- axis. We have experimented with a set of architectural models shown in Figure 1. We have chosen as (2-D) parts the doors and windows of different shapes and sizes in the models. We have chosen this set of models because of the large number of parts grouped into a few part-classes – this ensures a very high degree of interpretation ambiguity associated with a particular view of a few parts of the given object. Model LH (Figure 1(a)) has 167 parts, model DS (Figure 1(b)) has 170, while model GH (Figure 1(c)) has 122. Figure 4 shows the 7 different part-classes these 459 parts (of different sizes) correspond to. The 7 part-classes, with the number of parts corresponding to each, are $DW4(374)$, $DW6L(24)$, $DW6R(24)$, $OPEN(21)$, $DW8HANDLE(6)$, $DW8T(6)$, and $DW12(4)$, respectively. Given a particular view of the object, we first segment the image using sequential labeling. Then we detect corners as intersection of lines on the boundaries of 'dark' regions. We use 2-D projective invariants using the canonical frame construction method [12] for recognizing all part-classes (except the 4-cornered ones, for which we use grey-level information near their centroids). Figures 5 – 8 show some results of experimentation with the objects in our model base. The detected corners and parts are shown superimposed. Each of these experiments shows that *the planning to get to the centre of the expected part (Section 5.3) provides some immunity to small movement errors and changes in the camera's field of view.* For our experiments, we have adopted a stricter criterion for program termination than the probability of a particular hypothesis in an observed node being above a threshold. We stop when there is exactly one hypothesis possible for the observed node.

The initial view in Figure 5 shows two parts with part-classes $DW8T$ and $DW4$. For the first part

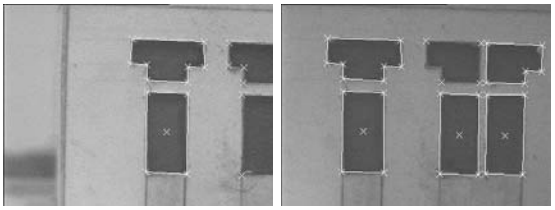


Figure 5: Experiment 1: The sequence of moves required to identify the object and its pose. The failure to detect a part does not affect the system (details in text).

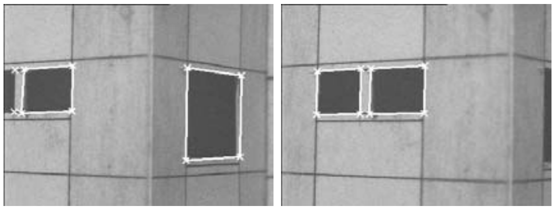
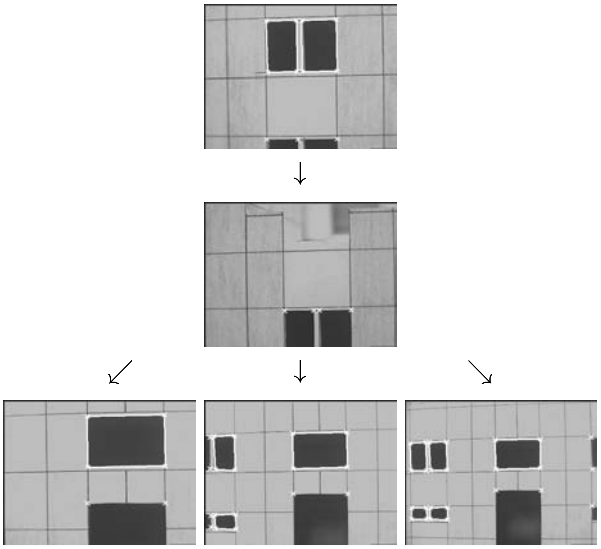


Figure 6: Experiment 2: The sequence of moves required to identify the object and its pose. The parts in the initial view do not lie in the same plane.



(a) ⟶ (b) ⟶ (c)

The camera progressively zooms out

Figure 7: Experiment 3: For the same first two views, we progressively zoom out the camera in three stages. (a), (b) and (c) depict the three views which the camera sees, for the third view. This does not affect the recognition system in any way (details in text).

alone, there are 6 hypotheses. Of the 6 hypotheses formed on including the information from the second part, 4 are pruned out since the estimated pose of the second part comes out to be invalid (Section 5.1). The system takes the required movements, and the view observed is the second image in Figure 5. This view contains the expected part, as well as a couple of neighbouring parts. Here, the system fails to detect the part corresponding to part-class $DW6$. The presence of the neighbouring parts (their part-classes, and their pose information) is consistent with that of the expected part (centre of the bottom row). *Thus, this feature detection error does not affect the performance of our algorithm in any way.*

Scenes containing a small number of parts belonging to part-class $DW4$, have a very high degree of ambiguity associated with their interpretation. This is due to the large number of parts belonging to part-class $DW4 - 374$. Due to this reason, we use a depth-restricted search tree expansion method. We present results of experiments where the first view contains a pair of $DW4$ parts (Experiment 2), and finally, only one $DW4$ part (Experiments 3 – 4).

Here, we present an example where the parts in the initial view do not come from the same plane. Figure 6 shows the moves taken by the system to identify the object, and the pose of the camera with respect to it. For the first part in the first image, 374 hypotheses are proposed, out of which the part pose estimation prunes out all but 115 hypotheses. The information from the second part results in a hypothesis list of size 87. The system plans a move to disambiguate between the different hypotheses. This corresponding move takes us to a view (the second image in Figure 6), whose view interpretation is unique.

For the next experiment (Figure 7), we changed the zoom parameter of the camera, thus changing the effective focal length of the camera system and consequently, its field of view. The first view could have come from 257 configurations of two adjacent parts with part-class $DW4$. We need three image processing operations (2 moves) to recognize the object and its pose, uniquely. In this case, we repeated the experiment for various values of the camera zoom-out at the third camera station. The expected part is the large 4-cornered window, GH_W_15. Since our strategy does not make any assumptions about the field of view of the camera, the recognition results are the same in each of the cases — (a), (b) and (c) in Figure 7. Further, the camera pose with respect to part GH_W_15 in these three cases are $\langle\ 9.425^o,\ -22.000mm,\ -9.999mm,\ 150.000mm\ \rangle$, $\langle\ 9.888^o,\ -22.000mm,\ -9.999mm,\ 150.000mm\ \rangle$, and $\langle\ 9.896^o,\ -22.000mm,\ -9.999mm,\ 150.000mm\ \rangle$, re-

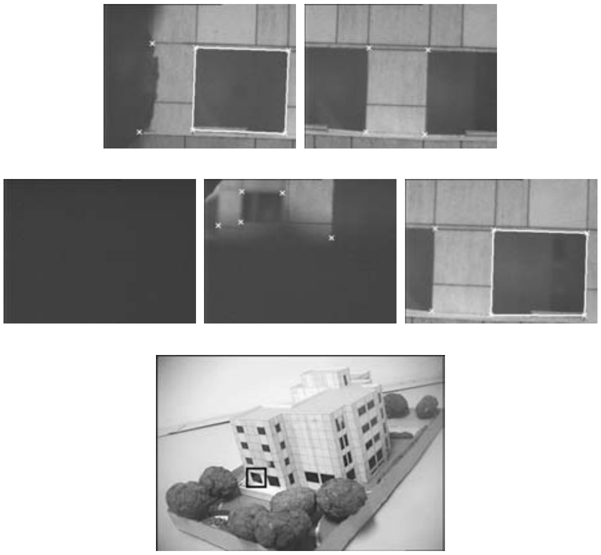


Figure 8: Experiment 4: The first, third and fourth views are cluttered by the presence of a tree. The image at the bottom shows an overall view. The corresponding window is highlighted with a black square.

spectively.

In Experiment 4, The first-level pruning results in 304 hypotheses, which reduces to 5 for the third view. The presence of a tree (an unmodeled object) accounts for clutter in the first, third and fourth view of Figure 8. *Here, our system is able to perform correct recognition even in the case of clutter.*

7 Conclusions

This paper presents a new scheme for the recognition of an isolated 3-D object through on-line next view planning, when only a portion of it is visible to the camera. The system uses an uncalibrated camera, and uses inner camera invariants for pose recognition. Our knowledge representation scheme is used both for probabilistic hypothesis generation, as well as in planning the next view. Experiments show ability of the system to correctly identify objects and their pose even when there is a high degree of interpretation ambiguity associated with the initial view.

References

[1] S. Dutta Roy, S. Chaudhury, and S. Banerjee, "Isolated 3-D Object Recognition through Next View Planning," *IEEE Trans. on Systems, Man and Cybernetics - Part A: Systems and Humans*, vol. 30, no. 1, pp. 67 – 76, January 2000.

[2] S. Dutta Roy, S. Chaudhury, and S. Banerjee, "Aspect Graph Based Modeling and Recognition with an Active Sensor: A Robust Approach," *Proc. Indian National Science Academy, Part A*, (Accepted for Publication).

[3] J. Maver and R. Bajcsy, "Occlusions as a Guide for Planning the Next View," *IEEE Trans. on Pattern Anal. and Machine Intell.*, vol. 15, no. 5, pp. 76 – 145, May 1993.

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

[5] K. D. Gremban and K. Ikeuchi, "Planning Multiple Observations for Object Recognition," *Int. Journal of Computer Vision*, vol. 12, no. 2/3, pp. 137 – 172, April 1994.

[6] 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.

[7] S. J. Dickinson, A. P. Pentland, and A. Rosenfield, "From Volumes to Views: An Approach to 3D Object Recognition," *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 55, no. 2, pp. 198 – 211, March 1992.

[8] C. Y. Huang, O. I. Camps, and T. Kanungo, "Object Recognition Using Appearance-Based Parts and Relations," in *Proc. IEEE Int. Conf. on CVPR*, 1997, pp. 877 – 883.

[9] O. I. Camps, C. Y. Huang, and T. Kanungo, "Hierarchical Organization of Appearance-Based Parts and Relations for Object Recognition," in *Proc. IEEE Int. Conf. on CVPR*, 1998, pp. 685 – 691.

[10] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, The MIT Press, 1996.

[11] M. Werman, S. Banerjee, S. Dutta Roy, and M. Qiu, "Robot Localization Using Uncalibrated Camera Invariants," in *Proc. IEEE Int. Conf. on CVPR*, 1999, pp. II: 353 – 359.

[12] C. A. Rothwell, *Recognition using Projective Invariance*, Ph.D. thesis, University of Oxford, 1993.