Recognizing Large 3-D Objects through Next View Planning using an Uncalibrated Camera

Sumantra Dutta Roy
Department of EE
IIT Bombay, Powai
Mumbai-400076, INDIA
sumantra@ee.iitb.ac.in

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

Subhashis Banerjee
Department of CSE
IIT Delhi, Hauz Khas
New Delhi-110016, INDIA
suban@ee.iitd.ernet.in

Abstract

We present a new on-line scheme for the recognition and pose estimation of a large *isolated 3-D object, which may not entirely fit in a camera's field of view. We do not assume any knowledge of the internal parameters of the camera, or their constancy. We use a probabilistic reasoning framework for recognition and next view planning. We show results of successful recognition and pose estimation even in cases of a high degree of interpretation ambiguity associated with the initial view.*

1. Introduction

In this paper, we present a new next view planning-based recognition and pose estimation scheme for an isolated large 3-D object. A large 3-D object may not fit into a camera's field of view. Figure 1(b) shows an image of a portion of a building obtained using an *active camera* (one whose parameters can be changed purposively *e.g.*, Figure 1(a)). Such a view could have come from any of the three models, different views of which are shown in Figure 1(c), (d) and (e), respectively. Further, even if the identity of the object were known, the same view 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.

We consider a view of an object to contain 2-D or 3-D **parts** (which are detectable using 2-D or 3-D projective invariants, for example), and other 'blank' or 'featureless' regions (which the given set of feature detectors cannot identify). We present a new reactive object recognition scheme for large 3-D objects. The scheme uses a hierarchical part-based knowledge representation scheme, and a probabilistic framework for both recognition and planning. The planning scheme is independent of the particular nature of a 2-D/3-D part, and the method used to detect it. A novel feature of our work is the use of *inner camera invariants* [12] for pose estimation – image-computable functions which are independent of the internal parameters of a camera.

Active recognition systems such as [9], [8], [6], [2], [3], [4] assume that the object completely fits into the camera's field of view. Active part-based object recognition systems such as [2], [7], [1] assume that the object to be identified is partitioned into a set of identifiable parts. The active planning in these systems incurs the overhead of tracking the region of interest through successive views. Volumetric primitives used in [2] are associated with a high feature extraction cost, while appearance-based methods [7], [1] require the object of interest to be segmented out from the background. None of these handle the case when internal parameters of the camera are allowed to vary, either unintentionally or on purpose.

The rest of the paper is organized as follows. Section 2 describes our method of pose estimation using inner camera invariants. We describe our hierarchical part-based knowledge representation scheme in Section 3. Section 4 describes our scheme of object recognition through next view planning. We present results of experiments with our system, in Section 5.

2. 3-D Euclidean Pose Estimation using Inner Camera Invariants

We use inner camera invariants to estimate the pose of parts present in a view of an object. The system uses this information to plan the next view, if the given view does not correspond to a unique pose of a particular object.

A commonly used projective camera model is [5]:

$$\lambda \mathbf{m} = \mathbf{P}\mathbf{M} = \mathbf{A}\left[\mathbf{R} \mid \mathbf{t}\right]\mathbf{M} \qquad (1)$$

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)

(b) (c) (d) (e)

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

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:

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

The skew parameter s may often be considered negligible [5]. 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 $\mathbf{m}_p = (u_p, v_p, 1)^T$, $p \in \{i, j, k\}$. Eliminating the internals of the camera,

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

where J_{ijk} and K_{ijk} are *image measurements* that are functions of $[\mathbf{R} \mid \mathbf{t}]$ $(= [\mathbf{r_1}\ \mathbf{r_2}\ \mathbf{r_3}]^T)$ and $\mathbf{M}_p$ ($p \in \{i, j, k\}$), and are independent of camera internals. J_{ijk} and K_{ijk} are *Inner Camera Invariants* [12] – image-computable invariants of the homography $\mathbf{A}$.

We use inner camera invariants for estimating the pose of a part ($\mathbf{R}$ and $\mathbf{t}$). Suppose we know the Euclidean coordinates $(X_i, Y_i, Z_i, 1)^T$ of 5 points (in general position) in the world coordinate system. *Six* independent inner camera invariant measurements give us six equations (of the type in Equation 3) in 6 unknowns: 3 rotations and translations each. We solve these equations to get the pose, using a suitable non-linear optimization routine (`constr/fmincon` in `MATLAB`). For a system with 4 degrees of freedom (hereafter, DOF) (*e.g.*, a setup with one rotational and all three translational DOF) as in Figure 1(a), we adopt the same procedure with *four* independent (inner camera) invariant measurements from four equations.

3 The Knowledge Representation Scheme

We propose a part-based hierarchical knowledge representation scheme that encodes domain knowledge about the objects in the model base. Figure 2 illustrates an example of our knowledge representation scheme. $\mathbf{O}$ represents the set of all objects $\{O_i\}$. An object node O_i stores its probability of occurrence, $P(O_i)$. 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 $[X_i, Y_i, Z_i]^T$, $1 \leq i \leq n$ (*e.g.*, $n \geq 6$ for a 6-DOF case: Section 2). It has $\mathbf{R}$ and $\mathbf{t}$ links with its neighbouring parts. We define a **Part-Class** as a set of parts, equivalent with

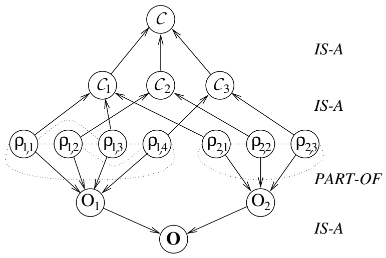


Figure 2. The knowledge representation scheme: an example

respect to a feature set. The set of parts is partitioned into different equivalence classes with respect to a given feature set: these equivalence classes are part-classes. $\mathcal{C}$ 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.

4 The Object Recognition Scheme

The system starts with an arbitrary view of an object in our model base. Our aim is to identify the given object, and the viewer pose with respect to it. There are three main components of our recognition scheme:

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

We discuss these three topics in the following sections. *Our scheme is independent of the particular technique to identify a part-class. The only requirement is that it should contain at least 5 points of interest for pose computation.* Figure 3 describes the main steps in our algorithm.

4.1. Hypothesis Generation

Let the given view of an object contain m parts – ρ_{i,j_1}, $\rho_{i,j_2}, \ldots\ \rho_{i,j_m}$. This view could correspond to any of the n objects in the model base. Further, this configuration of parts could have come from many different positions within the same object O_i. 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 system generates different part configuration hypotheses corresponding to the given view: We compute the estimated pose of each part (Section 2), and check if the relative poses of parts in the configuration are consistent with the **R** and **t** values in the knowledge representation scheme, within error limits (we use $\pm\ 5^o$ and $\pm\ 20mm$, respectively). The next section describes the process of computing probabilities associated with each part configuration hypothesis.

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,[I|0]);
5.  compute_hypothesis_probabilities(
             search_tree_root);  (* Eq. 5 *)
6.  IF the probability of some hypothesis
    is ≥ a pre-determined thresh THEN
             exit & call success;
7.  expand_search_tree_node(search_tree_root,
         MAX_LEVELS);  (* Section 4.3 *)
          (* ------ SECOND PHASE ------ *)
      previous:=search_tree_root;
      expected:=get_best_leaf_node(
             search_tree_root);
8.  {[R|t]}:=compute_movements(expected,previous);
    make_movements({[R|t]});
    image:=get_image_of_object();
9.  part_class_info:=identify_part_classes(image);
    IF NO part_class observed THEN
       (* — backtrack — *)
       undo_movements({[R|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,{[R|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 ≥ 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

4.2. Probability Calculations

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 (Step 1 in Figure 3).

$$P(\rho_{i,j_1},\ \rho_{i,j_2},\ \ldots\ \rho_{i,j_m})\ = \\ P(O_i)\ \cdot\ P(\rho_{i,j_1},\ \rho_{i,j_2},\ \ldots\ \rho_{i,j_m}\mid O_i) \qquad (4)$$

We may form estimates of $P(\rho_{i,j_1},\ \rho_{i,j_2},\ \ldots\ \rho_{i,j_m}\mid O_i)$ from 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. We have experimented with objects having planar faces. For such a case, one may approximate the probability of a part by its relative area in the 3-D model.

We use the Bayes rule to compute the *a posteriori* probability of each hypothesized configuration (Step 5 in Figure 3):

$$P(\rho_{i,j_1},\ \rho_{i,j_2},\ \ldots\ \rho_{i,j_m}\mid \mathcal{C}_{k_1},\ \mathcal{C}_{k_2},\ \ldots\ \mathcal{C}_{k_m}) \\ =\ Numerator/Denominator \qquad (5)$$

where $Numerator$ is given by

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

and $Denominator$, by

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

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 3), each of the terms $P(\mathcal{C}_{k_1},\ \mathcal{C}_{k_2},\ \ldots\ \mathcal{C}_{k_m}\mid \rho_{l,j_1},\ \rho_{l,j_2},\ \ldots\ \rho_{l,j_m})$ 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:

$$P(O_l) = \sum P(\rho_{l,j_1},\ \rho_{l,j_2},\ \ldots\ \rho_{l,j_m}\mid \mathcal{C}_{k_1},\ \mathcal{C}_{k_2},\ \ldots\ \mathcal{C}_{k_m}) \qquad (6)$$

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 uses Equation 6 to updates its probability. If the *a posteriori* probability of no hypothesis (Equation 5) is above a predetermined threshold, we have to take the next view to try to disambiguate between the competing hypotheses.

4.3. Next View Planning

We describe the state of the system in terms of the competing view interpretation hypotheses, and the set of $\mathbf{R}$ and $\mathbf{t}$ movements made thus far. We use a search tree node to represent the system state. One needs to plan the best move out of the current state to disambiguate between the competing hypotheses, subject to memory and processing limitations, if any. Search tree expansion proceeds according to the $\mathbf{R}$ and $\mathbf{t}$ relations in the knowledge representation scheme. Each search tree move is to get to the centre/centroid of the expected part. Thus, the expected part is more likely to be in the camera's field of view even in the event of a zoom-in/zoom-out, for example – thus providing *robustness to small movement errors*. The planning process aims to get to a leaf node of the corresponding search tree – one corresponding to a unique part-configuration. One may also employ a limited memory search tree expansion (MAX_LEVELS in Figure 3). We use three stages of filtering to get the best leaf node (Step 7 in Figure 3). First, we consider those leaf nodes which lie along a path from the most probable hypothesized view interpretation in the previously observed node. The algorithm assigns each search tree node, a 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. From among these leaf nodes, we select those associated with with minimum total path weight. We resolve remaining ties in favour of one of with a smaller number of rotational movements. We are selecting the move with the best discriminatory ability, at each stage. Hence, it is consistent in a decision-theoretic sense [11].

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 (Step 8). Similar to the process in Section 4.1, we generate different interpretation hypotheses corresponding to this view. The non-detection of some parts in the vicinity of the expected part (we do not predict a view) 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 corresponds to the expected search tree node, 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 (Step 12). If the current observation does not correspond to the expected search tree node, the system searches for the node corresponding to this observation among all leaf nodes corresponding to the movements made from the previous viewpoint. If we find one, then we repeat the process described above. If not, we undo the current movements, get the next best leaf node, and proceed (Step 9). If the probability of no hypothesis is above the threshold, this node is expanded further (Step 14).

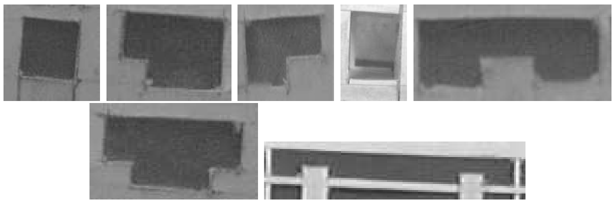


Figure 4. The 7 part-classes which the 459 parts belong to, for our model base: $DW4$, $DW6L$, $DW6R$, $OPEN$, $DW8HANDLE$, $DW8T$, and $DW12$, respectively in row-major order.

5 Experimental Results & Discussion

Figure 1 shows the set of architectural models we have experimented with. We have a 4-DOF experimental setup: translations along the $\mathbf{X}$-, $\mathbf{Y}$- and $\mathbf{Z}$- axes, and rotation about the $\mathbf{Y}$- axis (Figure 1(a)). We have chosen as (2-D) parts the doors and windows of different shapes and sizes in the models. Such an experimental setup is consistent with our 4-DOF pose estimation procedure 2. The first step in processing a given view of the object involves a segmentation of 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 [10] and grey-level information for recognizing part-classes. We emphasize however, the *our recognition strategy is independent of the types of the parts and part-classes, or the method to detect them.* There is 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. We describe four illustrative experiments with our system.

The initial view in Figure 5 shows the two detected parts with part-classes $DW8T$ and $DW4$. Of the 6 possible hypotheses, our part pose estimation procedure (Section 4.1) prunes out 4 of them. The system plans a disambiguating move: the second view contains the expected part (bottom row, centre). This move results in correct recognition and pose estimation, in spite of *the failure to detect a neighbouring part* (top row, centre). Further, *parts in a view need not be coplanar*. Figure 6 shows an example of correct recognition in such a case.

The first view in Figure 7 could have come from 257 configurations of two adjacent parts with part-class $DW4$. Two moves from this position were sufficient to recognize the object, the third view containing the expected part (the large 4-cornered window). For the same first two views, we performed two zoom-out operations at the the third camera position. The recognition results are the same in each of the cases — Figure 7 (a), (b) and (c). Further, the camera pose with respect to the expected part 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 $\langle\ 9.896^o,\ -22.000mm,\ -9.999mm,\ 150.000mm\ \rangle$, respectively.

In Experiment 4, the presence of a tree (an unmodeled object) accounts for clutter in the first, third and fourth view of Figure 8. The system plans the next move on the basis of a part: it does not predict an entire view. Hence, recognition performance is not affected by the presence of unmodeled objects or the non-detection of parts in the vicinity of the expected part.

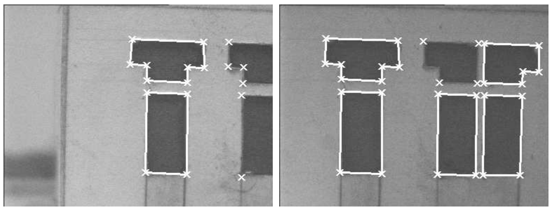


Figure 5. Experiment 1: The sequence of moves required to identify the object and its pose. The failure to detect a part does not affect the system (details in text).

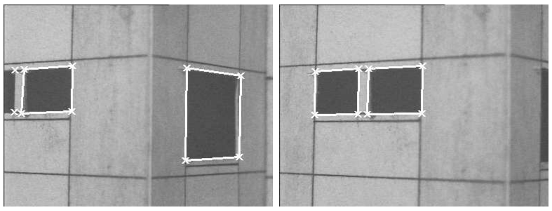
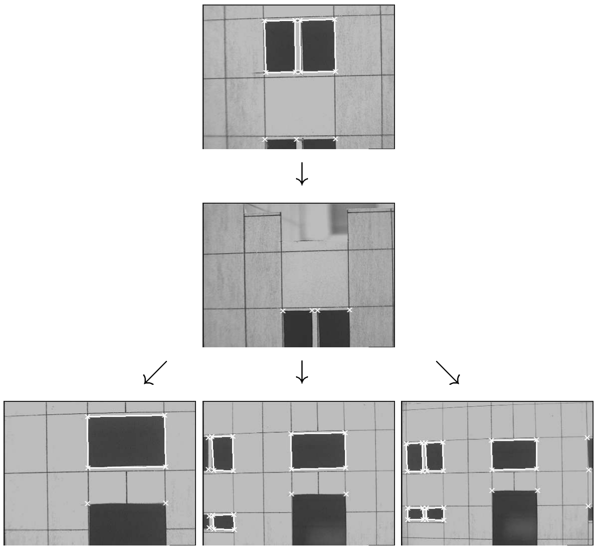


Figure 6. Experiment 2: The sequence of moves required to identify the object and its pose. The parts in the initial view do not lie in the same plane.

6 Conclusions

We present a new on-line scheme to identify large 3-D objects which do not fit into a camera's field of view, and finds the pose of the (uncalibrated) camera with respect to the object. The system does not assume any knowl-



(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).

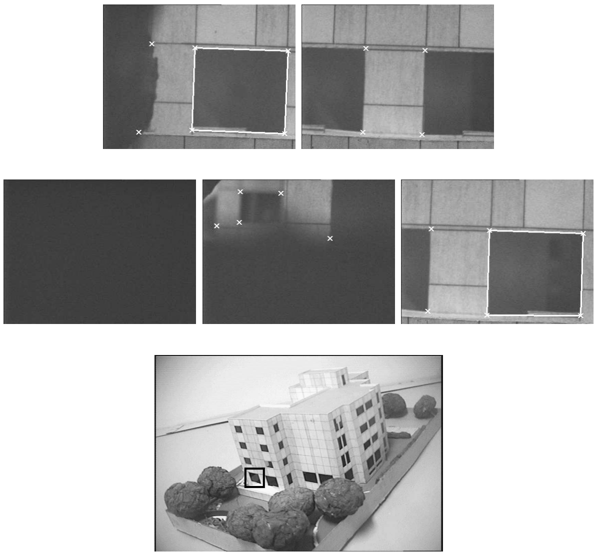


Figure 8. Experiment 4: The first, third and fourth views are cluttered by the presence of a tree. The image at the bottom shows an overall view. The corresponding window is highlighted with a black square.

edge of the internal parameters of the camera, or their constancy (permitting a zoom-in/zoom-out operation, for example). The part-based knowledge representation scheme is used both for probabilistic hypothesis generation, as well as in planning the next view. We show results of successful recognition and pose estimation even in cases of a high degree of interpretation ambiguity associated with the initial view.

References

[1] 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*, pages 685 – 691, 1998.

[2] 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*, 67(3):239 – 260, September 1997.

[3] 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*, 30(1):67 – 76, January 2000.

[4] 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).

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

[6] K. D. Gremban and K. Ikeuchi. Planning Multiple Observations for Object Recognition. *Int. Journal of Computer Vision*, 12(2/3):137 – 172, April 1994.

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

[8] 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*, 5(6):765 – 783, December 1989.

[9] J. Maver and R. Bajcsy. Occlusions as a Guide for Planning the Next View. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 15(5):76 – 145, May 1993.

[10] C. A. Rothwell. *Recognition using Projective Invariance*. PhD thesis, University of Oxford, 1993.

[11] S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, Inc., 1995.

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