Active 3-D Object Recognition using Appearance-Based Aspect Graphs

Sumantra Dutta Roy Nirupama Kulkarni
Department of Electrical Engineering, IIT Bombay, Powai, Mumbai - 400 076, INDIA
{sumantra, nirupama}@ee.iitb.ac.in

Abstract

We present a new active active recognition scheme (using an uncalibrated camera) based on a new idea, appearance-based aspect graphs. The scheme is robust to background clutter, and affine transformations of the object. We use a probabilistic reasoning framework which helps in probability calculations and planning the next view (when a view of the object does not contain sufficient features to recognise it unambiguously), in conjunction with a new hierarchical knowledge representation scheme. Preliminary experiments with the system show encouraging results.

1. Introduction

In this paper, we propose a new active 3-D object recognition strategy which used appearance-based aspect graphs. The strategy uses a hierarchical knowledge representation scheme which guides our probabilistic recognition scheme. We show results of experimentation with a standard object database in support of our proposed strategy.

Murase and Nayar [14] propose the use of parametric eigenspaces for isolated 3-D object recognition. The advantage of such a strategy over other feature-based ones is the use of all information present in an image, doing away with the (often noisy and error-prone) intermediate process of feature extraction. Existing appearance-based recognition schemes [14], [4] typically need accurate object-background segmentation, and normalization to illumination conditions and size. In this paper, we show how our formulation gets around these requirements, which constrain any appearance-based recognition strategy.

Most 3-D object recognition strategies (whether appearance-based, or feature-based) use information from a single view of an object [1], [6], [14]. A single view may not contain enough features to recognise an object unambiguously. Hence, there is a need to take one or more views around the object, in a planned manner [8], [9], [4], [10]. Borotschnig *et al.* propose an active appearance-based recognition strategy [4] - to the best of our knowledge, this is the only other work that uses planning for appearance-based active object recognition (the work of Winkeler *et al.* [16] deals with the selection of prominent views, rather than planning a set of views). In another paper, Borotschnig *et al.* perform a comparative study of using uncertainty calculi for an active appearance-based recognition strategy - using probability theory, the Dempster Shafer theory, and fuzzy logic [3]. While the authors report good results for a probabilistic approach under certain conditions, they use an information theoretic-criterion for termination. This paper presents a simpler and computationally efficient procedure for the same, using probabilities alone. Additionally, we derive a theoretical bound on the number of views required for recognition, too. Thus, this does nor incur the cost of using a large number of images for recognition, as in the active appearance-based method of Deinzer *et al.* [7], which uses a CONDENSATION-based tracker. Further, the authors assume an equal probability assumption of all views - this is not a tenable assumption for most objects.

The organisation of the rest of the paper is as follows. Section 2 proposes a hierarchical scheme to efficiently represent domain knowledge, and help in the planning process. We introduce our probabilistic recognition scheme in Section 3, and use it for next view planning in Section 4. We show results of experimentation with our system in Section 5.

2. View Recognition; Knowledge Representation Scheme

An Aspect Graph [12] partitions the space of viewpoints around an object into equivalence classes with respect to a set of features (*aspects*). As aspect graph has nodes as aspects, and links correspond to visual events - aspect transitions. Aspects can be further clustered into *classes* - equivalence classes of object appearances. In this paper, we propose the concept of an *Appearance-based Aspect Graph* (based on a parametric eigenspace representation of an object) and develop a hierarchical knowledge representation scheme based on the same idea. A view of an object corresponds to a class. The following section describes our

method of class recognition using an uncalibrated camera.

2.1. Appearance-based View Recognition

We propose a method for recognising a view of an object (a class), which does not suffer from the limitations of existing appearance-based recognition schemes such as [14], [4]. The first concerns size and orientation normalisation. Our scheme is independent of these factors - in fact, *any affine transformations* between the stored information about an object in the model base, and that presented to the recognition system. We adapt an idea from Black and Jepson's EigenTracker [2] for this purpose. We pose the problem as finding affine transformation coefficients $\mathbf{a} = [a_0\ a_1\ a_2\ a_3\ a_4\ a_5]^T$ and the eigenspace reconstruction coefficients $\mathbf{c}$, such that the robust error function between the parameterized image $\mathbf{I}$ (indexed by its pixel location $\mathbf{x}$) and the reconstructed one $\mathbf{Uc}$ (where $\mathbf{U}$ is the matrix of the most significant eigenvectors) is minimum, for all pixel positions $\mathbf{x} = [x\ y]^T$:

$$arg\ min_{\forall \mathbf{x}}, \rho(\mathbf{I}(\mathbf{x} + \mathbf{f}(\mathbf{x}, \mathbf{a})) - [\mathbf{Uc}](\mathbf{x}), \sigma) \qquad (1)$$

Here, $\rho(x, \sigma) = x^2/(x^2 + \sigma^2)$ is a robust error function, and σ is a scale parameter. The 2-D affine transformation is given by

$$\mathbf{f}(\mathbf{x}, \mathbf{a}) = \begin{bmatrix} a_0 \\ a_3 \end{bmatrix} + \begin{bmatrix} a_1 & a_2 \\ a_4 & a_5 \end{bmatrix} \mathbf{x} \qquad (2)$$

Section 3 examines our probabilistic class (view) recognition in detail. Background segmentation is another restrictive point in existing appearance-based recognition schemes (and is often done by hand, for the objects in the model base, or all experiments are done against a constant background). We get around this limitation using a simple background subtraction mechanism: we move the object by one step, and using a variant of a pyramidal motion segmentation scheme [11]. The above affine parameters have an added advantage - they help in accurate localisation of the object of interest in the given image, and ensure a *parallelogram bounding box* around the object in question - a tighter fit, as compared to a rectangle. A tighter fit ensures less background clutter, ensuring better chances of a match between the stored eigenspace, and the object in question. A third point is handling different illumination effects. For this, one may either learn the illumination parameters in the eigenspace, or assuming illumination source not to change their position, normalise the image brightness.

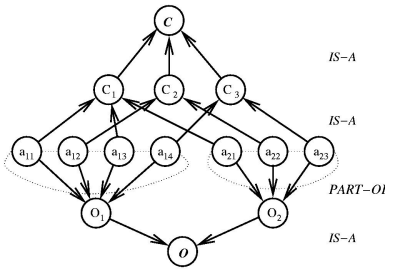

Figure 1. The Knowledge Representation Scheme: An Example

2.2. The Knowledge Representation Scheme

Rimey and Brown [15] propose the use of Bayes Nets for active planning tasks. We propose a new hierarchical knowledge representation scheme encoding domain knowledge about the model base. Figure 2.1 shows an example. An object node O_i is linked to its constituent aspects a_{ij} (a *PART-OF* relationship). Aspect a_{ij} has an angular extent θ_{ij}. Adjacent aspects have a link between them. Each node a_{ij}, O_i stores its *a priori* probability. Classes C_k have an *IS-A* relationship with the set of all classes $\mathbf{C}$. The same goes for objects O_i and $\mathbf{O}$. A class node C_k stores its *a priori* probability $P(C_k)$. Each class node C_k is linked with the set of aspects a_{ij} which correspond to it. The knowledge representation scheme helps in probability calculations (based on the instantiation of a node as evidence with its associated probability: Section 3. This evidence in turn, propagates across the levels, right down to the object nodes). It also plays an important role in next view planning (Section 4). We use an on-line eigenspace update mechanism [5] to build the aspect graph - we declare an aspect boundary when the reconstruction error is above a particular threshold.

A view of a particular object is an input to the recognition system. The next section describes our probabilistic hypothesis generation scheme.

3. Hypothesis Generation

3.1. *a priori* Probabilities

For a given set of N objects, we take the *a priori* probability of each object O_i is $1/N$. We first compute the *a priori* probability of an aspect a_{ij} of object O_i as

$$P(a_{ij}) = P(O_i)P(a_{ij}|O_i) \qquad (3)$$

(This equation computes the joint probability $P(a_{ij}, O_i)$, and serves as an indicator of the *a priori* probability of an aspect before the next observation is taken.) $P(a_{ij}|O_i)$ is initialized to $\theta_{ij}/360^\circ$ before the system takes the first view. From this equation, we initialize the *a priori* class probabil-

ity of class C_k as follows:

$$P(C_k) = \sum_p [P(O_p) \cdot \sum_q P(a_{pq}|O_p)] \quad (4)$$

The inner summation is for all aspects a_{pq} belonging to class C_k *i.e.*, *PART-OF*(a_{pq}, O_p) = *TRUE*. We also model cases of feature detection errors during the process of class determination. We use the term $P(C_l\ actual|C_k\ obs)$ to denote the probability that the class is actually C_l, given that class C_k is observed. We compute these estimates in the off-line aspect graph construction process [13]. The recognition system takes in an arbitrary view of an object as input. The first step is class recognition corresponding to the view of the object. The information about the classes that the given view could correspond to, along with their probabilities serves as an input to the next phase: aspect, and object recognition.

3.2. Class Recognition

The system is given a view of an object as an image **I**. We calculate the *a posteriori* probability of the observed class C_k, given that image **I** has been observed:

$$P(C_k\ obs|\mathbf{I}) = \frac{e^{-err_k}}{\sum_{\forall l} e^{-err_l}} \quad (5)$$

The summation in the denominator is over all classes C_l. Here, err_l is the reconstruction error incurred on projecting the image **I** onto the eigenspace of class C_l. We need $P(C_l\ actual|\mathbf{I})$, the probability of the class being actually C_l, given image **I** observed:

$$P(C_l\ actual|\mathbf{I}) = \sum_k P(C_l\ actual|C_k\ obs)P(C_k\ obs|\mathbf{I}) \quad (6)$$

Here, $P(C_l\ actual|C_k\ obs)$ is the probability of the observed class C_k actually being class C_l - this takes feature detection errors into account. The summation for k is over all classes C_k.

3.3. Object Recognition

We use the *a posteriori* class probabilities to calculate the *a posteriori* aspect and object probabilities. We generalize the winner-take-all approach in an earlier work [8] to include actual class probabilities. We compute the *a posteriori* aspect probabilities $P(a_{ij}|\mathbf{I})$ as follows:

$$\begin{aligned} P(a_{ij}|\mathbf{I}) &= \sum_k P(a_{ij}|C_k)P(C_k\ actual|\mathbf{I}) \\ &= P(a_{ij}|C_r)P(C_r\ actual|\mathbf{I}) \end{aligned} \quad (7)$$

We note that the above summation can be simplified by the observation that $P(a_{ij}|C_k)$ = 1 for *exactly one* class C_r such that *IS-A*(a_{ij}, C_r) = *TRUE*. The previous section (Section 3.2) gives the computation for the second term. Equation 8 below shows the computations for $P(a_{ij}|C_r)$:

$$P(a_{ij}|C_r) = \frac{P(C_r|a_{ij})P(a_{ij})}{\sum_{ij} P(C_r|a_{ij})P(a_{ij})} \quad (8)$$

The summation in the denominator is for aspects a_{ij} such that *IS-A*(a_{ij}, C_r) = *TRUE*. Thus, our knowledge representation scheme (Section 2) simplifies computations by having links between only the relevant terms. Finally, we compute the *a posteriori* object probabilities:

$$P(O_i|\mathbf{I}) = \sum_j P(a_{ij}|\mathbf{I}) \quad (9)$$

4. Next View Planning

The given view of an object could correspond to more than one aspect from more than one object in the model base. Due to this ambiguity, one has to search for the best disambiguating move, in order to recognise the object. This is of course, subject to memory and processing limitations. We use a search tree to search for this best move. Figure 2 outlines the basic steps in our object recognition algorithm. The first phase consists of steps described in detail in the previous section. The next view planning scheme is similar to that of an earlier work involving the first author [8], with one important difference. A search tree node represents the following information: the classes which could correspond to the given view along with their respective probabilities (in case the probability of no class is above a particular threshold - the earlier work [8] went ahead with the most probable class), the aspects corresponding to each class, and the possible range of positions within each aspect. The step size of movement is an important parameter, since too small a step size may cause us to remain within the same aspect - incurring wasteful image processing operations. A large move on the other hand, could miss out on a unique aspect, altogether. From a viewpoint, we categorise moves as:

Primary Move A primary move represents a minimum angle move out of an aspect.

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 unique aspect in each of the two directions We construct search tree nodes corresponding to both moves.

ALGORITHM identify_object

```
        (*------ FIRST PHASE ------*)
01. initialize_obj_probabilities(); (*1/N*)
02. im:=get_image_of_object();
03. class_list:=identify_class(im);
            (*Sec 3.2*)
    IF unknown_class(class_list) THEN exit;
04. st_root:=
    construct_search_tree_node(class_list,0);
05. compute_object_probabilities(
    st_root); (*Eqs 7,9*)
06. IF prob of some object ≥
    a thresh THEN exit & declare success;
07. expand_search_tree_node(
    st_root,0,class_list); (*Sec 4*)
    best_leaf:=get_best_leaf_node(st_root);
       (*------ SECOND PHASE ------*)
        prev:=st_root;
        expected:=best_leaf;
08. α:=compute_move_angle(expected,prev);
    make_movement(α);
    im:=get_image_of_object();
09. class_list:=identify_class(im);
    IF unknown_class(class_list) THEN exit;
10. new_node:=
    construct_search_tree_node(class_list,α);
11. compute_object_probabilities(new_node);
12. IF prob of some object ≥
    a thresh THEN exit & declare success;
13. expand_search_tree_node(
    new_node,α,class_list); (*Sec 4*)
    best_leaf:=get_best_leaf_node(st_root);
        prev:=new_node;
        expected:=best_leaf;
14. GOTO step 08
```

Figure 2. The Object Recognition Algorithm

2. **Type II move**: Exactly one out of α_{ij}^c and α_{ij}^a takes us to a single unique aspect a_{ip}. For the other direction, the aspect we would reach depends upon the initial position 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 from a node. If more memory/processing time is available, one can generate auxiliary moves also. 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 (Steps 07 and 13 in Figure 2). The system takes this movement, and the above process (the second phase of the algorithm, Figure 2) repeats till successful recognition. (We assume that the given object is one of the model base objects. Our system will report an unknown object for an object with an unknown class. The system will fail for three cases, as outlined in a previous work [8].)

It is important to note that we do not exhaustively generate all moves out of an assumed aspect. We strike a balance here, and use the next observation to look at the possible classes (and their associated aspects) this observation (given the past history of observations) could have corresponded to. Thus, our search tree expansion prunes out a large part of the search space, and resynchronises with each observation. As in [8], we can prove that the search tree node expansion is finite, and always terminates. Further, we can show that for a given view corresponding to a set of n aspects, the average number of moves needed to discriminate between them is $\mathcal{O}(log_e n)$, for a representative case [8].

5. Results and Discussion

We present results of some preliminary experiments with our system. In the absence of a standard modelbase of objects for active appearance-based recognition, we have chosen to experiment with the COIL-20 database (Columbia University), Figure 3. *While these objects have been used for single-view recognition, we simulate a case for active object recognition using the same objects, by considering a very small number of eigenvectors (Typically, 10% of the total number) and considering a suitable threshold for the similarity between two classes.* With this in mind, we examine our recognition strategy with regard to the following issues. This section presents the results of 111 experiments with our system. For our experiments, we have (empirically) chosen a threshold probability of 0.9 for object recognition.

Figure 3. The COIL-20 database (Columbia University)

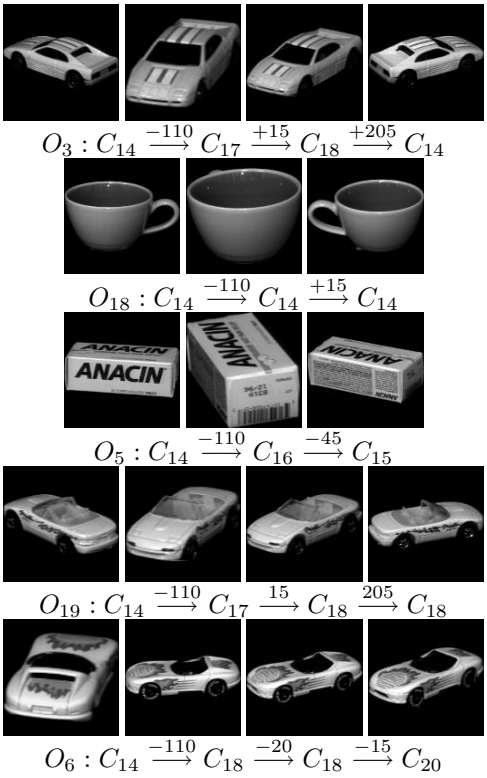

Figure 4. Active recognition with planning, corresponding to primary moves alone. The first image in each row, when projected onto the eigenspace of all classes, shows minimum distance from class C_{14}, with a sufficiently high probability (details in text). Each row shows the discriminating moves for the same starting class C_{14} for different objects in the modelbase.

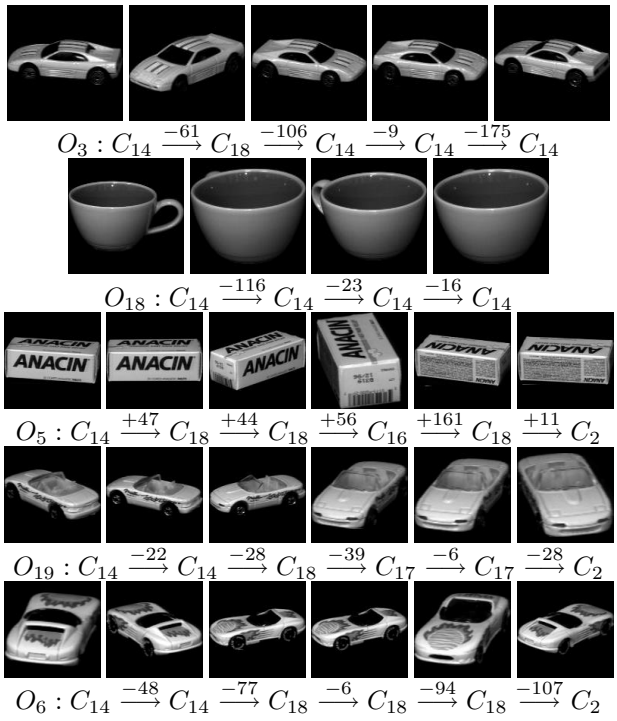

Figure 5. Active recognition with random unplanned moves: a representative case. Overall, planning reduces the average number of moves needed for unambiguous recognition. These experiments use the same initial class C_{14} as the same starting point as in Figure 5. Each row shows the (random) moves taken by the system before the probability of the corresponding object crosses a certain threshold value (0.9 for our experiments).

Planned Recognition vs Recognition with random views

Figure 5 shows a sample of recognition results for the same starting class C_{14} (which could have come from 8 aspects in 5 objects out of the 20 in the modelbase), for different objects in the modelbase. The trade-off between planning with both Primary and Auxiliary moves, and Primary moves alone, is one of completeness in the planning process (without incurring the prohibitive cost of an exhaustive search) versus consuming less memory and processing resources (at the cost of increasing the number of moves required for recognition). *In order to compare a the benefits of a planned recognition strategy with that of a random strategy (moves made at random, with the same probability updating process in place), we have worked with primary moves alone.* For the 111 experiments in each case, the average number of moves is 3.46, which increases to 5.40 for the random case. Figure 5 shows the corresponding moves for the same starting initial class C_{14}, for each of the objects in Figure 5. The number of moves for the random case is clearly more.

Multi-view recognition vs. single-view recognition

For the 111 experiments, we observed correct recognition results from a single view in only 65.70% of the cases, whereas the corresponding number for our active recognition strategy is 98.19%. This clearly shows the advantage of next view planning over a single view-based strategy. In our experiments, the cases where the system failed was with respect to the two cars in the COIL-20 model base. For our set of features, this corresponds to a case where our strategy is not guaranteed to succeed - objects with the same aspect structure (*i.e.*, the layout of classes in the aspect graph) but different aspect angles. where the system is not able to distinguish between objects with the same layout of aspects. It is important to note however, that the probabilities of all other objects were zero at the end of these experiments.

Variation of object probabilities

In Figure 5, we show the variation of *a posteriori* object probabilities for the sequences depicted in the first and

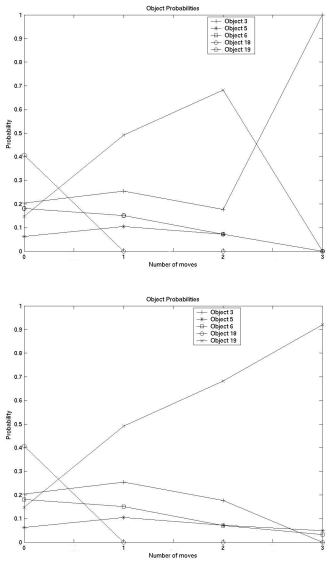

Figure 6. Variation of the *a posteriori* object probabilities for the sequences corresponding to the first and fourth rows in Figure 5: the first image corresponds to class C_{14}, and the two moves are for objects O_3 and O_{19}, respectively. The curves are for all objects with non-zero *a posteriori* probabilities after the first observation.

fourth rows in Figure 5. The top figure (for O_3) shows an interesting effect: the opportunistic nature of the system. Till the second observation, object O_5 is the most probable. The evidence after the third move results in the probability of O_3 going to 1, and all the rest becoming zero.

6. Acknowledgements

The first author is indebted to Profs. S. Chaudhury and S. Banerjee for useful discussions, which helped develop their feature-based active 3-D object recognition system at I.I.T. Delhi [8], which sowed the seeds for this work.

References

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

[2] M. J. Black and A. D. Jepson. EigenTracking: Robust Matching and Tracking of Articulated Objects Using a View-Based Representation. *Int. Journal of Computer Vision*, 26(1):63 – 84, 1998.

[3] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz. A Comparison of Probabilistic, Possibilistic, and Evidence Theoretic Fusion Schemes for Active Object Recognition. *Computing*, 62(4):293 – 319, 1999.

[4] H. Borotschnig, L. Paletta, M. Prantl, and A. Pinz. Appearance Based Active Object Recognition. *Image and Vision Computing*, 18(9):715 – 728, 2000.

[5] S. Chandrasekaran, B. S. Manjunath, Y. F. Wang, J. Winkeler, and H. Zhang. An Eigenspace Update Algorithm for Image Analysis. *Graphical Models and Image Processing*, 59(5):321 – 332, September 1997.

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

[7] F. Deinzer, J. Denzler, and H. Niemann. Improving Object Recognition by Fusion of Multiple Views. In *Proc. Indian Conf. on Computer Vision, Graphics and Image Processing (ICVGIP)*, pages 161 – 166, 2002.

[8] 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.

[9] S. Dutta Roy, S. Chaudhury, and S. Banerjee. Recognizing Large 3-D Objects through Next View Planning using an Uncalibrated Camera. In *Proc. IEEE Int'l Conf. on Computer Vision (ICCV)*, pages II:276 – 281, 2001.

[10] S. Dutta Roy, S. Chaudhury, and S. Banerjee. Active Recognition through Next View Planning: A Survey. *Pattern Recognition*, 37(3):429 – 446, March 2004.

[11] M. Irani, B. Rousso, and S. Peleg. Computing Occluding and Transparent Motions. *Int. Journal of Computer Vision*, 12(1):5 – 16, January 1994.

[12] J. J. Koenderink and A. J. van Doorn. The Internal Representation of Solid Shape with Respect to Vision. *Biological Cybernetics*, 32:211 – 216, 1979.

[13] N. Kulkarni. Appearance-Based Aspect Graphs: Construction and use in Active Object Recognition. B.Tech Thesis, Dept. of EE, IIT Bombay, 2004.

[14] H. Murase and S. K. Nayar. Visual Learning and Recognition of 3-D Objects from Appearance. *Int. Journal of Computer Vision*, 14:5 – 24, January 1995.

[15] R. D. Rimey and C. M. Brown. Control of Selective Perception using Bayes Nets and Decision Theory. *Int. Journal of Computer Vision*, 12(2/3):173 – 207, April 1994. Special Issue on Active Vision II.

[16] J. Winkeler, B. S. Manjunath, and S. Chandrasekaran. Subset Selection for Active Object Recognition. In *Proc. IEEE Int'l Conf. on Computer Vision and Pattern Recogntion (CVPR)*, pages II:511 – 516, 1999.