Aspect Graph Construction with Noisy Feature Detectors

Sumantra Dutta Roy, Santanu Chaudhury, Subhashis Banerjee

Abstract

Many 3-D object recognition strategies use aspect graphs to represent objects in the model base. A crucial factor in the success of these object recognition strategies is the accurate construction of the aspect graph, its ease of creation, and the extent to which it can represent all views of the object for a given setup. Factors such as noise and non-adaptive thresholds may introduce errors in the feature detection process. This paper presents a characterization of errors in aspect graphs, as well as an algorithm for estimating aspect graphs, given noisy sensor data. We present extensive results of our strategies applied on a reasonably complex experimental set, and demonstrate applications to a robust 3-D object recognition problem.

Keywords Aspect Graph, Feature Detection Errors, Noisy Sensor Data, Aspects, Classes, Aspect Graph Construction Algorithm

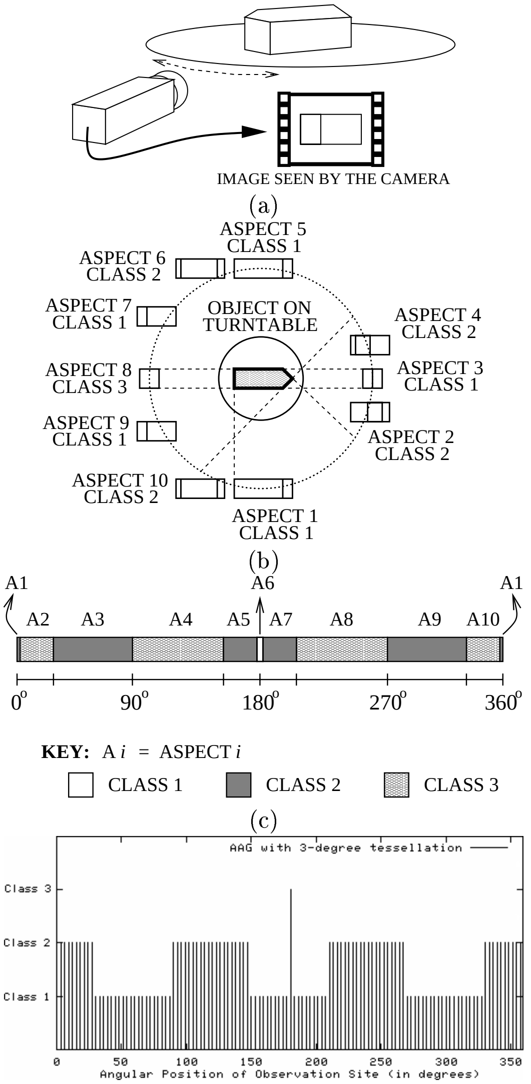
I. Introduction

Many 3-D object recognition strategies use aspect graphs ([1], [2], [3], [4], [5], [6], [7]). Any object recognition strategy, based on single or multiple views, requires robust identification of a view of the given object. The success of such a strategy crucially depends upon its ability to model and account for feature detection errors. This paper presents a new algorithm for aspect graph generation with noisy active sensors.

View-based 3-D object recognition strategies are based on representation schemes such as appearance-based parametric eigenspaces [8], and aspect graphs. The former method is associated with problems such as segmenting the object from the background, high storage requirements, and a time-consuming (though off-line) training phase. Further, an aspect graph-based approach is more general in that appearance-based information can also be used to construct an aspect graphs.

Depending on the method used to construct them, aspect graphs may be classified as: Exact aspect graphs, and Approximate aspect graphs (hereafter, AAGs)

Analytical approaches are used to construct exact aspect graphs – directly from object shapes and surface characteristics. A limitation of such an approach is its applicability to only a specific class of objects – transparent smooth objects([9]), piecewise-smooth objects and algebraic surfaces([10]), curved objects([11]), 2.5-D polyhedra under orthographic projection([12]), arbitrary polyhedra under orthographic projection([13]), convex polyhedra under perspective projection([14]), arbitrary polyhedra under perspective projection([15]) and objects with moving parts([16]). In the context of aspect graphs, an 'error' is a non-conformance of the observed feature data at a position in the viewpoint space with the one predicted by the aspect graph. Algorithms for aspect graph construction from CAD data generally do not address the issue of errors in aspect graphs. For analytical approaches, one would need very precise models of not only the noise process, but also the sensors and detectors, and the imaging process. Lu, Shapiro and Camps consider a very restricted class of feature detection errors in [17]. Even the elaborate feature modeling and prediction module in PREMIO [18], [19] is not able to account for many physical effects. The scale space approach of Eggert *et al.* [20], [21] and Pae and Ponce [22] model some types of errors through the notion of scale. Wilkes, Dickinson and Tsotsos [23], [24] present a computational model for view degeneracy. Weinshall and Werman [25] analytically derive the 'view stability' and 'view likelihood' measures for two feature-based 2-D metrics. These approaches do not account for all types of errors in aspect graphs. Further, the work on exact aspect graphs concentrates solely on their construction, rather than their use in a recognition task [26].



(a)

(b)

(c)

(d)

Fig. 1. (a) An example of the 1-DOF case; (b) the object with its aspects and classes; (c) Gantt chart representation of the aspect graph; and (d) an AAG of the object, shown as a class-distribution

AAG construction approaches usually tessellate the viewpoint space into uniform partitions. A site is a representative viewpoint for a partition, at which sensor data is collected. Adjacent viewpoints which give the same appearance of the object with respect to a feature set, are grouped together to form an aspect.

S. Dutta Roy is with the Dept. of EE, IIT Bombay, Mumbai - 400 076, INDIA. E-mail: sumantra@ee.iitb.ac.in

S. Chaudhury is with the Dept. of EE, IIT Delhi, New Delhi - 110 016, INDIA. E-mail: santanuc@{cse,ee}.iitd.ernet.in

S. Banerjee is with the Dept. of CSE, IIT Delhi, New Delhi - 110 016, INDIA. E-mail: suban@cse.iitd.ernet.in

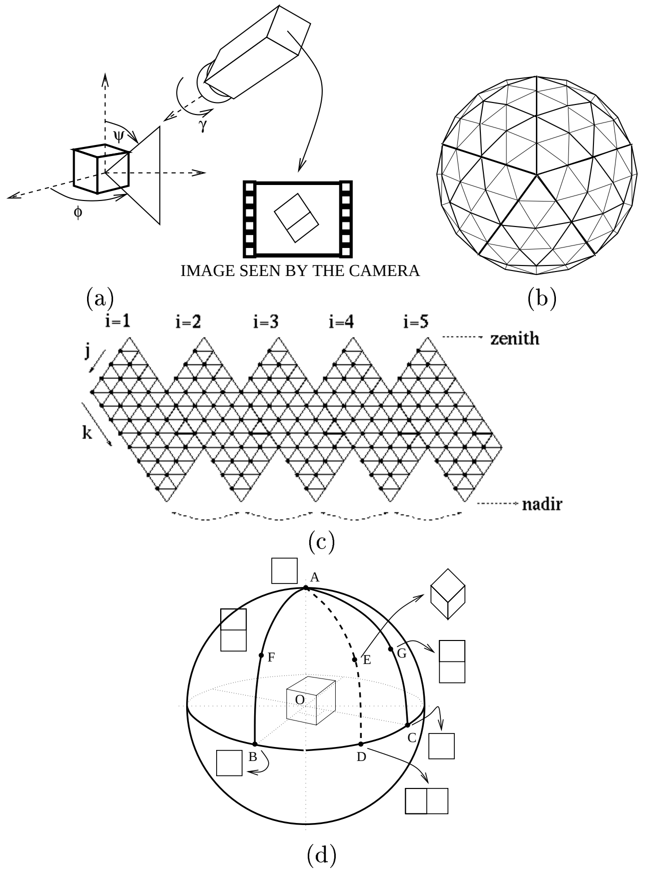


Fig. 2. (a) An example of the 3-DOF case; (b) The tessellated viewing sphere, and (c) its flattened-out representation; and (d) the object with its aspects and classes

Examples of partitioning strategies, and related issues are [27], [28], [3], [29], and [26]. The uniform partitioning approach is independent of the object shape and structure, the sensor, or the feature set. Using an AAG, it is possible to get information about noise and other feature detection errors from observed data directly. No related work accounts for feature detection errors in aspect graph construction. The authors in [3] mention only one type of error – errors on aspect boundaries.

This paper characterizes different types of errors in AAGs. We present an algorithm to construct an aspect graph, given noisy sensor data obtained from sites in the viewpoint space. We consider the uniform partitioning approach in this paper. The approach is independent of the specific features used. We propose a function to evaluate the output of aspect graph construction algorithms. We present results of over 100 experiments showing the effectiveness of our strategies. We use AAGs thus constructed in our work on robust 3-D object recognition using the same noisy feature detectors – details of which may be found in [5], [6] and [7].

The organization of the paper is as follows: Section II presents a characterization of errors in raw aspect data. We present our algorithm for aspect graph construction in Section III, as well as our evaluation function for comparing aspect graph construction algorithms. We present results of extensive experimentation in Section IV. Section V gives a brief overview of the application of our AAG construction algorithm to robust 3-D object recognition. In the concluding section, we summarize the salient features of our scheme.

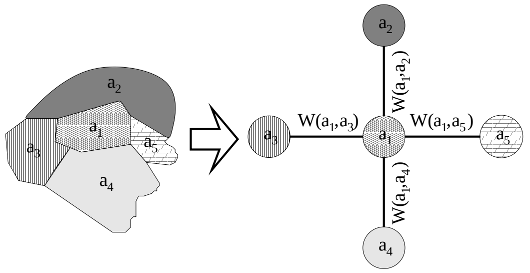


Fig. 3. Representation of a 3-DOF AAG: for aspect a_1

II. AAGs, Errors in Raw Aspect Data

Koenderink and van Doorn [30] define **aspects** as topologically equivalent classes of object appearances. Ikeuchi and coworkers generalize this definition to the output of different types of sensors [31]. An aspect is a collection of contiguous sites in viewpoint space which correspond to the same set of features. A **Class (or, Aspect-Class)** is a set of aspects, equivalent with respect to a feature set.

With the uniform partitioning approach, construction an AAG requires visiting each site in the tessellated viewing space around the object. In this paper, we consider the two cases, with different degrees of freedom (hereafter, DOF) between the object and the sensor: (i) the sensor can move around the object in a circle (1-DOF case), and (ii) a sphere (3-DOF case). Figure 1(a) shows an example of the 1-DOF case, along with an image taken by the camera (orthographic projection assumed). Figure 1(b) shows an example of an object with its associated aspects and classes. Figures 1(c) and 1(d) depict two ways of representing a 1-DOF aspect graph. Figure 1(c) shows a Gantt chart - each aspect of the object is represented by a shaded rectangle, proportional to its angular extent on the flattened-out perimeter of the viewing circle. Different shading patterns represent different classes. We may also represent an aspect graph as a plot of the class at each viewing position. The x-axis represents different viewing positions on the flattened-out perimeter of the viewing circle, while the non-metric y-axis represents different classes as different heights. Figure 1(d) shows an AAG of the example object represented as a class-distribution. We use a circular linked list as the data structure to represent an AAG. Each node (an aspect) stores information such as its associated class and its angular extent.

Figure 2 shows the corresponding examples for the 3-DOF case. The viewing space is a tessellated sphere (Figure 2(b)). Aspects and classes of this object are depicted in Figure 2(d), for one octant of the viewing sphere. We use the icosahedron-based tessellation algorithm of Chen and Kak [29]. We use a weighted graph as the data structure to represent a 3-DOF AAG (Figure 3). Each node represents an aspect. An adjacent node is one with which it shares a common boundary (in the tessellated viewing space). The weight of the link between them is the length of this common boundary.

Factors such as noise and non-adaptive thresholds may corrupt the output of a feature detector. The result is an incorrect feature at a viewpoint. Let the term 'raw aspect data' denote the collection of features obtained at the set of sites in the tessellated viewing space. We refer to aspects and classes obtained from raw aspect data as **aspect-candidates** and **class-candidates**, respectively. Thus, we can have erroneous aspect-candidates and class-candidates, but no erroneous aspects and classes. Let us use the terms $\mathcal{A}$ and $\mathcal{C}$ to denote the set of all aspect-candidates and the set of all class-candidates, for a particular model base, respectively. We define the following function:

$$CLASS\_CAND : \mathcal{A} \rightarrow \mathcal{C}$$

Thus, for an aspect-candidate α, the function $CLASS\_CAND(\alpha)$ returns its corresponding class-candidate. We use the notation $\mathcal{A}_c$ to denote the set of all aspect-candidates corresponding to class-candidate c. Formally,

$$\mathcal{A}_c \overset{\triangle}{=} \{ \ \alpha \ \in \mathcal{A} \mid CLASS\_CAND(\alpha) = c, \qquad c \in \mathcal{C}\}$$

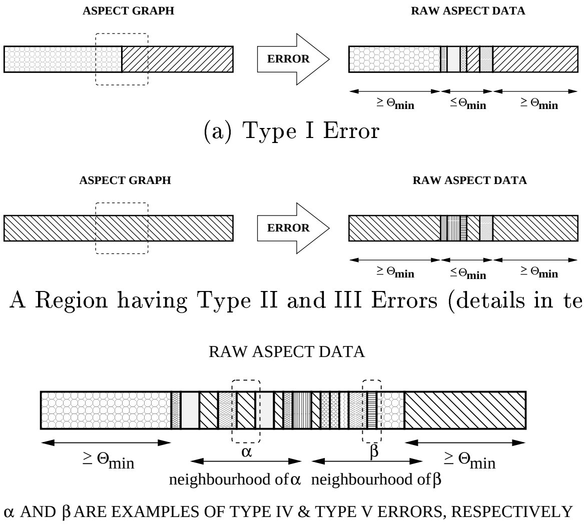
A. A Classification of Errors in Raw Aspect Data

An error-free AAG has aspects, whose corresponding angular extents are not too small in size. The discontinuities in the smoothness correspond to aspect boundaries. Thus, an error-free AAG is characterized by piecewise continuity in the class-candidate labels. We base our error classification on experimental observations – the feature data obtained with an active sensor. We first present an intuitive explanation behind our error classification. An error corresponds to a small region whose class-candidate label is different from the labels in its neighbourhood. Our error classification considers the sizes of the error region, and the properties of the class-candidates in the error region compared to those in their neighbourhood. Raw aspect data can contain a small error region sandwiched between two large aspect candidates. If the surrounding aspect-candidates have different class-candidates, the error could indicate a transition between two aspects. If the surrounding aspect-candidates have the same class candidate, we could have cases when some erroneous class-candidate c may occur as error regions exclusively in regions corresponding to a particular class c' of an exact aspect graph. This could be due to characteristics of the particular experimental setup – sensor response characteristics, sensor positioning, lighting arrangements, the imaging process and the feature detection mechanism. We term this phenomenon as an **association error**. For large error regions may contain a large proportion of small regions corresponding to the same class-candidate – an indication that they were all part of the same aspect. Such considerations form the basis of our classification of errors in AAGs into Types I – V.

For the 1-DOF case, the space of viewpoints is a circle. Three parameters characterize an aspect-candidate α – its corresponding class-candidate $CLASS\_CAND(\alpha)$, its angular extent θ_α (in terms of the number of sites it occupies), and its position in the aspect graph. In order to characterize an error region, we need to define size parameters. We define the following terms:

Θ_{min} : the minimum extent which an aspect-candidate
must have to be called a **Valid aspect-candidate**
Θ_p : the minimum extent which an aspect-candidate
must have to be called
a **Prominent aspect-candidate** ($\Theta_p \geq \Theta_{min}$)
$\mathcal{A}_c^g \overset{\triangle}{=} \{ \ \alpha \mid \theta_\alpha \geq \Theta_{min}, \text{where } \alpha \in \mathcal{A}_c\}$

A class-candidate present at a large number of sites is not likely to correspond to an error. We would like to identify those class-candidates which correspond to aspects in an error-free aspect graph – **valid class-candidates**. If a class-candidate is *not* a valid class-candidate, the total number of sites corresponding to it would be less than a threshold, N_{min}. Further, $\mathcal{A}_c^g = \phi$ *i.e.*, it has no valid aspect candidate corresponding to it. We clarify the significance of having two separate parameters N_{min} and Θ_{min}. N_{min} is a property of a class-candidate, whereas Θ_{min} is a property of an aspect-candidate. The presence of closely-spaced small fragments (aspect-candidates which are not valid *i.e.*, extents $< \Theta_{min}$) corresponding to the same class-candidate c is an indication of the presence of an aspect in the corresponding error-free aspect graph. Thus, c can still be a valid class-candidate, if its total number of occurrences is not less than N_{min}. We provide further explanations in the context of a Type II error, below.



(a) Type I Error

(b) A Region having Type II and III Errors (details in text)

(c) Types IV and V Errors

Fig. 4. A pictorial representation of some types of errors (Different shading patterns represent different aspect candidates): the 1-DOF case

We classify errors in aspect data into five categories. The classification is based on the width of the region $\mathcal{R}$ between two valid aspect-candidates, their properties, and the properties of the (non-valid) aspect-candidates present in region $\mathcal{R}$.

Type I Error A Type I error is present as a small transition region between two aspects of the corresponding error-free aspect graph. This corresponds to the "border effect" in [3]. In the raw aspect data, a Type I error can be described as follows: (Figure 4(a) illustrates an example of this type of error for the 1-DOF case.)

$$\theta_{a_i}, \theta_{a_j} \geq \Theta_{min}$$
$$CLASS\_CAND(a_i) \neq CLASS\_CAND(a_j), \text{ and}$$
$$\sum_{a_k} \theta_{a_k} < \Theta_{min}, \ \forall \ a_k \text{ between } a_i \text{ and } a_j$$

Here, a_i and a_j are two valid aspect-candidates belonging to different class-candidates such that there is a small region of width $\leq \Theta_{min}$ between them.

Type II Error First, we characterize the region in which a Type II error can lie. Type II errors can be present in a small isolated error region inside an aspect of the corresponding error-free aspect graph. Figure 4(b) shows an example of such a region In the raw aspect data, we observe a small region of width $\leq \Theta_{min}$ in between two valid aspect-candidates a_i and a_j. The two enclosing aspect-candidates a_i and a_j correspond to the same class-candidate (say, c'). Formally,

1. $\theta_{a_i}, \theta_{a_j} \geq \Theta_{min}$
2. $\sum_{a_k} \theta_{a_k} < \Theta_{min} \ \forall \ a_k$ between a_i and a_j
3. $CLASS\_CAND(a_i) = CLASS\_CAND(a_j) = c'$

We now define a Type II (association) error. For the small enclosed error region, those aspect-candidates constitute a Type II error, whose class-candidates observed are associated with the class-candidates of the enclosing aspect-candidates. Let us consider aspect-candidates a_k in the small enclosed region (described above). We may describe this as follows:

1. $\sum_{\alpha \in \mathcal{A}_c} \theta_\alpha < N_{min}$, where $CLASS\_CAND(a_k) = c$
2. $\mathcal{A}_c^g = \phi$, and
3. $P(c'_{actual} \mid c_{observed}) \geq$ a threshold T_1, $0 < T_1 < 1$

For an aspect-candidate a_k to constitute a Type II error, its corresponding class-candidate c should not be a valid class-candidate. In other words, the class-candidate c is observed at less than N_{min} sites in the tessellated viewing space. Further, the relative extent to which its corresponding valid aspect-candidates (if any) occupy the viewpoint space is also low. The most important condition for a Type II error is the third one – given that this class-candidate c is observed, the probability that it actually is another class-candidate c', is high. Our algorithm for AAG construction (Section III-B) describes the process of keeping estimates of association error probabilities.

A particular object in the model base may have a distinctive feature, or a combination of features that permit it to be clearly distinguished from other models. Additionally, this may be present over a small range of viewing angles. Conditions 2 and 3 taken together allow for such cases to be treated as aspects, and not as errors. For a distinctive feature, one would expect either the total number of sites at which the class-candidate is observed, to be greater than the N_{min} threshold; or the class-candidate to have most of its aspect-candidates with extent not less than Θ_{min}. Both the above conditions may also be simultaneously true. If none of these conditions hold, the distinctive aspect will be treated as an error condition.

Type III Error A small isolated error region inside a region corresponding to an aspect of an error-free aspect graph (as defined above) may contain either Type II or Type III errors. For a Type III error, the only requirement is $P(c'_{actual} \mid c_{observed}) \leq T_1$, *i.e.*, the probability of recognizing class candidate c as c' is minimal.

Type IV Error Due to noise, an aspect of an error-free AAG could be fragmented such that the raw aspect data has closely spaced aspect-candidates with the same class-candidate (Figures 4(c) shows an example of this situation). Such regions contain Type IV and Type V errors. Formally, Type IV and Type V errors can be present in a 'large' region between two valid aspect-candidates:

1. $\theta_{a_k} < \Theta_{min}$, $\forall\, a_k$ between a_i and a_j
2. $\sum_{a_k} \theta_{a_k} \geq \Theta_{min}$

In other words, no aspect-candidate a_k in the region is a valid aspect-candidate. However, the size of the entire region $\geq \Theta_{min}$. As aspect-candidate α constitutes a Type IV error if the following error conditions hold:

1. $\max_{\beta \in \mathcal{A}_c} \theta_\beta \geq \Theta_{min}$, where $CLASS\_CAND(\alpha) = c$
2. $\exists \gamma \in \mathcal{G}_\alpha : CLASS\_CAND(\gamma) = CLASS\_CAND(\alpha)$

Here, term $\mathcal{A}_c$ denotes the set of all aspect-candidates corresponding to class-candidate c. $\mathcal{G}_\alpha$ denotes the set of all aspect-candidates in the neighbourhood of aspect-candidate α.

This type of error considers those aspect-candidates which themselves do not have enough extent to be considered valid aspect-candidates. However, there are other valid aspect-candidates associated with this class-candidate, elsewhere. In Figure 4(c), aspect-candidate α shows an example of a Type IV error. A Type IV error indicates that the aspect-candidates with the same class-candidate may be part of a single aspect.

Type V Error Like Type IV errors, Type V errors too can be present in a 'large' region between two valid aspect-candidates (as defined above). An aspect-candidate α constitutes a Type V error if either

- $\forall \gamma \in \mathcal{G}_\alpha$, $CLASS\_CAND(\gamma) \neq CLASS\_CAND(\alpha)$, or
- $\exists \gamma \in \mathcal{G}_\alpha : CLASS\_CAND(\gamma) = CLASS\_CAND(\alpha)$, but

$\max_{\beta \in \mathcal{A}_c} \theta_\beta < \Theta_{min}$, where $CLASS\_CAND(\alpha) = c$

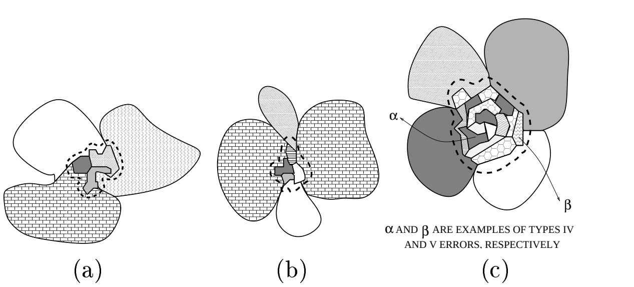


Fig. 5. A pictorial representation of some types of errors (Different shading patterns represent different aspect candidates): the 3-DOF case. (a) Type I error, (b) A region with Types II & III Errors, and (c) Types IV and V Errors. The thick dashed curve encloses aspect-candidates in the region of interest $\mathcal{R}$.

In Figure 4(c), aspect-candidate β illustrates an example of a Type V error. Type V errors are very difficult to correct since they give very little indication as to which aspect (in the corresponding error-free aspect graph) they might have come from.

For the 3-DOF case, an error region $\mathcal{R}$ consists of either a single small aspect-candidate, or a group of adjacent small aspect-candidates (the regions bounded by thick dashed curves in Figure 5.) The terms corresponding to θ_α, Θ_{min} and Θ_p are ω_α, Ω_{min} and Ω_p, respectively. We need to look at neighbourhood relationships of $\mathcal{R}$ with its adjacent aspect-candidates. Let $\mathcal{B}(\alpha)$ denote the set of sites on the *boundary* of aspect-candidate α. We recall the weighted graph representation of an AAG (Section II). In this context, we define the following terms:

$$W(\alpha_1, \alpha_2) \overset{\triangle}{=} \min(|S_1|, |S_2|), \ (|\cdot| \text{ denotes cardinality})$$
$$S_i = \{s \mid s \in NEIGH(s'), s' \in \mathcal{B}(\alpha_i)\} \cap \mathcal{B}(\alpha_{(i+1)\ mod\ 2}),$$
$$ADJ(\alpha) \overset{\triangle}{=} \{\beta \mid W(\alpha, \beta) \neq 0\}, (i.e., \text{nodes linked to } \alpha)$$

The five-category error classification for the 3-DOF case follows on lines similar to the 1-DOF case. This is based on the size of an aspect-candidate, its boundary relations with adjacent aspect-candidates (relative size of the common boundary), and relations with aspect-candidates in its neighbourhood. We only present examples here using figures (Figure 5(a), (b) and (c)). [32] presents a complete mathematical description for the 3-DOF case.

III. AAG Construction from Erroneous Raw Aspect Data

Let us define the terms 'smoothness' of model base data($\mathcal{S}(A)$), and the total model base error($\mathcal{E}(A)$) as follows:

$$\mathcal{S}(A) \overset{\triangle}{=} (1/M) \cdot \sum_{i=1}^{M} \sum_{j=1}^{G} d(c_{ij}, c_{ij+1})$$
$$\mathcal{E}(A) \overset{\triangle}{=} (1/M) \cdot \sum_{i=1}^{M} \sum_{j=1}^{G} d(c_{ij}, D_{ij})$$

where M is the number of objects in the model base and G is the number of tessellated viewpoints for the aspect data. Here, D_{ij} refers to the original raw aspect data at the jth site in model number i. c_{ij} is the corresponding class-candidate label assigned to it by the AAG construction algorithm. $d(\ ,\ )$ denotes the distance between two class-candidates. Depending on how class-candidates are defined, this distance may be the 'distance' between two graphs representing topological relations, or the Euclidean distance between two normalized feature vectors.

A. A New Evaluation Function for AAGs

As mentioned in Section II-A, an error-free AAG has aspects, whose corresponding extents are not too small in size. Further, an error-free AAG is characterized by piecewise continuity in the class-candidate labels (as in Figure 1(d)). Fidelity to the original data is another desirable feature. In view of these conflicting requirements, we define the 'Demerit Coefficient' $\eta(\mathcal{M}, i, \tau)$ for the AAG of model i in model base $\mathcal{M}$ as a non-convex sum of three terms. The third term penalizes too many prominent discontinuities in the aspect data. ρ_{ij} is defined to be 1 at if the distance between class-candidate labels at two adjacent sites $d(c_{ij}, c_{ij+1}) >$ threshold τ, and 0 otherwise. The first term, thus takes the piecewise smoothness criterion into account. The second term considers the fidelity between the original class-candidate at a site D_{ij} and the one assigned to the site by the algorithm c_{ij}. μ, ν and σ are constants.

$$\mu \sum_j (1 - \rho_{ij})\ d(c_{ij}, c_{ij+1}) + \nu \sum_j d(c_{ij}, D_{ij}) + \sigma \sum_j \rho_{ij}$$

(For our experimentation, we have chosen the constants μ, ν and σ such that all the three terms have the same order of magnitude.) We define the Demerit Coefficient for the set of AAGs for the entire model base as the average of the Demerit Coefficients of the M individual models' AAGs, using the same constants and threshold for each model:

$$\eta_{model\ base}(\mathcal{M}, \tau) \triangleq (1/M) \sum_{i=1}^{M} \eta(\mathcal{M}, i, \tau)$$

B. AAG Construction Algorithm

This section proposes a low order polynomial time-complexity algorithm for building an AAG from noisy aspect data in the 1-DOF case. Extensive experiments with two model bases (Section IV) shows that the output of the algorithm to have far lower values of the Demerit Coefficient compared to the noisy data input to it. Section III-B.5 briefly describes the corresponding algorithm for the 3-DOF case.

B.1. ASSOC_TABLE *& Association Probability Estimates.* The algorithm maintains estimates of the probability with which one class-candidate is observed as another. For example, one expects valid class-candidates to be observed correctly with a high probability. To keep estimates of these association values, the algorithm uses an $N_\mathcal{C} \times N_\mathcal{C}$ matrix, the $ASSOC\_TABLE$. The [i][j]th entry stores a relative count of the number of times class-candidate c_j is observed as c_i. We use the $ASSOC\_TABLE$ estimates for our work on robust 3-D object recognition (Section V).

We have a chicken-and-egg problem here. Proper $ASSOC\_TABLE$ estimates of $P(c'_{actual} \mid c_{observed})$ would be available only after the completion of the algorithm execution. However, the algorithm needs proper (qualitative) estimates of $P(c'_{actual} \mid c_{observed})$ for various processing operations. Hence at each stage, we need a good current estimate of $P(c'_{actual} \mid c_{observed})$ values. Therefore, we structure the phases in our algorithm in such a way that we fulfill this requirement.

Figure 1(c) shows the AAG of a polyhedral object. Figure 6 shows an example of this AAG corrupted with errors. With reference to this example, we discuss the three phases of our algorithm as follows:

B.2. Algorithm Phase I. Phase I of our algorithm is primarily concerned with *identification of valid class-candidates*. The algorithm clusters the raw aspect data into aspect-candidates using a 1-D version of Horn's sequential labeling algorithm ([33]). Prominent aspect-candidates correspond to valid class-candidates. We initialize the $ASSOC\_TABLE[i][i]$ entries for such class-candidates c with the sum of the extents of all prominent aspect-candidates corresponding to c. Hence for such an AAG, this phase is expected to identify most of the valid class-candidates for the given model base. Phase I does not remove any errors from the raw aspect data. This phase requires one pass through the raw aspect data at each site for each model in the model base.

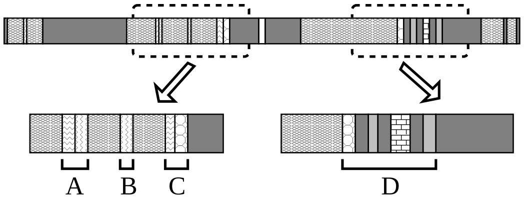


Fig. 6. The AAG of Figure 1(c) corrupted with errors. Some error regions are shown marked as A, B, C and D (Details in text).

B.3. Algorithm Phase II. Phase II is primarily concerned with identification of prominent aspect-candidates after removing interspersed errors. In this phase, we consider small isolated regions between two valid aspect-candidates. We specifically consider the case when the two aspect-candidates correspond to the same class-candidate. In Figure 6 for example, Phase II considers regions marked A and B. We handle Type II and Type III errors in this phase. In this phase, we consider pairs of **proximal valid aspect-candidates**, with the same class-candidate, say c. We define a pair of valid aspect-candidates (a_i, a_j) as proximal valid aspect-candidates if

$$\theta_{a_k} \in (0, \Theta_{min})\ \forall\ a_k \text{ between } a_i\ \&\ a_j \text{ in the direction of traversal of the aspect-candidate list.}$$

For each pair of proximal valid aspect-candidates with the same class-candidate separated by a gap of width $\leq \Theta_{min}$, we integrate both the valid aspect-candidates and those in between them, into one. For the class-candidates corresponding to the valid aspect-candidates, we increment the $ASSOC\_TABLE[i][i]$ count by the size of the aspect-candidates. For the aspect-candidates a_k between a_i and a_j, the algorithm updates the entry corresponding to $CLASS\_CAND(a_k)$ being observed as $CLASS\_CAND(a_i)$ by the size of aspect-candidate a_k. Phase II of our algorithm removes Type II and Type III errors. (Whether the isolated error removed is a Type II or Type III error will be clear from the $ASSOC\_TABLE$ conditional probabilities). Our experimentation shows that these estimates of association errors made in Phase II are reasonably correct.

B.4. Algorithm Phase III. The third phase of our algorithm handles the rest of the raw aspect data. There are two passes through Phase III. The first is a logical pass, done in order to get further (better) estimates for $ASSOC\_TABLE$ entries. Figure 7 gives an outline of the steps performed in this phase.

In Phase III again, we consider pairs of proximal valid aspect-candidates a_i and a_j. Depending on the gap between a_i and a_j in the direction of traversal, we consider two cases:

Case 1: $\mathbf{gap}(a_i,a_j) < \Theta_{min}$

In such a case, we cannot have any valid aspect-candidate in between a_i and a_j (such as region C in Figure 6). For this reason, we obtain the minimum square-error decision boundary for the region of gap δ ($\delta < \Theta_{min}$). We now take the minimum-error decision boundary considering the association information collected thus far. Of the two decision boundaries calculated so far, the algorithm takes the one with the minimum error. This process has quadratic ($O(\delta^2)$) time complexity as one has to

ALGORITHM: Phase_III

```
(* Two passes are made through Phase_III. *)
(* After the first(LOGICAL) pass, keep ASSOC_TABLE  *)
(* unchanged. Then, make the second (ACTUAL) pass.  *)
FOR EACH model IN model_base REPEAT the following step
 FOR EACH proximal valid aspect_candidate pair $\{a_i, a_j\}$ DO
  (* consider region from $a_i$ to $a_j$ *)
   IF gap($a_i$, $a_j$) < $\Theta_{min}$ THEN
    BEGIN (* create single decision boundary *)
     a.  get the minimum-error decision boundary from
         raw aspect data; and considering
         ASSOC_TABLE info for each class_candidate
     b.  of the two, select the one which incurs min sq error
     c.  integrate aspect_candidates from $a_i$ till just before
         before the decision boundary, into $a_i$; and those
         between the decision boundary & $a_j$, into $a_j$;
    END (* create single decision boundary *)
   ELSE
    BEGIN (* handle large region *)
     1.  create normalized freq histogram for class_candidates
     2.  (* — First Rule —  *)
         IF max(histogram) $\geq T_2$ AND
         the corresponding class_candidate
         == CLASS_CAND($a_i$) == CLASS_CAND($a_j$) THEN
          integrate all aspect_candidates from $a_i$ to $a_j$ into $a_i$;
          RETURN;
     3.  (* — Second Rule —  *)
         FOR EACH histogram entry $\geq T_3$ DO
          IF the corresponding class_candidate == CLASS_CAND
          of any/both surrounding aspect_candidates THEN
           integrate aspect_candidates in between to that/those
           surrounding aspect_candidate(s); RETURN;
     4.  *( — Third Rule —  *)
         FOR EACH class_candidate $c$ with histogram entry $\geq T_4$ DO
          FOR EACH pair of aspect_candidates $(a_k, a_l)$ between
          $a_i$ and $a_j$, with class_candidate $c$
           IF gap($a_k$,$a_l$) $\leq$ THEN
            integrate aspect_candidate from $a_k$ to $a_l$ into $a_k$;
           ELSE
           IF current_size($a_k$) < $\Theta_{min}$ THEN UNDO changes to $a_k$;
     5.  FOR EACH remaining region between $a_i$ and $a_j$ DO
         create single decision boundary (* as in steps a - d *)
    END  (* handle large region *)
```

Fig. 7. The Algorithm: Phase III

calculate the error for the entire region considering all possible positions of the decision boundary. We can thus remove Type I errors.

Case 2: gap(a_i,a_j) $\geq \Theta_{min}$
Region D in Figure 6 shows an example of this situation. Here, we have a trade-off between the optimality of the algorithm in terms of one parameter affecting the optimality in terms of another. Taking a single decision boundary for a large region may not just be time-consuming, but also may not be optimal since there may be one or more aspects in the region under consideration. To handle these cases, we try heuristic rules 1 – 3 (details in Figure 7). These are based on the relative frequency of occurrence for different class-candidates in the region under consideration, and their neighbourhood relations. If one of them succeeds, we exit and consider the next pair of proximal valid aspect-candidates. Otherwise, we try the next one. These rules aim at removing Type IV errors, and some Type V errors. After trying out region growing, we may be still be be left with regions between a_i and a_j unaccounted for. We get a single minimum square-error decision boundary for all such regions, as in Case I above.

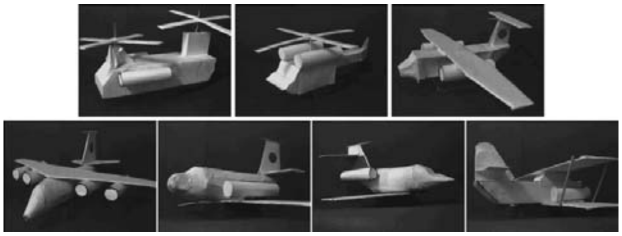


Fig. 8. Model Base I: The objects (in row major order) are heli_1, heli_2, plane_1, plane_2, plane_3, plane_4, and biplane.

B.5. Algorithm for 3-DOF AAG Generation. Our 3-phase algorithm for the 3-DOF case follows exactly on the same lines as the 1-DOF algorithm – [32] contains a detailed description of the same. As mentioned in Section II, we use an icosahedron-based geodesic tessellation (Figure 2(b), (c)) of the viewing space for the 3-DOF case. We formulate an algorithm whose time complexity is linear in the size of the AAG. We use the weighted graph representation of an AAG (Section II), and the related definitions (Section II-A). The algorithm considers the size of different aspect-candidates, and their relationships with neighbouring aspect-candidates. For the first phase, we formulate a raster-scan method to cluster the raw aspect data in a spherical array (Section II) into aspect-candidates, based on the similarity between neighbouring views, in terms of the features observed (Section III). The algorithm then considers regions $\mathcal{R}$ (Section II-A) and looks at their neighbouring regions. The second phase removes Type II and III errors, while the third phase tackles Type IV and V errors. The erroneous aspect-candidates are identified on the basis of their relative size, the weight of their links with adjacent nodes in the graph representation of the AAG, and the similarity of their class-candidates with those of their neighbouring nodes.

IV. Experimental Results and Discussion

This section presents experimental results with two model bases for the 1-DOF case. We also present results of experiments for the 3-DOF case, with synthetic data. Though our formulation is independent of any specific feature set, we use *simple image features* for our experimentation. We base our choice on the following issues.

It is difficult to extract topological information from images [34]. Nearly all object recognition systems which use aspect graphs, use simple image features *e.g.*, Hutchinson and Kak [2] (faces of a polyhedral object which are segmented from range images), Ikeuchi and Kanade [28], [35] use a binary feature vector based on the visibility of faces, Gremban and Ikeuchi [3] use information about the number and other features of specularities, and Chakravarty and Freeman [1] use a feature vector, whose components are the number of junctions of each of the 8 types. PREMIO [18], [19] uses topological information. However, it suffers from the overhead of having to match two hierarchical topological descriptions. Dickinson and co-workers [4], [36] use an Aspect Prediction Graph – their method has two important shortcomings. First, volumetric primitives are associated with a high feature extraction cost. In addition to the tracking overhead, the system performs many redundant image processing operations – in many successive images, when the aspect of the object remains unchanged.

A. The 1-DOF Case: Experimental Results

Our experimental setup has a camera connected to a MATROX Image Processing Card and a stepper motor-controlled

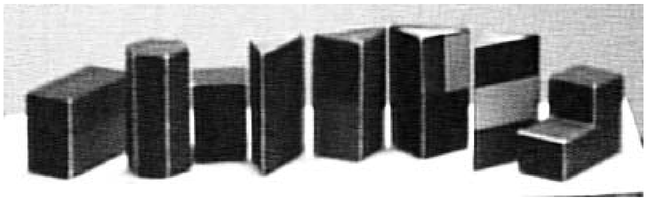


Fig. 9. Model Base II: The objects (from left) are O_1, O_2, O_3, O_4, O_5, O_6, O_7 and O_8, respectively.

turntable. The turntable moves by 200 steps to complete a 360 degree movement. We have experimented extensively with two object sets as model bases. Some details of the object sets are as follows:

1. **Model Base I: 7 Aircraft Models**
We use as features, the number of horizontal lines ($\langle h \rangle$), the number of vertical lines ($\langle v \rangle$), and the number of circles ($\langle c \rangle$). We represent a class-candidate as $\langle hvc \rangle$. We have chosen this relatively feature-rich model base to demonstrate the effectiveness of our system on raw aspect data with very low smoothness in the raw aspect data. Figure 8 shows the objects in this model base.
2. **Model Base II: 8 Polyhedral Objects**
We use as features, the number of horizontal lines ($\langle h \rangle$), the number of vertical lines ($\langle v \rangle$), and the number of non-background segmented regions in an image($\langle r \rangle$). We represent a class-candidate as $\langle hvr \rangle$. The raw aspect data for this model base has higher smoothness compared to the aircraft models. We have chosen this model base to compare the results of our system with those on the other model base. Figure 9 shows the objects in this model base.

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

Let the term 'Input Smoothness' ($\mathcal{S}(I)$) refer to the smoothness expression (Section III). for the raw aspect data. Thus, c_{ij} is D_{ij} here i.e., the raw aspect data item for the ith model at site number j. Similarly, we use the term 'Output Smoothness' ($\mathcal{S}(O)$) to refer to the smoothness expression for the output of the aspect graph construction algorithm. Thus, c_{ij} refers to the class-candidate label assigned to the jth site in the ith model by the algorithm.

The aspect data for Model Base I (aircraft models) has a very high values of the Demerit Coefficient $\eta_{model\ base}$ and $\mathcal{S}(I)$ as compared to the aspect data for the other. Hence, we first present results of 100 experiments with the first model base. Each experiment considers a set of raw aspect data from each object in the model base. Then, we compare some figures with those of Model Base II (polyhedral objects).

Output of the Aspect Graph Construction Algorithm:
Figure 10 shows a comparison of the raw aspect data and the output of our algorithm, for one instance of the aspect data for object plane_2 in Model Base I. A visual inspection of the lower graph shows that the aspects produced are prominent and not too large in number, the graph is piecewise smooth and at the same time, fidelity to the original data is high. Figure 11 shows an example for Model Base II.

Input and Output Smoothness:
Figure 12(a) shows a comparison of the input and output smoothness for 100 sets of aspect data for the aircraft model base. The mean smoothness values for the input and output data are $\mathcal{S}(I) = 138.85$ and $\mathcal{S}(O) = 20.99$, while the variances are 4.51 and 0.74 respectively. This clearly shows that $\mathcal{S}$ values have greatly decreased. Even though the raw aspect data has a large variation in $\mathcal{S}$ values, the variation in in $\mathcal{S}$ values for the output data is very small.

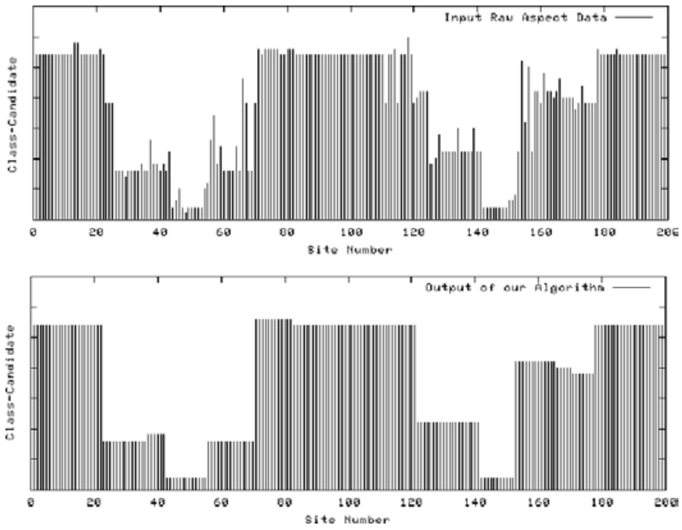


Fig. 10. Raw aspect data and the output of our algorithm: plane_2, Model Base I. On the y-axis, each class-candidate is represented by an index. Different heights represent different class-candidates. The tesselated viewing space has 200 sites.

Total Model Base Error:
$\mathcal{E}$ is a measure of fidelity of the output data to the input raw aspect data. Figure 12(b) shows the variation in $\mathcal{E}$ with the input smoothness for the 100 data sets.

Number of Aspects
Figure 13(a) shows the number of aspects obtained as a result of applying the algorithm on 100 instances of raw data for the model base. The figure shows the variation in the number of aspects with the input 'smoothness' for the 100 data sets.

Demerit Coefficients for Input and Output Aspect Data:
Figure 13(b) shows the variation of the Demerit Coefficient for the input aspect data, with the Demerit Coefficient for the output of the AAG construction algorithm for the 100 data sets. Our AAG construction algorithm greatly reduces the Demerit Coefficient. Further, the variation in the Demerit Coefficient for the output data is quite less compared that for the input raw aspect data.

Single decision boundary (percentage of sites):
The only part of our algorithm which has quadratic time complexity is where a single decision boundary has to be taken over a set of adjacent sites. The rest of it runs in linear time. Figure 14(a) shows the percentage of sites where a single decision boundary had to be taken, for 100 sets of aspect data. This is quite low (mean=29.47%, variance=1.85), even for aspect data with high $\mathcal{S}(I)$ values. This shows the efficiency of our algorithm, since one of the desirable characteristics of an AAG construction algorithm is that it should be fast i.e., have low-order polynomial time complexity.

Correctness of Phase II ASSOC_TABLE *estimates:*
The *ASSOC_TABLE* maintains estimates of the probability with which one class-candidate is observed as another(Section III-B). If *ASSOC_TABLE*$[i][j]$ remains above a threshold after Phase II as well as after Phase III (or equivalently, below it after both phases) – we refer to this as 'correctness'. Figure 14(a) shows the variation in percentage correctness of Phase II estimates, with the input 'smoothness' for 100 data

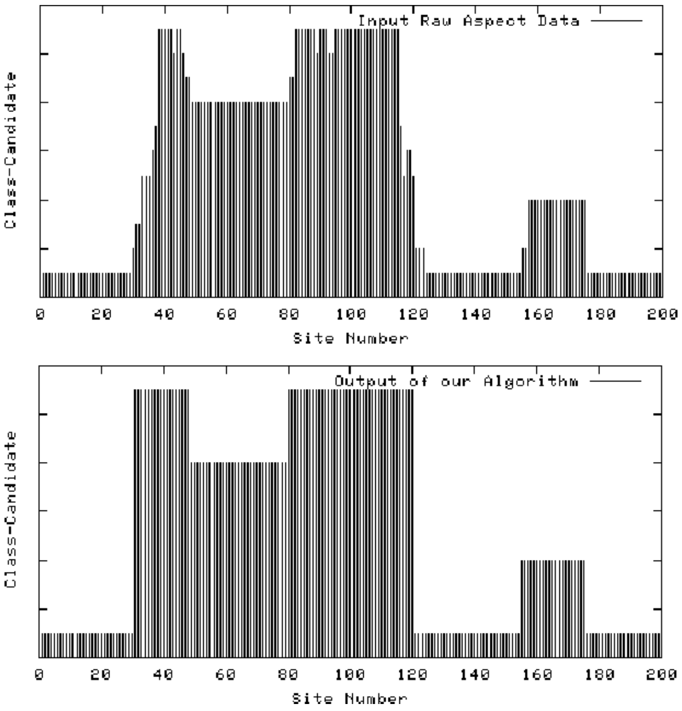


Fig. 11. Raw aspect data and the output of our algorithm: O_6, Model Base II On the y-axis, each class-candidate is represented by an index. Different heights represent different class-candidates.

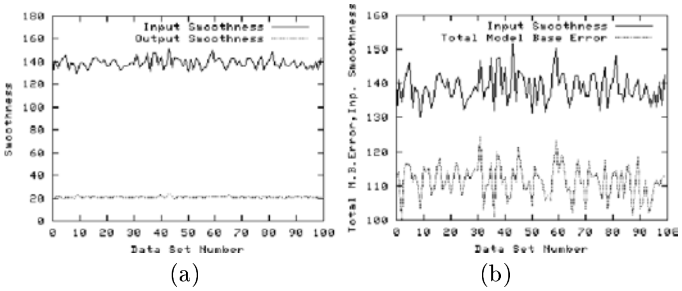


Fig. 12. (a) Variation in input 'smoothness' with the output 'smoothness', and (b) Variation in the total model base error with the input 'smoothness', both for 100 data sets

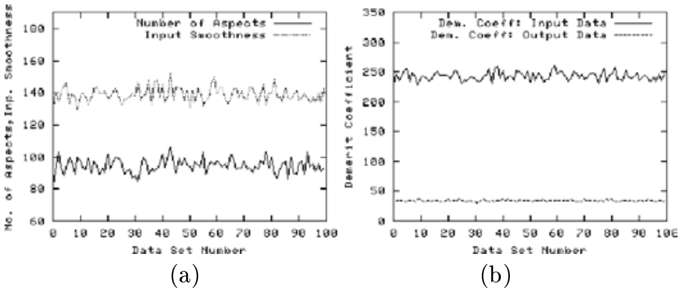


Fig. 13. (a) Variation in the number of aspects with the input 'smoothness', and (b) Variation in the Demerit Coefficient for the input raw aspect data, with the Demerit Coefficient for the output of the input raw aspect data: for 100 data sets

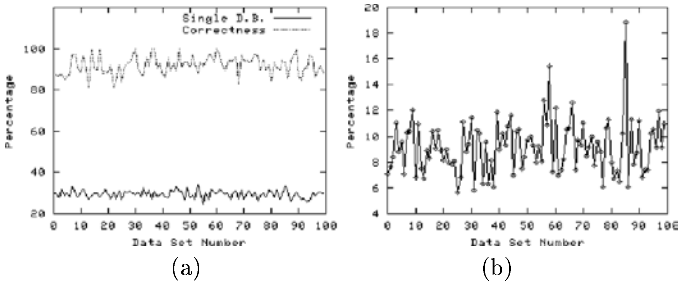


Fig. 14. (a) Percentage of sites where a single decision boundary had to be taken, and correctness in Phase II estimates in the $ASSOC\_TABLE$ (b) Percentage reduction in total model base error with $ASSOC\_TABLE$ data

	Model Base I		**Model Base II**	
PARAMETERS	*Mean*	*Var.*	*Mean*	*Var.*
$\mathcal{S}(I)$	138.85	4.51	43.99	0.47
$\mathcal{S}(O)$	20.99	0.74	18.93	0.34
$\mathcal{E}(A)$	111.49	5.01	27.97	1.35
# aspects	94.51	4.55	80.5	1.12
η (input data)	242.84	7.32	67.69	2.59
η (output data)	34.13	1.17	28.99	0.48
O. D. B.	29.47%	1.85	7.47%	0.41
Correctness	91.91%	4.46	94.74%	3.72
Err. redn.	9.18%	2.08	4.85%	0.37

TABLE I
Summary of AAG construction algorithm performance parameters for the two model bases: the 1-DOF case. 'O.D.B.' denotes the percentage of the total AAG size where an optimal decision boundary needed to be taken (Details in Section III-B.4). The term 'Correctness' denotes the correctness of Phase II $ASSOC\_TABLE$ estimates 'Err. redn.' denotes the percentage reduction in the model base error if $ASSOC\_TABLE$ estimates are used.

sets.

Percentage reduction in model base error:
If $P(c_{actual} \mid c'_{observed})$ (Section II-A) is above a particular threshold, we use this fact in order to obtain a minimum-error decision boundary (Section III-B.4), Our experiments show that this reduces the total model base error, $\mathcal{E}$. The model base error $\mathcal{E}(A)$ is reduced if one used the association data from the $ASSOC\_TABLE$. Figure 14(b) shows the percentage reduction in error for 100 instances of model base data.

Comparison of performance factors on two model bases:
Table I shows the comparison between the two model bases. The figures for Model Base I are for 100 experiments, whereas those for Model Base II are for 4. Though the feature detectors used for the two model bases are different, the range of values taken by the feature-classes for the two model bases are comparable.

The raw aspect data from objects in the first model base has more errors than that from the second. In spite of this, the output smoothness obtained in both cases is comparable, and small, especially for the first case. Since the $\mathcal{S}(I)$ for the first model base is large, its model base error $\mathcal{E}$ is large compared to the corresponding data for the second model base. The first model base is such that the number of aspects is larger than that obtained for the second model base. Due to the large $\mathcal{S}(I)$ values for the first model base, the variance in the number of aspects is larger in the first case. The Demerit Coefficient for the output data is much less than that for the input data. Since the second

	Model Base I		**Model Base II**	
	Mean	*Variance*	*Mean*	*Variance*
Type I	1.03%	0.19%	3.97%	0.03%
Type II	1.19%	0.27%	0.19%	0.00%
Type III	1.87%	1.26%	0.50%	0.00%
Type IV	11.60%	1.45%	0.46%	0.02%
Type V	19.38%	0.62%	2.09%	0.02%

TABLE II
A comparison of the relative extents of different types of errors in AAGs in terms of the percentage of the total number of sites ('Size'), and the relative percentages of the errors ('Relative Size') (Section II)

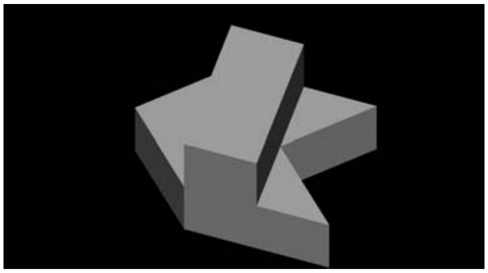


Fig. 15. The CAD model of a polyhedral object, for experimentation with the 3-DOF case

model base data has less feature detection errors, the percentage of sites in the AAG where a single decision boundary had to be taken is less than the corresponding value for the first model base. The correctness of Phase II *ASSOC_TABLE* estimates is more for the second model base, since the first one has more errors. Hence, the variance in the correctness values is more for the first model base compared to the second. Both cases show a reduction in model base error if the *ASSOC_TABLE* estimates are used to determine the correct class-candidate at a site.

A Comparison of Different Types of Errors

Table II presents a relative comparison of the different categories of errors (Section II) for the two model bases. We list the mean and variance for each type of error for the 100 data sets in the first model base, and the 4 data sets for the second. *We emphasize here that the relative importance of different types of errors may vary across different model bases, and feature sets.* For a particular setup and a set of models, a particular error type may be more prominent compared to others. For example, while a Type V error is more common for the first model base, a Type I error is more common for the second. *However, from the point of view of robust object recognition, it is equally important to account for all types of errors.*

B. The 3-DOF Case: Experimental Results

For our experiments with the 3-DOF case, we have used CAD data for a polyhedral object(Figure 15) to construct it's AAG. We use as features:

1. The number of corners with the number of incident edges as 2, 3, 4 and 5, respectively, and
2. The number of faces of the object visible in the given view

Thus, a 5-dimensional vector of the above features represents a class. For the tessellation of the viewing space around the object, we choose Q=14 as the frequency of geodesic division. Thus, there are 1962 sites in the viewpoint space.

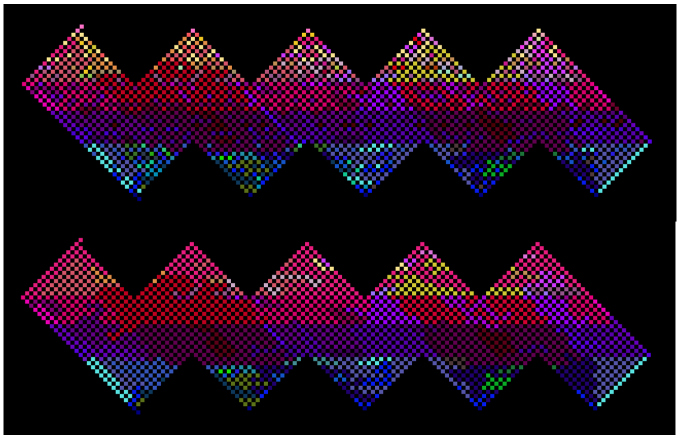


Fig. 16. Raw aspect data with 30% noise, and the output of our algorithm for the 3-DOF case (Spherical array representation). Different colours represent different class-candidates.

In order to generate the experimental data sets, we have perturbed the aspect data with different amounts of random noise. We randomly chose k% of the total 1962 sites for changing their class-candidate data. Each feature at such a site was replaced by a random number (in the range of the observed values of the feature in the aspect data obtained from the CAD model), We obtained 8 different data sets by taking $k = 0$, 1, 5, 10, 15, 20, 25, and 30, respectively.

Figure 16 shows a comparison of the raw aspect data with 30% noise, and the output of our algorithm, in the spherical array representation. The lower figure shows prominent aspects created by our algorithm. Table III shows the results of running our algorithm on these data sets. The first column describes the results obtained by processing the aspect data directly obtained from the CAD model. The $\eta(I)$ value corresponding to this 0% error column represents the quality of the original aspect data with respect to the desirable criteria mentioned in Section III-A. Examining the data in the table, we make the following observations. As we increase the amount of added noise to the aspect data, $\mathcal{S}(I)$ and $\eta(I)$ obviously increase. However, the increase in the $\mathcal{S}(O)$ and $\eta(O)$ values is considerably less. In fact, the values of $\eta(O)$ show that there has been considerable improvement in the quality of the output data in spite of the noise added (e.g., the results for 20%, 25% and 25% noise.) In spite of varying the amount of noise added to the raw aspect data from 1 to 30%, the number of aspects does not vary greatly. The variation in the number of aspects obtained can be attributed to the fact that the sites for the injection of noise have been randomly chosen. The percentage of sites where an optimal decision boundary would need to be taken is only a small percentage (under 6%) of the total number of sites. To avoid the exponential time complexity of taking an optimal decision boundary, we perform the following compromise – We consider the errors induced when the region $\mathcal{R}$ is assigned the class-candidates corresponding to each adjacent aspect-candidate. We also consider the error considering the *ASSOC_TABLE* estimates, and take the lower of the two. The correctness of Phase II estimates of association pairs is high, above 86%. The table also shows figures for the reduction in the total model base error due to the use of *ASSOC_TABLE* estimates.

V. Application to Robust 3-D Object Recognition

Our work on isolated 3-D object recognition ([5], [6], [7]) uses the output of our AAG construction algorithm, and the

	NOISE							
	0%	1%	5%	10%	15%	20%	25%	30%
$\mathcal{S}(I)$	1405.71	1578.89	2281.07	2945.86	3338.91	3629.05	4347.85	4344.85
$\mathcal{S}(O)$	1183.51	1203.18	1208.42	1306.09	1238.04	1605.85	1619.39	1817.89
$\mathcal{E}(A)$	329.31	411.99	851.48	1370.09	1661.98	1893.88	2385.50	2204.72
# aspects	59	59	57	56	57	61	63	60
$\eta(I)$	2762.44	2979.42	3857.72	4621.23	5090.32	5419.53	6212.21	6239.06
$\eta(O)$	2427.16	2469.35	2457.10	2553.13	2504.53	2846.96	2962.20	3214.79
O.D.B.	4.08 %	5.05 %	4.18 %	5.86 %	5.40 %	3.36 %	5.81 %	5.35 %
Correctness	97.78 %	88.68 %	90.79 %	88.31 %	87.84 %	86.25 %	95.00 %	86.42 %
Err. redn.	0.00 %	0.18 %	0.00 %	0.01 %	0.70 %	0.25 %	0.00 %	0.60 %

TABLE III

Summary of AAG construction algorithm performance parameters for aspect data perturbed by different amounts of noise: the 3-DOF case. The parameters are the same as in Table I.

ASSOC_TABLE estimates. To the best of our knowledge, our work on 3-D object recognition is the only one that accounts for feature detection errors at both stages – model base construction, and the object recognition.

The above papers present a probabilistic framework for recognition and planning. Our hierarchical knowledge representation scheme encodes domain knowledge and relations between objects(O_i), their aspects, classes and feature data. We use the *ASSOC_TABLE* estimates to compute the probability with which a feature appears as another. A given view of an object corresponds to a class. Given a view of a 3-D object, our probability calculation scheme (using *ASSOC_TABLE* estimates) maps the features for the view, onto the correct class. It then maps class probabilities onto object probabilities. If the probability of no object is above a predetermined threshold, the next view planning algorithm plans the best view to distinguish between the competing aspects at a stage. We demonstrate that our recognition strategy works correctly even under processing and memory constraints due to the incremental reactive planning strategy.

Figure 17 shows two examples of experiments with objects from the aircraft model base. For Figure 17, the initial class could have come from 10 aspects. In Figure 17(b), the shadow of the left wing on the fuselage of the aircraft, the feature detector detects 4 vertical lines instead of 3, the correct number. (This line is not shown superimposed, for clarity.) *ASSOC_TABLE* estimates enable the system to recover from this feature detection error.

	Model Base II		**Model Base I**	
	Raw	*AAG*	*Raw*	*AAG*
Feature det. errors	24	26	77	40
Cases of Recovery	–	6	–	16
Cases of Failure	24	20	77	24
Im. Proc. Opns. (avg.)	2.322	2.242	2.368	1.784
Search tree nodes (avg.)	450.373	136.468	264.684	24.431

TABLE IV

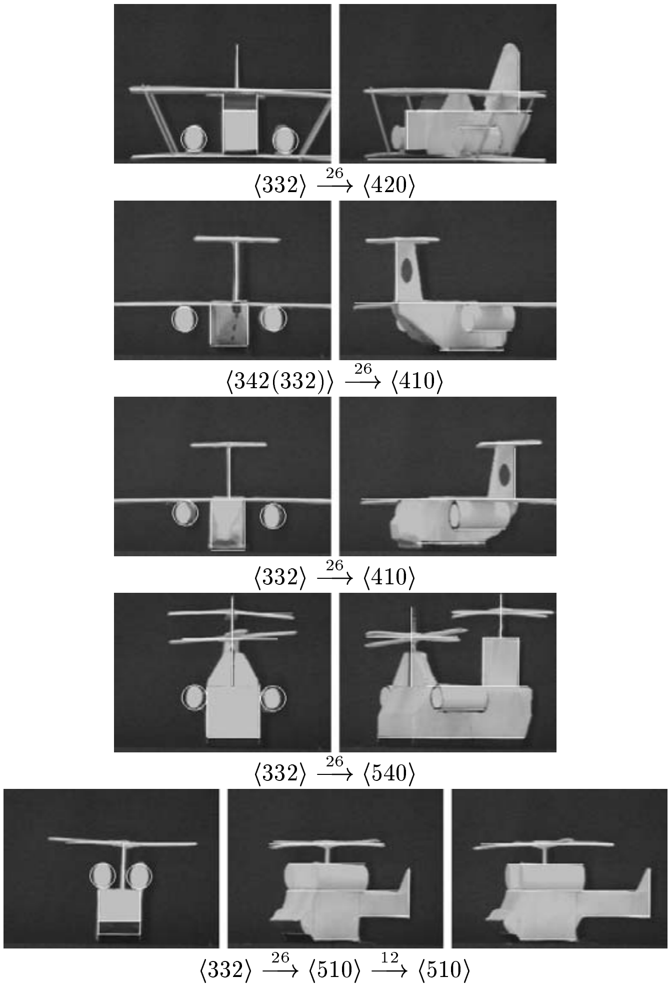
A comparison of our object recognition algorithm (citations in text) on raw aspect data ('Raw') and the output of our AAG construction algorithm ('AAG'), for the two model bases (Model Base II: Polyhedral Objects, and Model Base I: Aircraft Models)

A. The Effect of Errors on Recognition Performance

An AAG consisting of raw aspect data has a very large number of aspects associated with it. Many of these correspond to feature detection errors, and are very small in size. Hence, the degree of ambiguity associated with a view will be very large, as compared with the case when an error-free AAG is used. Errors due to noise often do not occur at a fixed position in an AAG. Hence in a recognition experiment, what may be actually observed at a viewpoint may not correspond to what it was when the raw aspect data was collected. Our algorithms not only aim at reconstructing the corresponding error-free AAG, they also store estimates of feature detection errors. We use these estimates to recover from cases of feature detection errors in our object recognition algorithm [5], [6], [7]. Table IV presents the results of experiments comparing the recognition performance with raw aspect data, and the output of our 1-DOF AAG construction algorithm. For the 83 experiments with the polyhedral objects, the number of cases of feature detection errors is nearly comparable (24 and 26, respectively for the raw aspect data, and the AAG constructed using our algorithm). However, there are 6 cases of recovery from feature detection errors in the latter, which is not so for the raw aspect data. The average number of image processing operations is more for the raw aspect data (2.322), as compared with 2.242 for the latter. The average number of search tree nodes for the raw aspect data is about 3 times the corresponding value for the output of our AAG construction algorithm. has a high value of Demerit Coefficient $\eta_{model\ base}$ and $\mathcal{S}(I)$. The number of feature detection errors is significantly higher for the raw aspect data – 77, as compared with 40 for the latter. Further, there are 16 instances of recovery from feature detection errors with the latter, using the *ASSOC_TABLE* estimates to advantage. The average number of image processing operations for the two types of inputs are 2.368 and 1.784, respectively. The average number of search tree nodes is significantly higher for the raw aspect data – nearly 11 times the corresponding value for the output of our AAG construction algorithm.

VI. Conclusion

This paper presents an integrated approach for handling feature detection errors for use in robust active 3-D object recognition. First, we present a new algorithm for AAG construction with noisy feature detectors. We propose an evaluation function for comparing the output of different AAG construction algorithms. We characterize the suitability of a feature detector for aspect graphs – in terms of the entire setup, and the given model base. We consider both the 1-DOF as well as the 3-DOF case. No related work has addressed these issues. We present the results of extensive experimentation on a reasonably complex experimental set, in support of our strategy. Since it is not possible to prevent feature detection errors, our strategy



$\langle 332\rangle \xrightarrow{26} \langle 420\rangle$

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

$\langle 332\rangle \xrightarrow{26} \langle 410\rangle$

$\langle 332\rangle \xrightarrow{26} \langle 540\rangle$

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

Fig. 17. Some experiments with objects in the aircraft model base, with the same initial class. The numbers above the arrows denote the number of turntable steps. (The figure in parenthesis shows an example of recovery from feature detection errors)

can be used for efficient and robust 3-D object recognition.

VII. Acknowledgements

We are thankful to the reviewers for their constructive suggestions, which have greatly helped in improving the quality of this paper.

References

[1] I. Chakravarty and H. Freeman, "Characteristic Views as a Basis for Three Dimensional Object Recognition," in *Proc. SPIE Conference on Robot Vision*, 1982, vol. 336, pp. 37 – 45.

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

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

[4] 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 (ECCV)*, 1994, pp. 3 – 14.

[5] S. Dutta Roy, S. Chaudhury, and S. Banerjee, "Isolated 3-D Object Recognition through Next View Planning," in *Intelligent Robotic Systems*, M. Vidyasagar, Ed., pp. 494 – 501. Tata McGraw-Hill Publishing Company Limited, 1998.

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

[7] 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).

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

[9] J. J. Koenderink and A. J. van Doorn, "The Singularities of the Visual Mapping," *Biological Cybernetics*, vol. 24, pp. 51 – 59, 1976.

[10] J. H. Rieger, "On the Complexity and Computation of View Graphs of Piecewise Smooth Algebraic Surfaces," *Phil. Trans. R. Soc. London. A*, vol. 354, pp. 1899 – 1940, 1996.

[11] D. Eggert and K. Bowyer, "Computing the Perspective Projection Aspect Graph of Solids of Revolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 2, pp. 109 – 128, February 1993.

[12] G. Castore and C. Crawford, "From Solid Models to Robot Vision," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 1988, pp. 90 – 92.

[13] I. Shimshoni and J. Ponce, "Finite-Resolution Aspect Graphs of Polyhedral Objects," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 315 – 327, April 1997.

[14] J. H. Stewman and K. W. Bowyer, "Direct Construction of the Perspective Projection Aspect Graph of Convex Polyhedra," *Computer Vision, Graphics and Image Processing*, vol. 51, no. 1, pp. 20 – 37, July 1990.

[15] J. Stewman and K. Bowyer, "Creating the Perspective Projection Aspect Graph of Polyhedral Objects," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 1988, pp. 494 – 500.

[16] K. Bowyer, M. Sallam, D. Eggert, and J. Stewman, "Computing the Generalized Aspect Graph for Objects with Moving Parts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 6, pp. 605 – 610, June 1993.

[17] H. Lu, L. G. Shapiro, and O. I. Camps, "A Relational Pyramid Approach to View Class Determination," in *Proc. IEEE Workshop on Interpretation of 3D Scenes*, 1989, pp. 177 – 183.

[18] O. I. Camps, L. G. Shapiro, and R. M. Haralick, "PREMIO: An Overview," in *Proc. IEEE International Workshop on Directions in Automated CAD Based Vision*, 1991, pp. 11 – 21.

[19] O. I. Camps, *PREMIO: The Use of Prediction in a CAD-Model-Based Vision System*, Ph.D. thesis, Department of Electrical Engineering, University of Washington, 1992.

[20] D. W. Eggert, K. W. Bowyer, C. R. Dyer, H. I. Christensen, and D. B. Goldgof, "The Scale Space Aspect Graph," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1992.

[21] D. W. Eggert, K. W. Bowyer, C. R. Dyer, H. I. Christensen, and D. B. Goldgof, "The Scale Space Aspect Graph," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1114 – 1130, November 1993.

[22] S. Pae and J. Ponce, "Toward A Scale-Space Aspect Graph: Solids of Revolution," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999, pp. II: 196 – 201.

[23] D. Wilkes, S. J. Dickinson, and J. K. Tsotsos, "A Quantitative Analysis of View Degeneracy and its Use for Active Focal Length Control," in *Proc. IEEE International Conference on Computer Vision (ICCV)*, 1995, pp. 938 – 944.

[24] S. J. Dickinson, D. Wilkes, and J. K. Tsotsos, "A Computational Model of View Degeneracy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 8, pp. 673 – 689, August 1999.

[25] D. Weinshall and M. Werman, "On View Likelihood and Stability," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 97 – 109, February 1997.

[26] P. J. Flynn and A. K. Jain, "CAD-Based Computer Vision: From CAD Models to Relational Graphs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 2, pp. 114 – 132, February 1991.

[27] M. R. Korn and C. R. Dyer, "3-D Multiview Object Representations for Model-Based Object Recognition," *Pattern Recognition*, vol. 20, no. 1, pp. 91 – 103, 1987.

[28] K. Ikeuchi and T. Kanade, "Automatic Generation of Object Recognition Programs," *Proceedings of the IEEE*, vol. 76, no. 8, pp. 1016 – 1035, August 1988.

[29] C. H. Chen and A. C. Kak, "A Robot Vision System for Recognition 3-D Objects in Low-Order Polynomial Time," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 19, no. 6, pp. 1135 – 1563, November/December 1989.

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

[31] K. D. Gremban and K. Ikeuchi, "Appearance-Based Vision and the Automatic Generation of Object Recognition Programs," in *Three-Dimensional Object Recognition Systems*, A. K. Jain and P. J. Flynn, Eds., pp. 229 – 258. Elsevier-Science Publishers, 1993.

[32] S. Dutta Roy, S. Chaudhury, and S. Banerjee, "Aspect Graph Construction with Noisy Feature Detectors," *(Preprint full version)*, http://www.ee.iitb.ac.in/∼sumantra/publications/aspect.ps.gz.

[33] B. K. P. Horn, *Robot Vision*, The MIT Press and McGraw-Hill Book Company, 1986.

[34] K. W. Bowyer (Organizer), O. D. Faugeras, J. Mundy, N. Ahuja, C. R. Dyer, A. P. Pentland, R. C. Jain, and K. Ikeuchi (Participants), "Why Aspect Graphs Are Not (Yet) Practical for Computer Vision (Workshop Panel Report)," *Computer Vision, Graphics and Image Processing: Image Understanding*, vol. 55, pp. 212 – 218, March 1992.

[35] K. Ikeuchi and T. Kanade, "Modeling Sensors: Towards Automatic Generation of Object Recognition Programs," *Computer Vision, Graphics and Image Processing*, vol. 48, pp. 50 – 79, 1989.

[36] 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.