Aspect Graph Construction with Noisy Feature Detectors

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

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

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

Abstract

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 correctness 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 the error process as well as an algorithm for constructing 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 method.

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

1 Introduction

Many 3-D object recognition strategies use aspect graphs (for example, [1], [2], [3], [4], [5]). The success of such a strategy crucially depends upon its ability to handle feature detection errors – the ability of the aspect graph construction algorithm to model and account for such errors, as well as the mechanism in the view-identification algorithm to handle feature detection errors. In analytical approaches, aspect graphs are constructed directly from object shapes and surface characteristics. A limitation of such an approach is its applicability only to a specific class of objects. (e.g., [6], [7], [8], etc.) Uniform partitioning approaches tessellate the viewpoint space into uniform partitions (e.g., [9], [10], [11]). Adjacent viewpoints which give the same appearance of the object with respect to a feature set, are grouped together to form an aspect. A site is a representative viewpoint for a partition, at which sensor data is collected. The uniform partitioning approach is more general than the analytical approach since it is independent of the object shape and structure, the sensor, or the feature set.

*Author to whom all correspondence is to be addressed

This paper presents a new algorithm for aspect graph generation with active sensors, using a uniform partitioning approach. No related work accounts for feature detection errors in aspect graph construction. We show its applicability for robust 3-D object recognition in [4] and [5]. We present results of over 100 experiments on two sets of models, showing the effectiveness of our strategies.

2 Errors in Raw Aspect Data

Let the term 'raw aspect data' denote the collection of feature vectors obtained at the set of sites in the tessellated viewing space.

Aspects and Classes Koenderink and van Doorn [12] define aspects as topologically equivalent classes of object appearances. Since sensors may be of different types, Ikeuchi and co-workers generalize this definition – Object appearances may be grouped into equivalence classes with respect to a feature set. These equivalence classes are aspects [10]. A Class (or, Aspect-Class) is a set of aspects, equivalent with respect to a feature set.

Aspect-Candidates and Class-Candidates 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.

2.1 A Classification of Errors in Raw Aspect Data

The result of the error process may be characterized by its position(site), content(value, or feature vector label) or both. One can also characterize an error in terms of the raw aspect data available, in terms of aspect-candidates, class-candidates and their parameters. For our formulation, we assume that the sensor can move around the object in a circle in fixed angular increments, at a fixed height. We define the following

terms:

$\mathcal{A}$: The set of all aspect-candidates

$\mathcal{C}$: The set of all class-candidates

$\mathcal{A}_c(\mathcal{A}_c \subseteq \mathcal{A}) \triangleq \{\ \alpha \mid CLASS\_CAND(\alpha) = c,$ where $\alpha \in \mathcal{A}, c \in \mathcal{C},\ CLASS\_CAND : \mathcal{A} \rightarrow \mathcal{C}\}$

θ_α : angular width of aspect-candidate α, in terms of the number of sites it occupies

N_{min} : the minimum total number of sites at which a class-candidate should be present in the entire model-base to be called a 'Valid class/class-candidate'

Θ_{min} : the minimum extent which an aspect-candidate must have in order for it to be called a 'Valid aspect-candidate'

Θ_p : the minimum extent which an aspect-candidate must have in order for it to be called a 'Prominent aspect-candidate'($\Theta_p \geq \Theta_{min}$)

$\mathcal{A}_c^g \triangleq \{\ \alpha \mid \theta_\alpha \geq \Theta_{min}, \text{where } \alpha \in \mathcal{A}_c\}$

$\mathcal{A}_c^l \triangleq \{\ \alpha \mid \theta_\alpha < \Theta_{min}, \text{where } \alpha \in \mathcal{A}_c\}$
(Thus, $\mathcal{A}_c^g \cup \mathcal{A}_c^l = \mathcal{A}_c,\ \mathcal{A}_c^g \cap \mathcal{A}_c^l = \phi$)

$\mathcal{G}_a$: set of aspect-candidates in the neighbourhood of aspect-candidate a

$P(c' \mid c)$: the probability of the class-candidate actually being c', given that class-candidate c has been observed

At the end of the aspect graph construction algorithm, a prominent aspect-candidate will either constitute an aspect by itself, or have other aspect-candidates integrated into it to constitute an aspect.

We classify errors in aspect data into the following five categories:

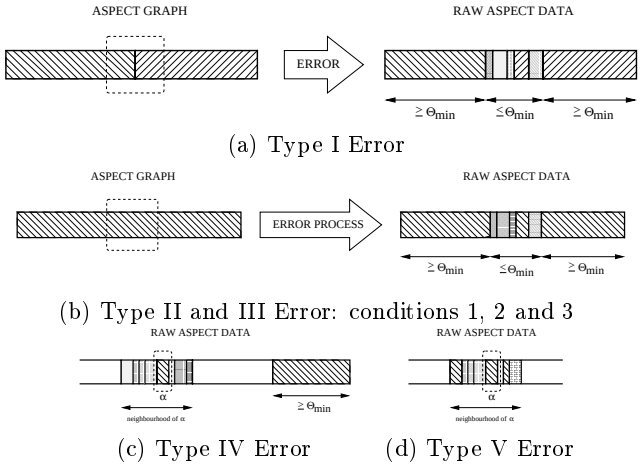
Type I Error A Type I error is present in a region of width $\leq \Theta_{min}$ between two aspects of the correct aspect graph. This corresponds to the "border effect" in [2]. In the raw aspect data, a Type I error can be described as follows:

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

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. Figure 1(a) illustrates an example of this type of error.

Type II Error The error process may introduce an isolated error inside an aspect such that the following conditions hold in the raw aspect data between two aspect-candidates a_i and a_j:

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(say)
4. $\sum_{\alpha \in \mathcal{A}_c} \theta_\alpha < N_{min}$



(a) Type I Error

(b) Type II and III Error: conditions 1, 2 and 3

(c) Type IV Error (d) Type V Error

Figure 1: A pictorial representation of some types of errors(Different shading patterns represent different aspect candidates)

5. $\frac{\sum_{\beta \in \mathcal{A}_c^g} \theta_\beta}{\sum_{\alpha \in \mathcal{A}_c} \theta_\alpha} <$ a threshold $T_1,\ 0 < T_1 < 1,$ and
6. $P(c|CLASS\_CAND(a_k)) \geq$ a threshold T_2, $0 < T_2 < 1,\ a_k$ between a_i and a_j

The probability estimates are characteristics of the given raw aspect data and are obtained from it. Figure 1(b) illustrates conditions 1, 2 and 3, which are common to Type II and Type III Errors.

For a Type II error, we observe a small region of width $\leq \Theta_{min}$ in between two valid aspect-candidates a_i and a_j, in the raw aspect graph. For the small enclosed error region, class-candidates observed are associated with the class-candidates of the enclosing aspect-candidates, for the particular model or the entire model base. Such 'association errors' characterize the response of a feature detector to a particular feature, or a combination of features present in the model being viewed.

Type III Error We define a Type III error as one introduced by the error process such that conditions 1 – 5 of a Type II error hold, but condition 6 does not i.e., there is no association of an error with a particular class-candidate.

Type IV Error Type IV errors constitute those errors scattered in an aspect such that the following additional conditions hold good for aspect-candidates α in the raw aspect data:

1. $\max_{\beta \in \mathcal{A}_c} \theta_\beta \geq \Theta_{min},$ where $CLASS\_CAND(\alpha) =$ c, but
2. $\theta_\alpha < \Theta_{min}$, and
3. $\forall \gamma \in \mathcal{G}_\alpha,\ \theta_\gamma < \Theta_{min}$
4. $\forall \gamma \in \mathcal{G}_\alpha,$ $CLASS\_CAND(\gamma) \neq CLASS\_CAND(\alpha)$

Figure 1(c) illustrates an example of a Type IV Error.

Type V Error Type V errors also involve errors scattered in an aspect with the conditions $1-3$ for an aspect-candidate α in the raw aspect data. The fourth condition is different: $\exists\gamma \in \mathcal{G}_\alpha$ for which

$$CLASS\_CAND(\gamma) = CLASS\_CAND(\alpha)$$

A Type V error indicates that the aspect-candidates with the same class-candidate may be part of a single aspect. Figure 1(d) illustrates an example of this type of error.

3 Building the Aspect Graph

3.1 A New Evaluation Function for Aspect Graphs

We propose a new coefficient to evaluate the output of aspect graph construction algorithms. We define the 'Demerit Coefficient' for the aspect graph of model i in model base $\mathcal{M}$ as follows:

$$\eta(\mathcal{M}, i, \tau) \stackrel{\triangle}{=} \mu \sum_j (1 - \rho_{ij})\ d(c_{ij}, c_{ij+1}) + \nu \sum_j d(c_{ij}, D_{ij}) + \sigma \sum_j \rho_{ij}$$

where D_{ij} is the original raw aspect data item at site j of model i, c_{ij} is the class-candidate assigned to site j by the aspect graph construction algorithm, $d(\ ,\ )$ denotes the Euclidean distance operator for two normalized feature vectors, and μ, ν and σ are constants which depend upon the aspect data characteristics. ρ_{ij} is defined to be 1 at if $d(c_{ij}, c_{ij+1}) >$ threshold τ, and 0 otherwise.

We define the Demerit Coefficient for the set of aspect graphs for the entire model base as the average of the Demerit Coefficients of the M individual models' aspect graphs, using the same constants and threshold for each model:

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

To evaluate the performance of an aspect graph construction algorithm, we consider the values of the Demerit Coefficient before and after the application of the algorithm. For the raw aspect data, c_{ij} is the same as D_{ij} for this case. The second term of $\eta(\mathcal{M}, i, \tau)$ is zero, while the presence of errors in the raw aspect data causes the first and third terms to have large values. The output of a good aspect graph construction algorithm is smooth, piecewise continuous, and is close to the original data. Hence for such a case, all the three components of $\eta(\mathcal{M}, i, \tau)$ have low values.

3.2 Aspect Graph Generation Algorithm

This section proposes a low order polynomial time-complexity algorithm for building an aspect graph from noisy aspect data. The algorithm maintains estimates of the probability with which one class-candidate is observed as another. To keep estimates of these association values, the algorithm uses an $N_C \times N_C$ matrix, the ASSOC_TABLE. The [i][j]th entry stores the probability of the class-candidate actually being c_j, given that class-candidate c_i is observed. The ASSOC_TABLE stores these values as counts of the number of times one class-candidate appears as another.

Our algorithm is divided into three phases:

3.2.1 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 sequential labeling algorithm. For an aspect graph which is not heavily corrupted with errors, one expects the class-candidates corresponding to prominent aspect-candidates to occupy more than N_{min} sites. Hence for such an aspect graph, this phase is expected to identify most of the valid class-candidates for the given model base. At the end of the entire algorithm, all valid class-candidates constitute the list of all classes for the given model base. A class-candidate node has the property that no node changes its label from VALID to INVALID, or vice versa during the course of the algorithm.

3.2.2 Algorithm Phase II

Phase II is primarily concerned with identification of prominent aspect-candidates after removing interspersed errors. 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 $\theta_{a_k} \in (0, \Theta_{min})\ \forall\ a_k$ lying in between a_i and a_j 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 $\leq \Theta_{min}$, we integrate both the valid aspect-candidates and those in between them, into one. The correct class-candidate for the aspect-candidates in between the valid aspect-candidates a_i and a_j, is considered to be that of a_i and a_j. The algorithm updates the ASSOC_TABLE with the information about the class-candidates of the valid aspect-candidates and those in between them. 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 from the raw aspect data – both of which consider isolated errors in an aspect for spurious class-candidates (Whether the isolated error removed is a Type II or Type III error will be clear from the ASSOC_TABLE conditional probabilities). This phase also removes some Type IV and Type V errors: that of aspect-candidates associated with a valid class-candidate, but having less angular extent. It is able to remove those errors regions which fall between two valid aspect-candidates having the same class-candidate. The algorithm uses the ASSOC_TABLE to account for association errors. If the value of $P(c|c')$ is above a particular threshold (c and c' are not necessarily different), the algorithm interprets an instance of c' to be c.

3.2.3 Algorithm Phase III

There are two passes through Phase III. The first is a logical pass, done in order to get estimates for ASSOC_TABLE entries. The actual pass, which follows the logical pass uses ASSOC_TABLE estimates made both during Phase II as well as the logical pass of Phase III. 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: 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. For this reason, we obtain the minimum square-error decision boundary for the region of gap δ ($\delta < \Theta_{min}$). The error for a decision boundary is the sum of Euclidean distances between the original class-candidate label and the one just assigned for the entire region, both suitably normalized. We select the one which incurs minimum error. This process has quadratic ($O(\delta^2)$) time complexity. This is one point where we use the association of one class-candidate with another. For each class-candidate in the region of width δ, its probability of being some other class-candidate is above a particular threshold, we replace it with this class-candidate for the purpose of getting an alternative decision boundary. Of the two decision boundaries calculated so far, the algorithm takes the one with the minimum error.

Case 2: gap(a_i,a_j) $\geq \Theta_{min}$ We construct a normalized histogram for the class-candidates in the region between valid aspect-candidates a_i and a_j. If the maximum histogram value exceeds a threshold, and a_i and a_j have the same class-candidate, then we integrate all aspect-candidates from a_i to a_j into a_i,a nd return. Otherwise, for all histogram entries above a threshold, if a_i and a_j have the same class-candidate, we repeat the above procedure. If the above test fails, we try to grow aspect-candidates a_i whose histogram entry exceeds a threshold, if its distance to the next aspect-candidate a_j (of the same class-candidate) is less than the current extent of a_i. For regions still unaccounted for, we get a single minimum-error decision boundary.

4 Experimental Results & Discussion

Our experimental setup has a camera connected to a MATROX Image Processing Card and a stepper motor-controlled turntable. The turntable moves by 200 steps to complete a 360 degree movement. We have experimented extensively with two object sets as model bases.

1. **Model Base I: 7 Aircraft Models** (Fig 2)
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 ($\langle c \rangle$).

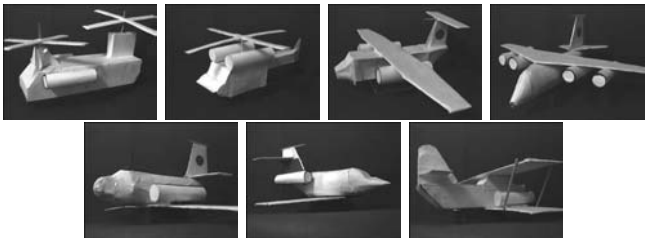


Figure 2: Model Base I: The objects (in row major order) are heli_1, heli_2, plane_1, plane_2, plane_3, plane_4, and biplane.

2. **Model Base II: 8 Polyhedral Objects** (Fig 3)
We use 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($\langle r \rangle$) as features.

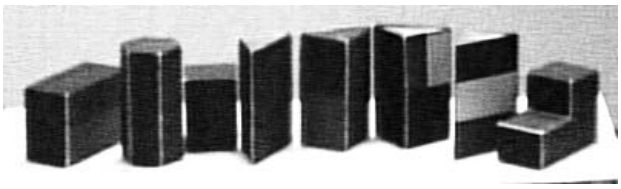


Figure 3: Model Base II: The objects (from left) are O_1, O_2, O_3, O_4, O_5, O_6, O_7 and O_8, respectively.

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.

By the term 'smoothness' of model base data(A), we mean

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

where M is the number of objects in the model base and G is the number of tessellated viewpoints for the aspect data. Let the term 'Input Smoothness' ($\mathcal{S}(I)$) and 'Output Smoothness' ($\mathcal{S}(O)$) refer to the smoothness expression for the raw aspect data and the output of the algorithm, respectively. (For $\mathcal{S}(I)$, $c_{ij} = D_{ij}$.)

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. Then, we compare some figures with those of Model Base II (polyhedral objects).

Output of the Aspect Graph Construction Algorithm: Figure 4 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

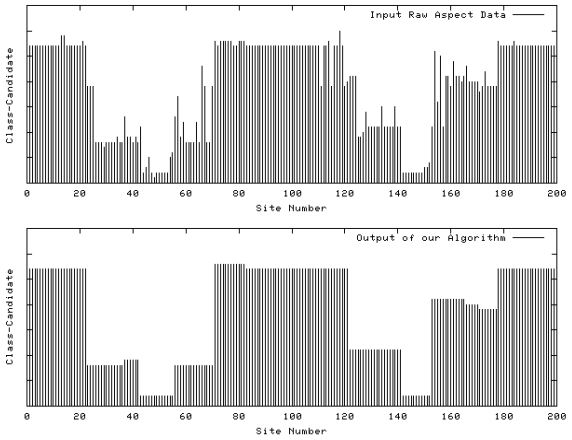


Figure 4: Raw aspect data and the output of our algorithm:plane_2, Model Base I. On the y-axis, each class-candidate is represented by an index. Different heights represent different class-candidates.

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 5 shows an example for Model Base II.

Input and Output Smoothness: Figure 6(a) shows a comparison of the input and output smoothness for

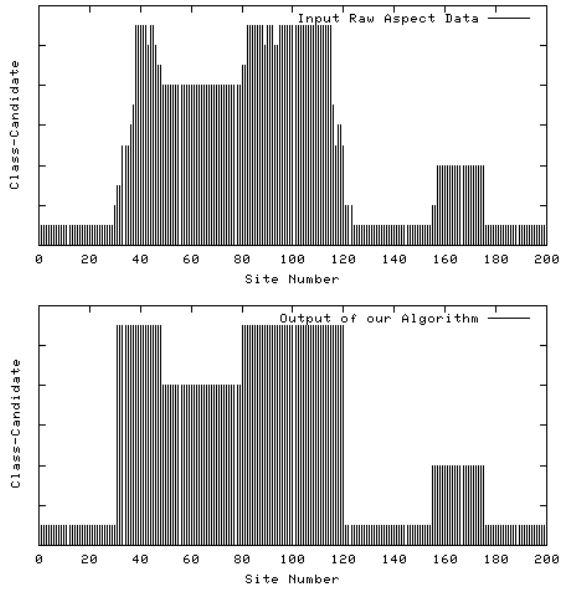


Figure 5: Raw aspect data and the output of our algorithm:O_6, Model Base II. On the y-axis, each class-candidate is represented by an index. Different heights represent different class-candidates.

100 sets of aspect data for the aircraft model base. Even though the raw aspect data has a large variation

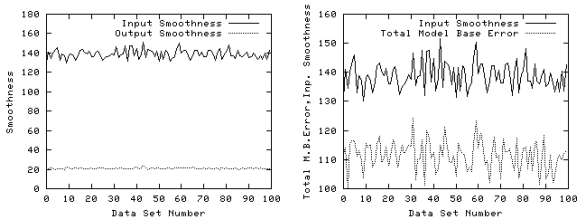


Figure 6: (a) Variation in input 'smoothness' with the output 'smoothness', (b) Variation in the total model base error with input 'smoothness' over 100 observations

in $\mathcal{S}$ values, the variation in in $\mathcal{S}$ values for the output data is very small.

Total Model Base Error and Number of Aspects: We define total model base error as:

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

where M is the number of objects in the model base and G is the number of tessellated viewpoints for the aspect data. Figures 6(b) and 7(a) show the variation in $\mathcal{E}$ and the number of aspects, respectively with the input smoothness for the 100 data sets.

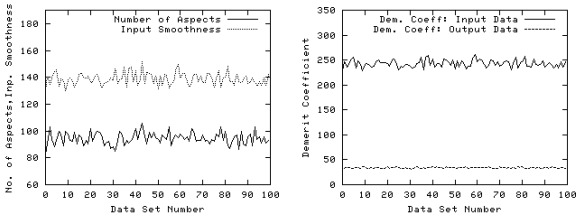


Figure 7: (a) Variation in the number of aspects with input 'smoothness' over 100 observations (b) Variation in the Demerit Coefficient for the input raw aspect data with the Demerit Coefficient for the output of the aspect graph construction algorithm

Demerit Coefficients: Figure 7(b) shows the variation of the Demerit Coefficient for the input aspect data, with the Demerit Coefficient for the output of the aspect graph construction algorithm for the 100 data sets. Our aspect graph construction algorithm greatly reduces the Demerit Coefficient. Further, the variation in the Demerit Coefficient for the output data is low compared with that for the input raw aspect data.

Percentage of sites where a single decision boundary had to be taken: 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 8(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

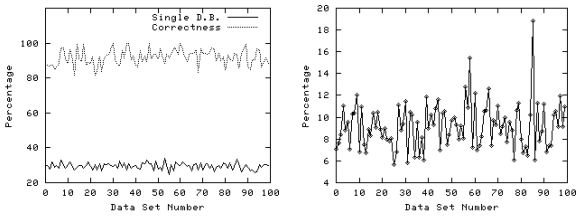


Figure 8: (a) Percentage of sites where a single decision boundary had to be taken, and correctness in Phase II estimates in the ASSOC_TABLE (b) Percentage reduction in total model base error with ASSOC_TABLE data

for aspect data with high $\mathcal{S}(I)$ values.

Correctness of Phase II ASSOC_TABLE estimates: We compare the probability values in the ASSOC_TABLE at the end of Phase III, to what they were at the end of Phase II. We refer to this as 'correctness'. Figure 8(a) shows the variation in percentage correctness of Phase II estimates, with the input 'smoothness' for 100 data sets.

Percentage reduction in model base error with ASSOC_TABLE estimates: The model base error $\mathcal{E}(A)$ is reduced if one used the association data from the ASSOC_TABLE. Figure 8(b) shows the percentage reduction in error for 100 instances of model base data.

Comparison of performance factors of our aspect graph construction algorithm on the two model bases: Table 1 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, their range of output values for the two model bases are comparable.

5 Application to Robust 3-D Object Recognition

Any aspect graph-based object recognition system needs well-defined aspects, which are not too large in number. Secondly, a probabilistic strategy can make the system more robust and resistant to errors compared to a deterministic one. Our work on isolated 3-D object recognition ([4], [5]) uses the output of our aspect graph construction algorithm, and 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 in an object recognition algorithm.

A view of a 3-D object corresponds to a class.

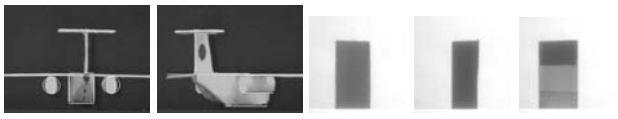
Feature-Class A Feature-Class is a set of equivalent aspects defined for one particular feature.

The papers [4], [5] present a probabilistic framework for recognition and planning. The $N_{\mathcal{C}} \times N_{\mathcal{C}}$ ASSOC_TABLE stores the number of times a class-candidate is observed as another. Let p_{jlk} represent the probability that the feature-class present is f_{jl}, given that the detector for feature F_j detects it to be f_{jk}. We define p_{jlk} as the ratio of the number of times the detector for feature F_j interprets feature-class f_{jl} as f_{jk}, and the number of times the feature detector reports the feature-class as f_{jk}. We use the ASSOC_TABLE to compute p_{jlk} values for each feature-class. The system uses the p_{jlk} values for computing the correct *a posteriori* class probabilities. 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.

Figure 9 shows two examples of experiments with objects from the two model bases. For Figure 9(a), the initial class could have come from 10 aspects. The corresponding number for Figure 9(b) is 18. The sys-

	Model Base I		Model Base II	
PARAMETERS	*Mean*	*Variance*	*Mean*	*Variance*
Input Smoothness	138.85	4.51	43.99	0.47
Output Smoothness	20.99	0.74	18.93	0.34
Total Model Base Error	111.49	5.01	27.97	1.35
No. of Aspects	94.51	4.55	80.5	1.12
Demerit Coefficient (input data)	242.84	7.32	67.69	2.59
Demerit Coefficient (output data)	34.13	1.17	28.99	0.48
Quadratic Complexity Regions	29.47%	1.85	7.47%	0.41
Correctness of Phase II Estimates	91.91%	4.46	94.74%	3.72
Error reduction with ASSOC_TABLE	9.18%	2.08	4.85%	0.37

Table 1: Aspect graph construction algorithm performance parameters for the two model bases



$\langle 342(332)\rangle \xrightarrow{26} \langle 410\rangle$ $\langle 221\rangle \xrightarrow{77} \langle 221\rangle \xrightarrow{72} \langle 423\rangle$

(a) Moves for object plane_1, initial class $\langle 332\rangle$
(b) Moves for object O_7, initial class $\langle 232\rangle$

Figure 9: Some experiments with objects in the two model bases. The numbers above the arrows denote the number of turntable steps. (The figure in parenthesis shows an example of recovery from feature detection errors)

tem needs 1 and 2 moves, respectively to correctly identify the given object. Figure 9(a) shows an example where the p_{jlk} values help the system in recovering from feature detection errors. Due to 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 superposed, for clarity.)

6 Conclusion

This paper presents an integrated approach for handling feature detection errors for use in robust active 3-D object recognition. We present the results of extensive experimentation on a reasonably complex experimental set, in support of our strategy. Scope for future work involves an extension to the general 3-D case.

References

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

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

[3] S. J. Dickinson, H. I. Christensen, J. Tsotsos, and G. Olofsson, "Active Object Recognition Integrating Attention and View Point Control," in *Proc. ECCV*. 1994, pp. 3–14, Springer-Verlag.

[4] S. Dutta Roy, S. Chaudhury, and S. Banerjee, "Isolated 3-D Object Recognition through Next View Planning," in *Proc. International Symp. on Intelligent Robotic Systems, Bangalore, India*, January 1998.

[5] S. Dutta Roy, S. Chaudhury, and S. Banerjee, "Isolated 3-D Object Recognition through Next View Planning," *IEEE Trans. on SMC B*, (Accepted for Publication).

[6] D. J. Kriegman and J. Ponce, "Computing Exact Aspect Graphs of Curved Objects: Solids of Revolution," *IJCV*, vol. 5, no. 2, pp. 119–135, November 1990.

[7] Z. Gigus and J. Malik, "Computing the Aspect Graph for Line Drawings of Polyhedral Objects," *IEEE Trans. on PAMI*, vol. 12, no. 2, pp. 113–122, February 1990.

[8] J. Stewman and K. Bowyer, "Creating the Perspective Projection Aspect Graph of Polyhedral Objects," in *Proc. ICCV*, 1988, pp. 494–500.

[9] M. R. Korn and C. R. Dyer, "3-D Multiview Object Representations for Model-Based Object

Recognition," *PR*, vol. 20, no. 1, pp. 91–103, 1987.

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

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

[12] 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.