ORACLE: AN INTEGRATED LEARNING APPROACH FOR OBJECT RECOGNITION

JOHN MING and BIR BHANU*

College of Engineering, University of California
Riverside, CA 92521, USA
E-mail: bhanu@shivish.ucr.edu

Model-based object recognition has become a popular paradigm in computer vision research. In most of the current model-based vision systems, the object models used for recognition are generally *a priori* given (e.g. obtained using a CAD model). For many object recognition applications, it is not realistic to utilize a fixed object model database with static model features. Rather, it is desirable to have a recognition system capable of performing automated object model acquisition and refinement. In order to achieve these capabilities, we have developed a system called ORACLE: Object Recognition Accomplished through Consolidated Learning Expertise. It uses two machine learning techniques known as Explanation-Based Learning (EBL) and Structured Conceptual Clustering (SCC) combined in a synergistic manner. As compared to systems which learn from numerous positive and negative examples, EBL allows the generalization of object model descriptions from a single example. Using these generalized descriptions, SCC constructs an efficient classification tree which is incrementally built and modified over time. Learning from experience is used to dynamically update the specific feature values of each object. These capabilities provide a dynamic object model database which allows the system to exhibit improved performance over time. We provide an overview of the ORACLE system and present experimental results using a database of thirty aircraft models.

Keywords: Dynamic object model database, model acquisition and refinement, multi-strategy learning.

1. INTRODUCTION

Prior attempts to automate object recognition systems have suffered from the lack of an ability to automatically acquire new object models, to adapt to changing environmental conditions, and to modify system behavior based on the context of the situation in which the systems are operating.[1–4] Due to recent advances in machine learning technology, some of these problems are resolvable by effectively combining machine learning and machine vision technologies. Learning allows an intelligent vision system to use situation context in order to understand images. This context, as defined in a machine learning scenario, consists of a collected body of background knowledge as well as environmental observations which may impact the processing of the scene. The resulting system dynamically reacts to the appropriate stimuli in the environment, continuously adapting its internal knowledge to improve overall performance levels. This improvement may come in the form of faster recognition times, improved recognition accuracy, and higher confidence in system results.

Machine learning technology provides two benefits for an object recognition domain: automatic knowledge base acquisition (e.g. object model database) and

*Corresponding author.

International Journal of Pattern Recognition and Artificial Intelligence Vol. 11 No. 6 (1997) 961–990

continuous knowledge base refinement. Although, significant work in automatic object model acquisition have been reported in the literature, particularly in the area of CAD-based vision in which computer vision systems are interfaced with existing CAD model databases, these approaches are not adequate enough for some real-world outdoor applications of computer vision for the following reasons.[14] The appearance of an object under varying imaging conditions (typical of outdoor scenes) can be significantly different, none of which may be suitably described using a single representational scheme.[6] Features of representation may not be robust for matching or may even prove to be difficult to extract; obtaining statistical distribution of object features is difficult when the object appearance cannot be predicted well; the representation may not be optimized for the task in hand; and, finally, new object models cannot be added to the static database as previously unseen objects are encountered. Further, the inability to adjust or update the descriptions of objects that are already modeled is another serious limitation of the existing approaches. The above two modifications (automatic knowledge acquisition and continuous knowledge base refinement) to the existing object recognition systems using machine learning techniques are the focus of this paper. Our work emphasizes that an effective recognition performance in unconstrained, outdoor scenarios require the capability to automatically acquire object model descriptions, refine those descriptions, and learn from experience.

Although machine learning has been used in many applications, its incorporation into the computer vision field is just beginning, particularly for object model acquisition (e.g. Ref. 5) and refinement. Further, within the machine learning field, very little effort has been made to combine several learning techniques together. Typically, learning methodologies are used independently to provide adaptive ability and improved system performance. Our multistrategy approach to object recognition[11] presented in this paper, called **ORACLE** (Object Recognition Accomplished through Consolidated Learning Expertise), incorporates two important learning techniques, known as explanation-based learning (EBL) and structured conceptual clustering (SCC). These techniques filter and structure the information present in positive concept examples to create useful knowledge structures. We have synergistically combined the EBL and SCC learning methodologies in the ORACLE system to offer a consolidated technique which employs the best features of each method to address the object model recognition, acquisition, and refinement requirements.

2. ORACLE LEARNING SYSTEM FOR OBJECT MODEL RECOGNITION, ACQUISITION, AND REFINEMENT

Learning systems based on a single learning paradigm, called *monostrategy* learning systems, are not adequate for complex problems like object model acquisition and refinement. Thus, the ORACLE object recognition system, which is a *multistrategy* learning system, integrates multiple learning techniques — EBL and SCC — to achieve its goals. The integration helps to overcome the inherent limitations present in the individual learning approaches when applied to the object recognition problem. In this section, we review the component learning methods of the ORACLE system, following which the description of the integration and the system details are presented.

EBL,[8,12] which is classified as a *learning by observation* technique, uses inference to construct a useful concept description from a single example (or a small number of examples) of that concept. Unlike the classical *learning from examples* techniques,[16] EBL uses a collection of applicable *background knowledge* to generate a useful object description from a single example. However, classification of objects using EBL can be an extremely slow process, particularly when the model database is large. The exact manner in which EBL is used in our system is discussed in Sec. 2.1.3.

SCC[9,10,15] is a method for grouping objects into classes similar to traditional numerical clustering techniques. However, instead of using predefined measures of object similarity to determine class boundaries, SCC uses a conjunction of conceptual attributes to group objects into conceptually simple classes. This process utilizes important contextual information relevant to the objects to assist in the classification process. SCC can handle complex, structural descriptions of objects, which is ideal for object recognition tasks since most objects are represented using structural descriptions. However, SCC has problems with model biases when the number of object class examples is small. Further details about our use of SCC are given in Sec. 2.1.4.

The *nugget* of the ORACLE object recognition system is that we have combined the ability of EBL to characterize an object using a single training example with SCC's efficient method of organizing objects once they have been properly modeled. This approach yields an integrated learning system which effectively handles the object recognition task.

2.1. ORACLE System Description

Figure 1 shows the configuration of the components in the ORACLE object recognition system. The processing elements (indicated by rectangular boxes) utilize object-specific data (indicated by rounded boxes) and knowledge databases (indicated by oval boxes) during the object recognition process. The input image is assumed to contain objects of interest (e.g. aircraft) and may include objects that are not currently in the object model database. The Segmentation and Symbolic Feature Extraction component identifies the regions of interest that contain objects in the input image and extracts symbolic feature information from these regions. The Symbolic Feature Definitions (e.g. wing span, fuselage length, number of engines) are used during this step to identify important object features which are useful for object recognition. The Knowledge-Based Matching component parses the Object Classification Tree (see Sec. 2.1.4) which represents a structured hierarchy of all objects (e.g. different aircraft types) known by the ORACLE system. Using the extracted symbolic object features, the matching component identifies the various recognition states (complete, incomplete, occluded, or failed recognition) of the ORACLE system. It also initiates the proper learning cycle (model acquisition or refinement) based on the object recognition results. The Feature Value Monitor modifies the object feature (e.g. wing span) values in the classification tree based on the features which are used to identify the object during the recognition cycle.

The Explanation Based Learning (EBL), when invoked by the matching component, selects the relevant object features from the symbolic feature information during the object model acquisition process. EBL also identifies new, pertinent object features for refining object models already present in the classification tree. The Background Knowledge is accessed by the EBL component to select object features during both model acquisition and refinement operations. The Object Model Database stores the complete model of each object encountered by the system, including every feature (relevant or not) defined on each object. Relevant object features, as determined by EBL, are tagged for future reference in the object model. The Structured Conceptual Clustering (SCC) is responsible for constructing and maintaining the object classification tree using the symbolic features selected by EBL. SCC makes use of the Goal Dependency Network (discussed in Sec. 2.1.4) while constructing or modifying the classification tree in order to compute the optimal clustering of the objects at the current level in the object hierarchy.

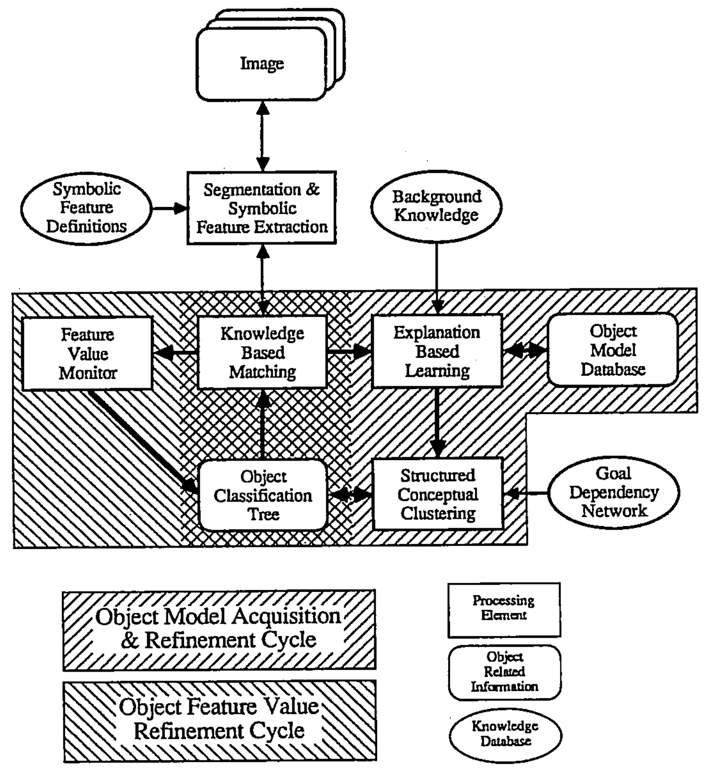


Fig. 1. Multistrategy machine learning approach for object recognition.

Figure 1 also highlights the two distinct *learning cycles* which are present in the ORACLE system. The first learning cycle is the object model acquisition and refinement process. The components used in this loop include the knowledge-based matching, explanation-based learning, and structured conceptual clustering along with the corresponding object and knowledge databases. The second learning cycle is the object feature value refinement process. This operation utilizes the knowledge-based matching and feature value monitor components.

The various processing components of the ORACLE system and the manner in which they interact will now be briefly described.

2.1.1. Segmentation and symbolic feature extraction

A segmentation algorithm is used to initially segment the image. When working with gray-scale images, we have used a "refocused"[13] algorithm which analyzes the large aerial images using domain knowledge at multiple resolutions to localize and segment objects.[7] Among the segmented image regions, the ones related to the objects of interest are identified and their boundaries are approximated using piecewise linear segments. Next, each object region is assigned a label based on its size, shape, and relationships with neighboring regions. A hypothesize-and-test approach is used to identify the rough orientation of the object during the region labeling operation. For example, a region may be hypothesized as being an aircraft fuselage based on its shape properties (narrow, elongated region). This hypothesis is verified by finding a symmetric pair of regions adjacent to the fuselage with wing-like properties. Similarly, the tail regions of the aircraft are labeled and used to support the current hypothesis.

When a hypothesis has been verified using surrounding regions as additional evidence, all contributing regions are tagged and used in the symbolic feature extraction operation. ORACLE computes symbolic feature information from the region borders using a knowledge-based approach. *Symbolic features* represent conceptual descriptions of an object's properties that would be used by a human in characterizing the object's appearance. The rules use distances and orientations of line segments on the region borders to compute the various object features.

2.1.2. Knowledge-based matching

The knowledge-based matching component receives an object schema (an enumerated list of symbolic features) from the segmentation and symbolic feature extraction component. This schema is utilized by the matching element to traverse the classification tree in an attempt to reach a leaf node of the tree. If the latter happens, then an input object is correctly recognized as the object model present at the leaf node. If at any point in the tree traversal, a feature is missing from the unknown object schema, the system spawns a set of multiple *viewpoints*. Each viewpoint represents a different interpretation of the data. A separate viewpoint is created for each feasible branch at the current level in the classification tree. This action allows the tree parsing process to evaluate many hypothetical alternatives.

The survival of any given viewpoint is governed by the matching success that is achieved during the processing of successive tree nodes in that viewpoint. At a later time in a particular viewpoint, the tree parsing process may terminate due to feature incompatibility. This condition results in the removal of the corresponding viewpoint from further consideration. Viewpoint removal allows the search process to prune branches from the classification tree when it becomes clear further search will be useless.

If the knowledge-based matching component is unable to parse the tree using the available symbolic feature data, the feature set is passed to the EBL component. In these situations, the failure of the matching process is due to one of two conditions. First, the feature information may represent a new object model that is not currently represented in the classification tree and which can potentially be acquired by the EBL component. Alternatively, the feature data may be faulty, incomplete, or inconsistent with the system's current object recognition domain, in which case EBL will not be able to acquire a new model. However, the matching process does not distinguish between these two cases. The matching component also sends the feature information to the EBL component when it detects the presence of a new feature in a correctly recognized object. By consulting the information stored in the Object Model Database, the matching component can detect when a new feature is present. It adds the new feature to the current object model and passes the revised model to the EBL component in order to determine the relevance of the new feature.

2.1.3. Explanation-based learning

The Explanation-Based Learning component utilizes domain-specific Background Knowledge (Fig. 1) to draw inferences from the symbolic feature data that are not possible for the matching process using the classification tree alone. Since, EBL and its associated knowledge base are only invoked in situations where the classification tree fails, the ORACLE system remains efficient by accessing the information in the knowledge base only when necessary.

The EBL component requires three inputs in order to acquire an object model:

Goal Concept: It is a definition of the concept to be learned. For example, in the aircraft recognition domain, a goal concept may be defined as Aircraft(X) for (Wing-Features(X) & Fuselage-Features(X) & Engine-Features(X) & Tail-Features(X)). In other words, an object X can be assumed to be an aircraft if it contains wing-like features, fuselage-like features, engine-like features, and tail-like features.

Training Example: It is an instantiation of the goal concept provided as an input to EBL. In the aircraft recognition scenario, a training example takes the form of a set of symbolic object features such as Wing-Span(Obj1,74'), Fuselage-Length(Obj1,69'), etc.

Domain Theory: It is a set of rules and facts to be used in explaining how the training example is an instance of the goal concept. Earlier, this information has been referred to as background knowledge. One of the rules in the aircraft recognition domain theory may state that if an object contains a Wing-Span feature (e.g., Wing-Span(Obj1,74')), then it exhibits wing-like features (e.g., Wing-Features(Obj1)).

As part of the EBL process, the generated explanation (of how the training example satisfies the criteria represented in the goal concept) structure is generalized by removing the intermediate level subgoals (Wing-Features, Fuselage-Features, etc.) of the explanation and by replacing the numeric values associated with each feature into a range of acceptable feature values. This generalized explanation is represented as an object model schema and can be used to recognize future instances of that object. The details of the EBL process are given in the report by Ming and Bhanu.[11]

The EBL component is responsible for four separate tasks within the ORACLE object recognition system:

(1) *Processing the training examples during system initialization.* EBL applies the Background Knowledge (Fig. 1) to each object schema using a generic object prototype to guide the explanation process. Once the explanation has been created, it is generalized to create an object model that contains the relevant object features. All object models created during system initialization are sent to the SCC component, which generates the object classification tree.
(2) *Acquiring new object models.* When the object classification tree is unable to process an unknown object schema, the EBL component is given the feature data in order to determine if a new object model can be constructed from the available features. The new model acquisition process is identical to the system training process described above. If a new model can be successfully derived, it is added to the object model database and is passed on to SCC for addition into the current object classification tree.
(3) *Refining existing object models.* EBL is also invoked to determine the relevance of new feature information that is present for an existing, correctly recognized object model. The presence of a new feature can be detected since EBL maintains a list of all previous symbolic features defined on each object in the object model database (Fig. 1). EBL adds the new feature to the current object model feature set and reprocesses the feature data. If the new feature is found to be relevant, it is tagged in the object model and is sent to the SCC component for addition into the object classification tree. Otherwise, the feature is simply left in the object model database as non-relevant.
(4) *Identifying recognition failures in the ORACLE system.* EBL is responsible for determining cases of recognition failure. When the knowledge-based matching component is unable to process a set of feature data, EBL is given the chance to acquire a new model using the available features. However, if EBL cannot construct an appropriate model from the feature information, the feature set is incomplete or the background knowledge is insufficient to understand the feature data. In either case, the situation is reported as a recognition failure.

2.1.4. Structured conceptual clustering

The Structured Conceptual Clustering component constructs the Object Classification Tree (OCT) from the relevant feature data generated by EBL (see Fig. 1; for an example see Fig. 7). The organization of the OCT closely matches the categorization of objects in human cognition. Coarse, high level features are used to identify broad object classes while the more specialized features are used to provide subclass discrimination. As described earlier, traversal of the classification tree allows the matching component to understand and compensate for missing information in unknown object schemata during the recognition process. The classification tree also provides efficiency in the object recognition task since the matching process does not have to compare the unknown object schema with every object model currently in the object model database.

During the construction or modification of the OCT, SCC accesses the information present in the Goal Dependency Network (GDN) in order to select potentially useful object features (Fig. 1). The GDN is an association list of potential relevant object features and the level in the OCT at which each feature is best used. Global object characteristics (e.g. fuselage length or wing span) are specified by the GDN at high levels (near the root) in the tree because they usually categorize coarse object classes. Within these classes, the GDN suggests more specialized object features (e.g. the number of engines, leading or trailing wing angles, etc.) that are used to determine subclass assignments. At each node in the tree, beginning with the root node, the SCC computes the intersection of the feature lists for all the object models at that node. The SCC then removes all features used at higher nodes in the OCT so they are not reused. If at any position in the tree, this resulting list is null, the object models at that node are each placed in a leaf node descending from the current tree location. Otherwise, the SCC clusters the object models using each of the valid features and selects the most distinguishing one based on cluster quality.

Although the GDN *suggests* several features to use at a particular position in the tree, the SCC process must still select the best feature for the specific situation. To perform this task, each suggested feature is used to generate a clustering of the objects. The quality of each clustering is based on the *conceptual simplicity* of the clustering results. ORACLE uses several factors in determining the conceptual simplicity of a proposed clustering including: the number of clusters into which the objects have been placed; the inter-cluster and intra-cluster distances of the clustering results; and the GDN's ranking of the selected feature at the current level in the tree. These measures of clustering quality are combined to evaluate the clustering results of each feature. The feature that provides the highest clustering quality value is selected and the current branch of the tree is defined accordingly. The SCC component continues to cluster the objects at each branch in the tree until every single object has been placed into a separate leaf node in the tree.

SCC provides an adaptive capability to the ORACLE system since it never relies on predefined measures of class similarity, but rather, it computes the feature that best distinguishes a set of objects at any given level in the OCT. Over time, the choice of the distinguishing feature at a particular level may dynamically change as a result of the new objects and revised objects which are continually being placed in the OCT. An analysis of the tree structure across many successive recognition cycles of the ORACLE system shows that it dynamically responds to the objects which are added or modified by automatically restructuring the appropriate tree branches to obtain an optimal object categorization.

The SCC component has three functions within the ORACLE object recognition system:

(1) *Construction of the initial OCT during system training.* SCC takes all the object models created by the EBL component and constructs the classification tree. At each branch in the tree, the GDN is used to suggest a set of appropriate object features from which one is selected by measuring the conceptual simplicity.
(2) *Addition of a new object model into the OCT during the object model acquisition process.* SCC attempts to retain as much of the original structure of the tree as possible. SCC traverses the tree using the new object model until a branch is encountered that is not compatible with the new object's features. The tree is then reclustered at that location. If a leaf node is encountered, a new branch is created to distinguish the object model currently stored in the leaf node from the new object model.
(3) *Modification of the current OCT structure during the object model refinement process.* This process is similar to the new object model situation since SCC minimizes the required changes to the tree. At each node in the tree, SCC determines if the new feature produces a better clustering quality than the distinguishing feature used at the current branch. If the new feature is better, the tree is reclustered at the current location. Otherwise, the appropriate branch is selected and the process continues. If a leaf node is reached, the new object model feature is simply inserted at the leaf node.

2.1.5. Feature value monitor

The Feature Value Monitor updates the quantitative feature values of an object model, if and when that model is used to recognize an unknown object. This process allows the ORACLE system to gradually modify the feature values of an object in order to overcome any initial bias that may have been acquired during the initial construction of the object model. Changes in object models made by the feature value monitor will be very gradual compared with the changes which result from activating the EBL-SCC object model refinement process described earlier. In the latter case, symbolic features are added or removed from the relevant feature list of the object model. The feature value monitor simply modifies the *relevant* quantitative feature values of the object. Further, the feature value monitor does not modify any qualitative object features present in the model.

The feature values are changed by shifting the range of numeric values produced during the EBL generalization process in the direction of the new object feature value. Each range is characterized by a central feature value with endpoints, a prescribed distance away from this value. For example, the central feature value for the range (100'-106') is 103'. The feature value monitor moves the entire feature value range in the direction that more closely aligns the central value of range with the new object feature value. The width of the feature value range remains the same. To avoid wild fluctuations in the feature value ranges, the range is moved only one unit (one foot, one degree, etc.) during any given recognition cycle, regardless of the discrepancy size. This approach is preferable to the alternative method of aligning the numeric range on the current object's feature value because it prevents potential misclassification results from adversely affecting the actual location of the feature value range. The approach is also more in tune with the notion that adaptation should be a gradual, rather than abrupt, process.

2.2. Recognition and Learning in ORACLE

During every recognition cycle, the ORACLE system identifies one of the following *recognition states* (See Fig 2):

(1) *Complete Recognition* – The unknown object schema is correctly classified with a high degree of confidence using the classification tree. The knowledge-based matching component and the feature value monitor are involved in the complete recognition operation.

(2) *Incomplete Matching* – The unknown object schema is partially classified using the classification tree. The matching component identifies multiple object models in the classification tree which meet the limited constraints imposed by the available unknown object features. A recognition confidence is produced for each matched object model. Only the knowledge-based matching component is used in this operation.

(3) *Object Occlusion* – Although occluded, the identity of the unknown object schema is predicted with some confidence level using the classification tree. This operation involves only the knowledge-based matching component. The main difference between this and incomplete matching is one of global vs. local nature of missing object features.

(4) *Object Model Acquisition* – The unknown object schema cannot be classified using the current classification tree, so the object model is acquired by the EBL-SCC learning cycle and added to the classification tree. The model acquisition process involves the knowledge-based matching, EBL, and SCC components of the ORACLE system.

(5) *Object Model Refinement* – After correctly classifying the unknown object schema using the classification tree, a new feature is identified in the unknown object schema. The object model and the classification tree are updated to indicate the relevance of this new object feature. The model acquisition process involves the knowledge-based matching, EBL, and SCC components of the ORACLE system.

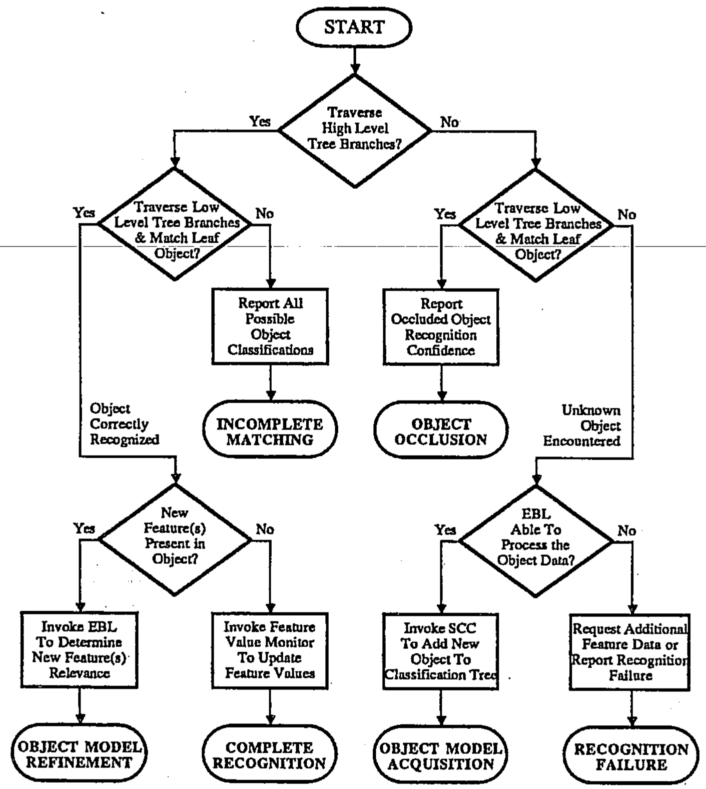


Fig. 2. Decision diagram specifying the conditions through which the six different *recognition states* of the ORACLE system are identified.

(6) *Recognition Failure* – The unknown object schema cannot be classified using the information in the classification tree or by EBL with the use of the background knowledge database.

3. EXPERIMENTAL RESULTS

We have conducted a series of experiments to test the object recognition and learning concepts, and capabilities of the ORACLE system for the recognition of 2D aircraft. The imagery (binary) used for these experiments was generated by digitizing technical diagrams of various commercial aircraft ranging in size from small single engine private aircraft (Cessna Caravan) to large passenger airliners (Boeing 747). Eleven aircraft were selected for the initial set of experiments on the ORACLE system. In order to simulate the degraded appearance of object region boundaries typical of segmented real images, Gaussian noise (mean = 0, variance = 1-20) was added to the border points of a binary image and the resulting image was distorted using morphological operations of erosion and dilation. Next, a border following routine was applied to generate a list of pixels that comprise the outline of the aircraft. Finally, the outline was represented with a piecewise polygonal approximation

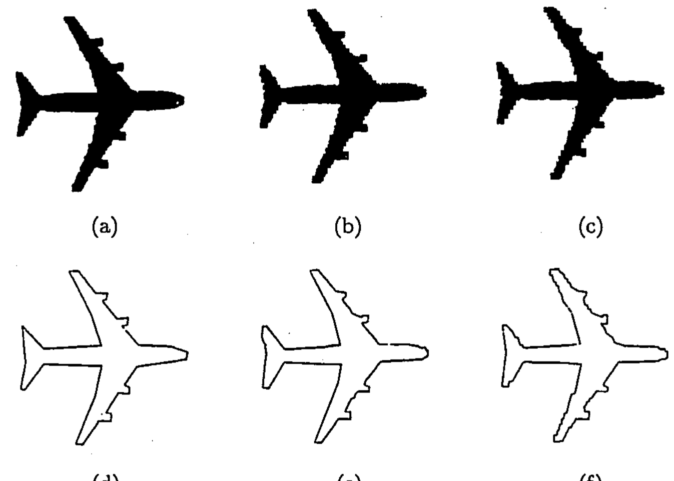


Fig. 3. Various levels of noise and distortion added to an aircraft. (a) Noise level = 3. (b) Noise level = 11. (c) Noise level = 17. (d) Polygonal approximation for (a). (e) Polygonal approximation for (b). (f) Polygonal approximation for (c).

using a split-merge approximation algorithm. Figure 3 provides an example of the border distortion process and the corresponding polygonal approximation results for a typical aircraft image.

The polygonal approximation of an aircraft contour is processed by a knowledge-based algorithm to create the list of symbolic object features needed by the ORACLE system. This operation makes use of the symmetry properties of the aircraft's shape in determining many of the symbolic features. When feature values obtained for a specific feature (e.g. wing span or leading wing angle) vary significantly on opposite sides of an aircraft (usually due to distortion in the aircraft image), the feature is not extracted due to the ambiguity of the situation. This approach insures that object misclassification does not result from the presence of uncertain feature information.

In the examples described below, the image distortion, border following, and polygonal approximation algorithms have all been implemented and executed on a SUN 3/60 workstation. The polygonal approximation data is transferred to a Symbolics 3670 workstation, which performs the symbolic feature extraction operation and hosts the ORACLE object recognition system.

3.1. System Training

The first step in the object recognition process is automatic construction of an initial object model database from user supplied images. Figure 4 shows the set of aircraft used to initialize the ORACLE system. These aircraft images have not been distorted since most training operations utilize high quality training data to insure accuracy. Figure 5 provides the set of symbolic object features obtained for each aircraft in Fig. 4. Missing features are due to inconsistencies in feature values or other aircraft anomalies.

Once the symbolic feature information for all training examples is available, the ORACLE system must construct the object classification tree. To do so, the EBL component is invoked on each aircraft schemata (the symbolic feature lists shown in Fig. 5) to select relevant object model features. EBL utilizes the background knowledge base which for these examples consists of a generic aircraft prototype that specifies the presence of wing, fuselage, engine, and tail features in order to generate an aircraft object model. The knowledge base contains 23 different rules which specify the allowable combinations of the symbolic features to satisfy the wing, fuselage, engine, and tail requirements. Figure 6 illustrates the resulting object models created by the EBL component for the system initialization phase. Notice that the specific feature values have been generalized into ranges of values and that EBL has generated a weight (user supplied) associated with each feature in the object model. The weights are used during matching to compute object recognition confidence.

Following the selection of relevant object features by the EBL component, the ORACLE system invokes the SCC process to construct the initial object classification tree. All seven object models are given to SCC, which builds the classification tree shown in Fig. 7. The nodes in the tree are labeled $TN - *$, which stands for $TREE - NODE - *$. Note that the aircraft have been classified into intuitively

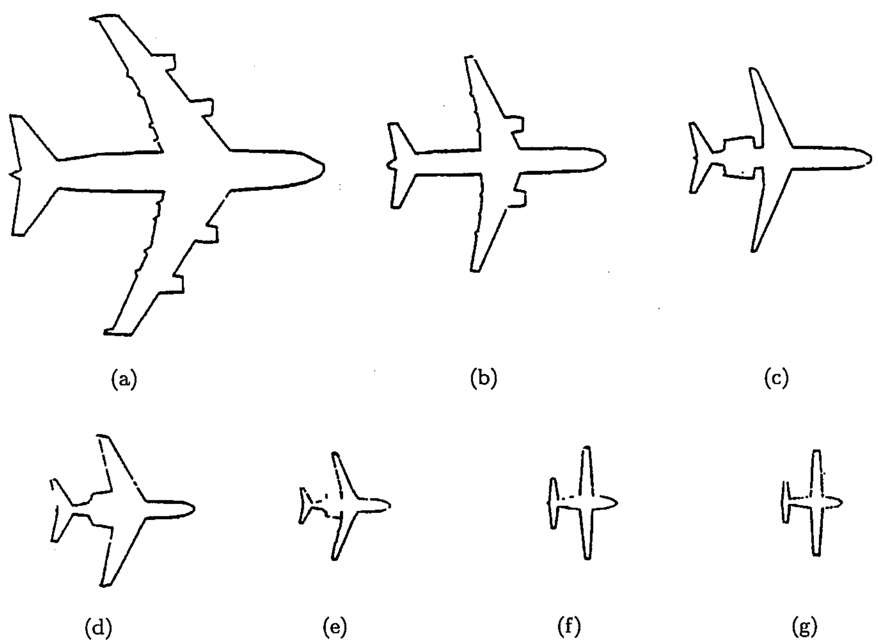


Fig. 4. Aircraft used during the initialization phase of the ORACLE system. (a) Boeing 747 (B-747). (b) Boeing 757 (B-757). (c) McDonnell Douglas MD-87 (MD-87). (d) Gulfstream Aerospace (Aerospace). (e) Cessna Citation (Citation). (f) Cessna Caravan (Caravan). (g) Piper Malibu (Malibu).

Feature	B-747	B-757	MD-87	Aerospace	Citation	Caravan	Malibu
Wingspan	245′	160′	140′	103′	69′	67′	53′
Wing sweep, leading	129°	114°	115°	119°	116°	95°	97°
Wing sweep, trailing	112°	97°	99°	102°	101°	88°	86°
Wing base chord	53′	28′	22′	22′	10′	7′	6′
Wing tip chord	12′	6′	4′	—	3′	—	3′
Wing taper, base/tip	4.37	4.86	6.00	—	3.33	—	2.26
Fuselage length	222′	155′	130′	88′	55′	38′	32′
Fuselage width	27′	18′	16′	11′	8′	8′	7′
Length, wing-to-nose	62′	60′	58′	32′	20′	12′	11′
Length, wing-to-tail	75′	47′	—	—	—	14′	10′
Nose shape	ROUND	—	ROUND	ROUND	—	—	POINTED
Position of engines	ON-WING	ON-WING	FUSELAGE	FUSELAGE	FUSELAGE	NOSE	NOSE
Number of engines	4	2	2	2	2	1	1
Tailspan	89′	62′	52′	42′	24′	26′	24′
Tail sweep, Leading	126°	118°	120°	120°	120°	96°	97°
Tail sweep, trailing	99°	99°	102°	102°	101°	87°	87°
Tail base chord	28′	15′	12′	10′	6′	5′	4′
Tail tip chord	8′	6′	4′	4′	2′	3′	2′
Tail taper, base/tip	4.03	2.40	2.92	2.73	3.17	1.88	2.40
Wingspan/tailspan	2.76	2.59	2.68	2.45	2.91	2.53	2.22

Fig. 5. Symbolic features extracted from the aircraft in Fig. 4.

obvious groups by the SCC component. The tree is used to recognize subsequent instances of aircraft modeled during training as discussed in the following three subsections.

3.2. Complete Recognition

Figure 8 shows an example of an "unknown" aircraft (*Malibu*) that must be recognized by the object recognition system. The aircraft image (Fig. 8(a)) is moderately distorted and thus, in the polygonal approximation (Fig. 8(b)), it is more irregular than in the training example. The distortion is apparent when the list of extracted symbolic features, shown in Fig. 8(c), and that of Fig. 6(g) are compared, e.g. only a few of the tail features are available in the former case.

The model matching component uses the list of features in Fig. 8(c) to parse the classification tree of Fig. 7. At the *ROOT-NODE*, the unknown aircraft is compatible with the leftmost branch, so the matching component traverses the tree to node

Feature	Value	Weight
Wingspan	(Range 239′–251′)	0.10
Wing sweep, leading	(Range 127°–131°)	0.08
Wing sweep, trailing	(Range 110°–114°)	0.07
Wing base chord	(Range 50′–56′)	0.06
Wing tip chord	(Range 11′–13′)	0.04
Fuselage length	(Range 216′–228′)	0.12
Fuselage width	(Range 25′–29′)	0.04
Length, wing-to-nose	(Range 59′–65′)	0.06
Length, wing-to-tail	(Range 72′–78′)	0.08
Position of engines	ON-WING	0.10
Number of engines	4	0.05
Tailspan	(Range 86′–92′)	0.08
Tail sweep, leading	(Range 124°–128°)	0.06
Tail sweep, trailing	(Range 97°–101°)	0.04
Tail base chord	(Range 26′–30′)	0.02

(a)

Feature	Value	Weight
Wingspan	(Range 155′–165′)	0.11
Wing sweep, leading	(Range 112°–116°)	0.09
Wing sweep, trailing	(Range 95°–99°)	0.09
Wing base chord	(Range 26′–30′)	0.06
Fuselage length	(Range 150′–160′)	0.12
Fuselage width	(Range 17′–19′)	0.04
Length, wing-to-nose	(Range 57′–63′)	0.06
Length, wing-to-tail	(Range 45′–49′)	0.08
Position of engines	ON-WING	0.10
Number of engines	2	0.05
Tailspan	(Range 59′–65′)	0.08
Tail sweep, leading	(Range 116°–120°)	0.06
Tail sweep, trailing	(Range 97°–101°)	0.04
Tail base chord	(Range 14′–16′)	0.02

(b)

Feature	Value	Weight
Wingspan	(Range 136′–144′)	0.11
Wing sweep, leading	(Range 113°–117°)	0.09
Wing sweep, trailing	(Range 97°–101°)	0.09
Wing base chord	(Range 21′–23′)	0.06
Fuselage length	(Range 126′–134′)	0.14
Fuselage width	(Range 15′–17′)	0.06
Length, wing-to-nose	(Range 55′–61′)	0.10
Position of engines	FUSELAGE	0.10
Number of engines	2	0.05
Tailspan	(Range 49′–55′)	0.08
Tail sweep, leading	(Range 118°–122°)	0.06
Tail sweep, trailing	(Range 100°–104°)	0.04
Tail base chord	(Range 11′–13′)	0.02

(c)

Feature	Value	Weight
Wingspan	(Range 99′–107′)	0.11
Wing sweep, leading	(Range 117°–121°)	0.09
Wing sweep, trailing	(Range 100°–104°)	0.09
Wing base chord	(Range 21′–23′)	0.06
Fuselage length	(Range 85′–91′)	0.14
Fuselage width	(Range 10′–12′)	0.06
Length, wing-to-nose	(Range 30′–34′)	0.10
Position of engines	FUSELAGE	0.10
Number of engines	2	0.05
Tailspan	(Range 40′–44′)	0.08
Tail sweep, leading	(Range 118°–122°)	0.06
Tail sweep, trailing	(Range 100°–104°)	0.04
Tail base chord	(Range 9′–11′)	0.02

(d)

Fig. 6. EBL-generated object models for each of the symbolic feature lists shown in Fig. 5. (a) B-747, (b) B-757, (c) MD-87, (d) Aerospace, (e) Citation, (f) Caravan, (g) Malibu.

TN-1. At this location, the unknown object matches the leftmost branch again, so the matching process moves to node *TN-2.* Here, both branches are investigated by the matching process, due to the missing wing-to-tail feature, to determine whether either of them (or possibly both) are compatible with the unknown object. The right branch of *TN-2* is discounted due to differences in wing span, fuselage length,

Feature	Value	Weight
Wingspan	(Range 66′–72′)	0.11
Wing sweep, leading	(Range 114°–118°)	0.09
Wing sweep, trailing	(Range 99°–103°)	0.09
Wing base chord	(Range 9′–11′)	0.06
Fuselage length	(Range 52′–58′)	0.18
Length, wing-to-nose	(Range 19′–21′)	0.12
Position of engines	FUSELAGE	0.10
Number of engines	2	0.05
Tailspan	(Range 23′–25′)	0.08
Tail sweep, leading	(Range 118°–122°)	0.06
Tail sweep, trailing	(Range 99°–103°)	0.06

(e)

Feature	Value	Weight
Wingspan	(Range 64′–70′)	0.17
Wing sweep, leading	(Range 93°–97°)	0.10
Wing sweep, trailing	(Range 86°–90°)	0.08
Fuselage length	(Range 36′–40′)	0.12
Length, wing-to-nose	(Range 11′–13′)	0.08
Length, wing-to-tail	(Range 13′–15′)	0.10
Position of engines	NOSE	0.10
Number of engines	1	0.05
Tailspan	(Range 24′–28′)	0.08
Tail sweep, leading	(Range 94°–98°)	0.06
Tail sweep, trailing	(Range 85°–89°)	0.06

(f)

Feature	Value	Weight
Wingspan	(Range 50′–56′)	0.17
Wing sweep, leading	(Range 95°–99°)	0.10
Wing sweep, trailing	(Range 84°–88°)	0.08
Fuselage length	(Range 30′–34′)	0.12
Length, wing-to-nose	(Range 10′–12′)	0.08
Length, wing-to-tail	(Range 9′–11′)	0.10
Position of engines	NOSE	0.10
Number of engines	1	0.05
Tailspan	(Range 23′–25′)	0.08
Tail sweep, leading	(Range 95°–99°)	0.06
Tail sweep, trailing	(Range 85°–89°)	0.06

(g)

Fig. 6. (*Cont'd*)

wing-to-nose, tail span, and tail leading angle. However, the left branch, *TN-3*, which contains the *Malibu* aircraft model, is found to be compatible with the unknown aircraft with 74.6% confidence. This confidence is derived using the weights assigned to each object model feature and the error between the feature values in the object model and the unknown aircraft. Even though the aircraft feature set was missing two features specified in the *Malibu* object model (wing-to-tail and tail leading angle), the ORACLE system was able to correctly recognize the aircraft.

Additionally, since the recognition confidence of the aircraft is greater than the complete recognition threshold (70% for these experiments), the feature value monitor is invoked to update the values in the *Malibu* aircraft model. The revised *Malibu* model is shown in Fig. 8(d).

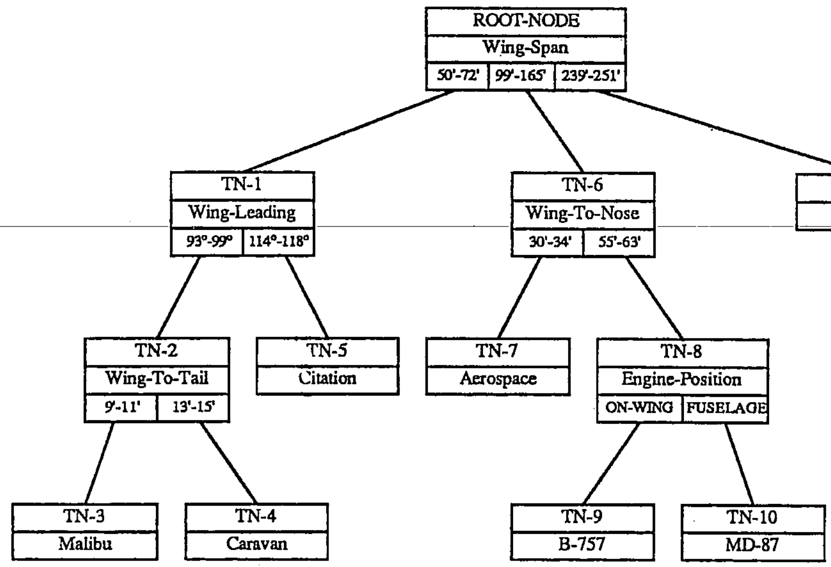


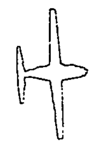
Fig. 7. SCC-generated object classification tree.

3.3. Incomplete Recognition

Figure 9 provides an example of an aircraft that causes the system to produce an incomplete recognition result. In this image (Fig. 9(a)), the tail of the aircraft and the engine regions are not connected to the main portion of the object. Such a situation is common in real images when there is low contrast between the object and the background. Since the border following algorithm is designed to locate only the largest object region, the system creates the polygonal approximation shown in Fig. 9(b). The corresponding set of extracted symbolic features are indicated in Fig. 9(c). Due to the lack of any tail information and discrepancies in the wing representation, very few reliable features have been obtained in this case.

The matching component begins at the *ROOT-NODE* of the classification tree (Fig. 7) as usual. Since the wing-span feature is missing, all three branches of the tree (Nodes *TN-1*, *TN-6*, and *TN-11*) are hypothesized as possible alternatives. *TN-11*, which contains the *B-747* aircraft model, is rejected due to differences in every single object model feature except wing-to-nose. At *TN-1*, the unknown aircraft's leading wing angle is compatible with the right branch of the node, so parsing continues down to *TN-5*. However, the *Citation* aircraft model contained in *TN-5* conflicts with the unknown aircraft in every feature except the leading wing angle. Thus, this hypothesis is also rejected.

(a)



(b)

Feature	Value
Wingspan	54′
Wing sweep, leading	97°
Wing sweep, trailing	87°
Wing base chord	5′
Wing tip chord	3′
Wing taper, base/tip	1.85
Fuselage length	30′
Fuselage width	7′
Length, wing-to-nose	10′
Position of engines	NOSE
Number of engines	1
Tailspan	23′
Tail sweep, trailing	87°
Tail tip chord	2°
Wingspan/tailspan	2.35

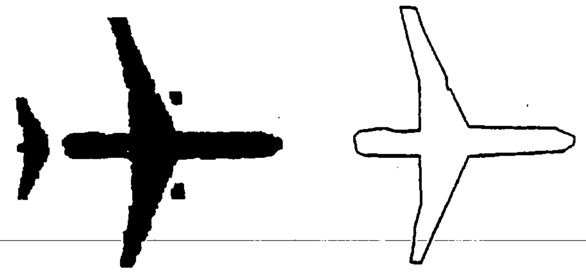
(c)

Feature	Value	Weight
Wingspan	(Range 51′–57′)	0.17
Wing sweep, leading	(Range 95°–99°)	0.10
Wing sweep, trailing	(Range 85°–89°)	0.08
Fuselage length	(Range 29′–33′)	0.12
Length, wing-to-nose	(Range 9′–11′)	0.08
Length, wing-to-tail	(Range 9′–11′)	0.10
Position of engines	NOSE	0.10
Number of engines	1	0.05
Tailspan	(Range 22′–24′)	0.08
Tail sweep, leading	(Range 95°–99°)	0.06
Tail sweep, trailing	(Range 85°–89°)	0.06

(d)

Fig. 8. Aircraft (Malibu) which illustrates the complete recognition state of the ORACLE system. (a) Distorted aircraft image. (b) Polygonal approximation of the aircraft. (c) Symbolic object features extracted from the aircraft. (d) Revised Malibu aircraft model after complete recognition cycle.

Looking at *TN-6*, the matching process selects the right branch and moves to the *TN-8* tree node. Since the engine position feature is missing, the matching process once again considers both branches as possible alternatives. Inspecting *TN-9*, the matching process finds that the *B-757* aircraft model is compatible with the unknown object (matching confidence = 25.9%). At *TN-10*, the unknown aircraft is also matched to the *MD-87* aircraft model (matching confidence = 27.5%). Since no additional feature information is available to select between these two alternatives, the ORACLE system reports both aircraft models as possible matches.



(a) (b)

Feature	Value
Wing sweep, leading	114°
Wing sweep, trailing	97°
Wing tip chord	4′
Fuselage width	17′
Length, wing-to-nose	60′
Nose shape	ROUND

(c)

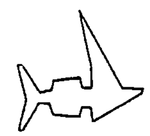
Fig. 9. Aircraft (B-757) which illustrates the incomplete recognition state of the ORACLE system. (a) Distorted aircraft image. (b) Polygonal approximation of the aircraft. (c) Symbolic object features extracted from the aircraft.

3.4. Occluded Recognition

Figure 10(a) provides an example of an aircraft image that illustrates the occluded recognition scenario. In this example, the nose and the port wing of the aircraft have been occluded. The polygonal approximation of this object is shown in Fig. 10(b). The symbolic feature extraction process is still able to derive a useful set of features from the aircraft, as indicated in Fig. 10(c).

The model matching component uses the list of symbolic feature information to parse the classification tree of Fig. 7. At the *ROOT-NODE*, the wing span value of the unknown aircraft is compatible with the center branch, so the matching component proceeds down to node *TN-6*. The wing-to-nose feature is missing in the feature list, so both branches (*TN-7* and *TN-8*) are hypothesized. Examining *TN-7*, the model matching process finds that the wing span and tail span feature values contradict those of the *Aerospace* object model stored in the node, although all other features are compatible. Thus, *TN-7* is discarded. At *TN-8*, the right branch of the node is compatible with the engine position feature in the feature list. Finally, at node *TN-10*, the matching process discovers that the *MD-87* object model is compatible with the feature list. The recognition confidence in this example is 65.9%. No changes are made to the object model since the recognition confidence is below the complete recognition confidence threshold.

(a)



(b)

Feature	Value
Wingspan	140′
Wing sweep, leading	117°
Wing sweep, trailing	99°
Wing base chord	22′
Position of engines	FUSELAGE
Number of engines	2
Tailspan	50′
Tail sweep, leading	120°
Tail sweep, trailing	102°
Tail base chord	11′
Tail tip chord	3′
Tail taper, base/tip	3.47
Wingspan/tailspan	2.65

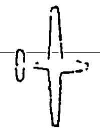
(c)

Fig. 10. Aircraft (MD-87) which illustrates the occluded recognition state of the ORACLE system. (a) Distorted aircraft image. (b) Polygonal approximation of the aircraft. (c) Symbolic object features extracted from the aircraft.

3.5. Object Model Acquisition

Figure 11(a) shows the image of an aircraft not encountered during training. Figure 11(b) illustrates the polygonal approximation of the aircraft image and Fig. 11(c) provides the list of symbolic object features extracted from the polygonal representation. During parsing of the classification tree (Fig. 7), only the left branch of the root node is hypothesized. Traversing the tree in standard fashion, the knowledge-based matching component eventually arrives at the sole hypothesized node *TN-3* and compares the symbolic feature list with the *Malibu* object model. However, differences in leading wing angle, trailing wing angle, fuselage length, and tail span cause the *Malibu* object model to be discarded. Thus, the current classification tree contains insufficient information to identify this aircraft.

(a)



(b)

Feature	Value
Wingspan	51′
Wing sweep, leading	93°
Wing sweep, trailing	84°
Wing base chord	5′
Wing tip chord	3′
Wing taper, base/tip	1.87
Fuselage length	27′
Length, wing-to-nose	10′
Length, wing-to-tail	9′
Position of engines	NOSE
Number of engines	1
Tailspan	13′
Tail sweep, trailing	88°
Wingspan/tailspan	3.92

(c)

Feature	Value	Weight
Wingspan	(Range 48′–54′)	0.17
Wing sweep, leading	(Range 91°–95°)	0.10
Wing sweep, trailing	(Range 82°–86°)	0.08
Fuselage length	(Range 25′–29′)	0.12
Length, wing-to-nose	(Range 9′–11′)	0.08
Length, wing-to-tail	(Range 8′–10′)	0.10
Position of engines	NOSE	0.10
Number of engines	1	0.05
Tailspan	(Range 12′–14′)	0.20

(d)

Fig. 11. Aircraft (Renegade) which illustrates the object model acquisition capabilities of the ORACLE system. (a) Distorted aircraft image. (b) Polygonal approximation of the aircraft. (c) Symbolic object features extracted from the aircraft. (d) EBL-generated object model for the aircraft.

The EBL process is subsequently invoked in an attempt to acquire the unknown aircraft as a new object model. Figure 11(d) illustrates the EBL-generated object model produced from the symbolic feature list in Fig. 11(c). The new object model is then handed to the SCC component so that it can be incorporated in the classification tree structure. SCC parses the tree using the new object model in an attempt to leave as much of the tree intact as possible. In the process, the tree is reclustered at *TN-3* node to distinguish between the current *Malibu* object model and the new *Renegade* object model. Tail span is found to be the best symbolic feature that separates the two object models. The revised classification tree, after insertion of the *Renegade* object model, is shown in Fig. 12.

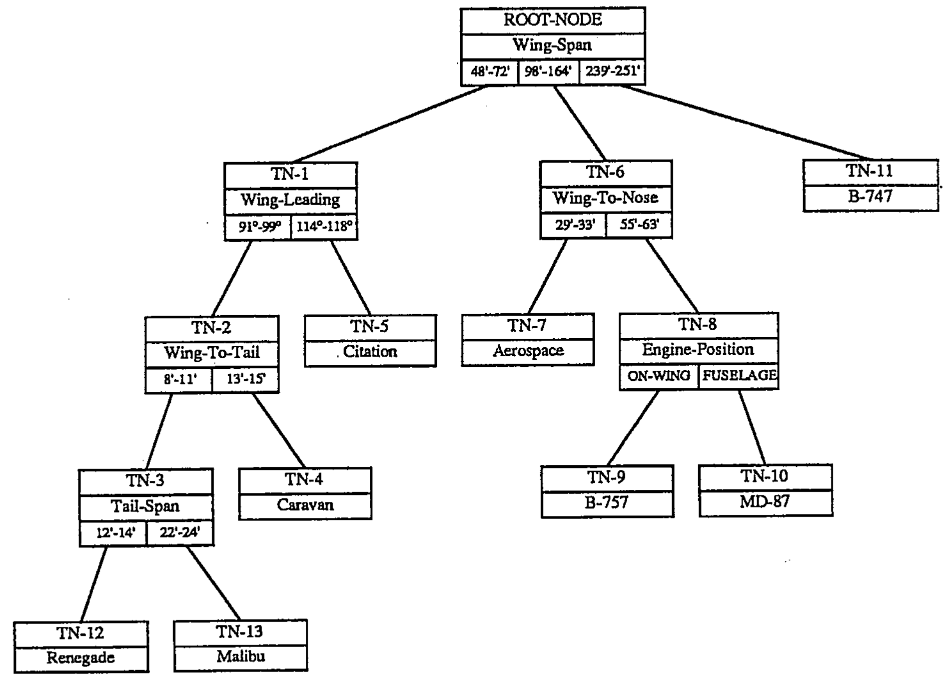


Fig. 12. Revised object classification tree after insertion of the Renegade aircraft model.

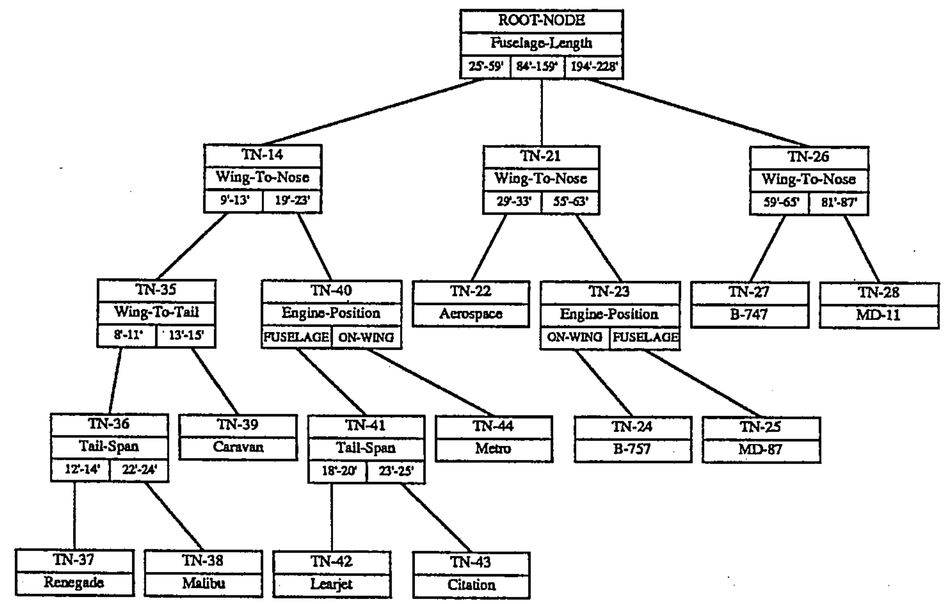


Fig. 13. Object classification tree for the object model refinement experiments.

3.6. Object Model Refinement

Object model refinement occurs when the presence of a new symbolic feature is detected for a recognizable aircraft. The classification tree used in these experiments is shown in Fig. 13. This tree was obtained from the one in Fig. 12 after three more aircraft models (*Metro*, *Learjet*, and *MD-11*) were acquired. Figure 14(a) shows a known aircraft whose model is to be refined and the corresponding polygonal approximation is indicated in Fig. 14(b). The list of symbolic object features obtained from this aircraft are presented in Fig. 14(c). The tree is parsed using this feature information and, although the matching process must hypothesize nodes *TN-36* and *TN-39* during the tree traversal, the aircraft is finally identified as an instance of the *Renegade* object model. The recognition confidence in this case is 86.1%. The nose shape, leading tail angle, and tail base features in Fig. 14(c) are discovered to be new model features and thus, the object model refinement operation is invoked.

(a)

(b)

Feature	Value
Wingspan	50′
Wing sweep, leading	93°
Wing sweep, trailing	85°
Wing base chord	5′
Wing tip chord	3′
Wing taper, base/tip	2.08
Fuselage length	28′
Length, wing-to-nose	10′
Length, wing-to-tail	9′
Nose shape	ROUND
Position of engines	NOSE
Number of engines	1
Tailspan	13′
Tail sweep, leading	89°
Tail sweep, trailing	89°
Tail base chord	4′
Wingspan/tailspan	3.68

(c)

Feature	Value	Weight
Wingspan	(Range 48′–52′)	0.17
Wing sweep, leading	(Range 91°–95°)	0.10
Wing sweep, trailing	(Range 83°–87°)	0.08
Fuselage length	(Range 26′–30′)	0.12
Length, wing-to-nose	(Range 9′–11′)	0.08
Length, wing-to-tail	(Range 8′–10′)	0.10
Position of engines	NOSE	0.10
Number of engines	1	0.05
Tailspan	(Range 12′–14′)	0.08
Tail sweep, leading	(Range 87°–91°)	0.06
Tail sweep, trailing	(Range 86°–90°)	0.06

(d)

Fig. 14. Aircraft (Renegade) which illlustrates the object model refinement capabilities of the ORACLE system. (a) Distorted aircraft image. (b) Polygonal approximation of the aircraft. (c) Symbolic object features extracted from the aircraft. (d) EBL-generated object model for the aircraft.

The EBL-SCC learning cycle determines if the new feature is relevant in recognizing the aircraft and, if so, where it should be placed in the object classification tree. EBL processes the *Renegade* aircraft model using the three new object features. In this case, the leading tail angle is found to be relevant along with the trailing tail angle that was present, but not relevant, in the initial *Renegade* symbolic feature list. The revised object model is shown in Fig. 14(d). As in the previous model refinement example, SCC is given the revised model for insertion into the classification tree. SCC finds that at *TN-36*, the new leading tail angle is a better distinguishing feature than the current tail span feature (Fig. 13), so the classification tree is reclustered at *TN-36*. The final structure of the classification tree, after the object model refinement process, is shown in Fig. 15. It also includes the result of refining another aircraft model, viz. MD-11.

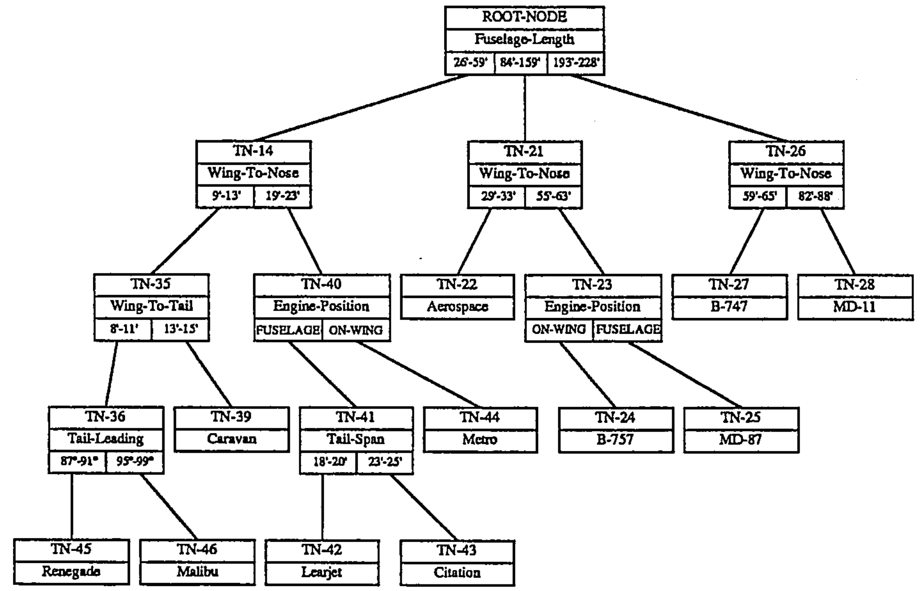
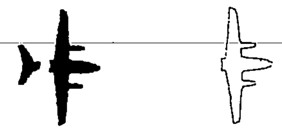


Fig. 15. Revised object classification tree, after refinement of the Renegade aircraft model.

3.7. Recognition Failure

As with any object recognition system, there will always be instances where the information processed by the system or the knowledge used to process the information is insufficient to perform the recognition task. Such an example is shown in Fig. 16(a) which contains a previously unknown aircraft image. The polygonal approximation (Fig. 16(b)) contains only the front part of the aircraft due to the separation in the fuselage portion of the image. The symbolic feature list obtained from this approximation is shown in Fig. 16(c). The knowledge-based matching component uses the feature data to parse the classification tree in Fig. 15. At the *ROOT-NODE*, the wing-span value is missing, so nodes *TN-14*, *TN-21*, and *TN-26* are hypothesized. However, at each of these nodes, none of the available branches are compatible with the aircraft's feature data, so the model matching process terminates. EBL is invoked to acquire the new aircraft, but is unable to generate an acceptable object model due to the absence of any tail features. Since EBL cannot process the available feature data, the aircraft is reported as a recognition failure.



(a) (b)

Feature	Value
Wingspan	63′
Wing sweep, leading	98°
Wing sweep, trailing	82°
Wing base chord	9′
Fuselage width	7′
Length, wing-to-nose	15′
Position of engines	ON-WING
Number of engines	2

(c)

Fig. 16. Aircraft (Merlin) which illustrates the recognition failure scenario in the ORACLE system. (a) Distorted aircraft image. (b) Polygonal approximation of the aircraft. (c) Symbolic object features extracted from the aircraft.

3.8. Evaluation of Recognition/Learning Performance

In order to quantify the performance of the ORACLE system, two sets of experiments were conducted to measure different aspects of system capability. The first set of tests allowed measurement of statistics relating the order of presenting object models to the system to the number of recognition/learning cycles. Twenty-four images (with varying levels of distortion, occlusion, etc.) of the aircraft models discussed earlier were used to perform these experiments. Nine images were used during system training to acquire the initial collection of aircraft models. Three of these training images were used simultaneously to acquire a single object model of the Boeing 747. Each of the remaining six images contained a different aircraft for a total of seven aircraft during training. Figure 17 summarizes the performance of the ORACLE software on the remaining fifteen (test) images.

Among the test images, five resulted in complete recognition, two in incomplete recognition, while three images provided correct occluded recognition results. The machine learning capabilities of the ORACLE system are evident from the fact that four images were used to acquire new object models and two of the five complete recognition cycles also caused the object model refinement process to be invoked.

Further, Fig. 17 indicates that in each of the five complete recognition cases, the feature value monitor was used to update the specific values of numeric features associated with the object models. Finally, one of the twenty-four images produced a recognition failure. However, none of the recognition results (complete, incomplete, or occlusion) resulted in a misclassification in these experiments.

Total number of images	24
Training images	9
Complete recognition	5
Incomplete recognition	2
Occluded recogntiion	3
Object model acquisition	4
Object model refinement	2
Feature value refinement	5
Recognition failure	1
Misclassification	0

Fig. 17. Summary of the ORACLE recognition experiments using the aircraft images. The final object classification tree contains 11 aircraft.

The second set of experiments was intended to evaluate the robustness and the extensibility of the ORACLE background knowledge when handling large model databases. Thirty different commercial aircraft (of which eleven were from the first experimental set) with ten images of each (total of 300 images) were used to generate the data set. An initial aircraft classification tree constructed from this data set is shown in Fig. 18. Each aircraft model was acquired by the EBL component during training using a good quality image of each object and the classification tree was computed in one pass. The EBL background knowledge as well as the SCC goal dependency network were not modified (with respect to the first set of experiments) in any way to handle the additional aircraft models. As in the earlier experiments, we see that the aircraft in the tree have been organized to differentiate between coarse categories of aircraft at the upper levels of the tree (passenger aircraft, business jets, private aircraft, etc.) and between subclasses of these categories at the lower levels of the tree. More tests will be performed to study the limits of the ORACLE system and enhance its capabilities further.

4. CONCLUSIONS

In this paper, we have presented the ORACLE, an object recognition system that exhibits the standard object recognition system functionalities (complete recognition, partial recognition, occluded recognition) as well as providing several new capabilities (object model acquisition and object model refinement). Since the ORACLE continuously adapts the features present in object models, it ensures timely response to changes in the dynamic environment. Additionally, through the use of the object classification tree, the ORACLE is able to recognize unknown

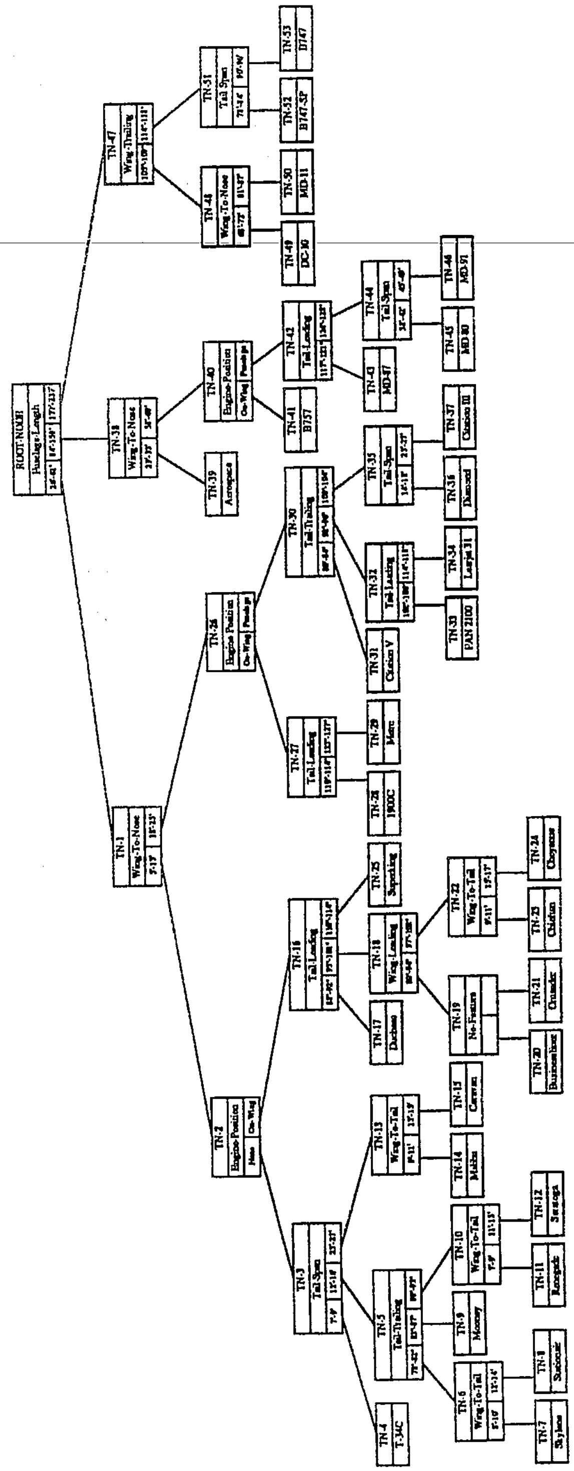


Fig. 18. Aircraft classification tree containing thirty distinct aircraft models.

objects without comparing them with the individual models of the database or processing them with extensive usage of knowledge. Consequently, the system can achieve high recognition speed. The contributions of the ORACLE to the field of machine learning are also noteworthy: modification of EBL to utilize multiple training examples to eliminate any model biases and to work with generalized feature ranges and weights in object models; simultaneous usage of symbolic information on feature utility and numeric information obtained from traditional clustering quality measures in the goal dependency network of SCC.

Although the experimental results presented in this paper are based on 2D object models, the ORACLE system can be extended to incorporate 3D models as well. The additional requirement to handle 3D imaging constraints, e.g. those posed by viewpoint location, is the availability of a feature extraction component that can locate and orient the position of a 3D object in the image and subsequently extract symbolic features. For example, one of our approaches utilizes a generic aircraft model to hypothesize various orientations and predict the appearance of specific target features.[11] Within the ORACLE itself, the only changes required are modifications of the EBL background knowledge to accommodate 3D features and that of the goal dependency network to allow proper structuring of the object classification tree. These issues are being currently investigated in the context of aircraft recognition in perspective aerial imagery.

ACKNOWLEDGMENTS

This work was supported in part by DARPA contract DACA 76-86-C-0017, DARPA, AFOSR grants F49620-95-1-0424, F49620-93-1-0624 and grants from other sponsors. The contents and information do not necessarily reflect the position or policy of the U.S. Government.

REFERENCES

1. P. J. Besl and R. C. Jain, "Three-dimensional object recognition", *ACM Comput. Surveys* **17** (1985) 75–145.
2. T. O. Binford, "Survey of model-based image analysis", *Int. J. Robotic Research* 1, 1 (1982) 18–64.
3. R. A. Brooks, "Symbolic reasoning among 3D models and 2D images", *Artif. Intell.* **17** (1981) 285–348.
4. R. T. Chin and C. R. Dyer, "Model-based recognition in robot vision", *ACM Comput. Surveys* **18**, 1 (1986) 67–108.
5. J. H. Connell and M. Brady, "Generating and generalizing models of visual objects", *Artif. Intell.* **31**, 2 (1987) 159–183.
6. S. Das, B. Bhanu and C.-C. Ho, "Generic object recognition using CAD-based multiple representations", *Proc. IEEE Second CAD-Based Vision Workshop* (Champion, PA, February 1994), pp. 202–209.
7. S. Das, B. Bhanu, X. Wu and N. Braithwaite, "Qualitative recognition of aircraft in perspective aerial images", *Image Technology*, ed. J. Sanz (Springer-Verlag, 1996) pp. 475–517.
8. G. DeJong and R. Mooney, "Explanation-based learning: an alternative view", *Mach. Learning* 1, 2 (1986) 145–176.

9. R. S. Michalski, "Knowledge acquisition through conceptual clustering: a theoretical framework and algorithm for partitioning data into conjunctive concepts", *Int. J. Policy Analy. and Inf. Syst.* 4 (1980) 219–243.
10. R. S. Michalski and R. E. Stepp, "Automated construction of classifications: conceptual clustering versus numerical taxonomy", *IEEE Trans. Patt. Analy. and Mach. Intell.* **5**, 4 (1983) 396–409.
11. J. C. Ming and B. Bhanu, "A multistrategy learning approach for target model recognition, acquisition, and refinement", *Proc. DARPA Image Understanding Workshop* (September, 1990), pp. 742–756.
12. T. M. Mitchell, R. M. Keller and S. T. Kedar-Cabelli, "Explanation-based generalization: a unifying view", *Mach. Learning* **1**, 1 (1986) 47–80.
13. H. Nasr, B. Bhanu and S. Lee, "Refocused recognition of aerial photographs at multiple resolution", *Proc. SPIE First Int. Conf. on Aerospace Pattern Recognition* (March, 1989), pp. 198–206.
14. *Proceedings IEEE Second CAD-Based Vision Workshop*, Champion, PA (February, 1994).
15. R. E. Stepp and R. S. Michalski, "Conceptual clustering of structured objects: a goal-oriented approach", *Artif. Intell.* **28**, 1 (1986) 43–69.
16. P. H. Winston, "Learning structural descriptions from examples", in *The Psychology of Computer Vision*, ed. P. Winston (McGraw-Hill, New York, 1975) pp. 157–209.

John Ming has worked at the NCR Human Interface Technology Center (HITC) as manager of the Image Understanding group since 1991. In this role, John has worked in multiple areas of imaging technology including: development of document image understanding applications for forms processing and high speed item processing applications; development of a new real-time image understanding system for monitoring customer behavior in retailing and banking environments, known as Customer Activity Analysis; and development of real-time image processing components to provide camera-based security for NCR's new Self Check Out Terminal for retail stores.

Prior to joining NCR, John worked as a research scientist and principal investigator at Honeywell's Systems and Research Center. In that role, John participated in several different projects which integrated computer vision and machine learning capabilities to provide solutions for different defense-related machine vision and photointerpretation problems.

Dr. Bhanu is currently a professor of electrical engineering and cooperative professor in computer science and Director of Visualization and Intelligent Systems Laboratory at the University of California, Riverside (UCR). Prior to joining UCR he was a Senior Honeywell Fellow at Honeywell Inc. He has been on the faculty of the Dept. of Computer Science, University of Utah, and has worked with Ford Aerospace & Communications Corporation, INRIA-France and IBM San Jose Research Laboratory. He has been the principal investigator of various programs from DARPA, NASA, NSF, AFOSR, ARO and other agencies and industries in the areas of object recognition, learning and vision, image understanding, image databases and machine vision applications. He has been the guest editor of the special issue of several IEEE transactions and other journals.

Dr. Bhanu is a Fellow of the IEEE for contributions to sensor-based navigation, automatic object recognition and closed-loop adaptive techniques for developing robust algorithms. Dr. Bhanu was the General Chair for the first IEEE Workshop on Applications of Computer Vision (Palm Springs, CA 1992), Chair for the DARPA Image Understanding Workshop, (Monterey, CA 1994), and General Chair for the IEEE Conference on Computer Vision and Pattern Recognition (San Francisco, CA, 1996). He is a member of ACM, AAAI, Pattern Recognition Society, and SPIE. Currently he is on the editorial board of the Journal of Mathematical Imaging and Vision, the Journal of Pattern Recognition, and International Journal of Machine Vision and Applications.

Dr. Bhanu received the S.M. and E.E. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology; the Ph.D. degree in electrical engineering from the Image Processing Institute, University of Southern California and the M.B.A. degree from the University of California, Irvine.

He also received the B.S. degree (with honors) in electronics engineering from Institute of Technology, BHU; the M.E. degree (with distinction) in electronics engineering from Birla Institute of Technology & Science, Pilani; and diploma in German from BHU.

His current research interests are computer vision, machine learning for computer vision, artificial intelligence, multimedia, pattern recognition, image processing, graphics, visualization, robotics and virtual reality.