Object Recognition and Performance Bounds*

J. K. Aggarwal and Shishir Shah

Computer and Vision Research Center
Department of Electrical and Computer Engineering, ENS 522
The University of Texas at Austin
Austin, TX 78712-1084, U.S.A.

Abstract. Object recognition is the classification of objects into one of many *a priori* known object classes. In addition, it may involve the estimation of the pose of the object and/or the track of the object in a sequence of images. Bayesian statistical pattern recognition, neural networks and rule based systems have been used to address the object recognition problem. In the case of statistical pattern recognition it is assumed that the *a priori* probability density functions are known or that they can be estimated from the given samples. For neural networks the samples may be used to train a network and the coefficients for the network function may be estimated. Whereas, in the case of the rule based system, rules may be given by an expert or they may be estimated from the samples. However, Bayesian framework provides a methodology for the estimation of error bounds on the performance of the recognition system. The paper discusses the Bayesian paradigm and contrasts its ability to provide performance bounds as compared to neural networks and rule based systems. Future direction of results on object recognition and performance bounds will also be discussed.

1 Introduction

Humans recognize objects and understand complex scenes with multiple objects, noise, clutter, occlusion, and camouflage with great ease. Humans are able to recognize as many as 10,000 distinct objects [Bie85] under varying viewing conditions, while a state-of-the-art object recognition system can recognize relatively few objects. We know very little about the physiological mechanisms with which the human visual system solves and uses solutions to lower-level processes such as depth and shape in the task of object recognition [CJR93]. Modeling human object recognition systems in terms of evidence-based systems accounts for the issues of view-independence, partial occlusions, variation between objects within object classes, and novel exemplar of object classes. As long as an object has enough similarity to the other objects in its class, the same set of evidence is accumulated, which helps in its recognition as a member of that object class. The evidence-based approach is also able to account for both perceptual and

* This work was supported by the Army Research Office Contracts DAAH-94-G-0417 and DAAH 049510494.

semantic considerations with explanatory efficiency. Due to the lack of working knowledge of the human visual system, there are no algorithmic descriptions for the human or other biological object recognition systems (ORS). Machine ORS have been driven to duplicate this diversity and remarkable performance. It is safe to say that machine ORS have progressed significantly in the past decade. A number of machine vision systems are now available in the marketplace for applications in inspection, target recognition, robotic manipulation, etc.

The dominant paradigm for object recognition in machine vision research is inverse optics, pioneered by Marr [Mar82]. Inverse optics is a bottom-up process where edges, surfaces, depth cues, etc. are identified before object recognition. While no precise definition of object recognition has been accepted, it is usually considered as the description of the three-dimensional object/scene that accounts for the two-dimensional imagery. It is perceived as a high-level task in computer vision, relating semantic knowledge in terms of a configuration of known objects [Ros84]. Object recognition is then achieved by comparing descriptions of *a priori* known object models, which are generalized descriptors that define object classes. In contrast to the bottom-up process, model-driven or top-down approaches to object recognition employ object models to predict image features and seek to find these features in the image or in a transformed feature space. In both approaches, the task of object recognition involves processing at all levels of computer vision. Typically, the input to the process is an image or a set of images from a sensor or multiple sensors. Some preprocessing is performed on the data and relevant information is extracted from the processed data and associated with a known description of the object. Therefore, object recognition involves lower-level vision, as with edge detection and image segmentation; mid-level vision, as with representation and description of pattern shape, and feature extraction; and higher-level vision, as with pattern category assignment or classification with an *a priori* known object descriptor.

In order to build a system that can achieve success in a realistic environment, certain simplifications and assumptions about the environment and the problem being tackled are generally made. This process of simplification introduces uncertainties into a problem that may create inaccuracies or difficulties in the system's reasoning abilities if these uncertainties are not represented and handled in a suitable manner. Some ways of dealing with uncertainty are by using: (1) methods that employ nonnumerical techniques, primarily nonmonotonic logic, (2) methods that are based on traditional probability theory, (3) methods that use neo-calculi techniques such as fuzzy logic, confidence factors and Dempster-Shafer calculus to represent uncertainties, and (4) approaches that are based on heuristic methods, where the uncertainties are not given explicit notations but are instead embedded in domain-specific procedures and data structures.

It is not the intent of the authors to present another review of object recognition systems. A number of good reviews of various paradigms and techniques have appeared in the past [AA93b, BJ85, CD86, SFH92]. The purpose of this paper is to look at the fundamental problems and discuss various ways of formulation in practical object recognition systems. This paper is organized into

the following sections: Section 2 briefly reviews the object recognition problem and some of the solutions proposed using both the model-based and bottom-up approaches. Next, a review of classification methods which allows for the incorporation of uncertainties into the system and provides a theoretical foundation for the inaccuracies in the reasoning ability of object recognition systems is presented in section 3. The classification paradigms considered are statistical or Bayesian, neural network based, and rule-based. We present a coherent comparison of the methods and discuss the ability of each in measuring the performance of the object recognition process by incorporating a degree of uncertainty. Finally, section 4 summarizes the trends of object recognition and discusses future directions of research.

2 Object Recognition

A wide range of approaches have been proposed and applied with limited success to the machine recognition of objects. Recognizing 3-dimensional (3D) objects from 2-dimensional (2D) images is an important part of computer vision [MA77]. The success of most computer vision applications (robotics, automatic target recognition, surveillance, etc.) is closely tied to reliable recognition of 3D objects or surfaces. The study of object recognition and the development of experimental object recognition systems has had a significant impact on the direction and content of computer vision research. Although a plethora of paradigms, algorithms and systems has been proposed over the past two decades, a versatile solution has not yet been developed; thus far, only partial solutions and limited success in constrained environments has been achieved. Practical implementation of an ORS can be viewed as a multi-stage process, as illustrated in Figure 1. Ideally, all objects of interest pass through each step and are included in the output list. As the data moves through the stages, the processing algorithms become more object specific and the number of data items processed and the number of false alarms decrease. The bottom-up approach mentioned briefly earlier has been successfully applied in a number of application. Here minimal amount of *a priori* information about the objects is used in the earlier part of the recognition process.

In model-based recognition, a 3D model(s) of the object(s) to be recognized is available. The 3D model contains concise and complete information about the object in terms of shape descriptions [VMA86], object parts information, relationship between object parts, etc. The 3D structure of an object is frequently represented by CAD models [AA93a], where volume-based representations of the object are built using primitives such as generalized cones, generalized cylinders and spheres. A method that uses a rectangular parallelepiped as the primitive volume element to represent objects was developed in [KA86]. Octrees [CA84] have also been used for the volumetric representation of objects. Typically, recognition involves extracting 3D information from the image and comparing it with the model features [AA93a], or deriving a 2D description from the image and then comparing it with 2D projections of the model. In using the former method,

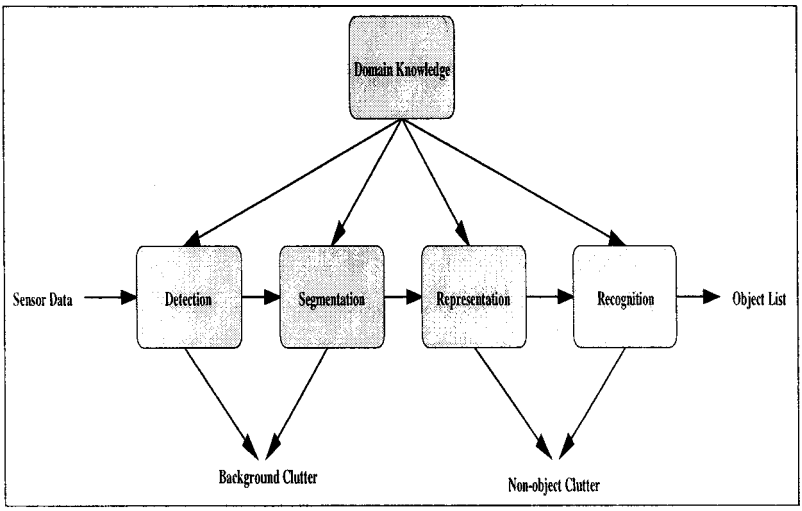

Fig. 1. Conceptual data flow in object recognition systems.

the sensing device should be able to provide 3D information in some form (such as range data or depth information using a stereo setup) which can then be compared with the model. In the latter case, the task is more difficult because (1) the effects of self-occlusions and perspective must be considered, and (2) the projection direction needs to be determined. In [WMA84], the 3D structure of an object is constructed using an observed sequence of silhouettes. During matching, the 3D structure of the unknown object is constructed from different image views, and more views are added to the construction process until features extracted from the object match one of the object models. A comprehensive survey of model-based vision systems using dense-range images is presented in [AA93b], and a recent survey is found in [Pop94].

View-based object recognition is often referred to as *viewer-centered* or *2D object recognition,* because direct information about the 3D structure of the object (such as a 3D model) is not available; the only *a priori* information is in the form of representations of the object viewed at different angles (aspects) and distances. Each representation (or characteristic view) describes the object from a single viewpoint, or from a range of viewpoints yielding similar views. Evidence shows that object recognition in human vision is viewer-centered rather than object-centered [KvD79]. The characteristic views may be obtained by building a database of images of the object or may be rendered from a 3D model of the object [PC93], [ZSB93]. Matching, in this case, is simpler than in model-based recognition because it involves only a 2D/2D comparison. However, considerable storage space is required to represent all of the characteristic views of an object. The number of model features to search among also increases, because each characteristic view can be considered to be a model. Methods have been developed to reduce the search space by grouping similar views [Pop94] [BR92], [PPK92].

Broadly speaking, there are two ways to approach this problem. The first is based on matching salient information, (e.g., corner points, lines, contours etc.,) that has been extracted from the image to the information obtained from the image database [MA77], [CJ93]. Based on the best match, the object is recognized and its pose estimated. The second approach extracts translation, rotation and scale invariant features (such as moment invariants [Hu62], Zernike moments [KH90] or Fourier descriptors [CH91]) from each image and compares them to the features that have been extracted from sample images of all the objects. The comparison is usually done in the form of a classification operation [DBM77].

Motivated by the human visual system, which strongly suggests a hierarchical approach to recognition, machine vision systems have been developed which attempt to mimic this process. Psychologists suggest that recognition of objects is guided by *perceptual organization* in the visual cortex. The principles of perceptual organization are the grouping of low-level, generic features to detect symmetry, collinearity, and parallelism from an input image. These principles have been shown to be useful in machine ORS, especially when no prior information of the image content is available [LA92]. Perceptual organization has been used to segment images into visible object surfaces [MN89]. Detection and recognition of various manmade objects in complex scenes has been accomplished using these principles. Most work in this area has concentrated on extracting groups of features, recognition of objects with exact models, and using additional sensing information.

In some sense, every object recognition algorithm is model based because every algorithm makes and uses *a priori* assumptions about the image and object characteristics. It would be difficult indeed to find an object about which we know nothing! With clutter, noise, occlusions, varying environmental conditions, and imperfect sensor information, these assumptions about the objects play an important role in the overall process. It becomes critical to incorporate a measure of uncertainty into the assumptions and algorithms we develop to evaluate the performance of the developed system. The final step for all recognition systems involves the classification of detected features to an *a priori* model. The success of this classification depends heavily on two main issues: (a) identifying the type of features to use in the matching, and (b) determining the best procedure to establish the correspondence between image and model features. The reliability and efficiency of an object recognition system directly depends on how carefully these issues are addressed.

3 Recognition Paradigms

A multitude of paradigms have been used to achieve success in constrained object recognition systems. Figure 2 shows the main technologies applied to this problem. Bayesian statistical pattern recognition, neural networks and rule based systems have been used extensively and successfully in addressing the object recognition problem. In this section we provide an overview of each of these methods and discuss their abilities to provide a performance measure.

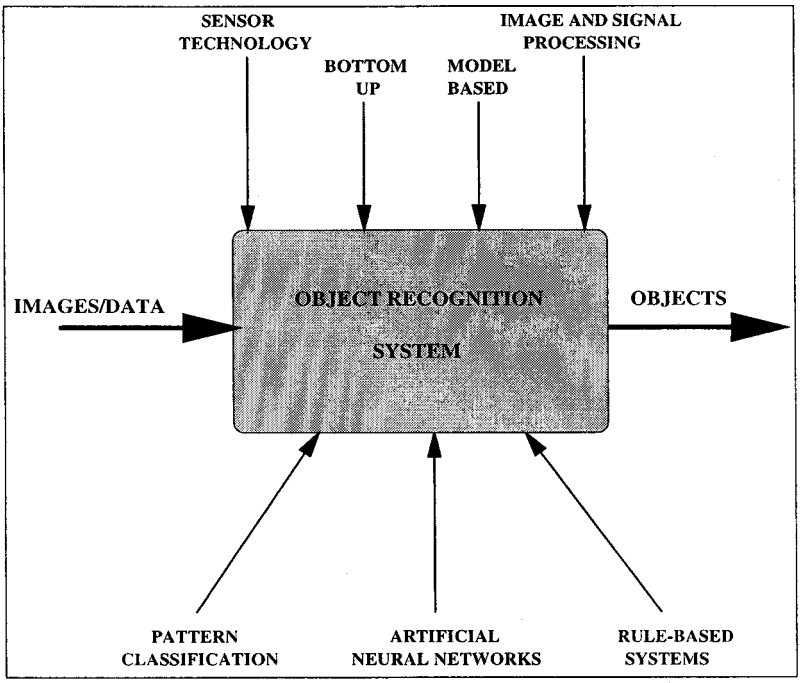

Fig. 2. Object recognition paradigms.

3.1 Bayesian Formulations

Bayesian methods provide a formal means to reason about partial beliefs under conditions of uncertainty [Pea88] [DH73]. Bayesian statistics have been used at various stages of the object recognition process to provide a firm theoretical footing as well as to improve performance and incorporate error estimates for the overall process. The biggest advantage of a Bayesian (or probabilistic) framework is in its ability to incorporate uncertainty elegantly into a process. Bayesian approaches also provide error estimates with their decisions, which give another perspective for analyzing systems. Bayesian statistics have been used in the object recognition paradigm for indexing, model matching and incorporating neighborhood relations under different contexts with some degree of success. In order to apply Bayes' theorem, one needs to have an estimate of the *prior* probabilities and also the underlying *likelihood* distributions. Depending on the application, different methods are used to determine these factors. *Prior* probabilities are usually estimated as the percentage of occurrence of the proposition over a period of time. The *likelihoods* are often estimated by making an assumption that simplifies the relationship between the hypothesis and the evidence. A commonly used assumption is that the evidence and the hypothesis are related by a normal (Gaussian) distribution.

Let us Consider a simple example. Suppose we are to recognize two objects, A and B, where the prior information is such that the object A occurs 70%

of the time and object B, 30% of the time. This provides the estimate of *a priori* probabilities, $P(A) = 0.7$, and $P(B) = 0.3$. Now consider that given the object data, we are able to extract a relevant feature for recognition, X. Thus the recognition problem can be posed as the identification of object A or B, given only the feature X. From a set of training samples, we can compute the parametrized density function that represents each of the objects. Assuming a normal distribution,

$$p(X|O) = \frac{1}{\sqrt{2\pi}\sigma_O} \exp \frac{-1}{2}(\frac{X - \mu_O}{\sigma_O})^2 \tag{1}$$

where O may be object A or B, μ_O and σ_O are the mean and variance for the respective object feature distribution. Given the prior probability and the likelihood, the posterior probability of recognizing the objects is given by the inversion formula,

$$P(O|X) = \frac{P(X|O)P(O)}{P(X)}, \tag{2}$$

The denominator $P(X)$, given by $P(X|A)P(A) + P(X|B)P(B))$, is a normalizing constant. Thus the recognition is based on deciding object A if $P(A|X) > P(B|X)$ and vice versa. In most practical formulations, the classification rule does not lead to perfect classification. One reason for this is that features are common to two or more classes and the regions for supports, or likelihoods, overlap. The Bayesian framework provides an estimate of the probability of classification error associated with each decision. These may take into account the significance of a classification error in addition to the probability of an error. The simple, two-object problem described above, which led to an intuitively appealing classification rule, can be extended to consider the probability of a classification error as a function of the measured feature X. We incur an error if we choose object B and the true class is object A or if we choose A and the true class is B. The error corresponding to this decision can be formulated as:

$$\begin{aligned} P(error|X) &= P(A|X) \; if \; we \; decide \; B \\ &= P(B|X) \; if \; we \; decide \; A \end{aligned} \tag{3}$$

The total error of classification can be expanded as:

$$\begin{aligned} P(error) &= P(error|A)P(A) + P(error|B)P(B) \\ &= P(X \in R_B|A)P(A) + P(X \in R_A|B)P(B) \\ &= P(X < \psi|B)P(B) + P(X > \psi|A)P(A) \\ &= P(B) \int_{-\infty}^{\psi} p(X|B)dX + P(A) \int_{\psi}^{\infty} p(X|A)dX \end{aligned} \tag{4}$$

The Bayesian formulation can easily be extended to n-classes, thus n different objects can be represented by parametrized density functions. The types and parameters of these functions can vary between different objects. To realize a Bayesian object recognition system, three main steps have to be followed:

1. Training, where the parameters θ_α, $1 \leq \alpha \leq n$, of the model density functions have to be estimated from a sample set of objects, A and B in our simple example.
2. Localization, where the image information is processed to estimate data that is most relevant to learned object models. This marks the use of relevant features X.
3. Recognition, where the localized image features are matched to the object model to determine the object class number α, by evaluating the discriminant function derived.

Generalizing for classification error in the n class decision problem, the expected risk or error is given by application of the total probability theorem [DH73]:

$$R[\psi(X)] = \int R[\psi(X)|X]p(X)dX \tag{5}$$

where $\psi(X)$ is the set of decision rules which maps the observed feature, X to its respective class.

For indexing formulation in object recognition, a feature set(s) (index vector) is identified that maps each unique object model (or part of a model) into a distinct point in the index space. This point is stored in a table with a pointer back to the object model. At runtime, the same type of feature set(s) are obtained from the image to form an index vector, which is then used to quickly access nearby pre-stored points. Thus a set of possible matches is found through correspondence of all possible image/model pairs. The distributions of the entries in the table could be organized based on similarities between object features or could be organized hierarchically such that the object classes are represented by a prototype table entry and further indexing is done to match the particular type of object within a class. Indexing using three points can be achieved using a probabilistic indexing scheme, which is based on the *probabilistic peaking effect* [BA90]. Alignment [HU90] and geometric hashing [GG92] are related techniques that are used for recognizing 3D object from 2D scenes. Both of these methods use a small number of points to find a transformation between the model space and the image space. Recognition then consists of finding evidence for instances of the models in the data, either by transforming the image into the model space and voting for an object's pose or by hypothesizing a pose and then transforming it into image space to guide the search. In [Wel93], a two-stage statistical formulation is used for feature-based object recognition. This work clearly shows how the Bayesian theory can be applied to model matching both in the *correspondence* space and the *transformation* space. A more detailed review of Bayesian techniques can be found in [AGNT96].

3.2 Neural Networks

Artificial neural networks (ANN) are motivated by biological systems which implement pattern recognition computations via interconnections of physical cells,

called neurons. The idea that the computations underlying the emulation of intelligent behavior may be accomplished by interactions of a large number of simple processing units is explored using ANNs. ANNs are highly parallel networks of simple computational elements (nodes) [JMM96], where each node performs operations such as summing the weighted inputs coming into it and then amplifying / thresholding the sum. The properties of the nodes, their interconnection topology (number of layers and number of nodes per layer), the connection strengths between pairs of nodes (weights) and the method used to update these weights (learning rule) characterize a neural network. Figure 3 shows a typical two-layer structure for an ANN. Neural networks are data-driven, and modifying patterns of internode connectivity as a function of the training data is the learning approach. In other words, the knowledge is stored in the form of network weights. Neural networks are trained so that subsequent associative behavior would recognize new patterns that are similar to the learned patterns. Learning in a neural network is usually performed using two distinct techniques: supervised and unsupervised. In supervised learning, the network is presented with both the input and the desired output for each input, and learning takes place to determine the weight structure that best realizes this input/output relationship. In unsupervised learning, the network is presented only with the input data and the network uses statistical regularities in the data to group it into categories.

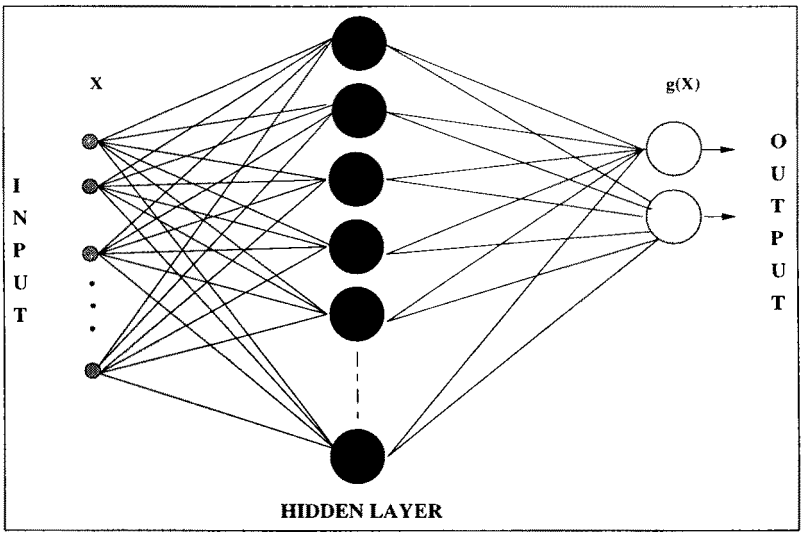

Fig. 3. A two-layer neural network.

Several types of neural networks can serve as adaptive classifiers that learn through examples and thus do not require a good *a priori* mathematical model for the underlying physical characteristics. These include feed-forward networks such as the Multi-Layer Perceptron (MLP), as well as kernel-based classifiers

such as those employing Radial Basis Functions (RBFs). A second group of neural-like schemes such as the Learning Vector Quantization (LVQ) have also received considerable attention. These are adaptive, exemplar-based classifiers that are closer in spirit to the classical K-nearest neighbor method. The strength of both groups of classifiers lies in their applicability to problems involving arbitrary distributions. Most neural network classifiers do not require the simultaneous availability of all training data and frequently yield error rates comparable to Bayesian methods without needing *a priori* information. Techniques such as fuzzy logic can be incorporated into a neural network classifier for applications with little training data. A good review of probabilistic, hyperplane, kernel and exemplar-based classifiers that discusses the relative merit of various schemes within each category is available in [NL91]. Although neural networks do not require geometric models, they do require that the set of examples used for training should come from the same (possibly unknown) distribution as the set used for testing the networks, in order to provide valid generalization and good performance on classifying unknown signals [GT94]. To obtain valid results, the number of training examples must be adequate and comparable to the number of effective parameters in the neural network. A deeper understanding of the properties of feed-forward neural networks has emerged recently that can relate their properties to Bayesian decision making and to information theoretic results [Bis95]. To compare the ANN structure to the Bayesian approach, consider the problem of recognizing 1 of C objects in an environment. The evidence of an observed object is given by the feature vector $X \in R^N$. Using the Bayesian decision, a network can be constructed where the hidden unit outputs represent the posterior estimates for each of the C classes and the final output unit performs the *max* operation. This is just the Bayes decision rule, as the probabilities are learned prior to their use in the network.

The common approach to learning in any feed-forward neural network is to perform gradient descent on a criterion function. For a simple two-class problem, the input features to the network are represented by $Z = (-1, X_1, X_2, \ldots, X_K)$ where -1 is provided as a bias term. The weights to H hidden units are initialized to be $\Lambda = (\theta, W_1, W_2, \ldots, W_J)$. The activation or output due to a single feature X is given as:

$$g(X) = W^t X + \theta \tag{6}$$

The weights to the network are learned over the entire training set, so the final decision function is:

$$g(Z) = \Lambda^t Z \tag{7}$$

and the class label is assigned based on the output sign. In order to learn the input-output relationship and update the weights, the criterion function to be minimized is chosen as the mean squared error. Thus, if the true output or class is d_i and the network output is $g_i(X)$, over M training patterns, the expected error cost is:

$$\Delta = E\sum_{i=1}^{M}[g_i(X) - d_i]^2 \tag{8}$$

If we consider a two-layer network with linear output units and hidden units with the logistic sigmoid activation, solving for a C class problem, the standard error function for each pattern to be minimized in an iterative manner is

$$E^m = \frac{1}{2}\sum_{k=1}^{C}(g_k - d_k)^2 \tag{9}$$

Using gradient descent, the derivative of the error is obtained by differentiating the error function with respect to the weights. As the output unit is linear, the error for each unit is simply given by

$$\delta_k = g_k - d_k \tag{10}$$

while for units in the hidden layer, the errors are

$$\delta_j = z_j(1 - z_j)\sum_{k=1}^{C} w_{kj}\delta_k \tag{11}$$

where z_j is the activation for the j^{th} hidden unit, w_{kj} are the weight connections between the hidden and output layer, and the sum runs over the output units. Thus the weight updates are given by

$$\begin{aligned} \Delta w_{kj} &= -\eta\delta_k z_j \\ \Delta w_{ji} &= -\eta\delta_j x_i \end{aligned} \tag{12}$$

for the output layer and hidden layer respectively, where η is the learning rate.

After learning the network weights, the resultant error indicates the performance of the network. Networks which can provide an estimate of probabilities associated with each decision can be used to determine the recognition/classification performance. Taking $P(X, C_j)$ to be the joint probability of input and the corresponding class label, and since $P(X, C_j) = P(C_j|X)P(X)$, the expected error cost can be evaluated as:

$$\begin{aligned} \Delta &= \int \sum_{j=1}^{C}\sum_{i=1}^{M}[g_i(X) - d_i]^2 P(X, C_j)dX \\ &= \int \sum_{j=1}^{C}\sum_{i=1}^{M}[g_i(X) - d_i]^2 P(C_j|X)P(X)dX \\ &= \int \sum_{k=1}^{C}[\sum_{j=1}^{C}\sum_{i=1}^{M}[[g_i(X) - d_i]^2 P(C_j|X)]P(X, C_k)dX \\ &= E[\sum_{j=1}^{C}\sum_{i=1}^{M}[[g_i(X) - d_i]^2 P(C_j|X)]] \\ &= E[\sum_{j=1}^{C}\sum_{i=1}^{M}[g_i(X)^2 P(C_j|X) - 2g_i(x)d_i P(C_j|X) + d_i^2 P(C_j|X)]] \end{aligned} \tag{13}$$

Simplifying using the expectation of true class given input features:

$$\Delta = E[\sum_{i=1}^{M}[g_i(X) - Ed_i|X]^2] + \underbrace{E[\sum_{i=1}^{M} Var[d_i|X]]}_{Independent\ of\ network} \tag{14}$$

The first term on the right is simply the mean squared error between the network outputs and the conditional expectation of the desired outputs. Thus, when the network parameters are chosen to minimize a squared error cost function, outputs estimate the conditional expectation. For a 1 of C classification, d_i equals 1 if the input X belongs to class C_i and 0 otherwise. Therefore,

$$\begin{aligned} E[d_i|X] &= \sum_{j=1}^{M} d_i P(C_j|X) \\ &= P(C_j|X) \end{aligned} \tag{15}$$

which are nothing but posterior probabilities. It has been demonstrated that classifiers provide outputs which accurately estimate known Bayesian probabilities and the outputs sum to one even though they are not explicitly constrained during training. More details regarding estimating probabilities in other networks and a survey of neural network approaches to machine inspection can be found in [Gho94].

3.3 Rule Based Approaches

Artificial Intelligence (AI) techniques have proven to fit well in high-level tasks that require reasoning capabilities and prior domain knowledge representation. A typical AI system has two main components: (1)*A knowledge-base* component which includes general facts about the application domain as well as task specific knowledge, and (2)*a control strategy* such as an *inference engine* which controls the reasoning or search process. The knowledge-base component of an AI system can be represented either as a set of procedures or in a declarative (i.e., non-procedural) fashion. Propositional logic, predicate calculus, decision trees, production rules, semantic nets, frames and slots, fuzzy logic and probabilistic logic are some of the commonly used knowledge representation techniques in the AI field. Although top-down (or goal-driven) and bottom-up (or data-driven) are the most commonly used control strategies, many successful AI systems use a hybrid top-down and bottom-up control strategy. Rule-based approaches have commonly been used in relation to object recognition systems due to their emergence from inductive learning and explanation-based learning. Reasoning under uncertainty is how humans perform object recognition, and it is rarely done with 100% certainty. Evidence supporting or refuting each particular decision is collected, examined, and weighed against all evidence supporting or refuting

other possible conclusions. Similarly, in real world complex problems such as machine ORS, some type of probabilistic or uncertain reasoning is required [SP90]. Consider a hypothetical rule, expressed in standard logical notation

$$a \wedge b \wedge c \wedge d \rightarrow O_1 \tag{16}$$

for recognizing an object O_1. An expert considering the same decision may choose the object despite lack of evidence for d if sufficient evidence exists for $a \wedge b \wedge c$. Without the knowledge of certainty in each of the evidences considered, it is hard to incorporate this notion into a rule-based ORS.

Rule-based paradigms provide a logical and understandable manner for using symbolic knowledge or domain knowledge in performing complex and heuristic tasks. Many object recognition systems have been developed based on these principles [Won87, Tou87, DMPA93, RH92]. In the overall structure of the object recognition paradigm, rule-based systems provide added advantages by increasing the system abstraction level, system maintainability, and uncertainty handling, providing reasoning and explanation capability, providing a built-in control strategy, and adding learning capabilities. Due to their use of symbolic representation, knowledge-based systems can be utilized to abstract many segmentation and labeling details. In rule-based ORS, the knowledge base and the matching criterion are two separate modules. Therefore, they both can be updated with little effort and time. Rule-based systems can handle uncertain decisions by attaching a measure of belief to each of their output decisions. In real object recognition applications it is important, if not essential, to have an explanation modality to clarify why and how a specific decision has been chosen over other decisions and to subsequently tune up the reasoning process. Reasoning and explanation capabilities are two unique features of rule-based systems. A rule-based system provides a built-in inference engine that can be used in a bottom-up, top-down or hybrid top-down and bottom-up fashion. A bottom-up control strategy can be used in a system when the noise level in the row data is low or when the search span in the solution space is large and hard to prune. In other cases, when there is a lot of interaction between data in the lower level tasks, a top-bottom or a goal-driven control strategy is more appropriate. However, in both cases, having a built-in control strategy with heuristic search criteria helps to reduce object recognition system complexity and implementation effort.

A system for multisensor image interpretation using a rule-based approach was developed by Chu and Aggarwal [CA95]. The AIMS (Automatic Interpretation system using Multiple Sensors) system has three main building blocks: (1) a segmentation module that integrates segmentation information from thermal, range, intensity, and velocity images and combines them into an integrated segmentation map; (2) a representation module, in which the outcome of the segmentation module is represented in a structural knowledge-based format that can be utilized by the KEE package; and (3) an interpretation module that uses KEE and supplementary LISP procedures, in a bottom-up manner to recognize different objects in an image. AIMS' reasoning process depends on knowledge in the form of rule-bases that are based on: (K1) knowledge of the imaging

geometry and device parameters, which are independent of the imaged scene; (K2) information on the segmented image regions, such as size, average temperature within the region, average distance, etc.; (K3) neighborhood relationships between the image regions; (K4) features and models of objects; and (K5) other general heuristics that are derived from known facts about the application domain and common sense. Using the above knowledge, a forward-chaining reasoning approach is adopted to recognize the objects that appear in an image, using six main consequent types of rules to: (R1) handle the difference between individual segmentation maps and the integrated segmentation map. These rules are also used to compute low-level attributes and place them in the corresponding knowledge structure. (R2) distinguish between man-made objects and background (MMO/BG). One such example is:

If (Segment A is relatively hot) AND
(Segment A has a compact contour)
Then (Segment A is a MMO, Confidence=Func(temperature, shape)),

(R3) to group similar segments (regions) into objects based on neighborhood relationships and other similarity measures. (R4) to classify back-ground (BG) into SKY, TREE, and GROUND types, and (R5) to classify man-made objects (MMO) into different types such as BULLETIN-BOARD, TANK, JEEP, APC, or TRUCK based on shape and size analysis. One such rule is:

IF (Segment A is of type MMO) AND
(Segment AS has a cool sub-region located at its lower-half) AND
(Segment A is about 2.0-2.5m high) AND
(Segment A has a trapezoidal contour) AND
THEN (Segment A is an APC with confidence of 0.8)

and finally, (R6) to verify the interpretation of an object and its surrounding objects. As example, a region recognized as a SKY cannot be surrounded by a region classified as GROUND. Any conflicting interpretations lead to reduced certainty factor recognitions.

Several algorithms have been developed for learning the domain knowledge from a set of learning examples in the form of a rule set. Sequential covering algorithms learn one rule at a time, subtracting out the covered examples and repeating the process on the remaining examples. In contrast, decision tree algorithms learn an entire set of disjuncts simultaneously as part of the single search for an acceptable decision tree. The main difference in the two approaches is in the partitions of data that they generate. Decision tree algorithms make fewer independent choices in selecting the precondition of each rule. Rule-based systems have been able to perform 3D shape recovery and orientation from a single view [SSY92]. The system uses some geometric regularity assumptions about perceived objects and image formation to recognize the objects from the 2D images. The system uses the expert system paradigms to perform some geometric reasoning from a given 2D image and form a set of possible 3D views and orientations that correspond to the given 2D object view. The reasoning process is

done in a forward-chaining fashion using OPS5, a production system language. The outcome of the reasoning process may result in more than one interpretation, each with an attached certainty factor that quantifies the system measure of belief in the recovered 3D object from the given prospective view.

Overall, current rule-based systems are limited in their ability to interpret typical knowledge bases in object recognition. Better outlier or exception dealing capabilities need to be explored along with retrieval or associative knowledge. Further, their capacity to evaluate error in the decision capability is limited and the authors are not aware of any means for characterizing the performance which can have a bound as provided through neural-network and Bayesian systems. Above all, better techniques to connect them with other kinds of representations need to be addressed, so that we can use rule-based approaches in object recognition systems in conjunction with different kinds of models and search procedures.

4 Future Directions

A number of distinct paradigms have been applied in our continuing attempts at development of machine object recognition systems. Most of them have been successful at least partially in constrained environments. A general purpose object recognition system is not in sight as yet. The object recognition problem is like an "elephant" being examined by a number of "visionless" persons. Each of the visionless persons gives a self-consistent and accurate description of the elephant. However, it would be difficult if not impossible to discover a complete description of the elephant from these partial "visionless" descriptions. Bayesian methodology is driven by probability theory. ANN methodology is motivated by the presumed behavior of a collection of biological neurons. Rule-based systems tend to emulate the presumed behavior of a human expert. An ideal ORS may be a combination of all three methodologies unified in a "visionary" fashion. Systematic methods and formalisms need to be developed for the design of hybrid systems consisting of the basic paradigms, performing characteristic tasks while simultaneously interacting with other modules. Such interaction would allow for the flow of information and decisions with competition and cooperation, all in the context of a global constraint, while minimizing the error in object recognition. The Bayesian paradigm in its formulation provides error estimates in the statistical formulation and yields similar estimates in the case of ANNs and possibly rule-based paradigms, especially if the relevant features have a Gaussian distribution.

References

[AA93a] F. Arman and J. K. Aggarwal. CAD-based vision: Object recognition in cluttered range images using recognition strategies. *Computer Vision, Graphics, and Image Procesing*, 58(1):33–47, 1993.

[AA93b] F. Arman and J. K. Aggarwal. Model-based object recognition in dense depth images - a review. *ACM Computing Surveys*, 25(1):5–43, 1993.

[AGNT96] J. K. Aggarwal, J. Ghosh, D. Nair, and I. Taha. A comparative study of three paradigms for object recognition: Bayesian, neural network and expert systems. In K. Bowyer and N. Ahuja, editors, *Advances in Image Understanding: A Festschrift to Azriel Rosenfeld*, chapter 15, pages 300–324. Springer-Verlag, 1996.

[BA90] J Ben-Arie. The probabilistic peaking effect of viewed angles and distances with application to 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(8):760–774, 1990.

[Bie85] I. Biederman. Human image understanding: Recent research and a theory. *Computer Vision, Graphics and Image Processing*, 32:29–73, 1985.

[Bis95] C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, New York, 1995.

[BJ85] P.J. Besl and R.C. Jain. Three-dimensional object recognition. *ACM Computing Surveys*, 17(1):75–145, March 1985.

[BR92] J. B. Burns and E. M. Riseman. Matching complex images to multiple 3D objects using view description networks. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 328–334, 1992.

[CA84] C. H. Chien and J. K. Aggarwal. A volume/surface octree representation. In *7th International Conference on Pattern Recognition*, pages 817–820, 1984.

[CA95] C.C. Chu and J.K. Aggarwal. The interpretation of a laser rader images by a knowledge-based system. *Machine Vision and Applications*, 4:145–163, 1995.

[CD86] R.T. Chin and C.R. Dyer. Model-based recognition in robot vision. *ACM Computing Surveys*, 18(1):67–108, March 1986.

[CH91] Z. Chen and S. Ho. Computer vision for robust 3D aircraft recognition with fast library search. *Pattern Recognition*, 24(5):375–390, 1991.

[CJ93] S. Chen and A. K. Jain. Strategies of multi-view multi-matching for 3d object recognition. *Computer Vision and Image Processing*, 57(1):121–130, 1993.

[CJR93] T. Caelli, M. Johnston, and T. Robinson. 3d object recognition: Inspiration and lessons from biological vision. In A. K. Jain and P. J. Flynn, editors, *Three-Dimensional Object Recognition Systems*, pages 1–16. Elsevier Science Publishers, 1993.

[DBM77] S.A. Dudani, K.J. Breeding, and R.B. McGhee. Aircraft identification by moment invariants. *IEEE Transactions on Computers*, C-26:39–46, 1977.

[DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. A Wiley-Interscience Publication, 1973.

[DMPA93] M. De Mathelin, C. Perneel, and M. Acheroy. IRES: an expert system for automatic target recognition from short-distance infrared images. In L.E. Garn and L.L. Graceffo, editors, *SPIE, Architecture, Hardware, and Forward-Looking Infrared Issues in Automatic Object Recognition*, volume 1957, pages 68–84, 1993.

[GG92] D. Gavrila and F. Greon. 3D object recognition from 2D image using geometric hashing. *Pattern Recognition Letters*, 13(4):263–278, 1992.

[Gho94] J. Ghosh. Vision based inspection. In C. H. Dagli, editor, *Artificial Neural Networks for Intelligent Manufacturing*, pages 265–297. Chapman and Hall, London, 1994.

[GT94] J. Ghosh and K. Tumer. Structural adaptation and generalization in supervised feedforward networks. *Journal of Artificial Neural Networks*, 1(4):431–458, 1994.

[Hu62] M. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, February:179–187, 1962.

[HU90] D. P. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with the image. *International Journal on Computer Vision*, 5(2):195–212, 1990.

[JMM96] A. Jain, J. Mao, and K. M. Mohiuddin. Artificial neural networks: A tutorial. In *Computer*, pages 31–44, March 1996.

[KA86] Y. C. Kim and J. K. Aggarwal. Rectangular parallepiped coding: A volumetric representation of three-dimensional objects. *IEEE Transactions on Robotics and Automation*, 2(3):127–134, 1986.

[KH90] A. Khotanzad and Y.H. Hong. Invariant image recognition by Zernike moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:489–497, 1990.

[KvD79] J. Koenderink and A. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[LA92] H. Q. Lu and J. K. Aggarwal. Applying perceptual organization to the detection of man-made objects in non-urban scenes. *Pattern Recognition*, 25(8):835–853, 1992.

[MA77] J. W. McKee and J. K. Aggarwal. Computer recognition of partial views of curved objects. *IEEE Transactions on Computers*, C-26(8):790–800, 1977.

[Mar82] D. Marr. *Vision*. W. H. Freeman, 1982.

[MN89] R. Mohan and R. Nevatia. Using perceptual organization to extract 3-d structures. *PAMI*, 11(11):1121–1139, November 1989.

[NL91] K. Ng and R.P. Lippmann. Practical characteristics of neural network and conventional pattern classifiers. In J.E. Moody R.P. Lippmann and D.S. Touretzky, editors, *Neural Information Processing Systems*, pages 970–976, 1991.

[PC93] A. Pathak and O. I. Camps. Bayesian view class determination. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 407–412, 1993.

[Pea88] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, Inc. San Mateo, California, 1988.

[Pop94] A. Pope. Model-based object recognition-a survey of recent research. *Technical Report*, TR-94-04, 1994.

[PPK92] S. Petitjean, S. Ponce, and D. J. Kriegman. Computing exact aspect graphs of curved objects: Algebraic surfaces. *International Journal on Computer Vision*, 9(3):231–255, 1992.

[RH92] E.M. Riseman and A.R. Hanson. A methodology for the development of general knowledge-based vision system. In C. Torras, editor, *Computer Vision: Theory and Industrial Applications*, pages 293–336. Springer Verlag, 1992.

[Ros84] A. Rosenfeld. Image analysis: Problems, progress and prospects. *Pattern Recognition*, 17(1):3–12, January 1984.

[SFH92] P. Suetens, P. Fua, and A.J. Hanson. Some computational strategies for object recognition. *ACM Computing Surveys*, 24(1):5–62, March 1992.

[SP90] G. Shafer and J. Pearl, editors. *Readings in Uncertain Reasoning.* Morgan Kauffman, Inc., 1990.

[SSY92] W.J. Shomar, G. Seetharaman, and T.Y. Young. An expert system for recovering 3D shape and orientation from a single view. In L. Shapiro and A. Rosenfield, editors, *Computer Vision and Image Processing*, pages 459–516. Academic Press, 1992.

[Tou87] J. T. Tou. Knowledge-based systems for robotic application. In A. Wong and A. Pugh, editors, *Machine Intelligence and Knowledge Engineering for Robotics Applications, Proc. NATO/ASI Workshop*, pages 145–189. Springer Verlag, 1987.

[VMA86] B. Vemuri, A. Mitiche, and J. K. Aggarwal. Curvature-based representation of objects from range data. *Image and Vision Computing*, 4(2):107–114, 1986.

[Wel93] W. M. Wells. *Statistical Object Recognition.* PhD thesis, Cambridge, MIT. November 1993.

[WMA84] Y. F. Wang, M. J. Magee, and J. K. Aggarwal. Matching three-dimensional objects using silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(4):513–518, 1984.

[Won87] A. Wong. Knowledge representation for robot vision and path planning using attributed graphs and hypergraphs. In A. Wong and A. Pugh, editors, *Machine Intelligence and Knowledge Engineering for Robotics Applications, Proc. NATO/ASI Workshop*, pages 113–143. Springer Verlag, 1987.

[ZSB93] S. Zhang, G. Sullivan, and K. Baker. The automatic construction of a view-independent relational model for 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(6):778–786, 1993.