A Comparative Study of Three Paradigms for Object Recognition - Bayesian Statistics, Neural Networks and Expert Systems. *

J. K. Aggarwal, Joydeep Ghosh, Dinesh Nair and Ismail Taha
Computer and Vision Research Center
The University of Texas at Austin, Austin, TX, USA
email: aggarwaljk@mail.utexas.edu

Abstract

Object recognition, which involves the classification of objects into one of many a priori known object types, and determining object characteristics such as pose, is a difficult problem. A wide range of approaches have been proposed and applied to this problem with limited success. This paper presents a brief comparative study of methods from three different paradigms for object recognition: Bayesian, Neural Network and Expert Systems.

1 Introduction

Recognizing 3-dimensional (3D) objects from 2-dimensional (2D) images is an important part of computer vision [1]. The success of most computer vision applications (robotics, automatic target recognition, surveillance, etc.) is closely tied with the reliability of the recognition of 3D objects or surfaces. The study of object recognition and the development of experimental object recognition systems has a great impact on the direction and content of research pursued by the computer vision community. Thus, it is not surprising that a plethora of paradigms, algorithms and systems have been proposed over the past two decades towards this problem. However, a versatile solution to this problem still evades the reach of even the best researchers, with only partial solutions and limited success in constrained environments being the state of the art. In fact, some researchers hold that it is not possible to design an object recognition system that is functional for a wide variety of scenes and environments and is still as efficient as a situation-specific system.

The difficulty in obtaining a general and comprehensive solution to this problem can be attributed to the complexity of object recognition in itself, as it involves processing at all levels of machine vision: 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, matching and reasoning (figure 1). The success of an object recognition system depends upon succeeding at all these levels. The task is made difficult by several factors, such as not knowing how many objects are there in an image, the possibility that the objects may be occluded, the possibility that unknown objects appear in the image, motion of the object, and variations

*This work was supported by the Army Research Office under Contracts DAAH 049510494 and DAAH 0494G0417.

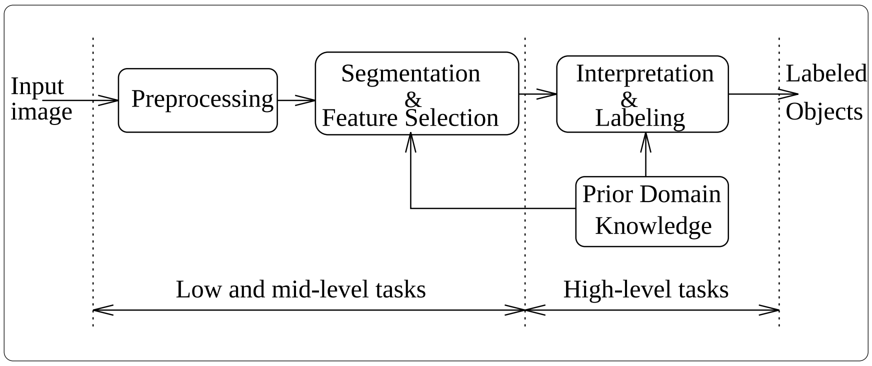

Figure 1. Levels of processing in an object recognition system.

in the sensing environment and the limits and accuracy of the sensor. In applications such as automatic target recognition (ATR), where targets have to be recognized in difficult outdoor scenes and adverse conditions, additional factors such as noise in the form of clutter, and deliberate mis-information such as camouflage, mislead the recognition system making the recognition process more difficult.

Recognition is the process of finding a correspondence between certain features in the image and similar features of the object model [2]. The most important issues involved in this process are: (a) identifying the type of features to use, and (b) determining the procedure to obtain the correspondence between image and model features. The reliability and efficiency of an object recognition system directly depends on how carefully these are addressed. Generally, recognition follows a bottom-up approach where the processing flows from the left to right in figure 1 and features extracted from an image are classified into one of many object types. However, attempts have also been made to approach the problem using a top-down perspective, where recognition is performed by determining if one of the many known objects appear in the image.

1.1 Model-Based Recognition

In this type of 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 [3], object parts information, relationship between object parts, etc. The 3D structure of an object is frequently represented by CAD models [4], where volume-based representations of the object is built using primitives such as generalized cones, generalized cylinders and spheres. Typically, recognition involves extracting 3D information from the image and comparing it with the model features [4], or deriving a 2D description from the image and then comparing it with 2D projections of the model. In using the former method, 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 a more difficult because the effects of self-occlusions and perspective must be considered, and the projection direction needs to be determined. In [5] 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 matches one of the object models. A comprehensive survey of model-based vision systems using dense-range images is presented in [6] and a recent survey is found in [10].

1.2 View-Based Object Recognition

View-based object recognition is often referred to as *viewed-centered* or *2D* object recognition, because direct information about the 3D structure (such as a 3D model) of the object 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 how the object appears from a single viewpoint, or from a range of viewpoints yielding similar views. There is evidence showing that object recognition by humans is viewer-centered rather than object-centered [7].

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 [8], [9]. Matching in this case is simpler than in model-based recognition because it involves only a 2D/2D comparison. However, the space requirements for representing all the characteristics views of an object tends to be considerable. Also, the number of model features to search among increases, because each characteristic view can be considered to be a model. Methods to reduce the search space have been addressed by grouping similar views [10] [11].

Broadly speaking, there are two ways to approach this problem. The first is based on matching salient information, such as corner points, lines, contours etc., that has been extracted from the image to the information obtained from the image database [1], [12]. 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, Zernike moments or Fourier descriptors) from each image and compares them to the features that have been extracted from example images of all the objects. The comparison is usually done in the form of a classification operation.

1.3 The Three Paradigms

In order to build a system that can achieve success in an 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 which may create inaccuracies/difficulties in the reasoning abilities of a system 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 non-numerical techniques, primarily non-monotonic 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. In this section we review three paradigms that are commonly used for handling uncertainties in realistic systems: Bayesian statistic, neural networks and expert systems.

1.3.1 Bayesian Statistics

Bayesian methods provide a formal means to reason about partial beliefs under conditions of uncertainty [13] [14]. Within this formulation, propositions (A) are given numerical parameters ($P(A|K)$) whose values signify the degree of belief given some knowledge (K) about the current environment or problem, and the parameters can be combined and manipulated according to the rules of probability. If the knowledge K remains unchanged then the degree of belief about a proposition is often represented by $P(A)$. In the Bayesian formalism, belief measures obey the axioms of probability theory: The essence of Bayesian techniques lies in the inversion formula,

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{1}$$

which states that the belief about any proposition A based on some evidence B can be computed by multiplying our previous belief about A by the *likelihood* that B will be true given that A is true. The denominator $P(B)$ is a normalizing constant. $P(A|B)$ is often called the posterior probability and $P(A)$ is often referred to as the prior probability. Bayes' theorem indicates that the overall strength of belief in a hypotheses A should be based on our previous knowledge and the observed evidence (B) and is obtained as a product of these two factors. In order to apply Bayes' theorem we need 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. Other attractive features of the Bayesian approach are: (1) the ability to pool evidence from different sources while making a hypothesis and (2) amenability to recursive and incremental computation schemes, specially when evidence is accumulated in a sequential manner.

1.3.2 Neural Networks

Artificial neural networks are inspired by biological neural networks. Artificial neural networks (ANNs) are highly parallel networks of simple computational elements (nodes) [15]. 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. Neural networks are data-driven, adaptive models that are mainly used for function approximation (classification) problems and optimization problems. This is normally done by defining an objective function that represents the status of the network and then trying to minimize this function using semi- or non-parametric methods to iteratively update the weights. 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 which 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. Generally, neural networks are often not used as stand-alone systems but as parts of larger systems as preprocessing or labeling / interpretation units.

1.3.3 Knowledge 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, as shown in Figure 2: (1)*A knowledge base* component which includes general facts about the application domain as well as task specific knowledge, and (2)*a*

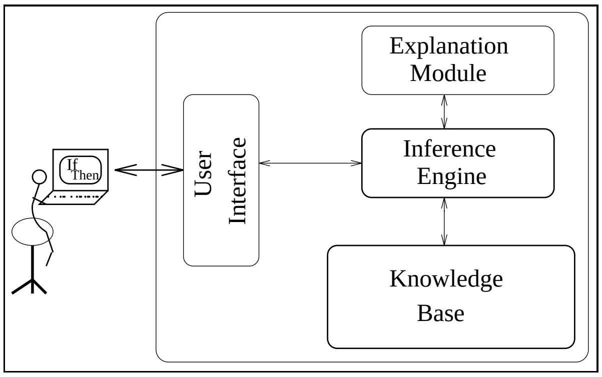

Figure 2. Main components of a knowledge-based system

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. Though 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.

In this article we shall largely restrict ourselves to expert systems. Our focus will be on expert systems, and see how this paradigm has been used as a tool in designing and developing powerful object recognition systems.

2 Bayesian Statistics in Object Recognition

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 into the recognition problem. The biggest advantage of a Bayesian (or probabilistic) framework is in its ability to incorporate uncertainty elegantly into the recognition process. Bayesian approaches also provide an error estimate with its decision, which gives another perspective for analyzing systems. Other advantages of using a Bayesian framework are [16]:

1. Modeling assumptions such as priors, noise distributions, etc., need to be explicitly defined. However, once the assumptions are made, the rules of probability give us an unique answer for any question that is posed.
2. Bayesian inference satisfies the likelihood principle where decisions are based on data that is seen and not on data that might have occurred but did not.
3. Bayesian model comparison techniques automatically prefer simpler models over complex ones (the **Occam's razor** principle).

On the other hand, since Bayesian decisions depend heavily on the underlying modeling assumptions, they are very sensitive to the veracity of these assumptions. Usually, to get a good description of the model a fairly large and representative amount of data is required.

Bayesian statistics have been used in object recognition for indexing, model matching and incorporating neighborhood relations under different contexts to some degree of success. Some representative applications of this framework are summarized in this section.

2.1 Indexing

Indexing is the process of finding the model from a database of models that best matches the features that have been extracted from an image. For indexing, 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 run time, feature set(s) of the same type 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 are found without actually comparing all possible image/model pairs.

An important issue in indexing is to make the extracted features relatively invariant to affine transformation and orthographic/perspective projections [17]. For indexing using three points on the object, an elegant approach based on the *probabilistic peaking effect* can be implemented. It has been observed that the probability density function of certain features in an image tend to peak at the values taken by the features in the model (*probabilistic peaking* effect) [18]. This means there is a large range of viewing directions over which these values change in the image by a small amount. For example, if the joint probability of the angle between segments from one of the three image points and the ratio of the lengths of these segments is determined over a viewing sphere, it peaks at the value of the corresponding features in the model. This information can be used to select only those matches whose feature values are fairly close to the image features and to disregard matches that have a small likelihood of being in actual correspondence.

The probabilistic indexing approach can also be extended to more that three points [19]. As a final note, the probabilistic indexing works well for most object recognition problems except when the objects are considerably foreshortened in the image due to rotation. In such cases they produce angles and distance ratios far from the probability peaks and will be difficult to recognize using probabilistic indexing techniques.

Alignment [20] and geometric hashing [21] [22] are related techniques that are used to recognize 3D object from 2D scenes. Both 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.

Alignment: is the process of determining a unique (up to a reflection) affine transformation between an image and the model by matching three image points with three model points. Once the transformation that brings them into alignment is determined, each of these transformations must be tested for correctness (verification). Speedup can be done using the probabilistic indexing scheme described earlier to determine which matches are most likely to be correct. Only these matches are considered and the rest are discarded. Other error measures can be used to further reduce the number of matches that need to be examined. Using this method the speedup obtained is equal to the fraction of the total matches indexed that are used for verification [19].

Geometric Hashing: for object recognition may be summarized as follows. Given m models $M_i, i = 1 \dots m$, each consisting of a pattern of n points ($p_i, i = 1 \dots n$), we pre-compute a hash table for these models where each entry is of the form $M(x, y, M_k, B_i, p_l)$, where a subset of the model points B_i (called a model *basis*) of the model M_k is combined with another point (feature) p_l of M_k not belonging to the set B_i, then it hashes to the location (x, y). The number of basis points is determined based on the application (e.g., 2 points for similarity-invariant recognition and 3 points for recognition under affine transformation and orthographic projection) and a separate entry in the hash table for all possible combinations of the distinct points (B_i, p_l) (called an *model group*) for every model M_k is generated. During the recognition phase, all features (points) in the image are detected and the following process is repeated until a recognition decision can be made: (1) an image basis is picked randomly and all image groups using this image basis is formed. The number of points in an image basis corresponds to the number of points in the model basis. (2) The location that each image group hashes to is located and all entries in the hash table (possible model group matches) register a weighted vote for each model group and the corresponding model basis. The weights depend on how close the image group hash location (u, v) is to the model groups location (x, y). (3) if a model (through the current model basis) receives sufficient votes, an object hypothesis has been found, otherwise the recognition process is repeated with a new image basis.

Geometric hashing is recast in probabilistic terms in [22] and geometric hashing with weighted voting is interpreted as a Bayesian maximum likelihood object recognition system. Using this interpretation, for the case where the basis comprises of two points, it is shown that voting weights (likelihoods) depend on the density functions of the hash values in the hash space under assumptions of a particular model/basis combination and a particular basis set selection in the scene. If it is assumed that the model points after undergoing translation, rotation and scaling are perturbed by a Gaussian-distributed noise before being projected to the scene, then the perturbations lead to an approximately Gaussian distribution of the hash locations, where the covariance matrix is diagonal and depends on both the location of the hash point and the separation between the basis points in the scene. Given a model M_k and its $(n-2)$ hash locations, the expected density function in hash space is simply the superposition of the $(n-2)$ Gaussian distributions, each centered around a distinct hash entry.

2.2 Model representation and matching

The matching process involves matching image features to model features. Matching can be performed either by trying to first get all good matches between the image features and the model features (commonly referred to as *correspondence* matching or *hypotheses* generation) or by trying to estimate the transformation of the image features needed to match the model features (i.e., trying to determine the object pose, commonly referred to as *transformation* matching), or a combination of both. Matching in the *correspondence* space is easily defined using Bayesian statistics. Also, the search in the correspondence space can by cast as an iterative estimation problem using the Bayesian theory.

Both techniques are exemplified by the work done by Wells [23], who uses a two stage statistical formulation for feature based object recognition. In the first stage (*correspondence*), the joint hypotheses of match and pose are evaluated in terms of their a posteriori probabilities for a given image. The model matching is cast as a maximum a posteriori (MAP) estimation problem using Bayesian theory and a Gaussian error model. The parameters that are to be estimated in the matching are the correspondences between the image and the object features, and the pose of the object in the image. The probability densities of image features, conditioned on the parameters of match and pose, are combined with the prior probabilities on the parameters using Bayes' rule to give the a posterior probability density of the parameters. An estimate of the parameters is then obtained by choosing them so as to maximize their a posteriori probability. The probability models for the features are built by assuming that matched features are normally distributed about their predicted positions in the image, and the unmatched features (considered as background features) are uniformly distributed in the image. The prior probabilities for the correspondences between the image and the object are assumed to be uniform for match features and constant for background features. Prior information on the pose is assumed to be supplied as a normal density.

The second stage of the statistical formulation (*transformation*), presents a method that builds on the earlier stage to provide a smooth objective function for evaluating the pose of the object without actually determining the correct match. To obtain the pose of the image, the posterior probability density of the pose is computed from the joint posterior probability on pose and match, by taking the marginal over all possible matches (for a given pose). Limited experiments show that this function is relatively smooth and its maximum is usually close to the correct pose. This maximum is then obtained by iteratively using a variant of the Expectation-Minimization (EM) algorithm to get the correct pose.

Markov random fields (MRFs): provide an efficient tool to incorporate neighborhood/ dependency constraints that can make the matching/recognition process more reliable and effective [24]. During the *hypothesis* generation stage, while trying to determine all possible matches between image features (such as regions or edges) and model features, it is important to include all possible correct matches and exclude as many incorrect ones as possible. Incorrect matches can be excluded by incorporating constraints based on prior knowledge about the problem (such as object models). Dependencies between hypotheses or features can be represented elegantly in the Markov random field framework. MRFs possess characteristics that are particularly suited to solution of spatially oriented vision problems that involve uncertainty (recognition). These include:

- MRFs fit conveniently into a Bayesian framework.
- Prior knowledge about local spatial interactions can be easily expressed.
- Probabilistic constraints based on arbitrary spatial dependencies and neighborhood relationships can be encoded easily.
- Local evidence about hypotheses is easily represented and manipulated.

For a review of MRFs and their applications to computer vision, the reader is referred to [25]. The use of a Markovian framework to improve the efficiency of object recognition is used in [26] where a probabilistic *hypothesis generation* or matching between features in an image and the model features is done. In this work, planar regions (R_i) extracted from range images are used as primitive features that are matched to faces on CAD models (M_j) of each object. Each planar region (feature) is represented by a fixed set of attribute values ($f^k_{R_i}, k = 1, \ldots n_1$) such as region area, second order moments, area diameter etc. Also, relationships between pairs of regions R_i and R_j are described by another set of attributes ($f^l_{R_i,R_j}, l = 1, \ldots, n_2$) such as simultaneous visibility, maximum distances between surfaces, etc. These region-based attributes are then then used to determine a set of model face matches using a Bayesian statistical approach. This is done by obtaining the statistical distribution of the observed attributes of each model face in the form of the conditional probabilities $P(f^k_{R_i}|M_j)$ and $P(f^k_{R_i,R_j})$ and the prior probabilities $P(M_j)$. During the hypothesis generation stage features are extracted from the image, and a set of possible matches of model features to image features is found. To reduce the number of incorrect matches, prior knowledge about the dependencies between different hypotheses is used to reduce the set of possible matches using Markovian framework. Each of the hypotheses can be thought of as being either correct (**ON**) or incorrect (**OFF**). The dependencies between hypotheses are incorporated in a Markovian framework as follows. Let X_{R_i,M_j} represent the hypothesis that the region R_i in the image is assigned to the model face M_j. Since each hypothesis can be either correct or incorrect, it has an associated value $\omega_{R_i,M_j} \in \{\mathbf{ON}, \mathbf{OFF}\}$. The variables X_{R_i,M_j} can be thought of as the MRF random variables (X) and the ω_{R_i,M_j} as the labels that these variables take. To define the dependencies between these MRF variables, two neighborhoods are defined: one for contradictory hypotheses and one for supporting hypotheses. Contradictory hypotheses are defined as those where the same image region is matched with two different model faces, while supporting hypotheses are those that are consistent with the prior constraints, i.e. $P(f_{R_i,R_k}|M_j, M_l) > 0$.

The matching process, which is equivalent to determining the most likely state of the MRF variables given a set of image regions $R_i, i = 1, \ldots, n$, reduces to determining the minimum of the posterior energy function:

$$U(\omega|R_1, R_2, \ldots, R_n) = \sum_{c \in C} V_c(\omega) - \sum_{X_{R_i,M_j} \in X} \log P(R_i|\omega_{R_i,M_j}). \tag{2}$$

The clique potentials $V_c(\omega)$ are easily obtained. For example, if only 1 and 2-cliques are considered, then for 1-cliques the potential at $c = \{X_{R_i,M_j}\}$ equals the prior probability $P(M_j)$ if the hypothesis is true and its complement otherwise. Similarly the 2-cliques energies are obtained by considering two related MRF variables $c = \{X_{Ri,M_j}, X_{R_k,M_l}\}$ in a neighborhood consisting of both contradicting and supporting hypotheses. The clique potentials are chosen according to the amount of supporting evidence that a hypothesis requires to survive the energy minimization process. The probabilities $P(R_i|\omega_{R_i,M_j})$ are found by determining the probability that attributes extracted for the region are similar to the model face M_j if the hypothesis is true ($P(f_{R_i}|M_j)$) and the complementary if the hypothesis is incorrect ($1 - P(f_{R_i}|M_j)$). Given a set of regions extracted from the image, the most likely set of hypotheses is obtained by minimizing the energy function in equation (2). Different techniques such as Highest Confidence First (HCF) [25] and Simulated Annealing [27] procedures can be used to get fairly good results. After the minimization process is completed using either of the procedures mentioned above, all matches that are **ON** are considered for verification. This verification is done by trying to estimate the pose of the object in the image.

View clusters: are often used to to reduce the time complexity involved with matching image features (regions) with all the possible model views. The clusters are obtained by grouping model views into equivalence classes using similarity metrics. Once the clusters are formed, each *view cluster* is represented by a prototype feature vector. This is a viewer-centered approach to object recognition in which matching is reduced to finding the closest view cluster(s) to which the image features may belong. Once the closest match(es) are found, the matching process can be refined to find the exact pose within each *view cluster*. The pose identification process can also be used to verify the initial match(es), as the candidate *view clusters* should have a good pose match with the image region (features). Bayesian statistics provide a complete framework for representing these *view clusters* and for the view class determination problem [8].

In [28] a hierarchical recognition methodology that uses salient object parts as cues for classification and recognition and a hierarchical modular structure (HMS) for parts-based object recognition is proposed. In this system, each level in the hierarchy is made up of modules, each of which is an *expert* on a specific part of a specific object. Each modular *expert* is trained to recognize the part under different viewing angles and transformations (translation, scaling and rotation). When presented with an input object part, each *expert* provides a measure of confidence of that part belonging to the object that the *expert* represents. These confidence estimates are used at the higher levels for refined classification. The modular *experts* (i.e., object parts) are modeled as a mixture density of multivariate Gaussian distributions. For each module, features (Zernike moments in the current implementation) obtained from the object part from all possible viewpoints are used in an *Expectation-Maximization* (EM) approach [29] to determine the module parameters.

When presented with an object part, each module computes the posterior probability of that part belonging to the object the module represents. These posterior probabilities are then pooled using a recursive Bayesian updating rule to compute the final object posterior probabilities given all the input parts. When computing the posterior probabilities of each object, the prior probability for each object part module is determined by its importance in the recognition process.

3 Neural Network Based Methods

Neural networks have been largely used as data driven models for function approximation or classification, or as networks which implicitly optimize a cost function that reflects the goodness of a match. Some promising neural approaches to feature extraction and clustering have also been proposed [36], which are adaptive on-line and may exhibit additional desirable properties such as robustness against outliers [30], as compared to more traditional feature extractors.

Several types of neural networks can serve as adaptive classifiers that learn through examples. Thus, they 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 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 [31].

Although neural networks do not require geometric models, they do 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 [32]. 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 relates their properties to Bayesian decision making and to information theoretic results [33]. A survey of neural network approaches to machine inspection can be found in [34].

3.1 Function Approximation for Object Recognition

In this section we describe neural network techniques that have been used for feature-based recognition and for indexing applications. These neural networks are mainly used to approximate certain functions, such as class optimizers (for classifiers) or for interpolation, using some training samples in a supervised fashion.

Feature-based object recognition: is a simplistic, direct approach where neural networks are used in the form of classifiers. Feature-based object recognition system using neural networks generally do the following:

1. Extract and select features from the objects that are *invariant* and/or *salient*. Different types of features (such as shape features, or intensity features, etc.) can be extracted from an image.
2. Features from a set of training images are then used to train a neural network classifier, either using supervised learning or in an unsupervised manner. During the training phase, the neural network can be made to learn different objects, and optionally, the pose of these objects. During the training phase, a set of images, not used for training, can be used to determine how well the neural network is performing.
3. Given a new image, the features extracted from the image are fed into the previously trained neural network, which then classifies the features and recognizes the object. The new features can also be used to further train the neural network.

Based on the above, the main issues in feature-based object recognition are (1) extraction and selection of invariant and salient features and (2) deciding on the type neural network classifier (architecture and size), and type of learning algorithms to use.

A good review of the various issues in feature selection/extraction for pattern recognition is presented in [35], [36]. Feature extraction and selection is used to identify the features which are most important in discriminating among different objects. Also, by retaining a small number of "useful" or "good" features, factors such as computational cost and classifier complexity are reduced. Feature *selection* implies the choosing a subset of the features. Feature *extraction* involves the transformation of the image and selecting a set of features from the transformed space. In object recognition, it is desirable to use features that are invariant to translation (i.e., position in the image), rotation and scale (viewing distance) of the object. A few of the commonly used features are invariant moments, log-polar transforms, shape descriptors such as Fourier descriptors and Zernike moments, and other local features such as curves and corner points, etc [38]. *Saliency* of a feature can be defined as the measure of the feature's ability to impact classification. One way to compare the saliency of features is by using the single probability of error criterion. This technique computes the probability of error separately for each individual feature. These errors are then used to rank the features and select a subset of them.

In [40], object recognition was performed using Zernike moments to represent the shapes of the object. Five different neural network classifiers were tested for this application, with the aim of comparing and evaluating their classification performances. These neural networks were a perceptron, a two-layer perceptron and the three-layer perceptron ART-2; an adaptive resonance theory based network; and a Kohonen associative memory [41]. The data set consisted of images of 6 tactical vehicles viewed from varying distances and angles and under conditions of noise and occlusions. Each object was represented by a vector of 23 features. For this application the multilayer perceptron with 2 hidden layers gave the best results.

In *view-centered* recognition, approaches based on *aspect graphs* are quite common. *Aspect graphs* [7] are created by representing 2D views of a 3D object along the nodes of the graph, with legal view transitions indicated by the arcs among the nodes. Each node represents a characteristic view (CV) of the object in which certain edges and faces are visible, as seen from a contiguous region of viewing angle and distances. Since each CV can be indicated by a binary vector with "1" for observable edges or faces and "0" for hidden ones, an object can be described as a mapping from (view angle, distance) to the CV vectors. In [43], an Radial Basis Function (RBF) network was used to learn this mapping and then predict the CV from a given view angle or to propose a view angle for a given CV.

Clustering techniques: are used in [44] to self-organize aspect graph representations of 3D objects from 2D view sequences. This architecture is based on a neural "cross correlation matrix" which was used to learn both 2D views and 2D view transitions and to associate the 2D views and 2D transitions with the 3D objects that produced them. The characteristic views of the different objects were represented using an Adaptive Resonance Theory (ART2) neural network through unsupervised learning and categorization. These 2D views were then fed into a series of cross-correlation matrices, or view graphs, one for each possible 3D object, so that views and view transitions could be learned by a 3D object categorization layer. The 3D categorization layer incorporated "evidence accumulation" nodes which integrate activations that they receive from learned connections to the correlation matrix. Decay terms in these integrator nodes determine how long they stay active without input support and, hence, determine the amount of evidence that is accumulated from different views of an object. The biggest drawback of this system was in its space (cost) requirements. To reduce the system cost, which is directly related to the complexity of the evidence accumulation parts of the architecture, the VIEWNET architecture proposed in [37] explores the problem of enhancing the preprocessing and categorizing stages in order to generate less ambiguous 2D categories and hence, rely, less on view transitions.

Indexing: as seen earlier, is an efficient method of recovering match hypotheses in model-based recognition. In a number of approaches, the indexing technique is viewed as obtaining indexing functions that associate each index vector from an image to some kind of probability measure with each of the indexed matches. One such method is presented in [45], where *indexing functions* are introduced to estimate these probabilities.

In [45], Radial Basis Functions (RBFs) are used to learn these functions. One advantage of using RBFs is that they smoothly interpolate between training examples (to fill in sections of the viewing sphere where no views exist). Also, a large number of training samples can be represented by a single center, thus reducing the storage requirements of the system. A drawback with using RBFs is that it is often difficult to determine the optimal number and positioning of the centers.

3.2 Matching as an optimization problem

As noted earlier, the main part of a recognition process is to establish the correspondence relationships between the information in the image and the object model. This may be posed as a graph matching problem, which in turn, is often formulated as an optimization problem where an energy function is minimized. In [46], a Hopfield network realizes a constraint satisfaction process to match visible surfaces to 3D objects. In [47], object recognition is posed an inexact graph matching problem and then formulated in terms of constrained optimization. In [48] the problem of constraint satisfaction in computer vision is mapped to a network where the nodes are the hypotheses and the links are the constraints. The network is then employed to select the optimal subset of hypotheses which satisfy the given constraints.

To convey the flavor of such optimization frameworks, we consider here the hierarchical approach of [49], where Hopfield networks are used to recognize objects at two levels: (a) coarse recognition, which is based on the surfaces of the objects, and (b) fine recognition, which is done by determining the correspondences between vertexes in the image and the model. At both levels, object recognition is viewed as a graph matching process differing only in the features being used for matching. At the coarse level, matching is done using surfaces as features, while at the finer level, matching is done using vertex (or corner points) information. The compatibility measures between the features are used to determine the network configuration and as the network iterates to a stable state, the number of active

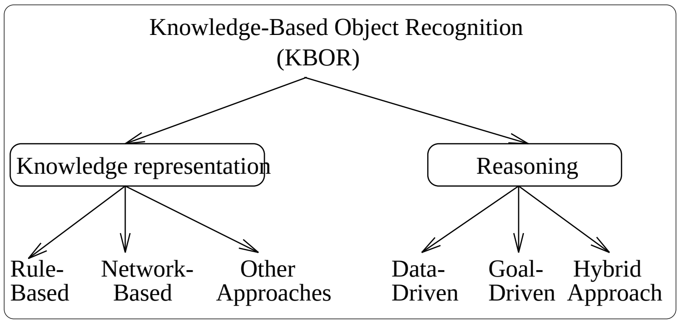

Figure 3. Most common classes of knowledge-based object recognition systems

neurons in the network can be used as a measure of matching between an image and a model. This information can then be used to select a few possible matching models from the model database for further verification. The verification stage (or fine recognition) is done by matching vertex (or corner points) of the image regions to the vertexes of the model in a similar manner.

4 Expert Systems

A Knowledge-Based Object Recognition (KBOR) system may be defined as an object recognition system that uses either a symbolic knowledge format to represent its domain knowledge and/or a knowledge-based inference engine to search its domain knowledge. Several KBOR systems were developed during the 1980s and 1990s [50, 51, 52, 53]. In most of these systems, the knowledge-based paradigm has helped to perform complex and heuristic tasks in a logical and understandable manner. Figure 3 shows some popular approaches taken by KBOR systems, which provide some of the following advantages:

1. *Increased system abstraction level*, due to symbolic representation.
2. *Increased system maintainability*, if the knowledge base and the matching engine can be updated separately.
3. *Better uncertainty handling*, by attaching a measure of belief to output decisions.
4. *Reasoning and explanation capability.*
5. *A built-in control strategy*, via the 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 KBOR system when the noise level in the raw data is low or when the search span in the solution space is large and hard to prune. In other cases, when there are many interactions among 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.

4.1 Examples

This section summarizes three object recognition systems that use a knowledge-based paradigm, specifically the expert system paradigm, as an integral part.

3D Shape and Orientation Recovery: Shomar et al. have implemented an object recognition system that depends on an expert system to perform 3D shape recovery and orientation from a single view [54]. This system has two main modules: an expert system module and a graphics display module. The system uses some geometric regularity assumptions about perceived objects and image formation to recognize the objects from 2D images. Geometrical reasoning is applied to each 2D image to form a set of possible 3D views and orientations corresponding to this given 2D object view. The search process is done in a forward-chaining fashion using OPS5, a production system language. The outcome of the reasoning process may result in multiple interpretations, each with an attached certainty factor that quantifies the system measure of belief in the recovered 3D object from the given perspective view. The main steps of this system can be summarized as:

1. *Representing geometrical rules:* The system has 35 geometrical heuristic rules that can be divided into five major categories: parallelism, perpendicular, right corners, parallel right corners, focal length, and hidden lines rules. These rules help in recognizing man-made objects since most man-made objects have some geometrical regularities. The first step in the system is to represent these rules in a production format using OPS5. Some supplementary functions were implemented using Pascal and Fortran procedures to facilitate low-level computation.
2. *The 3D reconstruction phase:* Each detected closed region in the given 2D view is assumed to correspond to a planar face in the 3D object. The system then utilizes the stored geometry regularities rules using the inference engine of the OPS5 to reconstruct the 3D object coordinates of the vertices. Each resulting reconstruction may result in different 3D recovery. However, a certainty factor is assigned to each recovered vertex based on the strength of the regularities used to reconstruct it.
3. *The graphical representation phase:* After constructing all possible 3D views from the given scene, the system uses a graphical illustration to display three orthographic views of the reconstructed model after applying some symmetry rules.

The key to the success of this system lies in constraining the domain to geometrical and unoccluded shapes. Regularities in this limited environment are then exploited to limit the search space.

A KB system for Image Interpretation: Chu and Aggarwal [55] developed a knowledge-based multi-sensor image interpretation system using KEE, an expert system shell development package. The AIMS (Automatic Interpretation 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 [56].
2. *A representation module* where the outcome of the segmentation module is represented in a structured knowledge-based format that can be utilized by the KEE package.
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 rules that are based on: (i) knowledge of the imaging geometry and device parameters, which are independent of the imaged scene; (ii) information on the segmented image regions, such as size, average temperature within the region, average distance etc.; (iii) neighborhood relationships between the image regions; (iv) features and models of objects; and (v) other general heuristics, which 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. Six types of rules are used sequentially: (i) pre-processing rules: to handle the difference between individual segmentation maps and integrated segmentation map and to compute low-level attributes and place them in the corresponding knowledge structure; (ii) coarse recognition rules: to distinguish between Man-Made Objects and Back-Ground (MMO/BG); (iii) grouping rules: to group similar segments (regions) into objects based on neighborhood relationships and other similarity measures; (iv) back-ground classification rules: to classify back-ground (BG) into SKY, TREE, and GROUND types; (v) man-made classification rules: to classify man-made objects (MMO) into different types such as BULLETIN-BOARD, TANK, JEEP, APC, or TRUCK based on shape and size analysis; (vi) consistency check rules: to verify the interpretation of an object and its surrounding objects. For example, a region recognized as a SKY cannot be surrounded by a region classified as GROUND. Any conflicting interpretations lead to reduced certainty factors. One such example is the rule:

IF (Segment A is of type MMO) AND
(Segment A 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)

This system is more versatile in that it allows for occluded objects and integrates knowledge of multisensor characteristics. However, the domain is again specialized, in this case to identify military objects from ground objects.

SIGMA: Matsuyama and Hwang have developed an image understanding system called SIGMA, which is a knowledge-based aerial image understanding system [57]. SIGMA uses expert systems in three different modules.

1. *A geometric reasoning expert* which extracts geometric structures and spatial relations among objects and represents them in a symbolic hierarchical knowledge. Bottom-up and top-down reasoning approaches are integrated into a unified reasoning approach, which is then used to construct a globally consistent description of the scene.
2. *A model selection expert* to reason about specialized objects that match the resulting general description of the geometric reasoning expert module. The model selection expert uses contextual goal reasoning to determined the most plausible object. However, this top-down reasoning approach is not enough to determine the most plausible object in each case. Thus, this module is used to compose objects into specific shape parts.
3. 3)*A low-level vision expert* is used to perform image segmentation and extract specific parts of objects features which the model selection expert has specified as its output. It uses a trial-and-error reasoning approach to find out segmentation features to help the other experts to reason about objects in the image. This module is the only expert in the system that uses domain-independent knowledge.

5 Future Directions in Object Recognition

In the preceding sections, through a review of some existing object recognition systems, we have highlighted the use of Bayesian statistics, neural networks and expert systems for object recognition. This discourse would be incomplete without mentioning that there are similarities among these paradigms and many of the reasoning/modeling abilities of one approach can be mimicked by the others. But, even more importantly, there are features of these approaches that are complementary in nature. To fully exploit these approaches to build robust and comprehensive object recognition systems, they have to be used concurrently in a mutually supportive manner. We now touch upon some areas of research that the authors believe are important and will influence the design of object recognition systems in the future.

5.1 Bayesian Methods and Neural Networks

Bayesian methods and neural networks share several similarities [16] [58] [59]. Both methods generate models that closely fit the data. Many popular artificial neural networks are essentially nonlinear parametric or semi-parametric estimators that are based on general and powerful functional forms such as a linear combination of sigmoidal or radial basis functions. The parameters are the weights which are "learned" or estimated using training data. Due to the specific types of non-linearities used, such functional forms are very flexible and can model complex variations in the data better than simple linear methods. However, with increased flexibility comes the potential problems of over-fitting and poor generalization. Bayesian methods, with their inherent preference for simpler models over complex ones, can complement neural networks by providing information about the amount of flexibility that is warranted by the data. This process can be facilitated by interpreting neural networks as probabilistic models which is possible with several neural networks that are used as regression networks as well as classifiers [16]. For example, the objective function (plus some regularization parameter) that is minimized during the training (weight change) of a neural network can be regarded as the negative log of the probability assigned to the observed data by the model with the current weights. As more data is seen, this objective function is updated to get the most probable weights given the data, using Bayes' theorem.

5.2 Combining Neural Networks and Expert Systems

The main motivation to combine expert systems and neural networks is to revise available domain knowledge and to augment the neural networks' output decision with explanation capabilities. Two popular methods of combining are presented in [60]. In one, an expert system is used to initialize a neural network. In this initialization, antecedents of rules are mapped into input and hidden nodes, certainty factors determine initial weights, while rule consequences are mapped into output nodes. The goal of this mapping is to embed all available prior domain knowledge into the network's initial internal architecture. Due to this embedded prior knowledge, the time required to train such networks is much less than those which are initialized randomly. In the other method, rules are extracted from trained neural networks. In this case, a neural network is mapped into a rule-based system. Rule extraction can provide trained (i.e., adapted) connectionist architectures with explanation power. Extracted rules can also be used to validate the connectionist networks' output decisions.

5.3 Pattern Theory: A Unifying Framework

The statistical framework of pattern theory provides mathematical representations of subject-matter knowledge that can serve as a basis for the algorithmic understanding of images [61] [62]. This powerful theory uses a modified Bayesian approach for hypothesis

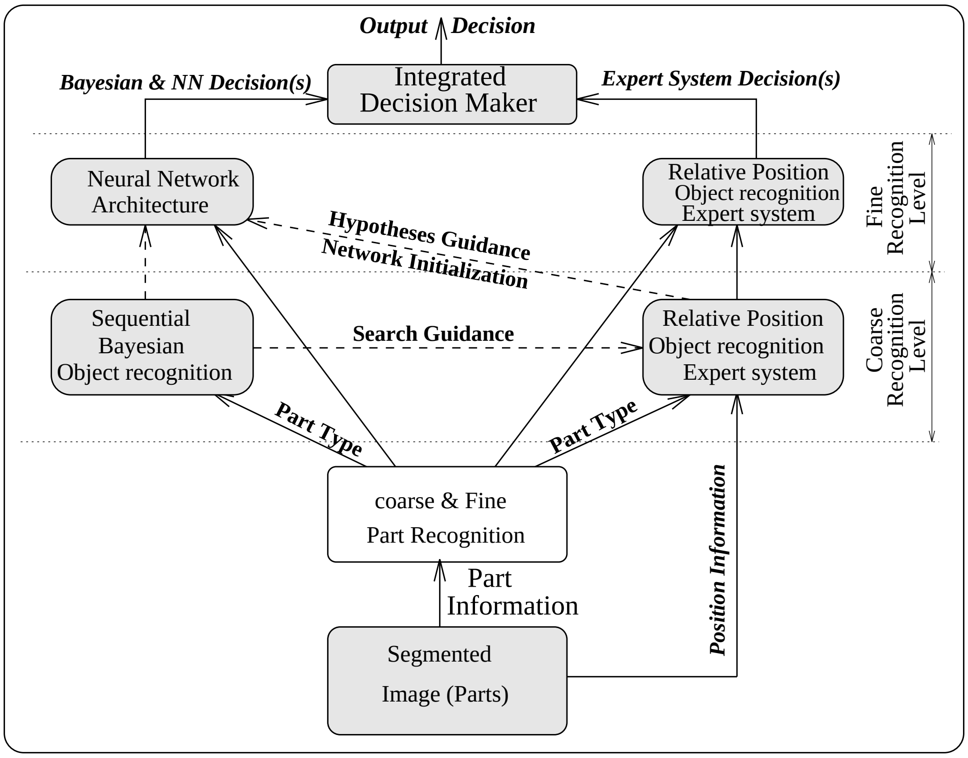

Figure 4. Schematic of a Mutually Guided Hybrid System.

formation that is capable of the creation or annihilation of hypotheses by jumps from one continuum to another in configuration (hypothesis) space. Also, rules can be represented by visual grammars that regulate transformations of or algebraic operations on pattern templates. Thus it provides a common language for both Bayesian and rule-based reasoning. Pattern theoretic approaches have met with success in a number of applications, from describing mitochondria ensembles to multi-target recognition and tracking [62] [63]. Moreover, mixed Markov models inspired by this theory are being suggested as basic tools in object recognition [64].

5.4 Combining all three paradigms: An illustration

To illustrate how all three paradigms can be tightly integrated, we briefly describe an ongoing project in which supplementary information from each sub-system helps to guide the working of the other, as exemplified by figure 4. We are currently implementing this hybrid system for object recognition from second generation Forward Looking Infrared (FLIR) images. This hybrid system uses a methodology based on a hierarchical, modular structure for object recognition by parts. Recognition is performed at different levels in the hierarchy, and the type of recognition performed differs from level to level. Each module is used to represent and recognize parts of objects. In the Bayesian sub-system, the final recognition of the object is based on the evidence accumulated from the sequential presentation of the different parts of the object. However, this does not exploit relative positional information of the different object parts while performing the final recognition. Relative positional information about object parts is readily incorporated into an Expert system, that uses the part evidence from the Bayesian part experts and the relative positional information from the image to recognize the object. However, searching through all possible part position combinations to arrive at a recognition is a costly process; a typical shortcoming of a bottom-up process. Some information about the type of the object can be used to guide the search, thus improving its efficiency. This information is provided from the Bayesian system at different levels. Recognition using relative spatial information of the parts as obtained using the Expert system is then used to inject all previously learned hypotheses into a neural network architecture, as described in [60].

6 Conclusions

In this paper, we present a comparative study of object recognition methods from three different paradigms: Bayesian, Neural Network and Expert Systems. Since object recognition is a difficult problem, a wide range of approaches, spanning across different theoretical paradigms, have been proposed and applied with limited success. In this paper, we have highlighted the use of these three different approaches to object recognition by reviewing some existing systems that display the features of these methods. These approaches have certain advantages and disadvantages, and the choice of a particular paradigm depends on the application at hand, the amount/accuracy of information about the environment, the available data and on the amount of "blind faith" in the system outputs that is tolerable.

Bayesian statistics seems a natural fit to object recognition problems because of its ability to handle uncertainties and provide error estimates. Also, given prior knowledge and some assumptions about the data, methods based on this approach always give consistent and concise solutions. However, the solutions are very sensitive to the underlying assumptions, and are only acceptable if the knowledge about the domain is quite reliable and well understood. Also, this approach can become computationally prohibitive if the size of the parameters needed to reasonably describe the problem becomes large. Neural networks, on the other hand, perform well in complex environments, with their data-driven ability to learn the underlying functionalities and their relationship to the application domain. However, neural networks suffer in their lack of interpretability and in incorporating prior domain knowledge. Another difficultly encountered when using either Bayesian statistics or neural networks is in incorporating high level (symbolic) reasoning capabilities into the system. This kind of reasoning is easily implemented using expert systems. Expert systems also provide an explanation for every decision that is made; which is a desirable property in any system. Based on the above discussion, it is clear that for systems to perform well in complex and dynamic environments, complementary features from each of these paradigms should be incorporated into a system in a mutually supportive manner. In this paper we have briefly addressed issues on how this may be done.

Even with the advance of technology and sophistication of object recognition algorithms, object recognition systems of today are still really limited when compared with human performance. Humans can recognize about 10,000 distinct objects [65] under varying conditions, while a state of the art object recognition system can recognize relatively a few objects and certainly are nowhere near the breadth and depth of the human performance. Automatic target recognition (ATR) is a good example of an exacting situation where the shortcomings of the state of the art systems are evident.

It is evident from the above discussion that object recognition remains an important problem to be resolved. This area has evolved significantly in the past two decades with applications into diverse areas ranging from recognizing targets in a battlefield to recognizing produce at checkout counters. However, we are still not able to design reliable object recognition systems. This may be partly attributed to the absence of theoretical underpinnings for object recognition systems which may enable us to analyze, synthesize and design such systems. It is envisaged that future efforts will be directed at fulfilling this need for theoretical underpinnings for pattern recognition and object recognition.

References

[1] J. W. McKee and J. K. Aggarwal, "Computer recognition of partial views of curved objects," *IEEE Transactions on Computers*, vol. C-26, no. 8, pp. 790–800, 1977.

[2] W. E. L. Grimson, *Object Recognition by Computer: The role of geometric constraints.* MIT Press, Cambridge, 1990.

[3] B. Vemuri, A. Mitiche, and J. K. Aggarwal, "Curvature-based representation of objects from range data," *Image and Vision Computing*, vol. 4, no. 2, pp. 107–114, 1986.

[4] F. Arman and J. K. Aggarwal, "CAD-based vision: Object recognition in cluttered range images using recognition strategies," *Computer Vision, Graphics, and Image Processing*, vol. 58, no. 1, pp. 33–47, 1993.

[5] Y. F. Wang, M. J. Magee, and J. K. Aggarwal, "Matching three-dimensional objects using silhouettes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, no. 4, pp. 513–518, 1984.

[6] F. Arman and J. K. Aggarwal, "Model-based object recognition in dense depth images - A review," *ACM Computing Surveys*, vol. 25, no. 1, pp. 5–43, 1993.

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

[8] A. Pathak and O. I. Camps, "Bayesian view class determination," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 407–412, 1993.

[9] 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*, vol. 15, no. 6, pp. 778–786, 1993.

[10] A. Pope, "Model-based object recognition-a survey of recent research," *Technical Report*, vol. TR-94-04, University of British Columbia, 1994.

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

[12] S. Chen and A. K. Jain, "Strategies of multi-view multi-matching for 3D object recognition," *Computer Vision and Image Processing*, vol. 57, no. 1, pp. 121–130, 1993.

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

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

[15] A. Jain, J. Mao, and K. M. Mohiuddin, "Artificial neural networks: A tutorial," in *Computer*, pp. 31–44, March 1996.

[16] D. J. MacKay, "Probable networks and plausible predictions - a review of practical bayesian methods for supervised neural networks." to appear in *Network*.

[17] Y. Lamdan, Y. Shwartz, and H. Wolfson, "Affine invariant model-based object recognition," *IEEE Transactions on Robotics and Automation*, vol. 6, no. 5, pp. 578–589, 1990.

[18] 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*, vol. 12, no. 8, pp. 760–774, 1990.

[19] C. F. Olson, "Probabilistic indexing for object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 5, pp. 518–522, 1995.

[20] D. P. Huttenlocher and S. Ullman, "Recognizing solid objects by alignment with the image," *International Journal on Computer Vision*, vol. 5, no. 2, pp. 195–212, 1990.

[21] D. Gavrila and F. Greon, "3D object recognition from 2D image using geometric hashing," *Pattern Recognition Letters*, vol. 13, no. 4, pp. 263–278, 1992.

[22] I. Rigoutsos and R. Hummel, "Distributed Bayesian object recognition," *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pp. 180–186, 1993.

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

[24] P. R. Cooper, *Parallel Object Recognition from Structure The Tinkertoy Project.* PhD thesis, University of Rochester, Rochester, New York, 1989.

[25] P. B. Chou and C. M. Brown, "The theory and practice of Bayesian image labeling," *International Journal on Computer Vision*, vol. 4, pp. 185–210, 1990.

[26] M. Wheeler and K. Ikeuchi, "Sensor modeling, probabilistic hypothesis generation, and robust localization for object recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 3, pp. 252–265, 1995.

[27] S. Geman and D. Geman, "Stochastic relaxation, gibbs distribution, and the bayesian restoration of images.," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 6, pp. 721–741, 1984.

[28] D. Nair and J. K. Aggarwal, "Hierarchical, modular architectures for object recognition by parts." submitted to *13th International Conference on Pattern Recognition.* November, 1996, Vienna, Austria.

[29] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm.," *Journal of the Royal Statistical Society*, vol. 39-B, pp. 1–38, 1977.

[30] L. Xu and A. L. Yuille, "Robust principal component analysis by self-organizing rules based on statistical physics approach," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 131–195, 1995.

[31] K. Ng and R. Lippmann, "Practical characteristics of neural network and conventional pattern classifiers," in *Neural Information Processing Systems* (J. M. R.P. Lippmann and D. Touretzky, eds.), pp. 970–976, 1991.

[32] J. Ghosh and K. Tumer, "Structural adaptation and generalization in supervised feed-forward networks," *Journal of Artificial Neural Networks*, vol. 1, no. 4, pp. 431–458, 1994.

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

[34] J. Ghosh, "Vision based inspection," in *Artificial Neural Networks for Intelligent Manufacturing* (C. H. Dagli, ed.), pp. 265–297, Chapman and Hall, London, 1994.

[35] A. K. Jain, "Advances in pattern recognition," in *Pattern Recognition Theory and Applications* (F. A. Denijver and J. Kittler, eds.), pp. 1–19, Springer-Verlag, 1986.

[36] J. Mao and A. K. Jain, "Artificial neural networks for feature extraction and multivari-

ate data projection," *IEEE Transactions on Neural Networks*, vol. 6, no. 2, pp. 296–317, 1995.

[37] G. Bradski and S. Grossberg, "Fast-learning VIEWNET architectures for recognizing three-dimensional from multiple two-dimensional views," *IEEE Transactions on Neural Networks*, vol. 8, no. 7/8, pp. 1053–1080, 1995.

[38] J. Wang and F. Cohen, "3D object recognition and shape estimation from image contours using B-splines, shape invariant matching, and neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 1–23, 1994.

[39] D. P. Casasent and L. M. Neiberg, "Classifier and shift-invariant automatic target recognition neural networks," *IEEE Transactions on Neural Networks*, vol. 8, no. 7/8, pp. 1117–1129, 1995.

[40] D. Nair, A. Mitiche, and J. K. Aggarwal, "On comparing the performance of object recognition systems," in *Proceedings of the Second IEEE International Conference on Image Processing*, (Washington D. C.), pp. 311–315, October 1995.

[41] S. Haykin, *Neural Networks: A Comprehensive Foundation.* MacMillan, 1994.

[42] T. Kohonen, *Self-Organization and Associative Memory.* Kluwer Academic Publishers, 1988.

[43] S. V. Chakravarthy, J. Ghosh, and S. Jaikumar, "Aspect graph construction using a neural network of radial basis functions," in *Proc. ANNIE 91*, pp. 465–472, Nov 1991.

[44] M. Seibert and A. Waxman, "Adaptive 3D object recognition from multiple views," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 3, pp. 107–124, 1987.

[45] J. S. Beis and D. G. Lowe, "Learning indexing functions for 3D model-based object recognition," *Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, pp. 275–280, 1994.

[46] B. Pravin and G. Medioni, "A constraint satisfaction network for matching 3D object," in *International Joint Conference on Artificial Intelligence*, vol. II, pp. 18–22, June 1989.

[47] E. Mjolsness, E. Gindi, and P. Anandan, "Optimization in model matching and perceptual organization," *Neural Computation*, vol. 1, pp. 218–219, 1989.

[48] R. Mohan, "Application of neural constraint satisfaction network to vision," in *International Joint Conference on Artificial Intelligence*, vol. II, pp. 619–620, June 1989.

[49] W. Lin, F. Liao, C. Tsao, and T. Lingutla, "A hierarchical multiple-view approach to three-dimensional object recognition," *IEEE Transactions on Neural Networks*, vol. 2, no. 1, pp. 84–92, 1991.

[50] A. Wong, "Knowledge representation for robot vision and path planning using attributed graphs and hypergraphs," in *Machine Intelligence and Knowledge Engineering for Robotics Applications, Proc. NATO/ASI Workshop* (A. Wong and A. Pugh, eds.), pp. 113–143, Springer Verlag, 1987.

[51] J. T. Tou, "Knowledge-based systems for robotic application," in *Machine Intelligence and Knowledge Engineering for Robotics Applications, Proc. NATO/ASI Workshop* (A. Wong and A. Pugh, eds.), pp. 145–189, Springer Verlag, 1987.

[52] M. De Mathelin, C. Perneel, and M. Acheroy, "IRES: an expert system for automatic target recognition from short-distance infrared images," in *Proceedings of SPIE, Architecture, Hardware, and Forward-Looking Infrared Issues in Automatic Object Recognition* (L. Garn and L. Graceffo, eds.), vol. 1957, pp. 68–84, 1993.

[53] E. Riseman and A. Hanson, "A methodology for the development of general knowledge-based vision system," in *Computer vision: theory and Industrial Applications* (C. Torras, ed.), pp. 293–336, Springer Verlag, 1992.

[54] W. Shomar, G. Seetharaman, and T. Young, "An expert system for recovering 3D shape and orientation from a single view," in *Computer vision and image processing* (L. Shapiro and A. Rosenfeld, eds.), pp. 459–516, Academic press, 1992.

[55] C. Chu and J. K. Aggarwal, "The interpretation of a laser radar images by a knowledge-based system," *Machine Vision and application*, vol. 4, pp. 145–163, 1995.

[56] C. Chu and J. K. Aggarwal, "The integration of image segmentation maps using region and edge information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 12, pp. 1241–1252, 1993.

[57] T. Matsuyama and V. Hwang, *SIGMA: A knowledge-based aerial image understanding system.* Plenum Press, New York, 1990.

[58] V. Cherkassky, J. Friedman, and H. W. (Eds.), *From Statistics to Neural Networks, Proc. NATO/ASI Workshop.* Springer-Verlag, 1995.

[59] I. Sethi and A. Jain, eds., *Artificial Neural Networks and Statistical Pattern Recognition.* Elsevier Science, Amsterdam, 1991.

[60] I. Taha and J. Ghosh, "A hybrid intelligent architecture for refining input characterization and domain knowledge," in *Proceedings of World Congress on Neural Networks*, vol. II, pp. 284–287, July 1995.

[61] U. Grenander, *General Pattern Theory.* Oxford Univ. Press, 1994.

[62] U. Grenander and M. I. Miller, "Representations of knowledge in complex systems," *Jl. of the Royal Statistical Society Series B*, vol. 56, no. 4, pp. 549–603, 1994.

[63] M. I. Miller, A. Srivastava, and U. Grenander, "Conditional-mean estimation via jump-diffusion processes in multiple target tracking/recognition," *IEEE Trans. Signal Processing*, vol. 43, pp. 1–13, November 1995.

[64] D. B. Mumford, "Pattern theory: a unifying perspective," in *Proc. 1st European Congress of Mathematics*, 1994.

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