Spatial Vision, Vol. 10, No. 1, pp. 87–103 (1996)

Machine learning paradigms for pattern recognition and image understanding

TERRY CAELLI[1] and WALTER F. BISCHOF[2]

[1]*Department of Computer Science, Curtin University of Technology, GPO Box U1987, Perth, WA 6001, Australia*

[2]*Department of Psychology, University of Alberta, Edmonton, Alberta, T6G 2E9, Canada*

Received 12 July 1995; accepted 30 October 1995

Abstract—In this paper some issues are considered related to the encoding of spatial information and associated perceptual learning algorithms which, it is claimed, are necessary for robust pattern and object recognition in multi-object (natural) scenes. The types of learning requirements within a 'recognition-by-parts' paradigm are contrasted with findings from alternative models.

1. INTRODUCTION

Over the past decades, an enormous effort has been devoted to determining how biological visual systems decompose the sensed world into salient features or parts. Questions about these processes have varied from very basic problems of 'foreground-background' segregation (for example, Julesz, 1984) to more complex problems of what constitute 'parts' of 3D objects (for example, Hoffman and Richards, 1986; Biederman, 1987). In a similar way, most computational models for computer pattern and object recognition are based on image segmentation, feature and part extraction procedures. Structures (patterns, textures, objects) are typically defined by descriptions that capture both first- and second-order attributes of parts and their relations (for example, Jain and Hoffman, 1988; Fan *et al.*, 1989; Bischof and Caelli, 1994). The net result of this work has been a large range of procedures for encoding attributes of image parts and part relations upon which recognition and interpretation processes are enacted. Just how such feature encoding strategies are useful for recognition is the topic of this paper—using a perceptual learning perspective.

One of the more fundamental issues in modeling perceptual learning is that of encoding or representation: the domain upon which learning occurs. For example, many theories of image coding are direct in so far as image information processing is proposed in terms of the successive transformations and extraction of image features which are indexed or stored in, essentially, an image format ('implicit' image or pixel (positional) indexing of information). Other models assume that images are encoded

or indexed by features, and their attributes, which are stored in some feature space or list: 'indirect' or 'explicit' coding schemes.

It is important to note that the connectionist approach to perception supports both coding models. Neural networks, associative memories, etc., are concerned with techniques for reducing redundancy in signals and with learning classification rules in the most general sense. The ensuing rules in such systems are contained within the various layers beyond the input and the learning (search) technique employed. That is, they apply to any representation of the signal and at any level—at least in principle. They also share in common the minimum requirement from a statistical pattern recognition perspective: patterns are represented as vectors of characteristic features which are chosen to optimize representational uniqueness of patterns belonging to different classes. Pattern classification can be achieved by partitioning attribute spaces into regions associated with different pattern classes such that classification rules minimize misclassification while, at the same time, maximize the simplicity of attribute space partitions. In a formal sense 'features', here, refer to 'parts' which have attributes such as the intensity, length, orientation and size of specific pattern blobs, as well as part relations.

Most current pattern recognition systems have both unsupervised and supervised components where the former typically are aimed at deriving parts or clusters of attributes which are of focal interest (see, for example, Poggio and Edelman, 1990; Milanese *et al.*, 1994) followed by a supervised domain where the aim is to determine the degrees to which each such grouping can evidence different types of patterns. What differentiates most learning procedures is the type of cost function, the search algorithm and the types of constraints placed upon rule generation (generalization). For example, with unconstrained backpropagation neural networks the rules correspond to arbitrarily oriented regions of attribute space (without part labeling) while for decision trees rules correspond to regions of attribute space (without part labeling) defined by conjunctions of attribute bounds. The hyper-BF system of Poggio and Girosi (1990) corresponds to fuzzy rules defined by the locations and spread of the fitted radial basis functions—a type of Bayesian clustering pre-processing stage described by Cheeseman *et al.* (1990).

The most common approach to visual learning has been inductive or supervised learning where the task is to generalize from training samples and classify new samples according to the learned rules. Such approaches have been quite successful for simple isolated patterns (see, for example, Rentschler *et al.*, 1994). However, they do not perform well when pattern complexity is high, as is the case with 3D object recognition, or with complex and highly similar 2D patterns. Most importantly, they do not perform well with complex scenes as they are developed to classify isolated patterns. To achieve pattern identification in complex scenes, it is necessary to group regions and parts.

To overcome such problems, we have considered 'visual learning' in relational terms. Here, patterns are explicitly described as being composed of constituent parts and pattern descriptions involve enumeration of both (unary) part and part-relation attributes. These attributes can be linked together into relational structures that define patterns uniquely. Pattern classification is achieved using relational graph matching

where a (new) sample pattern is matched to a model pattern by searching for a label assignment that maximizes some objective similarity function. Pattern classes are usually represented by enumeration of instances, and classification is achieved by searching through all model graphs to determine the one producing the best match. Indeed, this representation and the associated graph matching approach—in the form of interpretation trees and feature indexing—has been the preferred architecture for machine object recognition (Flynn and Jain, 1993). What is novel about our approach is that we have investigated the use of Machine Learning to actually 'pre-compile' the optimal search strategies for matching (including solving correspondence problems) and generalization from training samples—as described below.

The background to our work lies in that of endeavoring to integrate two quite different representations and technologies: Graphs and Evidence Theory. In the former case structures are represented by graphs where parts are defined by nodes and relations by edges. Solutions for matching models to data typically involve solving correspondence problems or the sub-graph isomorphism problem. This relational graph matching approach to pattern and object recognition has several weaknesses. First, the computational complexity is exponential. This is a significant problem since the cardinality (order) of such algorithms is defined by the number of model and sample parts. Second, pattern generalization is difficult to represent. One solution has been to represent pattern classes via a typical class member (prototype) and a distance measure between data and models (Shapiro and Haralick, 1982). See also Suetens *et al.* (1992) for a more general discussion on this issue.

On the other hand, Evidence Theory is inherently non-relational in the sense that models are defined by lists of rules pertaining to the attributes of parts and part relations without considering part labels (see below). Recognition occurs by comparing part and relational attribute states to rule bounds which, if satisfied, contribute evidence for different models. Combining this evidence then results in pattern recognition.

Our specific aim, here, is to describe how to combine the relational-structure representation with learning and generalization in three different ways: Evidenced-Based Systems (EBS), Rulegraphs (RG) and conditional rule generation (CRG). In all three cases, the system learns to describe or represent patterns in terms of rules about pattern part attributes and their relations which are encoded by region (volume) bounds in unary (part) and binary (relational) feature spaces.

The three approaches differ in their way of dealing with the label compatibility problem, that is, in the way they ensure compatibility between instantiations of unary and binary rules. Two of the approaches, Rulegraphs and Conditional Rule Generation, contain explicit label compatibility checks; the third, and weaker form, Evidence-Based Systems does so weakly. Finally, we will consider the additional structures and constraints required to apply such learning and recognition algorithms to the labeling of complex scenes as found in image understanding problems.

2. EVIDENCE-BASED SYSTEMS

Pattern and object recognition are often difficult problems because parts of different patterns or objects can be quite similar: sharing similar attributes (regions in feature

space). Within-class variance may thus exceed between-class variance substantially. The EBS solution to this problem involves the development of an intermediate representational stage where an attempt is made to capture the predominant characteristics of training samples by clustering (grouping) them into different regions of the pattern's attribute spaces (unary and binary). The bounds on such regions are used as conditions for rule activation, and *evidence weights* are determined for such bounds (regions) corresponding, typically, to the probabilities of a class, pattern or object, given such feature attributes. Such rules are of the general form:

if
(attribute—unary or binary—states are within bounds)
then
(evidence for class X is Y).

2.1. Rule generation

Generation of rules and evidence weights involves the use of clustering algorithms. In this model we have generated clusters defined by lower and upper bounds of feature attributes such that the rule conditions (see above) are defined by conjunctions of the respective bounds of feature attributes (hyper-rectangles oriented along the feature space axes)—consistent with earlier notions of perceptual feature integration involving the use of (logical) *conjunctions* of attribute states (Treisman and Gelade, 1980).

In Jain and Hoffman's (1988) 3D object recognition system, rules were generated by clustering the samples in feature space using a minimum spanning tree technique. In our work (Caelli and Pennington, 1993 and Caelli and Dreier, 1994), we have used minimum entropy clustering which endeavors to change the position and size of a fixed number of rectangles (clusters) to maximally separate the occurrences of class samples per cluster. In other words, we relabel the cluster membership of each sample i to minimize the entropy function

$$H_{\min} = \min_{i \in j} \left\{ - \sum_j \sum_k p_{jk} \ln p_{jk} \right\}, \tag{1}$$

where p_{jk} is the probability of class j occurring in cluster k, with the probability being determined from the relative frequency of class samples within a given cluster solution. The difference between clusters generated by the minimum spanning tree and the minimum entropy methods is illustrated in Fig. 1 in comparison to the types of rules generated from neural networks (Fig. 1(a)) where attribute regions are defined by zeros of each perceptron beyond the input layer (see, for example, Lippman, 1987). That is, the essential differences between neural networks and the types of rules defined here lie in how the conditions for rule activation are defined. Both, however, partition attribute spaces to enable generalization from training data and both permit weight estimation for the evidence vectors. Indeed, other forms of rule generation can also be used. For example, the Hyper-BF model of Poggio and Girosi (1990)

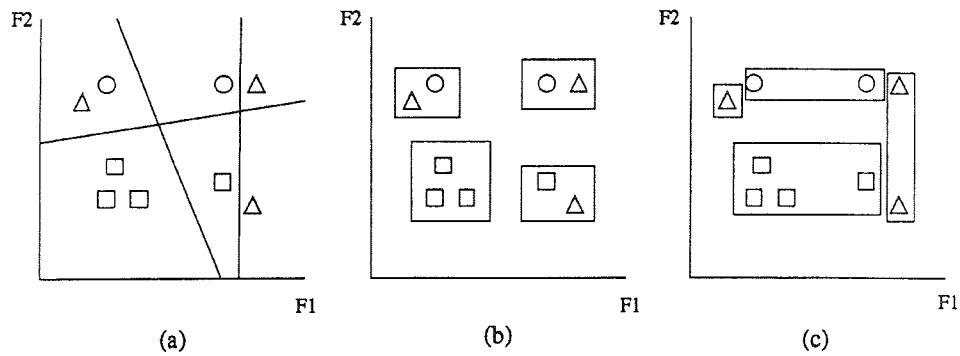

Figure 1. (a) Rule generation using a neural network. Each line corresponds to the zeros of a hidden unit element. Regions are generated by maximizing class classification with respect to the position and orientation of the lines (in general, hyperplanes). (b) Rule generation by an EBS using Minimum Spanning Tree clustering. Boundaries of clusters are orthogonal to feature axes to allow conjunctive rule forms. (c) Same as (b) except that classification performance was maximized using minimum entropy clustering.

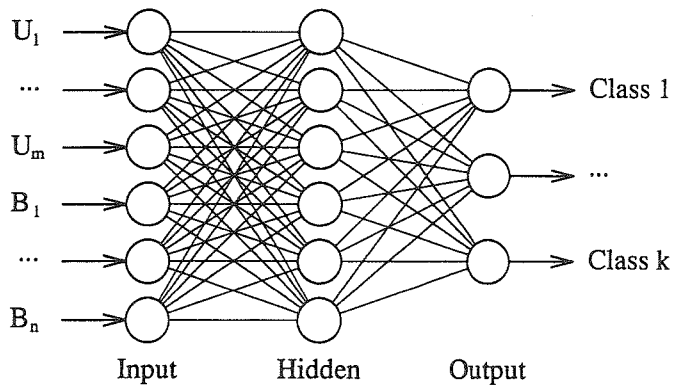

Figure 2. Neural network for evidence-weight estimation for a problem with six rules (hidden units) and three classes. The input is a (1, 0) vector representing rule activation, and the output node with greatest activation determines the classification. The hidden nodes receive input from unary and binary attribute nodes. This allows for the reinforcement of co-occurrences between unary and binary feature states and thus allows implicit learning of relational structures. (From Caelli and Dreier, 1994.)

replaces hard clustering by fuzzy clustering in terms of fitting radial basis functions to data.

The problem of estimating evidence weights has been approached from a number of perspectives including Bayesian (Jain and Hoffman, 1988) and, more recently, neural networks. Specifically, a neural network can be used to estimate evidence weights with input nodes corresponding to clusters, output nodes corresponding to classes, and one hidden layer, with the number of nodes being the larger of the input or output node numbers. Such a network architecture allows for the establishment of relations between unary and binary features (see Fig. 2).

That is, evidence weights are determined by the connections between input-hidden-output layer nodes. Each hidden layer node is connected to every unary and binary rule. This allows for the reinforcement of co-occurrences between unary and binary

feature states and thus allows implicit learning of relational structures. This model for encoding patterns is not unique in the sense that many different combinations of part and part-relations can satisfy the same conditions and so have the same evidence vectors. This is because 'structure' is not necessarily uniquely defined by the enumeration of parts and relations, alone, without considering just *what* parts and *what* relations are required (the indexing problem).

However, this approach to rule generation differs from more traditional neural network models in a number of ways. First, as mentioned above, feature (attribute) space partitioning is not the same as that obtained with multi-layered perceptrons (see Fig. 1). Second, we have defined constraints on the hidden layers to determine evidence weights that are consistent with the conjunctive rule form and involving the integration of unary and binary attributes. Most importantly, this system learns to describe patterns in terms of just what specific parts and specific relations are necessary and sufficient to evidence different classes or models.

Again, the main limitation of this EBS approach is that the representation is not unique: rules are generated without *explicitly* considering the relationships between specific (labeled) unary and binary feature states that define specific objects.

3. RULEGRAPHS

Again, for recognition-by-parts models, patterns are encoded as attributed and labeled graphs where parts correspond to labeled vertices and edges to relationships between parts. When matching data with learned or known models, checks on attributes and labels must be made—and the EBS system described above does not check for the latter completely as the neural network is attribute-indexed and not part-indexed. To solve this label-checking problem efficiently, we have introduced 'Rulegraphs'. The idea behind Rulegraphs (Pearce *et al.*, 1994) is to use EBS evidence weights together with *explicit* label compatibilities to prune the search space in graph matching. The technique relies on two simple principles: First, the initial sample (model or pattern) data are summarized via the EBS system in the form of rules. Second, search for subsets of compatible labels between rules is constrained using evidence weights produced by an EBS. The matching process involves graphs of cardinality (in this case, vertices) no greater than the number of unary rules and thus is more efficient than classical graph matching procedures.

A Rulegraph is a graph of rules in which vertices correspond to unary rules and edges correspond to binary rules according to the following connection criterion: Two unary rules U_i and U_j are connected by a binary rule B_k if there exist labels X, Y such that $X \in U_i$ and $Y \in U_j$ and $XY \in B_k$ (see Fig. 3).

The basic idea of graph matching using Rulegraphs is illustrated in Fig. 3. Given a set of training patterns, a set of unary and binary rules is generated (Fig. 3(a)). Unary and binary rules that share common labels are connected by a Rulegraph edge, according to the connection criterion above (Fig. 3(b)). When a new pattern is presented to the system, several rules are activated (Fig. 3(c)), each evidencing different parts of the training patterns: Sample part X could correspond to parts A

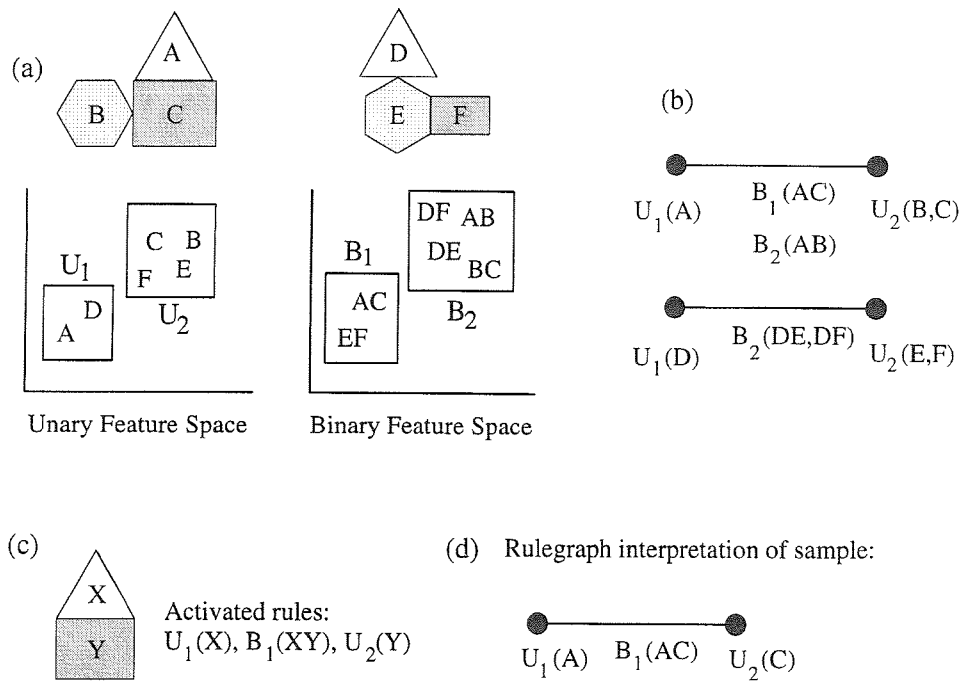

Figure 3. Training patterns are used in (a) to label the unary and binary rules according to the mapping of the parts and their relationships into each feature space. Unary rules are labeled with single labels and binary rules are labeled with label pairs. Rulegraph models may then be formed, according to the connection criterion, and these are shown in (b). At run time, parts in the sample pattern activate unary and binary rules based on their feature states as shown in (c). The search for label-compatible rules between the sample and the model results in Rulegraph interpretation (best match) as is seen in (d). (From Pearce, Caelli and Bischof, 1994.)

or D, Y could correspond to B, C, E, or F, and the binary relation $B(XY)$ could correspond to $B(AC)$ or $B(EF)$. Among these, only one interpretation is consistent with the Rulegraph model, namely the label mapping $X \rightarrow A$ and $Y \rightarrow C$, and this is the interpretation accepted by the Rulegraph system.

In Rulegraphs, several labels may exist in each rule vertex and this gives rise to multiple mapping states involving the same labels. To determine label compatibility between *rules* instead of *parts* we use a method which proceeds in five steps. In step 1, all possible mapping states are created for all the labels in the sample and model rules. In step 2, all incompatible label mappings between sample Rulegraph and model Rulegraphs are eliminated. In step 3, the (multiple) remaining mapping states are updated by instantiation (if the label is *not* yet mapped) or elimination (if the label *is* mapped) using the mappings generated in step 2 and the old mapping states. In step 4, the mapping states are updated in order of decreasing evidence of rules into which they map. This ensures that labels which have strongest evidence for a particular class will be mapped first. Finally, in step 5, we check that at least one binary rule is satisfied.

The label compatibility checking method offers a technique for checking compatibility between two rules. The problem of finding the *best* match now reduces to that of finding the largest (activated) set of rules which are all pairwise compatible. Furthermore, the evidence weights can be used to direct the search toward rules and models for which strong evidence exists. To achieve this, we use A* search (an optimal search tree which orders candidate nodes in terms of the current and expected cost) combined with the Bayesian evidence weight metric to allow probabilistic pruning of the search tree. (For further details see Pearce *et al.*, 1994.)

4. CONDITIONAL RULE GENERATION

In the EBS approach, label compatibilities between unary and binary rules are represented implicitly in the hidden layer of the neural network. In the Rulegraph approach, label compatibilities of evidence rules are checked a posteriori, i.e. during graph matching—a form of hypothesis verification and model projection. The idea of the third approach, Conditional Rule Generation (CRG), is to account for label compatibilities a priori, i.e. during rule generation. The CRG technique (Bischof and Caelli, 1994) searches for the occurrence of unary or binary feature states between connected components of the training patterns and creates trees of hierarchically organized rules for classifying new patterns. In contrast to EBS and Rulegraphs, the CRG method produces deterministic classification rules, at least in the current implementation.

Let each pattern be composed of a number of parts (pattern components). Each part c_i, $i = 1, \ldots, N$, is described by a set of unary features $\vec{u}(c_i)$, and pairs of parts (c_i, c_j) belonging to the same sample (not necessarily all possible pairs) are described by a set of binary features $\vec{b}(c_i, c_j)$. Below, $S(c_i)$ denotes the sample (in 3D object recognition, a *view*) a part c_i belongs to, and H refers to the information, or cluster entropy statistic:

$$H_i = -\sum_j p_{ij} \ln p_{ij}, \tag{2}$$

where p_{ij} is the probability that an element of cluster i belongs to class j.

We first construct the initial unary feature space for all parts over all samples and classes $U = \{\vec{u}(c_i), i = 1, .., N\}$ and segment this feature space into clusters U_i. The characteristics of the clustering procedure are not critical since recursive splitting is later used to refine clusters. Clusters that are unique with respect to class membership (with entropy $H = 0$) provide simple classification rules for some patterns (so-called U-rules). Each non-unique cluster U_i is further analyzed with respect to binary features by constructing the (conditional) binary feature space

$$UB_i = \{\vec{b}(c_r, c_s) \mid \vec{u}(c_r) \in U_i \text{ and } S(c_r) = S(c_s)\}.$$

This feature space is clustered with respect to binary features into clusters UB_{ij}. Again, clusters that are unique with respect to class membership provide classification rules for some patterns (UB-rules). Each non-unique cluster UB_{ij} is then analyzed with respect to unary features of the second part and the resulting feature space

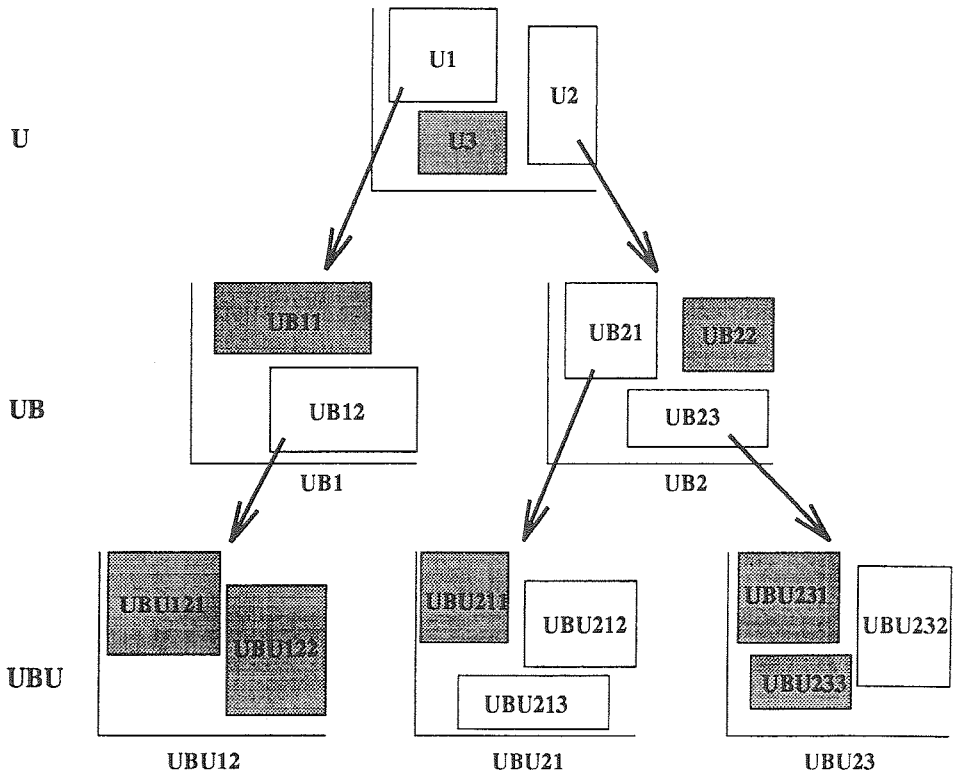

Figure 4. Cluster tree generated by the conditional rule generation procedure (CRG). The unresolved unary clusters ($U1$ and $U2$)—with element from more than one class—are expanded to the binary feature spaces $UB1$ and $UB2$, from where clustering and expansion continues until either all rules are resolved or the predetermined maximum rule length is reached, in which case rule splitting occurs. Shaded boxes denote resolved clusters (with $H = 0$) and white boxes denote unresolved clusters. (From Bischof and Caelli, 1994.)

$UBU_{ij} = \{\vec{u}(c_s) \mid \vec{b}(c_r, c_s) \in UB_{ij}\}$ is clustered into clusters UBU_{ijk}. In general, unique clusters provide classification rules for some patterns (UBU-rules), the other clusters have to be further analyzed to construct unique decision rules (see Fig. 4).

Uniqueness of pattern classification rules can be achieved either by repeated conditional clustering involving additional pattern parts or through cluster refinement. Refinement of a cluster C is achieved by finding the feature dimension F and the feature threshold T_F that minimizes the partition entropy $H_P(T_F)$

$$H_P(T_F) = n_1 H(P_1) + n_2 H(P_2), \tag{3}$$

where P_1 and P_2 denote the two partitions obtained using the threshold T_F, and n_1 and n_2 denote the number of elements in the two partitions, respectively. In addition, rather than splitting only leaf clusters (those at the bottom of the tree), one can split the cluster tree at any level, and the cluster minimizing Eqn (3) is considered optimal for refining the cluster tree.

The completely resolved cluster tree provides a set of deterministic rules for classification of patterns. Furthermore, partial rule instantiations (for example, the classification associated with cluster UB_{23} in Fig. 4) can provide partial evidence for class membership of incomplete or distorted patterns.

The rules for CRG are part-indexed as well as attribute-indexed and have the following form:

if
(this part has these attributes
and this part is related to that part with these attributes
and ...)
then
(evidence for class X is Y).

In summary, CRG has been developed to allow learning of patterns defined by parts and their relations. The method is particularly suitable for learning of patterns with variable complexity and the detection of these patterns in complex scenes. In this context, it should be noted that each feature space in the cluster tree corresponds to a standard decision tree (Quinlan, 1993). CRG thus produces a tree of decision trees that is indexed by sequences of pattern parts, i.e. it is part-indexed, whereas decision trees are purely attribute-indexed. The dynamic expansion of cluster trees constitutes a major advantage of CRG over decision trees: CRG can expand trees to the level optimized for a given data set whereas decision trees operate on fixed sets of features that have to be chosen *a priori*.

5. COMPLEX SCENES AND REGION LABELING

In the preceding sections, we have introduced three methods for learning rules for classifying relational structures, Evidence-Based Systems, Rulegraphs, and Conditional Rule Generation. Application of these classification rules is straightforward in the case of single, isolated patterns (see, for example, Bischof and Caelli, 1994; Caelli and Dreier, 1994; Pearce *et al.*, 1994) given that, by definition, all scene parts belong to the same pattern or object. In complex scenes composed of multiple objects, one is faced with the 'grouping problem': parts belonging to the same object should be grouped together for rule evaluation. This problem has been studied by Grimson (1990) and others in the context of model-based vision. However, in both machine and biological-vision research, insights into just how we perceptually solve such problems and the development of efficient algorithms to do so are still greatly lacking.

Our approach to this problem is based on the notion of 'critical focal features' or the notion that specific scene features yield evidence of classes or categories of objects (patterns) with varying degrees and, where clear evidence is not available with such features, further evidence is sought by examining spatially contiguous features and their relations within and between instantiated rules generated by the CRG method.

This is consistent with our claim about just how structure is encoded in images—by (learnt) rules about parts and their relations—a generalization of earlier work on how humans may go about encoding textural information (Julesz, 1984). In this case, however, the parts and relations are 'pre-compiled' in learning, and scene evaluation is enacted by propagating, evaluating and aggregating rules—a form of constraint propagation. The following discussion is concerned with the investigation of techniques for evaluating and integrating rules within complex scenes without assuming that part groupings (cliques) have been pre-selected.

5.1. Rule constraints

5.1.1. Initial rule evaluation. The first stage involves direct activation of the CRG rules in a parallel, iterative deepening method. Starting from each scene part, all possible sequences of parts, termed *evidence chains* or, simply, *chains*, are generated and classified using the rules. Expansion of each chain $S = \langle s_1, s_2, \ldots, s_n \rangle$ terminates if at least one of the following conditions occurs: (1) the part sequence $s_1, s_2, \ldots, s_n$ cannot be expanded without creating a cycle, (2) all CRG rules instantiated by S are completely resolved, or (3) the binary features $\vec{b}(s_n, s_{n+1})$ do not satisfy the features bounds of any CRG rule. If a chain S cannot be expanded, the evidence vectors of all rules instantiated by S are averaged to obtain the evidence vector $\vec{E}(S)$ of the chain S. Further, the evidence vectors of all chains starting at p can be averaged to obtain an initial evidence vector for part p.

Classification of scene parts based on simple averaging has one major problem. Snakes that are contained completely within a single 'object' are likely to be classified correctly, but chains that 'cross' two or more objects are likely to be classified in an arbitrary way, and they therefore bias classification. For this reason we have developed the following inter- and intra-constraints.

5.1.2. Snake permutation constraint. Every CRG rule encodes a set of model chains $\{M_k = \langle m_{k1}, m_{k2}, \ldots, m_{kn} \rangle, 1 \leqslant k \leqslant K\}$. When a chain $S = \langle s_1, s_2, \ldots, s_n \rangle$ instantiates a rule, each image part s_i indexes a set of model parts $\mathcal{M}(s_i) = \{m_{ki}, 1 \leqslant k \leqslant K\}$. The chain permutation constraint is based on the assumption that rule instantiations are invariant to permutations, that is, if two chains are permutations of each other, for example $S_1 = \langle A, B, C \rangle$ and $S_2 = \langle B, A, C \rangle$, their parts must index the same set of model parts, independent of chains and independent of instantiated rules.

5.1.3. Single classification constraint. The single classification constraint is based on the assumption that *at least one* chain among all chains starting at a scene part does not cross an object boundary and that at least one instantiated rule indexes the correct model parts. Given this assumption, if there is any scene part that initiates a single chain S_i and if S_i instantiates a single classification rule then the model parts indexed by S_i can be used to constrain all chains that touch S_i. These two deterministic constraints are very powerful in terms of eliminating inconsistent (crossing) chains. Their usefulness breaks down, however, for cases where the assumptions formulated earlier are not met for a given training and test data set.

5.1.4. Inter-chain compatibility analysis. The idea of the inter-chain compatibility analysis is as follows. The less compatible the evidence vector of a chain S_i is with the evidence vectors of all chains that S_i touches, the more likely it is that S_i crosses an object boundary. In this case S_i is given a low weight in the computation of the part evidence vectors. The overall compatibility of a chain S_i can be defined with respect to the set $\mathcal{S}_T$ of chains that share common parts with S_i:

$$w_{\text{inter}}(S_i) = \frac{1}{Z} \sum_{S \in \mathcal{S}_T} C(S_i, S), \tag{4}$$

where $C(S_i, S)$ denotes a measure of compatibility between two chains and Z is a normalizing constant (for further details see Bischof and Caelli, 1996), and the evidence vector for a part p becomes:

$$\vec{E}(p) = \frac{1}{Z} \sum_{S \in \mathcal{S}_p} w_{\text{inter}}(S) \vec{E}(S), \tag{5}$$

where $\mathcal{S}_p$ is the set of chains starting at part p, and Z is again a normalizing constant.

5.1.5. Intra-chain compatibility analysis. The last rule for detecting boundary-crossing chains is based on the following idea. If a chain $S_i = \langle s_{i1}, s_{i2}, \ldots, s_{in} \rangle$ does not cross boundaries of objects then the evidence vectors $\vec{E}(s_{i1})$, $\vec{E}(s_{i2})$, $\ldots$, $\vec{E}(s_{in})$ computed by Eqn (5) are likely to be similar, and dissimilarity of the evidence vectors suggests that S_i may be a 'crossing' chain. Similarity of any pair of evidence vectors can be measured by their dot product. Let w_{intra} denote the average of all intra-chain evidence vectors. The part evidence vectors can then be computed using the following iterative (relaxation) scheme:

$$\vec{E}^{(t+1)}(p) = \Phi\left[\frac{1}{Z} \sum_{S \in \mathcal{S}_p} w_{\text{inter}}(S) w_{\text{intra}}^{(t)}(S) \vec{E}(S)\right], \tag{6}$$

where Z is a normalizing factor and Φ a non-linear transducer function. Iterative computation of Eqn (6) is required since recomputation of $\vec{E}(p)$ affects the average intra-chain compatibility. As indicated above, the four rules presented in this section are evaluated sequentially, and the final part classification is given by the iterative scheme Eqn (6).

5.2. Example

In the following, we illustrate the CRG method and how the different compatibility constraints aid in solving relatively complex labeling problems with the *blocks* example shown in Fig. 5. It consists of configurations of toy blocks that are learned in isolation (Fig. 5, c1–c5) and have to be recognized in complex arrangements (Fig. 5, s1–s2). The training set consisted of 5 classes of block configurations, each with tree training examples with one of each being shown in Fig. 5 (c1–c5). The

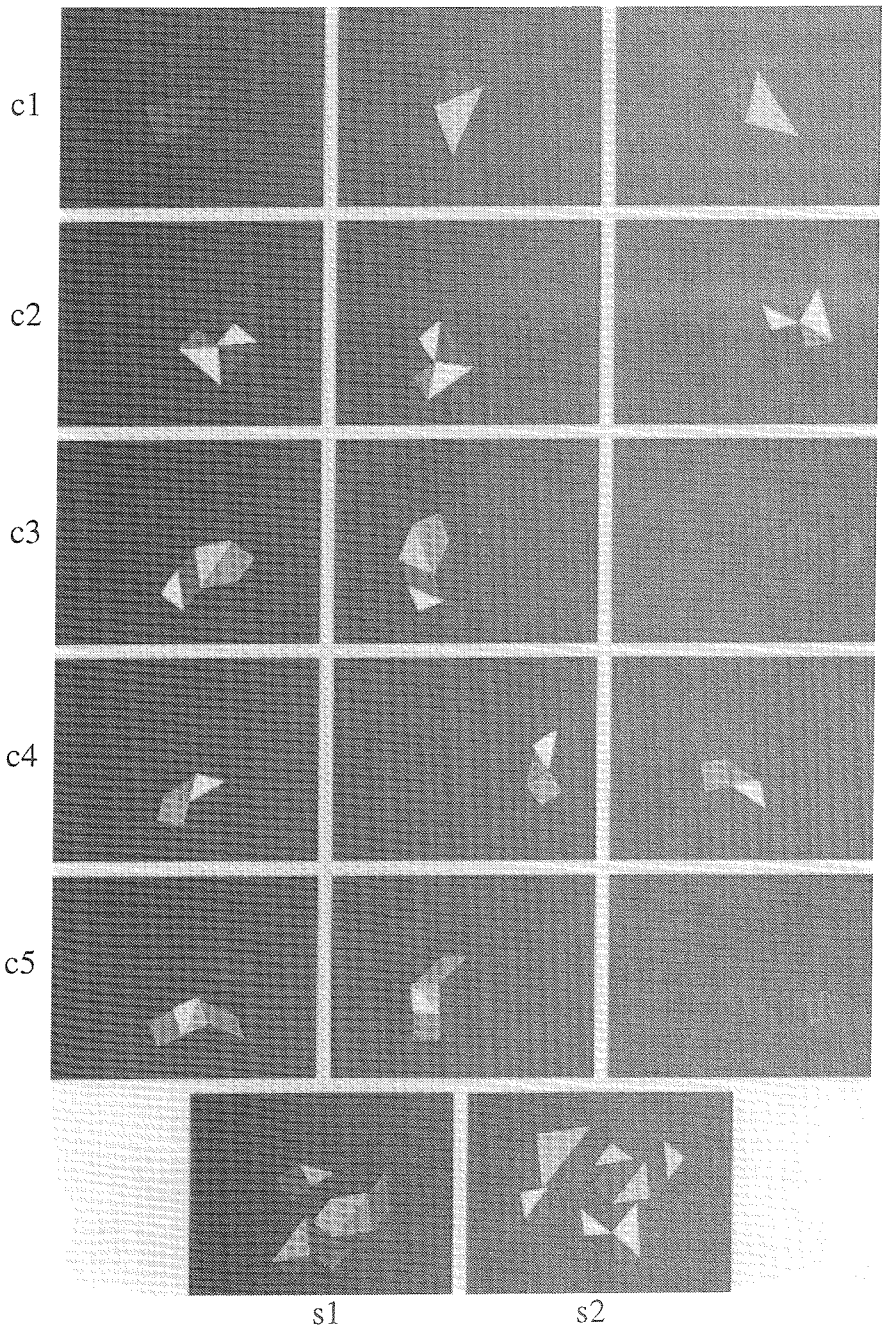

Figure 5. (a) Images of five classes of block configurations with three views each are shown in panels c1–c5. The image parts are described by the unary features size, eccentricity and the three normalized color coordinates. Pairs of image parts are described by the binary features of midpoint distance, area-normalized midpoint distance, minimum distance and normalized shared boundary length. (b) Panels s1 and s2 show images of two complex block configurations consisting of 16 blocks (s1) and 17 blocks (s2). Results for part classification (block faces) are shown in Table 1 with respect to the different compatibility models.

Table 1.
Number of chains (NS), number of correct classifications (NC: maxima for s1 and s2 are 16 and 17, respectively), and average entropy of classification vectors (AE) after application of each rule constraints described in Section 5.1, for the two complex scenes shown in Fig. 5

	Scene s1			Scene s2		
	NS	NC	AE	NS	NC	AE
After initial rule evaluation	81	13	0.26	93	11	0.33
After chain permutation constraint	55	15	0.26	66	13	0.32
After single classification constraint	32	16	0.10	36	15	0.21
After inter-chain compatibility analysis	32	16	0.10	36	15	0.18
After intra-chain compatibility analysis	32	16	0.0	36	15	0.0

complex scenes consisted of up to 20 blocks, two of which are shown in Fig. 5 (s1 and s2).

Images of the training and test objects were captured with a color camera and segmented using a form of K-means clustering on position (x, y) and color (r, g, b) attributes. Small clusters were merged with larger neighbour clusters in order to eliminate spurious image regions. The following unary features were extracted for each image region: size (in pixels), compactness (perimeter^2/area), and the normalized color signals $R/(R+G+B)$, $G/(R+G+B)$, and $B/(R+G+B)$. For pairs of image regions the following binary features were computed: absolute distance of region centers, minimum distance between the regions, distance of region centers normalized by the sum of the region areas, and length of shared boundaries normalized by total boundary length.

For the training data, CRG analyzed 240 different 'chains' and produced 25 rules: 11 *U*-rules, 3 *UB*-rules, and 11 *UBU*-rules. Classification performance was obtained for two scenes, s1 and s2, illustrated in Fig. 5. For scene s1, 16 out of 17 parts were classified correctly, and for the 17 parts of scene s2, 15 were classified correctly and one part was not classified. Relative merits of the four constraints discussed above are shown in Table 1 which shows the number of chains, the number of correct classifications, and the average entropy of all classification vectors after the (sequential) application of the four constraints. It is clear from these results that the constraints improve the classification of scene parts through elimination of crossing chains, and hence improve region and object classification in complex scenes.

6. DISCUSSION

In this paper we have argued that understanding the processes involved in pattern and object recognition include:

- signal encoding;
- feature (region, boundary) extraction;
- feature description (attribute generation);

- rule generation (learning structural descriptions from training, conditioning, examples);
- projection and evaluation of rules in new data.

Although we have argued in support of the Recognition-by-Parts (RBP) paradigm which has already been advocated by others in biological and machine vision (see e.g. Biederman, 1987), we have also argued that the associated perceptual learning processes (see earlier) need to be further developed before a real RBP paradigm is complete. Currently, in biological vision, RBP advocates have essentially restricted their investigations to the low-level processes such as multi-scale parsing of images (see e.g. Watt and Morgan, 1985). What is now required is research into just how humans integrate such low-level processes with knowledge acquisition or learning mechanisms. However, developing explicit models for the processes involved in perceptual memory and learning is not that easy as both low-level and high-level cognitive models must be explicated.

The claim of this paper is that a simple parallel neural network model is inadequate and incompatible with the RBP paradigm for the following reasons. First, the input to neural networks is typically attribute values and not image parts. Second, even if this problem were solved the direct neural networks cannot, in parallel, detect sets of patterns in complex scenes without first of all isolating candidate regions of images. Third, the types of rules generated by neural networks (without any additional constraints) to represent different patterns are not in the conjunctive form described above. We have overcome these limitations using the EBS system. However, it did not have the uniqueness and variable length rule-generation advantages of the RG and CRG methods.

So, we have presented three new approaches to an Evidence Theory of pattern and object recognition: EBS, EBS with optimal label checking (RG), and Conditional Rule Generation (CRG). These approaches are aimed at finding efficient and accurate methods for developing prototypical descriptions of shapes which involve the definitions of part and part relations. The approaches use techniques from machine learning to solve this problem, as well as to address the generalization problem and the problem of *pre-compiling* search strategies for matching. They involve various combinations of standard representation and search methods from the machine learning literature. What differentiates this work is just how we have compiled each method and how they have been adapted to solve problems in vision.

Rule generation. In all cases, rules are formulated in terms of conjunctions and disjunctions over both unary and binary feature values. Rule generation is, in each case, based on clustering. In the EBS and the RG approach, we have used standard clustering procedures, such as Leader Cluster or Minimum Entropy. In CRG, on the other hand, we have used a deterministic splitting procedure similar to those used in Decision Trees (Quinlan, 1993). The former approach leads to optimized rule conjunctions and disjunctions, but rules are nondeterministic. The latter approach generates rules only on demand, but it is difficult to control rule complexity.

Label compatibility. The main difference between the three approaches relates to the way in with label compatibilities between rules and data are enforced. In the

EBS approach, the emphasis is on generating rules which can filter evidence for the possible existence of given objects from the presence of specific unary and binary features. The RG method essentially checks whether there is an object or pattern in the data which not only satisfies the EBS constraints, but which also has the *specific* co-occurrences of unary and binary feature states. Finally, the CRG method puts both these characteristics of a relational structure (RS) together to *compile* rules which allow for generalization while, at the same time, guaranteeing the compatibility of the unary and binary feature states.

Uncertainty. Both EBS and RG use non-deterministic rules with evidence weights. CRG produces, at least in the current implementation, deterministic classification rules only. For this reason, the performance of CRG with highly complex data sets or noisy training data is clearly suboptimal.

Generalization. In all three approaches, pattern generalization is captured through clustering of the unary and binary feature states. Limits on clustering are determined a priori for the EBS and RG system. In CRG, clustering or, more precisely, cluster splitting is determined by the target classification performance for the training data sets. However, this does not necessarily optimize the ability of the system to generalize to new unseen examples. Indeed, there may be a range of solutions which can generalize data from least general to most general, and the selection of the particular generalization may be arbitrary.

What remains to be resolved is how to develop insightful experimental procedures which can clearly demonstrate (or not) that RBP is actually what human observers do and also expose the types of learning, features and attributes which are encoded in so doing. It is the authors' belief that RBP operates at the 'attentive' level of visual information processing and and so it is not intended to capture the more direct and semi-automated processes of image encoding at the 'pixel' level.

Acknowledgement

This project was funded by grants from the Australian Research Council and the Natural Science and Engineering Research Council of Canada.

REFERENCES

Biederman, I. (1987). Recognition-by-components: A theory of human imageunderstanding. *Psychol. Rev.* **94**, 115–147.

Bischof, W. F. and Caelli, T. (1994). Learning structural descriptions of patterns: A new technique for conditional clustering and rule generation. *Patt. Recognit.* **27**, 689–697.

Bischof, W. F. and Caelli, T. (1996). Scene understanding by rule evaluation. (submitted).

Caelli, T. and Dreier, A. (1994). Variations on the evidence-based object recognition theme. *Patt. Recognit.* **27**, 185–204.

Caelli, T. and Pennington, A. (1993). An improved rule generation method for evidence-based classification systems. *Patt. Recognit.* **26**, 733–740.

Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W. and Freeman, D. (1990). In: *Readings in Machine Learning*. Shavlik and T. Dietterich (Eds). Morgan Kaufman, pp. 296–306.

Fan, T., Medioni, R. and Nevatia, R. (1989). Recognizing 3-D objects using surface descriptions. *IEEE Trans. Patt. Anal. Machine Intell.* **11**, 1140–1156.

Flynn, P. and Jain, A. K. (1993). Three-dimensional object recognition. In: *Handbook of Pattern Recognition and Image Processing, Volume 2: Computer Vision.* Tzay Y. Young (Ed.). Academic Press, New York.

Grimson, W. E. L. (1990). *Object Recognition by Computer.* MIT Press, Cambridge.

Hoffmann, D. and Richards, W. (1986). Parts of recognition. In: *From Pixels to Predicates.* A. Pentland (Ed.). Ablex, New Jersey, pp. 268–294.

Jain, A. K. and Hoffman, D. (1988). Evidence-based recognition of objects. *IEEE Trans. Patt. Anal. Machine Intell.* **10**, 783–802.

Julesz, B. (1984). Towards an axiomatic theory of preattentive vision. In: *Dynamic Aspects of Neocortical Function.* G. Edelman, W. Gall and M. Cowan (Eds). Neuroscience Research Foundation, New York, pp. 581–611.

Lippman, R. P. (1987). An introduction to computing with neural nets. *IEEE: ASSP Magazine*, April 1987.

Milanese, R., Wechsler, H., Gil, S., Bost, J.-M. and Pun, T. (1994). Integration of bottom-up and top-down cues for visual attention using non-linear relaxation. In: *Proceedings of CVPR '94*, Seattle, Washington, pp. 781–785.

Pearce, A., Caelli, T. and Bischof, W. F. (1994). Rulegraphs for graph matching in pattern recognition. *Patt. Recognit.* **27**, 1231–1248.

Poggio, T. and Edelman, S. (1990). A network that learns to recognize three-dimensional objects. *Nature* **343**, 263–266.

Poggio, T. and Girosi, F. (1990). Regularization algorithms for learning that are equivalent to multilayer networks. *Science* **247**, 978–982.

Quinlan, J. R. (1993). *C4.5 Programs for Machine Learning.* Academic Press, New York.

Rentschler, I., Jüttner, M. and Caelli, T. (1994). Probabilistic analysis of human supervised learning and classification. *Vision Res.* **34**, 669–687.

Shapiro, L. G. and Haralick, R. M. (1982). Organization of relational models for scene analysis. *IEEE Trans. Patt. Recognit. Machine Intell.* **4**, 595–602.

Suetens, P., Fua, P. and Hanson, A. J. (1992). Computational strategies for object recognition. *ACM Computing Surveys* **24**, 5–61.

Treisman, A. and Gelade, G. (1980). A feature integration theory of attention. *Cognit. Psychol.* **12**, 97–136.

Watt, R. and Morgan, M. (1985). A theory of the primary spatial code in human vision. *Vision Res.* **25**, 1661–1674.