Integrating Numerical and Syntactic Learning Models for Pattern Recognition

Terry Caelli
Center for Mapping, The Ohio State University
E-mail:caelli@cfm.ohio-state.edu

Abstract. In this paper we consider how the recognition, interpretation of image structures, patterns, objects can be posed in terms of "Inductive Bayesian Networks" (IBN) which combine syntactic domain models with the numerical/statistical characteristics of what is sensed. The net result of this formulation is the production of contextual and relational rules which can be used to summarize, generalize structural descriptions from examples in ways which are consistent with domain knowledge. In this approach the associated algorithms are also constrained by principles of Minimum Description Length (MDL) which endeavor to produce structural descriptions which generalize over numerical data attribute while specializing over symbolic description length. Examples in pattern and object recognition are discussed.

Keywords: Pattern Recognition, Macine Learning, Image Annotation

1 Introduction

It is still the case that most pattern/object recognition methods in Computer Vision use techniques based upon the notion of determining the types of attributes, their characteristic ranges and associated rule structures which result in optimal classification of training and new test data. Bayesian classifyers, Decision Trees, Neural Networks, Kohonen Maps, in their basic forms, are all examples of such an approach where a "representative" attribute space is partitioned or clustered to attain this goal. For statistical, Neural Network and self-organising classifiers the rules are typically what we term "attribute-indexed" in so far as they are techniques for partitioning (categorically or using fuzzy membership) attribute spaces, resulting in rules of the form:

IF feature exists with these attributes
THEN it is (an unspecified) part of that pattern.

Figure 1c illustrates different types of attribute indexed techniques in terms of the resultent rule geometries generated from training data in the form shown in Figure 1a. Such rules are not relational in so far as they do not utilize feature

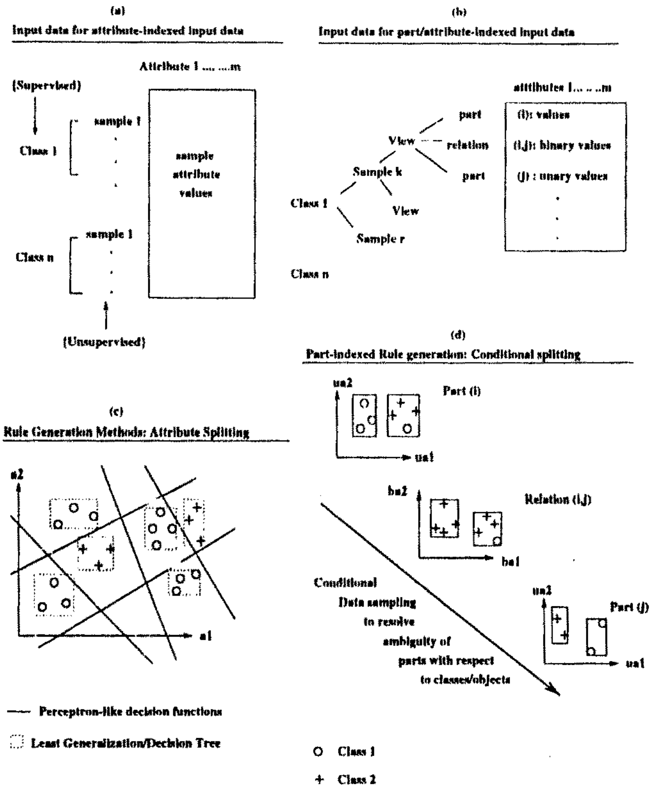


Fig. 1. Learning models: Different types of induction geometries. Top left shows training data for attribute-indexed induction models. Top right: training data for relational learning modes.Bottom left: typical automated rule generation techniques using Neural Networks and Least Generalisation methods. Bottom right: Relational Learning models (see text). a_i refers to attribute i; ua_i to unary (part i) attribute; ba_{ij} to relational (binary) attribute between parts i and j.

labels or indexing, beyond class membership, nor use them in the generation process. In contrast, "relational rules" or "relational learning" involves training data which preserves data indexing and the rule types show relational learning models where the feature indexing controls generalizing over attributes (Figure 1d).

There are a number of fundamental problems with "attribute-indexed" or traditional Pattern Recognition (PR)/learning techniques:

- They are not designed to index features as relational structure.
- They typically fail with only parts of patterns to be recognize.
- They are not designed to function in complex (multi-object) scenes.

- They rarely view PR as a process of transmitting syntactical and semantic structures through an information flow network which incorporates domain knowledge and models as well as sensed data.

In general, we see PR in these latter terms - akin to "explanation-based learning" [7] as envisaged in Figure 2 where the constraints are typically defined in terms of contingency relations. In this sense, then, data-driven sensing, feature extraction and measurement are tuned to reduce patterns to sets of (labelled) parts, relations and their attributes. Domain knowledge provides constraints on the network model and the symbolic data structures necessary for interpretation. Machine learning is seen as the class of techniques used to bind features with domain knowledge and update both according to external criteria - such as human task-demands, objective performance of the system, etc.

The operators listed in Figure 2 depict the classes of learning and inference techniques we have explored in pattern recognition over the past five years. These will be illustrated in a number of systems in the following sections and pertain to solving, in a robust and efficient way, the binding of sensed data with domain knowledge.

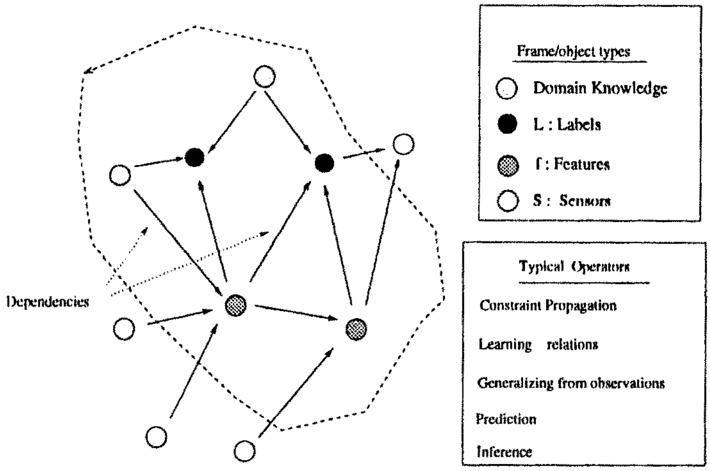


Fig. 2. General information flow perspective to Pattern Recognition. Here, a specific interpretation model involves the selection of sensing, feature extraction, labelling units, domain knowledge, learning operators and specific dependency relations.

Such network formulations can be posed in Bayesian terms - as extensions of past Bayesian network methods for PR (see, for example, [12, 9]). Using

[1] This project was funded by a grant from the Australian Research Grants Committee

[1] This project was funded by a grant from the Australian Research Grants Committee and NASA

this formulation for PR, an interpretation is defined by the joint probability distribution (left-hand side) of:

$$p(\mathbf{S},\mathbf{f},\mathbf{L},\mathbf{DK}) = p(L_k/f_I, f_j)p(L_r/DK_s)p(f_g/f_h, f_w)... \qquad (1)$$

where $\mathbf{S},\mathbf{f},\mathbf{L},\mathbf{DK}$ correspond to sensor (S) modalities, image features (f), labels (L) and domain knowledge (DK) types, respectively. The terms on the right hand side define the set of dependencies which determine the model and procedures. The aim is to produce the simplest and most robust model in terms of the smallest number of symbols, nodes and connections in the network which can produce an interpretation: the association of symbols with image features (Figure 2). This type of goal can be related to the Minimum Description Length (MDL) criterion[7] a perspective which will be developed in the following discussion.

The role of machine learning is to aide in the discovery of the dependencies with respect to process and domain constraints. For example, for strictly hierarchical pattern recognition systems, we have a lattice model defined by:

$$p(\mathbf{S},\mathbf{f},\mathbf{L},\mathbf{DK}) = p(f_i/S_k.., S_s)p(f_j/f_r..f_m)p(L_u/f_p..)p(L_m/L_e..)p(DK_g/L_k..).. \qquad (2)$$

where features are strictly dependent on sensors and other features, symbols on features and other symbols; domain knowledge on labels, etc. More general networks can have the following formulation and include feedback connections between levels of processing (not identical units) which allow for a probabilistic measure of sensor, feature and label updating:

$$p(\mathbf{S},\mathbf{f},\mathbf{L},\mathbf{DK}) = p(S_i/f_k.., L_s)p(DK_g/S_k..)p(S_r/DK_s..)... \qquad (3)$$

2 Learning and knowledge acquisition

As already discussed attribute-indexed learning models (such as Decision Trees[11] and Neural Networks like HyperBF [10]) do not generate descriptions in terms of labelled parts and relations ("this" feature and the relations between "those" features) and they typically fail when dealing with partial data and are not designed to function for interpreting multi-pattern(object) data. For this reason over the past five years we have developed a class of new learning procedures explicitly designed to satisfy these types of requirements and apply to relational data. Thel relational system described in Figure 1d is an example of one such system - Conditional Rule Generation (CRG) - which generates rules of the form:

IF this feature has these attributes
AND is related to that feature with those attributes ..etc..*AND*...
THEN this feature is likely to correspond to feature x of model y

This method splits feature attribute spaces (of varying arities) and expands the number of connected (labelled) features in ways which can resolve the class membership of specific objects. We will see other *relational learning methods* where the relations are defined by hierarchical lattice structures to constrain the generalisation process (see the CITE system below). Again, in Bayesian terms, these methods actually generate the conditional dependencies (right hand side of Equations 1,2) using the minimum number of features to uniquely resolve the class to which a given feature belong.

To repeat, this view of pattern recognition involves the recurrent evolution of symbolic descriptions which generalize overthe numerical attributes indexed by each symbol. For example, initial feature labels for object parts (roofs and walls...etc.) are propagated further to objects and groups of objects all which have their own symbolic descriptions (house, suburb ..etc..). From an Minimum Description Length (MDL) perspective, the benefit of hierarchies of symbolic representations lies in the trade-off between data and models. Recalling that the fundamental insight in MDL is that, maximising the log of the posterior probability of data (D) given Model (M) is determined by minimising the length of the encoded data given the model, since:

$$-log_2(P(M)P(D/M)) = -log_2(P(M)) - log_2(P(D/M)). \tag{4}$$

In other words, it is argued that one of the more important benefits of such hierarchical symbolic representations is that we trade the large amount of information required to encode the original image data by reduced data sets and the network model(Figure 2), resulting in:

$$MDL = CodeLength(Models) + CodeLength(Data/Model). \tag{5}$$

The various systems developed focus on how this may be accomplished in different problems, domain knowledge and sensed data. In the following sections we illustrate this perspective to PR in a number of systems developed over the past five years by our group[3].

3 Complex scenes as multi-graphs

As briefly described in Figure 1d, the aim of the Conditional Rule Generation (CRG) system was to use Machine Learning methods to generate descriptions of structures in terms of labelled part and relational attributes. That is, each model is defined by a labelled, attributed and directed graph where each vertex defines a part or relational feature having specified (derived) attribute bounds. Directed edges within each model define the feature (and attribute) dependencies necessary and sufficient to uniquely identify the model and feature to which a given vertex belongs out of multiple models.

Figure 1d illustrates how the rule generation process functions - and both categorical and fuzzy versions (FCRG) exist[1, 5]. Although, like Decision Trees, this method involves depth-first expansion of attributes, best-first versions have also been developed[8]. Rules define local (directed) cliques: lists of parts and relations associated with a given feature of minimum length and covering attribute bounds which enable part matching to multiple models. In multi-object scene recognition problems, training data typically consists of sets of views of different objects (see, for example, Figure 3). For each view of each object, segmentation or feature extraction procedures are enacted to result in lists of labelled parts over all objects and views - each part and relation having preselected attributes. The CRG algorithm then generates descriptions of each object part in terms of permissible attribute bounds for itself, its relation to others - and, in turn, their relations to others which are sufficient to resolve uncertainty about the object, view and part to which the initial part actually corresponds. Again, defined as an "Inductive Bayesian Network"(IBN) this model expresses dependencies in terms of a tree of conditional feature attribute states (Figure 1d) where we determine the degrees to which specific features (labelled parts and relations) condition each other to maximally evidence a given model, and having the general form:

$$p_{model(k)}(f_1,..,f_n) = \sum_{i=1}^{n} \Pi_{j=i+1}^{n} p_{model(k)}(f_i|f_j). \tag{6}$$

That is, each feature in the IBN is a random variable corresponding to an identified (labelled) part (unary feature) or part-relation (binary feature) of the (training) objects and the task of the IBN algorithm is to discover the lists of feature dependencies required to maximally evidence a given model or object. Our solution to the problem is analogous to Decision Trees [18] where, using the Minimum Description Length (MDL) version of maximum likelihood solutions to Bayesian classification[7], we expand the associated feature dependency tree to resolve class uncertainty. That is, we have formulated the IBN model in such a way to enable the optimisation procedure to be implemented via a feature tree search algorithm which expands the conditional feature list while, at the same time, determining feature attribute bounds which, together, can maximally evidence a given model — as illustrated in Figure 1d.

The resultant part-indexed rules can then be used to label each part of complex scenes in accord to the most likely view and model they belong to - without pre-supposing that part cliques or candidate objects, per se, have been isolated before classification. Results using this system are illustrated in Figure 4 - showing correctly and incorrectly labelled parts (see McCane and Caelli[5] for more details).

Once CRG has generated rules from training samples, the problem of scene labelling reduces to that of instantiating rules in data, grouping labels and checking for their compatibilities. Indeed, the very purpose of the CRG method has been to "pre-compile" the types and number of parts, their attribute and relational attribute states that are necessary and sufficient for recognition. We have also examined a number of methods for evaluating evidence from acti-

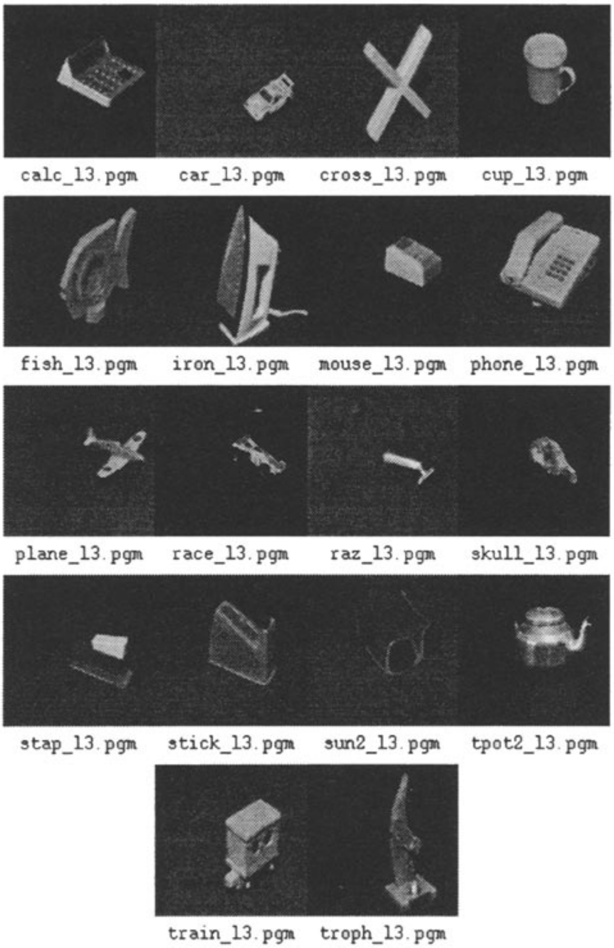


Fig. 3. Training data showing single views of 18 objects used for learning relational descriptions of objects from intensity and sparse depth maps from intensity and sparse depth maps. See text and [5] for more details.

vated rules. These include the SURE system (Scene Understanding using Rule Evaluation[2]), and a simpler version used in the follwing example.

3.1 Rule evaluation procedures

Recognition of objects in a scene typically involves locating an object in a scene which contains multiple objects.. Hence, there are two problems which need to be solved - scene partitioning, grouping, and object recognition. The CRG classifier can be used for performing all tasks using, in parallel, the following steps:

1. Unary features are extracted for all scene parts and binary features are extracted for proximate parts.

2. An adjacency graph is extracted from the scene and all non-cyclic paths up to a maximum length, *maxlength*, are extracted. The value of *maxlength* is equal to the maximum depth of the tree which was generated during the training stage.

3. Each path, $P =< p_i, p_j, ..., p_n >$, is classified using the classification tree generated during the learning phase - resulting in a set of classification vectors for each part.

4. The evidence vectors of all paths (evidence paths) starting at p_n determine the classification of part p_n and so determine rules for recognition purposes.

It is important to note that such rules are "part-indexed" insofar as it is the conjunction oi specific parts, relations and their associated attributes, which evidence objects! This is precisely how FCRG and CRG differ from other learning procedures - including traditional decision trees and neural networks.

Due to the nature of the CRG tree, each path can be classified in more than one way by choosing the best N clusters to descend, rather than just one cluster as in the crisp case. In the current implementation, the value of N given above is 2. This means that at each level of the cluster tree, two clusters are chosen from all the clusters and the path is expanded along both those clusters. The best clusters at a particular level are chosen by evaluating the fuzzy membership function.

Finally, not only can we discover the most likely clusters a given path belongs to, we can also extract a measure of how good a given path fits into the cluster tree. At each level in the tree, a path has fuzzy memberships associated with each of the clusters at that level in the tree. One well-known way for defining the decision value of a particular node in a fuzzy decision tree is via the product of the decision values of the branches composing the path from the root to the node. Consider a path $P =< p_1, p_2, ..., p_n >$, being classified in the decision tree along the cluster branch $< U_i, UB_{ij}, ..., (UBU...)_{ij...} >$, the fuzzy evidence vector for P is given by:

$$\mathbf{E}_f(P) = \mathbf{E}(P) \bigwedge_{c = U_i}^{(UBU...)_{ij...}} \mathcal{F}_c(p_c), \tag{7}$$

where $\mathbf{E}$ is the original evidence vector of path P characterised by the relative frequencies of each model part in the cluster $(UBU...)_{ij...}$, $\bigwedge$ is the product operator, and $\mathcal{F}$ is the fuzzy membership function of part p_c in cluster c given by Equation 7. Here, U_i, B_{ij}, correspond to unary (part) and relational (binary) attributes of parts i, relations ij, etc. There are two different interpretations of the product in Equation 7 - a real number product in the probability model, or as a minimum function in the max-min model (the minimum of all the operands is chosen as the result). Th‿ max-min model has been used here as it is more useful for comparing evidence vectors of instantiated rules, especially if the paths were of different length (using the probability model, shorter paths would typically have higher evidences since there would be less multiplications by numbers less than 1.0 involved in arriving at the final evidence vectors).

Step 4 (see above) deserves further consideration as evidence vectors for a given part may be incompatible, or non-unique. Incompatibility occurs when two evidence vectors indicate two separate class labels for the given part, while non-uniqueness indicates that a rule has been partially instantiated (for example, a path is only UB, while rules are UBU). Further, the problem of how best to utilise such evidence vectors of the paths to provide an optimal and consistent labeling of scene parts is non-trivial. Approaches to this problem fall into two categories. In the first approach, candidate "objects" are selected using perceptual grouping principles and the like. The second approach is the one adopted here which was initially described by Bischof and Caelli [1] and is extende‿ here. In ‿his approach evidence is propagated from parts and relations via compatibility measures, as described below.

3.2 Compatibility Analysis

The compatibility measure adopted here involves a measure of the compatibility of the evidence vector's of the constituent parts with the evidence vector of the path. More formally, this measure can be characterised by the following equation, for a path $P_i =< p_{i1}, p_{i2}, ..., p_{in} >$:

$$\mathbf{w}_{intra}(P_i) = \frac{1}{n} \sum_{k=1}^{n} \mathbf{E}(p_{ik}) \tag{8}$$

where $\mathbf{E}(p_{ik})$ refers to the evidence vector of part p_{ik}. Initially, this can be found by averaging the evidence vectors of the paths which begin with part p_{ik}. This compatibility measure can be used with a relaxation labeling scheme defined by:

$$\mathbf{E}^{(t+1)}(p) = \frac{1}{Z} \sum_{S \in S_p} \mathbf{w}_{intra}^{(t)}(S) \otimes \mathbf{E}(S), \tag{9}$$

where Z is a normalising factor:

$$Z = \sum_{S \in S_p} w_{intra}^{(t)}(S), \tag{10}$$

and the binary operator $\otimes$ is defined as a component-wise vector multiplication in the following way:

$$\begin{pmatrix} a \\ b \end{pmatrix} \otimes \begin{pmatrix} c \\ d \end{pmatrix} = \begin{pmatrix} ac \\ bd \end{pmatrix}. \tag{11}$$

The compatibility measure utilised above is a vector quantity by which the paths' evidence vectors are updated. The compatibility measure used by Bischof and Caelli ([?]) takes into account compatibility between the parts involved in a particular path (by using the dot product of average evidence vectors of each part), but not the compatibility between the average evidence vectors and the path's evidence vector. This problem is rectified in the above solution.

FCRG and other Machine Learning approaches to rule generation and recognition can be viewed as ways of "pre-compiling" search strategies for the verification of models in data. For this reason we have also considered a final hypothesis verification stage to resolve the types of ambiguities remaining. In all, then, the result of the FCRG classifier is a set of most likely labels for each part of the scene where the labels refer to a given object, its sample and sample part from the training data which can be used to infer the pose of the object!

Figures 3 and 4 show training objects and test scenes used to evaluation these types of rule evaluation methods. Here, initial views of objects are segmented using an adaptive multi-scaled edge/boundary extraction procedure and attributes such as colours, shape statistics and relational attributes such as distances and angles between parts are computed from the resultant parts (see McCane and Caelli[5] for more details). Rules are instantiated and evaluated in the test scenes (Figure 4) using the types of processes discussed above.

In an abstract sense, then, this view of PR is one of a multi-subgraph matching problem in so far as, at a given level of representation (features, symbols), the interpretation process involves matching model subgraphs (paths) with images which are defined by graphs. The CRG algorithm pre-compiles the types of parts and relations necessary and sufficient to identify a image part and the relaxation-based methods (described above) are then used to determine the complete labelling in terms of the consistency between .he part labels (Bischof and Caelli[2]). Related to this perspective are a number of recent results which also show how, by pre-compiling common subgraphs within models, the complexity of this search problem can be significantly decreased - see Messmer and Bunke[6], for example. However, the CRG method is quite different from these approaches as they are focused on generating trees of common subgraphs for fixed attributed model graphs. CRG considers a trade-off between description length (path lengths) and vertex attribute resolution (vertex colouring) - so allowing for graph proximity, shortest description length of common and discriminating subgraphs.

4 Complex scenes as lattices

In contrast to the previous object recognition problem, in the CITE system domain knowledge and image data is hierarchical and defined by a lattice structure (Figure 5) where parent nodes define higher-order structures depicting groupings of objects, function and the more general co-occurrences of components.

Initial feature (parts, regions) extraction is enacted and unary attributes computed for each feature (1,2 in Figure 5) of the current image. Such feature

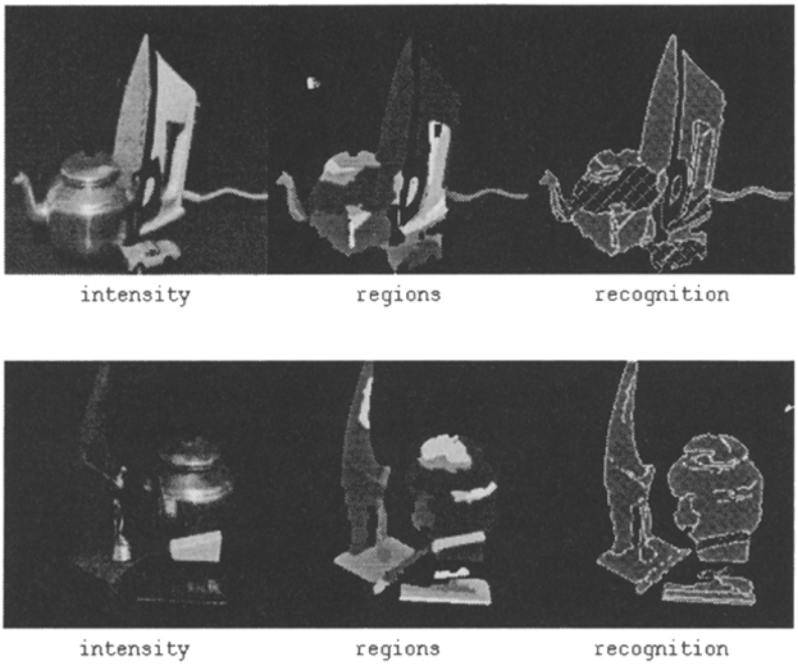


Fig. 4. Left: Shows input complex scene examples. Center: Shows images segmented by an adaptive multi-scaled segmenter as used on the initial single object images. Right: interpretation results. Here grey regions correspond to correctly labelled regions and hashed ones to errors[5].

attributes are matched with the current knowledge base which, in turn, generates initial hypotheses about the feature labels (3; Figure 5). This is called the *scene interpretation.* Feature grouping (clique resolving) is then computed from what is known in the knowledge base about the co-occurrence of features, their binary features (4: relational attributes) and their consistencies over the hierarchical image model (5: Figure 5). Since this process has arbitrary levels of abstraction (hierarchies), higher level scene hypotheses are added to the scene interpretation structure and a form of hierarchical relaxation labelling (see below) begins to resolve the multiple ambiguous labels for each object (6: Figure 5). As the labels begin to "resolve": relabelled to be consistent with the current domain knowledge, resegmentation occurs with respect to the process and parameters allocated to each particular object. The knowledge base is updated to include these new parameter states. The resultant new features replace the previous ones in the visual interpretation structure, resulting in repeating the extraction of unary and binary attributes for matching (2-6: Figure 5). The cycle continues till the interpretation becomes stable. If the interpretation is deemed as incorrect, the user can then choose to incrementally learn the correct object labelling (8: Figure 5) by selecting the incorrectly labelled nodes and the desired knowledge base node. The updated knowledge base is then available for viewing the next scene.

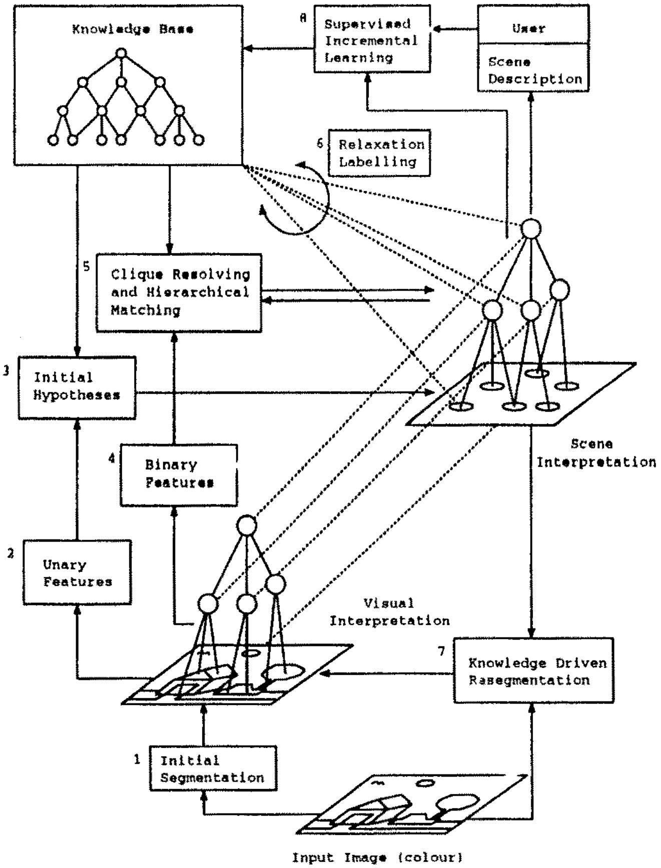


Fig. 5. Cite system showing the cooperative interaction between feature extraction, learning and domain knowledge units synergistically employed to obtain the most parsimonious interpretation of images[4]

4.1 Evidence evaluation via hierarchical relaxation labelling

There are typically multiple labelling and grouping hypotheses generated by *Cite* for any image region or set of image regions. Again, these multiple hypotheses are resolved by a process of relaxation labelling and constraint propagation. The iterative nature of the relaxation labelling process and its ability to propagate local constraints through the interaction of compatible or incompatible labels are ideal for the purposes of this hierarchical system. In this case we use the following formulation for relaxation labelling:

Let $\mathbf{B}$ be a set of objects $\{b_1, ..., b_n\}$, and Λ be a set of labels $\{1, ..., m\}$. For each object b_i we can obtain an initial label probability vector $\mathbf{P}_i^0 = (p_{i1}^0, ..., p_{im}^0)$ where $0 \leq p_{ij}^0 \leq 1$, for $i = 1...n$ and $j = 1...m$, and $\sum_j p_{ij}^0 = 1$, for $i = 1...n$.

Each initial probability vector is interpreted as the prior probability distribution of labels for that object, and can be computed from the initial unary and binary matching. The comparability constraints can be described as a $n \times n$ block matrix $\mathbf{R}$, where each R_{ij} is a $m \times m$ matrix of non-negative real-valued compatibility coefficients, denoted $r_{ij}(1..m, 1..m)$. The coefficient $r_{ij}(\lambda, \mu)$ is a measure of the compatibility between object b_i being labelled λ and object b_j being labelled μ. The relaxation labelling algorithm iteratively updates the probability vectors $\mathbf{P}$ using a normalised weighted sum equation:

$$p_{i\lambda}^{t+1} = \frac{p_{i\lambda}^t q_{i\lambda}^t}{\sum\limits_{\mu=1}^{m} p_{i\mu}^t q_{i\mu}^t} \tag{12}$$

where the denominator is the normalisation factor and:

$$q_{i\lambda}^t = \sum_{j=1}^{n} \sum_{\mu=1}^{m} r_{ij}(\lambda, \mu) p_{j\mu}^t. \tag{13}$$

In this form of relaxation labelling, the number of objects is constant and the number of iterations, t, is the same for each object. However, *Cite* contains hierarchical knowledge and can generate and remove image regions dynamically. For example, the Scene Interpretation(SI)-Knowledge Base(KB) Link is updated according to the level of support given by the SI children *and* SI parents, as follows:

$$\widehat{P}_{i,j}^{SK} = (1 - \alpha_{pc}) \sum_{\lambda \in S_i^C} P_\lambda^S P_{\lambda,i}^S \sum_{\xi \in K_j^C} P_{\lambda,\xi}^{SK} + \alpha_{pc} \sum_{\lambda \in S_i^P} P_\lambda^S P_{i,\lambda}^S \sum_{\xi \in K_j^P} P_{\lambda,\xi}^{SK} \tag{14}$$

The initial hypothesis value is set by the unary and/or binary matching from the operator which created the given SI-KB hypothesis. The update ratio of parent and child support (α_{pc}) reveals some asymmetry in the update procedure in terms of the relative importance of children and parents. This is necessary because, in general, there are fewer parents of a SI node than children.

In Equation 13, the double summations represent the summing over what are termed *compatibility cycles* between the scene interpretation and knowledge base graphs. The compatibility cycle is a cycle comprising four hypotheses and is computed through the parent and child chain (right half of Equation 14). There are only three terms, rather than four, in each half of the update equation because the parent-child knowledge base link has a set hypothesis weight of 1.0.

An extension to the knowledge base facilitating partial belief in the knowledge structure could be achieved by including a non-unity term into this equation.

Returning to the Bayesian formulation and Figure 2, the CITE system provides a lattice model as described in Equation 2 - in contrast to the more general graph model (Equation 3) and performance of the system is shown in Figures 6 to 8. First, Figure 6 illustrates the symbolic domain knowledge for the "Office" scene involving the definition of specific labels and their relations. The numbers next to each label correspond to nodes and regions in Figure 7. Figure 7, then, shows initial images and resultant labels derived from instantiating the observed feature attributes(bottom lattice: Scene Interpretation), and their relations, in the Knowledge Base (top lattice).

The learning component involves determining the part and relational attribute bounds which are consistent with domain knowledge models or expert interpretations. As with CRG rules are formed by bounding rectangles over unary and binary attribute values - so covering valid examples of the different structures.

Figure 8 shows the results of interpretation an outdoor scene image involving a domain knowledge base consisting of houses, fueltrucks, trees, roads and related basic objects; higher-order objects such as diary, ground, sky - all being composed of more basic parts and resulting in the interpretation shown in the bottom of the figure. Here the numbers associated with each label define the certainty (0-1) of the labelling.

5 Conclusions

Pattern Recognition and Image Understanding typically involves a variety of levels of data and knowledge representations and processing algorithms. For this reason it has typically been difficult to conceptualise, evaluate and compare in any systematic way. However, in this paper PR has been posed in a way, hopefully, which overcomes these problems. It is viewed in terms of a network involving specialised operators which, in this case, learn to bind domain knowledge representations with what is sensed, and vice-versa.

References

References

1. W. Bischof and T. Caelli. Learning structural descriptions of patterns: A new technique for conditional clustering and rule generation. *Pattern Recognition*, 27(5):689–697, 1994.
2. W. Bischof and T. Caelli. Sure: Scene understanding by rule evaluation. *IEEE: Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1284–1289, 1997.
3. T. Caelli and W. Bischof. *Machine Learning and Image Interpretation.* Plenum, 1997.

```
World[0] Consisting of:
├── stapler[57] Of Types:
│   ├── stapler2[53] Constructed from:
│   │   ├── base[54]
│   │   └── lever[55]
│   └── stapler1[49] Constructed from:
│       ├── base[50]
│       ├── lever[51]
│       └── head[52]
├── holepunch[56] Of Types:
│   ├── holepunch-old[26]
│   └── holepunch-new[23] Constructed from:
│       ├── base[24]
│       └── handle[25]
├── liquidpaper[30]
├── storage[62] Of Types:
│   ├── disk[10] Constructed from:
│   │   ├── case[11]
│   │   ├── slide[12]
│   │   └── label[13]
│   └── cdrom[1] Constructed from:
│       ├── case[2]
│       └── label[3]
├── duster[17] Constructed from:
│   ├── felt[18]
│   └── handle[19]
├── receptacle[61] Of Types:
│   ├── cup[60] Of Types:
│   │   ├── tea[9]
│   │   └── espresso[8]
│   ├── mug[59] Of Types:
│   ├── mug[59] Of Types:
│   │   ├── mug-soup[38]
│   │   ├── mug-molly[34] Constructed from:
│   │   │   ├── body[35]
│   │   │   ├── face[36]
│   │   │   └── rim[37]
│   │   └── mug-craig[31] Constructed from:
│   │       ├── body[32]
│   │       └── rim[33]
│   └── drink[14] Constructed from:
│       ├── bottle[15]
│       └── lid[16]
├── calculator[4] Constructed from:
│   ├── case[5]
│   ├── keypad[6]
│   └── display[7]
├── tool[63] Of Types:
│   ├── pliers[45] Constructed from:
│   │   ├── handle[46]
│   │   ├── jaws[47]
│   │   └── handle[48]
│   └── keyset[27] Constructed from:
│       ├── holder[28]
│       └── keys[29]
├── gluestick[20] Constructed from:
│   ├── base[21]
│   └── lid[22]
└── pen[58] Of Types:
    ├── pen-green[42] Constructed from:
    │   ├── body[43]
    │   └── lid[44]
    └── pen-blue[39] Constructed from:
        ├── body[40]
        └── lid[41]
```

Fig. 6. Example of CITE's hierarchical knowledge-base: text description of office knowledge base

4. C. Dillon and T. Caelli. Cite: A scene understanding and object recognition system. In *Asian Conference on Computer Vision: ACCV95*, volume I, pages 214–218, Singapore, Dec 1995.
5. B. McCane and T. Caelli. A fuzzy machine learning approach to recognising 3D objects in 2D s cenes. In *Asian Conference on Computer Vision:ACCV95*, volume III, pages 106–111, Singapore, Dec 1995.

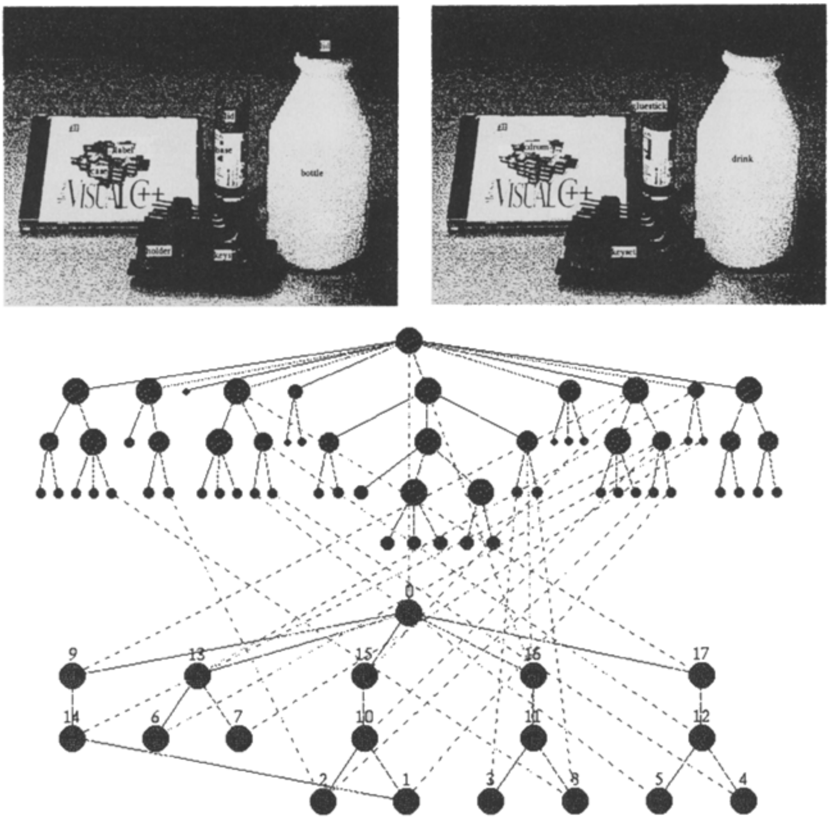
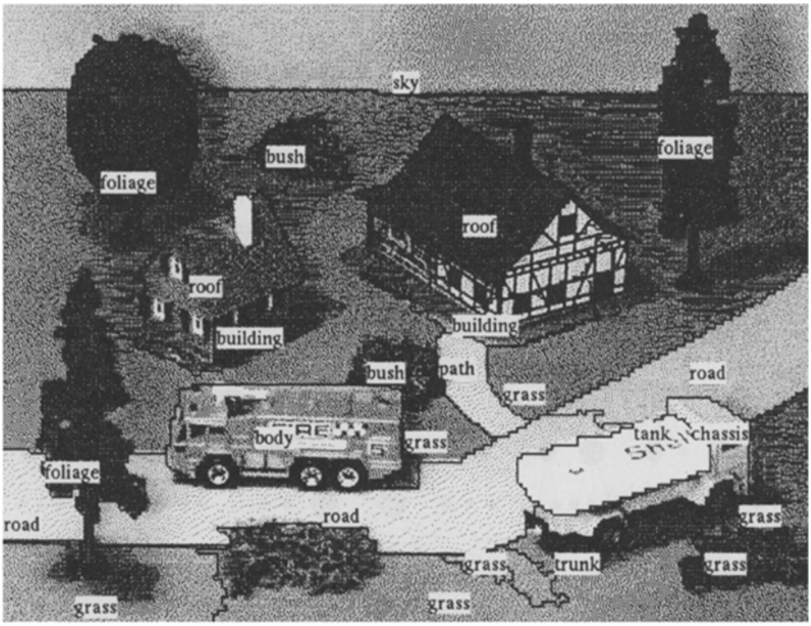


Fig. 7. Resultant interpretation represented as a match between different levels of representation. Here, observed features and their labels are shown in the bottom lattice while the current domain knowledge base corresponds to the top lattice. Labels on the bottom lattice correspond to the resultant interpretation according to the knowledge base labels defined in Figure 6.

6. B.T. Messmer and H. Bunke. Fast error-correcting graph isomorphism based on model precompilation. Technical Report IAM-96-012, University of Bern, Sep 1996.
7. T. Mitchell. *Machine Learning.* McGraw-Hill, 1997.
8. A. Pearce and T. Caelli. On the effiency of learning in spatial domains and relational evidence theory. volume 3, pages 290–294, Singapore, Dec 1995.
9. Prantl M. Ganster H. Pinz, A. and H. Kopp-Borot Schnig. Active fusion - a new method applied to remote sensing image interpreta tion. *Pattern Recognition Letters*, 17:1349–1359, 1996.
10. T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilay er networks. *Science*, 247:978–982, 1990.



```
World[0] (1.000) Consisting of:
 ├── sky[25] (0.876)
 ├── ground[26] (1.000) Of Type:
 │     └── grass[1] (0.825)
 ├── pencilpine[27] (1.000) Constructed from:
 │     ├── foliage[9] (0.932)
 │     └── trunk[5] (0.878)
 ├── pencilpine[29] (1.000) Constructed from:
 │     └── foliage[22] (0.922)
 ├── pencilpine[31] (1.000) Constructed from:
 │     └── foliage[23] (0.939)
 ├── house[36] (0.666) Constructed from:
 │     ├── building[19] (1.000)
 │     └── roof[18] (1.000)
 ├── fueltruck[44] (1.000) Constructed from:
 │     ├── chassis[11] (1.000)
 │     └── tank[10] (0.920)
 ├── dairy[45] (1.000) Constructed from:
 │     ├── roof[21] (0.934)
 │     └── building[20] (1.000)
 ├── ground[46] (1.000) Of Type:
 │     └── grass[2] (0.868)
 ├── ground[47] (1.000) Of Type:
 │     └── grass[3] (0.848)
 ├── ground[48] (1.000) Of Type:
 │     └── grass[4] (0.563)
 ├── ground[49] (1.000) Of Type:
 │     └── road[6] (1.000)
 ├── ground[50] (1.000) Of Type:
 │     └── grass[7] (0.687)
 ├── ground[51] (1.000) Of Type:
 │     └── road[8] (0.830)
 ├── firetruck[52] (1.000) Constructed from:
 │     └── body[12] (0.842)
 ├── ground[53] (1.000) Of Type:
 │     └── grass[13] (1.000)
 ├── ground[54] (1.000) Of Type:
 │     └── grass[14] (0.716)
 ├── ground[55] (1.000) Of Type:
 │     └── path[15] (1.000)
 └── ground[56] (1.000) Of Type:
       └── road[16] (1.000)
```

Fig. 8. Top: Labelled scene. Bottom: Text Description of Street Scene. Here labels in the instantiated knowledge base (bottom) refer to part labels in the knowledge base and numerical values correspond to certainty of the labelling (1:most certain).

11. J. R. Quinlan. Improved use of continuous attributes in c4.5. *Journal of Artificial Intelligence Research*, 1996.
12. S. Sarkar and K. Boyer. Using perceptual inference networks to manage visio n processes. *Computer Vision and Image Understanding*, 62(1):27–46, 1995.