

Parallel Techniques for Rule-Based Scene Interpretation

Walter F. Bischof and Terry Caelli

Department of Computing Science, University of Alberta, Edmonton, T6G 2H1,
Canada, (wfb,tcaelli)@ualberta.ca

Abstract. We consider a parallel, rule-based approach for learning and recognition of pattern and objects in scenes. Classification rules for pattern fragments are learned with objects presented in isolation and are based on unary features of pattern parts and binary features of part relations. These rules are then applied to scenes composed of multiple objects. We present an approach that solves, at the same time, evidence combination and consistency analysis of multiple rule instantiations. Finally, we introduce an extension of our approach to the learning of dynamic patterns.

1 Introduction

Over the last decades, research in computer vision has concentrated on recognizing simple, isolated objects in controlled situations, and consequently these systems often fail in complex, natural settings with many objects. More recently, researchers have realized that, in order to overcome these limitations, systems have to be enhanced with visual learning capabilities. Many of the learning techniques are investigated within symbolic, rule-based systems for recognizing specific and generic objects, and for recognizing events and complex scenes [6]. Such systems are suitable for incremental generation, modular organization and efficient application of recognition knowledge.

One successful approach to the learning of object recognition involves training a system with isolated objects in an interactive or supervised learning paradigm. Recognition rules are pre-compiled in the form of hashing schemes [7], interpretation tables [4], or decision trees [1, 5]. Most of these schemes rely on attribute hashing of single image regions or object parts, and use relational information only to a limited degree for hypothesis generation and model indexing. A generalized approach based on the use of (unary) attributes of parts and (binary) attributes of part relations is presented below [1, 10]. We show that relational pattern information can be generated adaptively and used efficiently for hypothesis generation.

The incorporation of relational pattern information into pre-compiled rules has important implications for rule application in complex scenes. It raises the question of how evidence from different rule instantiations should be combined, and, more importantly, how consistency between different rule instantiations

should be assessed. Consistency analysis and label updating can be easily done using the simple compatibility functions of classical relaxation labeling [13], but becomes non-trivial with complex classification rules. This is especially true when classification rules are applied to scenes composed of multiple objects where rules learned with single objects may be instantiated by pattern fragments "belonging" to different objects. To avoid misclassifications, parts belonging to the same object should be identified, and it has been traditionally assumed that this clique problem has already been solved (e.g. using perceptual grouping [8]). In contrast, we propose below an approach where classification of pattern fragments, evidence combination and the clique problem are solved at the same time.

The rule-based approach presented here shares many similarities with approaches based on inductive logic programming. However, the parallelism of our approach, both in rule learning and in rule application, is the major characteristics that sets our system apart from systems such as FOIL [11] or GOLEM. In rule learning, our system develops trees of decision-tree, and hence belongs to the class of parallel covering algorithms. In rule application, our system evaluates, again in parallel, all rule instations and thus is able to evaluate evidence combination, evidence consistency and the clique problem at the same time. It is this parallelism, we argue, that makes our approach feasible for learning complex visual data. A second major advantage of our approach is that it can be extended to fuzzy classifiers in a straightforward way, and experiments show that this can be done very effectively and efficiently [9].

In the following sections, we first present our approach to the generation and compilation of recognition rules, and then we discuss application of these recognition rules in scenes composed of multiple objects.

2 Learning of Spatial Patterns

We present an approach to pattern learning, termed Conditional Rule Generation (CRG, [1]) which is based on the following idea. Classification rules for patterns or pattern fragments are generated that include structural pattern information to the extent that is required for classifying correctly a set of training patterns. CRG analyzes unary and binary features of connected pattern components and creates a tree of hierarchically organized rules for classifying new patterns. Generation of a rule tree proceeds in the following manner (see Fig. 1):

First, the unary features of all parts of all patterns are collected into a unary feature space U in which each point represents a single pattern part. The feature space U is partitioned into a number of clusters U_i . Some of these clusters may be unique with respect to class membership and provide a classification rule: If a pattern contains a part p_r whose unary features $\mathbf{u}(p_r)$ satisfy the bounds of a unique cluster U_i then the pattern can be assigned a unique classification. The non-unique clusters contain parts from multiple pattern classes and have to be analyzed further. For every part of a non-unique cluster we collect the binary features of this part with all other parts in the pattern to form a (conditional) binary feature space UB_i . The binary feature space is clustered into a number of

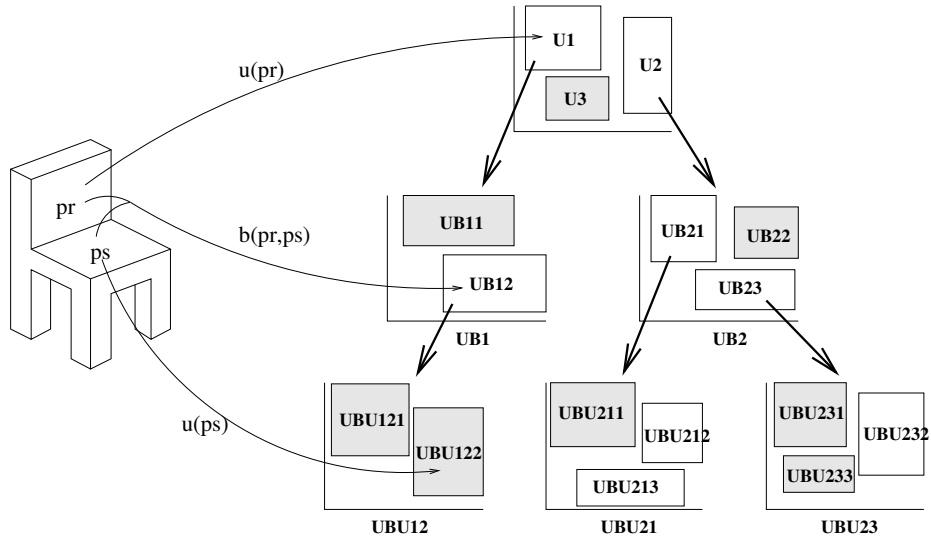


Fig. 1. Cluster tree generated by the CRG method. Grey clusters are resolved (i.e. contain elements of a single pattern class). Unresolved clusters (e.g. U_1 and U_2) are expanded to binary feature spaces (e.g. UB_1 and UB_2), from where clustering and expansion continues until either all rules are resolved or the predetermined maximum rule length is reached

clusters UB_{ij} . Again, some clusters may be unique and provide a classification rule: If a pattern contains a part p_r whose unary features satisfy the bounds of cluster U_i , and there is an other part p_s , such that the binary features $\mathbf{b}(p_r, p_s)$ of the pair $\langle p_r, p_s \rangle$ satisfy the bounds of a unique cluster UB_{ij} then the pattern can be assigned a unique classification. For non-unique clusters, the unary features of the second part p_s are used to construct another unary feature space UBU_{ij} that is again clustered to produce clusters UBU_{ijk} . This expansion of the cluster tree continues until all classification rules are resolved or maximum rule length has been reached.

If there remain unresolved rules at the end of the expansion procedure (which is normally the case), the generated rules are split into more discriminating rules using an entropy-based splitting procedure where the elements of a cluster are split along feature dimension such that the normalized partition entropy $H_P(T) = (n_1 H(P_1) + n_2 H(P_2)) / (n_1 + n_2)$ is minimized, where H is entropy. Rule splitting continues until all classification rules are unique or some termination criterion has been reached. This results in a tree of conditional feature spaces (as shown in Fig. 1), and within each feature space, rules for cluster membership are developed in the form of a decision tree. Hence, CRG generates a tree of decision trees.

A completely resolved rule tree provides a set of rules for classification of patterns. Every rule in the classification tree corresponds to a sequence $U_i -$

$B_{ij} - U_j - B_{jk} - \dots$ of unary and binary features associated with a chain of pattern parts and their relations. A pattern fragment $p_1 - p_2 - \dots - p_n$ can instantiate a classification rule of length m completely (if $n \geq m$) or partially (if $n < m$). In the former case, the pattern fragment is classified uniquely; in the latter case, classification uncertainty is reduced via the empirical class frequencies associated with nodes of the cluster tree.

It is important to note that the CRG algorithm is more general than classical decision trees, given that it develops descriptions in the form of Horn clauses $C \leftarrow U_1(X), B_1(X, Y), U_2(Y), B_2(Y, Z), \dots$ involving unary and relational attributes. At the same time, it is also more general than inductive logic programming approaches such as FOIL [11], given that the literals of the Horn clauses refer to bounded regions of continuous unary and binary feature spaces. Finally, it should be pointed out that CRG lends itself fairly naturally for extensions to fuzzy classifiers, and it has been shown that this can be done fairly effectively and efficiently [9].

3 Recognition of Spatial Patterns

CRG generates classification rules for (small) pattern fragments in the form of symbolic, possibly fuzzy Horn clauses. When the classification rules are applied to some new pattern one obtains one or more (classification) evidence vectors for each pattern fragment, and the evidence vectors have to be combined into a single evidence vector for the whole pattern. The combination rules can be learned [12], they can be knowledge-guided [3], or they can be based on general compatibility heuristics. In the latter approach, sets of instantiated classification rules are analyzed with respect to their compatibilities and rule instantiations that lead to incompatible interpretations are removed. This is particularly important in scenes composed of multiple patterns where it is unclear whether a chain $p_i - p_j - \dots - p_n$ of pattern parts belongs to the same pattern or whether it is “crossing the boundary” between different patterns. Our compatibility analysis makes only weak and general assumptions about the structure of scene and objects, and is based on the analysis of the relationships within and between instantiated rules [2]. The learning and test situation is illustrated in Fig. 2 that shows some objects in isolation, and a scene composed of multiple objects.

Initial Rule Evaluation The first rule application stage involves direct activation of the rules in a parallel, iterative deepening method. Starting from each scene part, all possible chains of parts are generated and classified using the CRG rules. The evidence vectors of all rules instantiated by a chain $S = \langle p_1 p_2 \dots p_n \rangle$ are averaged to obtain the evidence vector $\mathbf{E}(S)$ of the chain S , and the set \mathcal{S}_p of all chains that start at p is used to obtain an initial evidence vector for part p :

$$\mathbf{E}(p) = \frac{1}{\#(\mathcal{S}_p)} \sum_{S \in \mathcal{S}_p} \mathbf{E}(S), \quad (1)$$

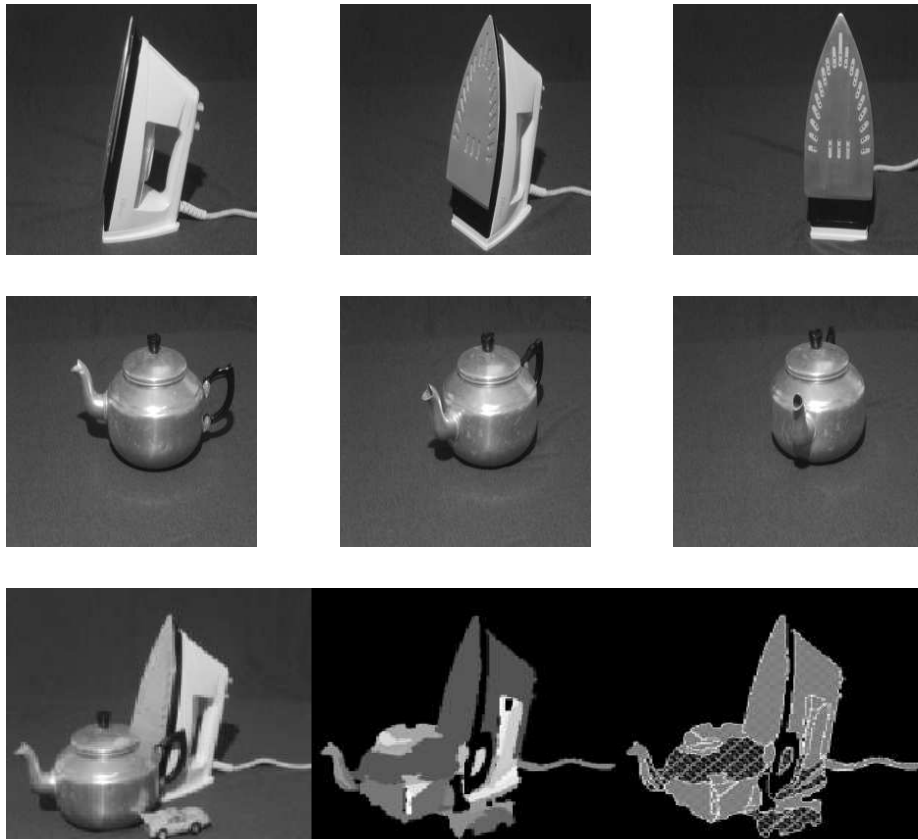


Fig. 2. The first two rows of show several views of objects that are used in the learning phase. The third row shows a scene composed of several objects, with the input image on the left, the segmentation result in the middle, and the classification result on the right (Adapted from [9])

where $\#(\mathcal{S})$ denotes the cardinality of the set \mathcal{S} . As discussed before, evidence combination based on (1) does not take into account the fact that some rule instantiations may be incorrect and incompatible with the rest. To the extent that such incompatible rule instantiations can be detected, the part classification (1) can be improved. Compatibility analysis involves an analysis of compatibilities between and within chains of pattern parts.

Inter-chain Analysis The inter-chain compatibility analysis is based on the following general idea: 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 (1). More formally, let $S_i = \langle p_{i1}p_{i2} \dots p_{in_i} \rangle$ and $S_j = \langle$

$p_{j1}p_{j2} \dots p_{jn_j} >$ be touching chains, and let T_{ij} be the set of common parts, i.e. $T_{ij} = \{p \mid \exists k p = p_{ik} \text{ and } \exists l p = p_{jl}\}$ with $\#(T_{ij}) > 0$. The compatibility of S_i and S_j , $C(S_i, S_j)$ is defined as

$$C(S_i, S_j) = \frac{1}{\#(T_{ij})} \sum_{p \in T_{ij}} \frac{\#(\mathcal{M}(p|S_i) \cap \mathcal{M}(p|S_j))}{\#(\mathcal{M}(p|S_i) \cup \mathcal{M}(p|S_j))}. \quad (2)$$

The overall compatibility of a chain S_i is then defined with respect to the set \mathcal{S}_T of chains that touch S_i , i.e. $\mathcal{S}_T = \{S_j \mid \#(T_{ij}) > 0\}$:

$$w_{\text{inter}}(S_i) = \frac{1}{\#(\mathcal{S}_T)} \sum_{S \in \mathcal{S}_T} C(S_i, S). \quad (3)$$

Using the inter-chain compatibility, we can now modify the original averaging for the part evidence vectors in (1) to

$$\mathbf{E}(p) = \frac{\sum_{S \in \mathcal{S}_p} w_{\text{inter}}(S) \mathbf{E}(S)}{\sum_{S \in \mathcal{S}_p} w_{\text{inter}}(S)}, \quad (4)$$

where \mathcal{S}_p is defined as in (1).

Intra-chain Analysis The intra-chain analysis for detecting boundary-crossing chains is based on the following idea. If a chain $S_i = \langle p_{i1}p_{i2} \dots p_{in} \rangle$ does not cross boundaries of objects then the evidence vectors $\mathbf{E}(p_{i1})$, $\mathbf{E}(p_{i2})$, \dots , $\mathbf{E}(p_{in})$ computed by (4) are likely to be similar, and dissimilarity of the evidence vectors suggests that S_i may be a ‘‘crossing’’ chain. 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 chain. This measure is captured in the following way. For a chain $S_i = \langle p_{i1}p_{i2} \dots p_{in} \rangle$,

$$\mathbf{w}_{\text{intra}}(S_i) = \frac{1}{n} \sum_{k=1}^n \mathbf{E}(p_{ik}), \quad (5)$$

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 chains which begin with part p_{ik} .

Relaxation Scheme Taking together inter- and intra-chain analysis, our compatibility measure can be used with a relaxation labeling scheme for updating the part evidence vectors of the following form:

$$\mathbf{E}^{(t+1)}(p) = \Phi \left(\frac{1}{Z} \sum_{S \in \mathcal{S}_p} \mathbf{w}_{\text{inter}}^{(t)}(S) \mathbf{w}_{\text{intra}}^{(t)}(S) \otimes \mathbf{E}(S) \right), \quad (6)$$

where Φ is the logistic function, Z is a normalizing factor, and the binary operator \otimes is defined as a component-wise vector multiplication $[a \ b]^T \otimes [c \ d]^T = [ac \ bc]^T$.

For a given chain $S_i = \langle p_{i1}, p_{i2}, \dots, p_{in} \rangle$ of parts, the updating scheme (6) not only takes into account the compatibility between evidence vectors of all parts p_i but also the compatibility between the average evidence vectors and the chain's evidence vector. The updating scheme (6) (together with (5), (3), and (4)) defines a (possibly fuzzy) inference procedure that can be executed in parallel for all parts of a scene, and that solves at the same time, evidence combination and consistency analysis of rule instantiations as well as the clique problem.

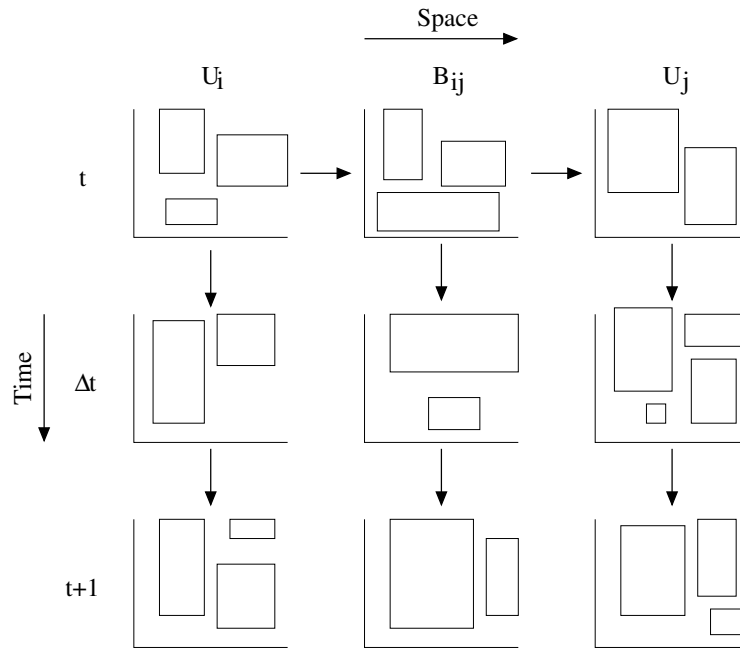


Fig. 3. A sketch of the overall organization of a spatiotemporal cluster tree. Spatial expansions are along the horizontal, temporal expansions along the vertical. The cluster tree shown in Fig. 1 is sketched along the top row (where two unary and a (spatial) binary feature space are shown), temporal dependencies and expansions are shown vertically (where again two unary and a (temporal) binary feature space are shown). See text for further explanations

4 Dynamic Patterns and Scenes

The previous sections introduced our approach to the learning and classification of spatial patterns. In this section, we sketch a generalization of CRG into the temporal domain, CRG_{ST} , for learning dynamic ("spatiotemporal") patterns and its application to animated scenes. For a set of spatiotemporal

patterns $P_t = \{p_{1t}, \dots, p_{nt}\}$, $t = 1, \dots, T$, we define the following features: spatial unary features $\mathbf{u}(p_{it})$ (e.g. area, brightness, position), spatial binary features $\mathbf{b}(p_{it}, p_{jt})$ (e.g. distance, relative size), temporal changes in unary features $\Delta\mathbf{u}(p_{it}, p_{it'})$ (e.g. velocity, acceleration), and temporal changes in binary features $\Delta\mathbf{b}(p_{it}, p_{jt}, p_{it'}, p_{jt'})$ (e.g. relative velocity). As before, pattern classification is learned in a supervised learning paradigm, and learning of classification rules proceeds in the following way (see Fig. 3):

First, the unary features of all parts (of all patterns at all time points), $\{p_{it}\}$, $i = 1, \dots, n$, $t = 1, \dots, T$, are collected into a unary feature space U in which each point represents a single pattern part at any time point $t = 1, \dots, T$. From this unary feature space, cluster tree expansion proceeds in two directions, in the spatial domain and in the temporal domain. In the spatial domain (along the horizontal direction in Fig. 3), cluster tree generation proceeds exactly as described in Section 2. Each of these feature spaces now also be expanded in the temporal domain by analyzing recursively temporal changes in unary ($\Delta\mathbf{u}$) and binary ($\Delta\mathbf{b}$) attributes within a limited temporal window. Rule expansion and refinement proceeds along the same lines as discussed in Section 2 and [1], leading to a set of rules for classifying spatiotemporal fragments, i.e. for pattern fragments and their changes within a restricted temporal window.

References

1. W. F. Bischof and T. Caelli. Learning structural descriptions of patterns: A new technique for conditional clustering and rule generation. *Pattern Recognition*, 27:1231–1248, 1994.
2. W. F. Bischof and T. Caelli. Scene understanding by rule evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:1284–1288, 1997.
3. C. Dillon and T. Caelli. Cite – scene understanding and object recognition. In T. Caelli and W. F. Bischof, editors, *Machine Learning and Image Interpretation*, pages 119–187. Plenum, New York, NY, 1997.
4. P. Flynn and A. K. Jain. 3D object recognition using invariant feature indexing of interpretation tables. *Computer Vision, Graphics, and Image Processing*, 55:119–129, 1992.
5. L. Grewe and A. Kak. Multi-hash: Learning object attributes and hash tables for fast 3-d object recognition. In K. Ikeuchi and M. Veloso, editors, *Symbolic Visual Learning*, pages 9–47. Oxford University Press, New York, NY, 1997.
6. K. Ikeuchi and M. Veloso, editors. *Symbolic Visual Learning*. Oxford University Press, New York, NY, 1997.
7. Y. Lamdan and H. Wolfson. Geometric hasing: A general and efficient model-based recognition scheme. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1407–1413, 1988.
8. D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31:355–395, 1987.
9. B. McCane and T. Caelli. Fuzzy conditional rule generation for the learning and recognition of 3d objects from 2d images. In T. Caelli and W. F. Bischof, editors, *Machine Learning and Image Interpretation*, pages 17–66. Plenum, New York, NY, 1997.

10. A. Pearce and T. Caelli. Relational evidence theory and interpreting schematics. In T. Caelli and W. F. Bischof, editors, *Machine Learning and Image Interpretation*, pages 67–118. Plenum, New York, NY, 1997.
11. J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.
12. D. H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.
13. S. Zucker, E. V. Krishnamurty, and R. L. Haar. Relaxation processes for scene labeling: Convergence, speed, and stability. *IEEE Transactions on Systems, Man and Cybernetics*, 8:41–48, 1978.