An Object Recognition Model Based on Visual Grammars and Bayesian Networks

Elias Ruiz and Luis Enrique Sucar

Instituto Nacional de Astrofísica, Óptica y Electrónica
Departamento de Ciencias Computacionales
Luis Enrique Erro 1, Tonantzintla, Puebla, México
{elias_ruiz,esucar}@inaoep.mx
http://www.inaoep.mx

Abstract. A novel proposal for a general model for object recognition is presented. The proposed method is based on symbol-relational grammars and Bayesian networks. An object is modeled as a hierarchy of features and spatial relationships using a symbol-relational grammar. This grammar is learned automatically from examples, incorporating a simple segmentation algorithm in order to generate the lexicon. The grammar is created with the elements of the lexicon as terminal elements. This representation is automatically transformed into a Bayesian network structure which parameters are learned from examples. Thus, recognition is based on probabilistic inference in the Bayesian network representation. Preliminary results in modeling natural objects are presented. The main contribution of this work is a general methodology for building object recognition systems which combines the expressivity of a grammar with the robustness of probabilistic inference.

Keywords: Visual Grammars, Bayesian Networks, Object Recognition

1 Introduction

Most current object recognition systems are centered in recognizing certain type of objects, and do not consider their structure. This implies several limitations: (i) the systems are difficult to generalize to any type of object, (ii) they are not robust to noise and occlusions, (iii) the model is difficult to interpret.

This paper proposes a model that achieves a hierarchical representation of a visual object in order to perform object recognition tasks, based on a visual grammar [3] and Bayesian networks (BNs) [8]. Thus, we propose the incorporation of a visual grammar in order to develop an understandable hierarchical model so that from basic elements (obtained by a simple image segmentation algorithm) it will construct more complex forms by certain rules of composition defined in the grammar, in order to achieve object recognition in a limited context (e.g. images of natural objects).

The importance of addressing this issue from a hierarchical approach is that it can build a more robust model which can represent variability in a class of

R. Klette, M. Rivera, and S. Satoh (Eds.): PSIVT 2013, LNCS 8333, pp. 349–359, 2014.

objects and can handle occlusions and partial information. This is combined with the advantages of a BNs for dealing with uncertainty, such as incomplete information and noise. Additionally, a model expressed as a symbolic grammar provides a transparent and understandable representation.

We propose a method for learning the grammar from examples and then transforming this representation to Bayesian network for object recognition using probabilistic inference. The terminal elements are also learned from examples, based on simple and general visual features (edges and uniform regions), which provide the lexicon for the grammar. The grammar is a symbol-relation grammar which incorporates spatial relationships.

A symbol-relation grammar is learned for each class of object, and then transformed automatically to a Bayesian network which incorporates the symbols and relations as nodes, and the arcs represent the structure derived from the grammar rules. Intermediate nodes in this BN structure are hidden, so we learn the parameters of the model using Expectation-Maximization algorithm (EM). Once the structure and parameters of the BN are obtained, it can be used for recognizing a class of object using probabilistic inference.

The proposed method has been evaluated experimentally with several classes of natural objects with promising results. The main contribution of this work is a general methodology for building object recognition systems which combines the expressivity of a grammar with the robustness of probabilistic inference.

Next we present a brief review of alternative hierarchical approaches for object recognition and contrast them with our approach. Then we describe in detail the proposed model, including the model building and recognition methods. We present experimental results in learning visual grammars for several object classes, and then using these for recognition. We conclude with a summary and directions for future work.

2 Related Work

There are several works using a hierarchical approach for object recognition based on visual grammars [1,2,6,7,9]. In these studies, there is a clear consensus in the usage of a certain kind of grammar to represent hierarchically the terminal elements (Lexicon). However, they differ in what terminal elements to use and how to handle the uncertainty in order to perform object recognition. In addition, they are usually designed for specific types of objects (e.g., car plate recognition, pedestrian or face recognition) so that the models developed are difficult to generalize. In these works the grammar is described manually. Similarly, the model which handles the uncertainty is restricted to a fixed structure.

The proposed model differs in several aspects from previous work:

- It is based on a symbol-relation grammar which incorporates spatial relationships.
- The grammar is learned from example images of a class of objects.
- The terminal elements are simple and general so they can be used for different types of objects, and the lexicon is also learned from examples.

– The grammar is automatically transformed to a BN which provides a robust and efficient techniques for object recognition.

We consider the use of Symbol-Relation grammars because of the convenience of putting the relationships in predicate logic, which is natural in this kind of grammar. Also, it is desirable that the grammar is automatically learned from examples, for greater generality; the grammar is independent of the lexicon definition used. Finally, the transformation to a model that considers uncertainty must also be automatic and analogous (but independent) to the hierarchy described in the grammar.

We use Bayesian networks to represent the information given by the grammar incorporating uncertainty. Other studies use different schemas or even probabilistic grammars. Bayesian networks have several advantages, such a preserving the structure given by the grammar and providing efficient algorithms for parameter learning and probabilistic inference.

3 Object Recognition Model

The proposed method compromises two phases: (i) model construction and transformation to a BN (Fig. 1); and (ii) image pre-processing and object recognition using probabilistic inference, (Fig. 2). Next we describe each phase in detail.

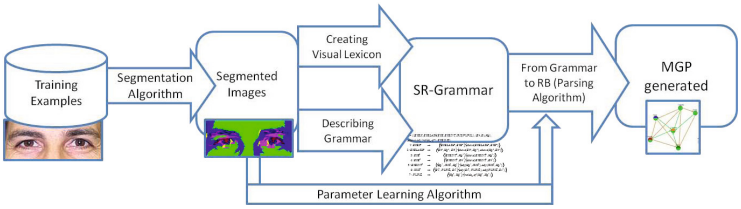


Fig. 1. Model construction. Starting from training images, these are segmented and a lexicon and a visual grammar are induced, obtaining a description of the objects in terms of the lexicon and a SR grammar. Then the model is transferred to a BN, whose parameters are also learned from examples.

3.1 Model Construction

Segmentation and Lexicon. Segmentation is performed with simple RGB quantization (32 levels) and edge extraction using Gabor Filters [4]. Small regions are deleted by fusion with other regions. The idea is to use simple and general features as basic elements so they can be applied for different classes of objects. These regions define a visual dictionary. Every region is described with

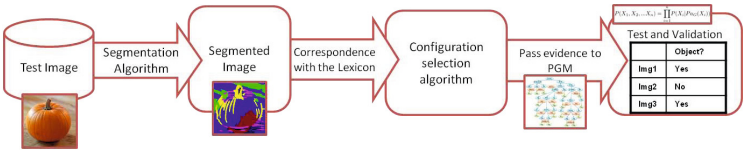


Fig. 2. Object recognition. The test image is segmented with the visual dictionary and we obtain correspondences of regions with the visual lexicon. After that, the algorithm evaluates subsets of those regions with their spatial relationships that are candidates to be evaluated in the previously trained BN in order to do inference. At the end, we obtain a result by probabilistic inference in the BN obtaining the probability of the presence of an object in the image.

Fig. 3. In the picture, the color segments are examples of terminal elements found. Each segment is attached with an element of the Lexicon. The Lexicon is composed of edge type elements and homogeneous regions, given by a segmentation algorithm based on quantization of the RGB channels. In this example two orientations of edge elements were used. (Best seen in color.)

its color histogram and shape features. From a segmented dataset with negative and positive images, we use a *k-means* algorithm in order to select the clusters which appear more often in the positive images (two times in positive images against negative images). The centroids of these clusters are considered as terminal elements in our grammar. All the terminal elements constitute the *Visual Lexicon*. An example of terminal elements obtained by this method are illustrated in the Fig. 3. According to our model, the Lexicon can be improved by incorporating local features or a better segmentation algorithm. These changes may provide better results in recognition tasks. This will not affect other layers of our model like grammar learning or its transformation into a Bayesian Network.

Spatial Relationships and Candidate Rules. Although there are different types of spatial relationships, in our model we use topological and order

relationships. The relationships used in our model were: $Inside\_of(A, B)$ (A region is within B region), $Contains(A, B)$ (A region covers completely B region), $Ady(A, B)$ (A is touched by B and A is located left from B), $Above(A, B)$ (A is touched by B and A is located above from B), $Invading(A, B)$ (A is covering partially B more than $Above$ and $Left$ but less than $Contains$). In each relationship we also consider that the two regions are also adjacent. We use these relationships because they preserve the coherence when we subsume two regions in another new one. The new non-terminal elements generated preserve all the relationships from its children with other elements, and loose its internal relationships.

Learning the Grammar. A visual grammar describes objects hierarchically. It can represent a diagram, flowchart, or a geometric drawing. For example, the description of a flowchart is made by decomposition: complex elements are decomposed in simple elements (from the complete image to arrows or simple boxes). For our model, we need a grammar that allows us to model the decomposition of a visual object into its parts and how they relate with another parts. Symbol-Relation grammars *(SR-grammars)*, which are described in [3], provide this type of description and incorporate the possibility of adding rewriting rules (R) to specify relationships between terminals and non-terminals symbols after decompositions from all the non-terminals. A Symbol Relation-Grammar is defined as:

$G = (V_N, V_T, V_R, S, P, R)$ where:

- V_N is a finite set of non terminal symbols.
- V_T is a finite set of terminal symbols.
- V_R is a finite set of relational symbols between $V_N \cup V_T$.
- $S \in V_N$ is the starting symbol.
- P is a finite set of labelled rewriting rules, called s-item productions of the form:

$$l : Y^0 \rightarrow \langle \mathbf{M}, \mathbf{R} \rangle$$

  where:
  - l is a integer labelling the s-production.
  - $\langle \mathbf{M}, \mathbf{R} \rangle$ is a sentence on V_R and $V_N \cup V_T$
    - $\mathbf{M}$ is a set of s-items (v, i) with $v \in V_N \cup V_T$ and i is a natural number used to distinguish different ocurrences of the same symbol.
    - $\mathbf{R}$ is a set of r-items of the form $r(X^i, Y^j)$, with X^i, $Y^j \in \mathbf{M}$and $r \in V_R$
  - $Y \in V_N$, $Y^0 \notin \mathbf{M}$

As a convention, the index "0" will only be used to denote the symbol on the left-hand side of every s-production.

The next step is to generate the rules that make up the grammar. Using the training images, we search the most common relationships between the clusters obtained. Such relationships become candidate rules to build the grammar. This is an iterative process where the rules are subsumed and converted to a new

non-terminal elements of the grammar. This new non-terminal element can be seen as a product of the composition process: two terminal regions in the image are merged into a new one. This new region is the new non-terminal element in the grammar. If we repeat this process, the starting symbol of the grammar represents the object that we want to recognize.

The stop criterion (to learn each rule) is a frequency threshold for the rule (the rule needs to be found in at least n images of the training set). This criterion also avoids generating a higly complex grammar. As an example, the relationship $Inside\_of(C_1, C_2)$ is subsumed into a new non-terminal element named NT_1. The rule obtained is: $1 : NT_1^0 \rightarrow < \{C_1^2, C_2^2\}, \{Inside\_of(C_1^2, C_2^2)\} >$. As a convention, the superscript zero will only be used to denote the symbol on the left-hand side of every s-production, the superscript 2 is used on the right side. Superscript 3 or higher are used when there are two or more instances of one terminal or non-terminal element in the same rule. Superscript 1 is not used.

We incorporate a restriction in SR-grammars in order to avoid circular productions (the PGM generated would have infinite structure). For example, these two rules are not allowed: $A^0 \rightarrow \langle B^2 \rangle$ and $\mathrm{B}^0 \rightarrow \langle A^2 \rangle$, where A produces B and B produces A. In a formal sense, the restriction is as follows: for every rule of the form $Y^0 \rightarrow \langle \mathbf{M}, \mathbf{R} \rangle$, where $\mathbf{M}$ is a set of terminal and non terminal elements, and $\mathbf{R}$ is a set of relationships between elements of $\mathbf{M}$, we have that, $\forall x \in \mathbf{M}$ is not a daughter of Y^0. Thus, the learned grammar will not have cycles.

Transformation of the Grammar. We transform the grammar into a BN, using the following procedure. For every production rule, $Y^0 \rightarrow \langle \mathbf{M}, \mathbf{R} \rangle$, we produce the node Y^0 in the grammar and connect this node with all $x \in \mathbf{M}$. For every relationship $r(a, b) \in \mathbf{R}$ we produce the node r connected with its parents $a, b \in \mathbf{M}$.

The transformation procedure is illustrated with the following example. If we consider the next grammar (we have deleted the superscripts for simplicity):
G = (VN, VT, VR, S, P, R);
$VN = \{NT\_R2\_C24\_C12, NT\_R1\_C9\_C25, NT\_R1\_C9\_C26, NT\_R7\_C9\_C31,$
$NT\_R1\_C42\_C9, NT\_R1\_C49\_C19, NT\_R1\_C9\_C69, NT\_R3\_C11\_C96\}$;
$VT = \{Term\_C24, Term\_C12, Term\_C9, Term\_C19, Term\_C11\}$;
$VR = \{Above, Ady, Invading, Inside\_of\}$;
$S = NT\_R3\_C11\_C96$

where the rule productions are defined by P :

1. $NT\_R3\_C11\_C96 \rightarrow < \{Term\_C11, NT\_R1\_C9\_C69\}, \{Inside\_of(Term\_C11, NT\_R1\_C9\_C69)\} >$;
2. $NT\_R1\_C9\_C69 \rightarrow < \{Term\_C9, NT\_R1\_C49\_C19\}, \{Ady(Term\_C9, NT\_R1\_C49\_C19)\} >$;
3. $NT\_R1\_C49\_C19 \rightarrow < \{NT\_R1\_C42\_C9, Term\_C19\}, \{Ady(NT\_R1\_C42\_C9, Term\_C19)\} >$;
4. $NT\_R1\_C42\_C9 \rightarrow < \{NT\_R7\_C9\_C31, Term\_C9\}, \{Ady(NT\_R7\_C9\_C31, Term\_C9)\} >$;
5. $NT\_R7\_C9\_C31 \rightarrow < \{Term\_C9, NT\_R1\_C9\_C26\}, \{Invading(Term\_C9, NT\_R1\_C9\_C26)\} >$;
6. $NT\_R1\_C9\_C26 \rightarrow < \{Term\_C9, NT\_R1\_C9\_C25\}, \{Ady(Term\_C9, NT\_R1\_C9\_C25)\} >$;
7. $NT\_R1\_C9\_C25 \rightarrow < \{Term\_C9, NT\_R2\_C24\_C12\}, \{Ady(Term\_C9, NT\_R2\_C24\_C12)\} >$;
8. $NT\_R2\_C24\_C12 \rightarrow < \{Term\_C24, Term\_C12\}, \{Above(Term\_C24, Term\_C12)\} >$;

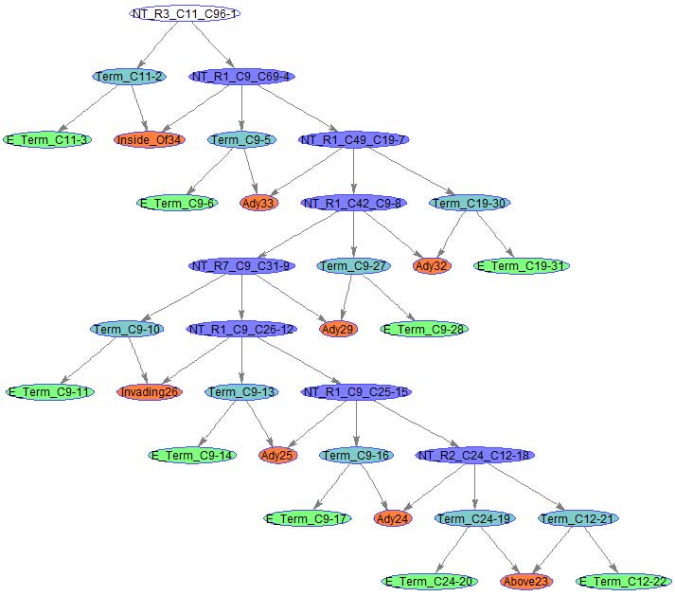


Fig. 4. Bayesian Network generated by the example grammar. Evidence is given only to leaf nodes. Nodes with two parents (in red) represent relationship nodes. Leaf nodes with only one parent (in green) represent terminal elements. Other nodes (in dark blue and light blue) represent non-terminal and terminal elements (but evidence is not given to these ones).

The algorithm generates the structure of a Bayesian network illustrated in Fig. 4.

Parameter Learning. Once the BN is obtained, its parameters are learned from examples, using positive and negative validation sets. The intermediate elements in the BN (nodes which are related to non-terminal elements in the SR-grammar) are considered as hidden nodes, so we use Expectation-Maximization (EM) to learn the parameters of the network. In each experiment we iterated the EM algorithm ten times at most. We use a subjective initialization for the probabilities of the hidden nodes, and the parameters of the evidence (terminal) nodes are initialized from the training image features.

3.2 Object Recognition

For object recognition, an image is initially segmented and converted to regions which are mapped to the terminal elements of the lexicon. Finding a valid configuration means to discover a relationship that has a match with the grammar rule as represented in the BN. This match is converted to evidence in the BN. The relationship is subsumed in the image and the process is repeated until the

grammar is completed or the image has no configuration applicable to the BN. If the complete grammar is found, there is a high probability that the object learned with the grammar appears in the image. The method is briefly described in Alg. 1. An example of some configurations obtained in a sample image are illustrated in the Fig. 5.

Algorithm 1: Grammar Search. BN is the built Bayesian Network from de Grammar.

```
Data: BN, Image ;                                   /* Segmented Image */
Result: Candidate-Grammar ;        /* corresponding with the image */
foreach Relationship of V_T's in BN do
    foreach Relationship applicable in Image do
        Call to Expand with BN and Image
        Check Evidence in Relationships and nodes
        Seek relationships in Image that match with
        neighbor relationships in BN
        // side, below and upward
        if No relationships in Image or BN has no more nodes then
            Accept Candidate-Grammar and finish
        if there are n relationships in Image then
            Call to Expand for every relationship (Recursion)
```

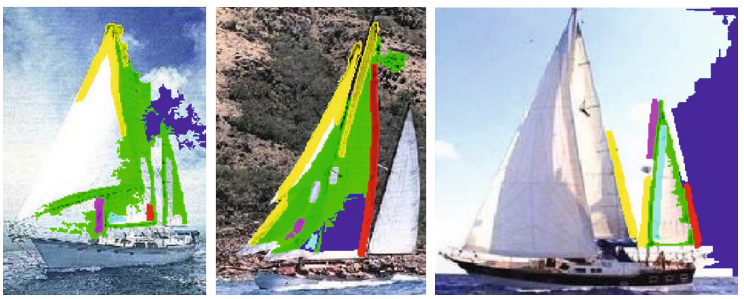
4 Results

To evaluate experimentally the proposed model, we consider different object classes from the Caltech database [5]: bikes, sailboats, faces, airplanes, compact-disc, soccer-ball, boombox. A grammar is learned for each object class using a training set (45 images), and the parameters for the corresponding BN are estimated from a validation set (other 45 images). Then the obtained model is tested using a different set of test images (the number varies between 90 and 668). Recognition is evaluated based on the posterior probability given by probability propagation in the BN (according to a predefined threshold that provides a compromise between precision and recall)

Examples of detected objects are illustrated in Fig. 5, 6 and 7; as in previous images, different colors in these images are used for a better differentiation only. As we can see, the method discovered certain regions (terminal elements) that are category specific. Also, the grammar and corresponding BN are particular for each class of object.

(a) bike 1 (b) bike 2 (c) bike 3

Fig. 5. Regions detected for a bike example. The colored regions are the terminal elements detected and provided as evidence to the BN. Different colors are used for a better differentiation only. In spite of the simple segmentation algorithm, the grammar helps to detect parts of the object. The grammar has detected the background and edges of the bike, because these were the most invariant elements along the images of the training set.



(a) sailboat 1 (b) sailboat 2 (c) sailboat 3

Fig. 6. Example of sailboat images. The sails (white regions) and its edges are detected. Although the model detects only part of the object (the sail), this part is more invariant than the rest of the object and provides a good clue for sailboat recognition; this was discovered automatically by the method.

Recognition results are evaluated in terms of accuracy, precision and recall:

$$\begin{aligned} Accuracy &= \frac{TP+TN}{TP+FP+TN+FN} \\ Precision &= \frac{TP}{TP+FP} \\ Recall &= \frac{TP}{TP+FN}, \end{aligned}$$

where TP is the true positive, TN the true negative, FP the false positive and FN the false negative rate in each experiment. The model obtained for each

Fig. 7. Example of face images. The detected terminal elements are depicted, which correspond to different parts of the face that the system discovered were appropriate for recognizing faces.

Table 1. Recognition results of the model in Caltech 256 database. Positive examples are obtained from the specified class and negative examples are obtained from the background class (257-clutter).

Class	Accuracy	Precision	Recall	Time	Num Examples
Faces101	84.88	88.44	80.23	50 min.	668
Airplanes	75.6	69.2	92.22	12 min.	180
compact-disc	75.5	83.82	63.33	14 min.	180
Soccer Ball	74.4	84.33	60.00	13 min.	180
Boombox	72.7	73.56	71.11	12 min.	180

class of object was evaluated with a set of test images, that include positive and negative examples. The negative examples were obtained from the background dataset images, provided in the 256-Caltech Database. The results for several classes of objects are summarized in the Table 1; with an average of nearly 80% precision with similar average recall. Although these results are in general not superior to other methods in the state of the art, we consider that they are promising as the proposed method provides a general framework that still needs to be optimized. For example, the definition of the terminal elements can be improved by extending the types of uniform and edge regions. Also, currently the grammar is restricted to binary relations and does not incorporate alternatives ("OR" rules). Lastly, the learned relations are currently hard and can be extended to consider partial relations.

5 Conclusions and Future Work

A novel and general model for object recognition based on visual grammars and Bayesian Networks was described. This approach combines Symbol-Relation grammars and Bayesian networks to describe an object in an image. The Bayesian Network is generated automatically from the grammar, and the grammar is

learned automatically from examples. The terminal elements (lexicon) and parameters are also learned from examples. Object recognition is done via probabilistic inference in the BN model. We have performed preliminary experiments with several natural object classes with promising results.

The main contribution of this work is proposing a general methodology for developing object recognition systems, that combines the richness and expressivity of formal grammars and the robustness and efficiency of Bayesian networks. We consider that this work contributes to the final goal of developing more general vision systems, analogous to those developed for voice and language.

There are several avenues for future research:

1. Improve and extend the visual dictionary by enriching the types of terminal elements.
2. Extend the grammar to incorporate rules with more than two elements as well as OR rules.
3. Consider partial relations when learning the grammar.
4. Evaluate the model with other classes of objects.

References

1. Chang, L., Jin, Y., Zhang, W., Borenstein, E.: Context, Computation, and Optimal ROC Performance in Hierarchical Models. International Journal of Computer Vision 93(2), 117–140 (2011)
2. Felzenszwalb, P.F.: Object Detection Grammars. In: ICCV Workshops, p. 691. IEEE, Barcelona (2011)
3. Ferrucci, F., Pacini, G., Satta, G., Sessa, M.I., Tortora, G., Tucci, M., Vitiello, G.: Symbol-relation grammars: a formalism for graphical languages. Inf. Comput. 131(1), 1–46 (1996)
4. Gabor, D.: Theory of Communication. JIEE 93(3), 429–459 (1946)
5. Griffin, G., Holub, A., Perona, P.: Caltech-256 Object Category Dataset. Technical Report 7694, California Institute of Technology (2007)
6. Meléndez, A., Sucar, L.E., Morales, E.F.: A Visual Grammar for Face Detection. In: Kuri-Morales, A., Simari, G.R. (eds.) IBERAMIA 2010. LNCS, vol. 6433, pp. 493–502. Springer, Heidelberg (2010)
7. Ommer, B., Buhmann, J.M.: Learning Compositional Categorization Models. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3953, pp. 316–329. Springer, Heidelberg (2006)
8. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann, San Francisco (1988)
9. Zhu, S.C., Mumford, D.: A Stochastic Grammar of Images. Foundations and Trends in Computer Graphics and Vision 2(4), 259–362 (2006)