Languages for the structural models for constrained image segmentation

Jan Čech and Radim Šára
cechj@cmp.felk.cvut.cz
tel.: +420 2 2435 5726

Center for Machine Perception
Prague, Czech Republic

TN-eTRIMS-CMP-03-2007
version 1.0

May 10, 2007

Abstract

We first review grammar approaches in the literature. The review shows the notion of two-dimensional grammars is not stabilized yet, since it is understood differently by various authors. Moreover, there are not too many successful computer vision applications based on 2D grammars.

Then, we compare two-dimensional grammar approach to other type of structure modeling, the labeling framework under which we implemented a windowpane detection task. The labeling is similar to 2D grammars to some extent, since it also strongly restricts a possible solution. However, there are principal differences between these two approaches leading to their different properties.

1 Introduction

Modeling the *structure* of the system, i.e. modeling the relations among individual sub-parts, improves the posedness of tasks like recognition, segmentation, detection compared to modeling individual sub-parts independently. This is the fundamental motivation for syntactical pattern recognition [10]. In our considerations, the sub-parts may be regions in the image corresponding to real world objects. The simplest sub-parts are defined in pixel-level, where each individual pixel has its semantic meaning. Or the sub-parts are larger image regions which may or may not have some hierarchical structure.

One of the basic tool of syntactic pattern analysis is the *grammar*. The grammar describes a language which represents a possible system configuration. A typical task is to decide if a string representing estimated configuration

belongs to the language induced by the grammar. This task is called the exact matching or simply *parsing*. Or, in the stochastic generalization, the task is to find the closest string belonging to the language induced by the grammar given a sequence of observations. This is called the *best matching* task.

Although one-dimensional grammars generating strings are nowadays well established, this is not the case of *two-dimensional grammars*. An image has a natural 2D topology, and cannot be easily handled by one-dimensional string description. Therefore various authors defined several kinds of 2D grammars which differ by allowed form of terminal and non-terminal symbols, and production rules, this is e.g. [17, 24, 7]. The problem is with the algorithms for solving the parsing or best matching task in the generalized two-dimensional grammars. Often, the algorithms are not known. Even worse, it is known that these task are NP-complete for certain grammars. In [20], there are defined 2D grammars which are provably solvable.

Schlesinger in [21, 22] defines a context-free two-dimensional grammar as a direct generalization of a context-free one-dimensional grammar in the Chomsky normal form, see Sec. 2 in more details. They propose to use a straightforward generalization of Cocke-Younger-Kasami (CYK) algorithm [12] for both parsing and best matching task. A successful application of this approach is their recognition of printed notes [23].

Another application of 2D grammars is [1], where facades of modern buildings are interpreted using combined Bayesian model from stochastic context-free grammar and MCMC sampling to approximate the posterior given an image.

Two-dimensional grammars are successfully used in computer graphics for automated synthesis of realistic building/city models [18]. But, this is rather orthogonal, since we are interested in the inverse process, image analysis.

Another possibility to model the structure offer Markov Random Fields [26]. The individual sub-parts are in unknown (hidden) state, which is expressed by a set of possible labels assigned to the sub-parts. The sub-part labels are interconnected with compatibility edges which express allowed neighbourhood configuration, i.e. a likelihood the adjacent labels appears together. The task is to find the labeling which is the most probable given the observations.

Such problems are studied in e.g. [16, 6], and in [25, 5] on the pixel-level. Finding the MAP labeling leads to a discrete optimization task which is NP-complete in general. There are solvable sub-classes [9], which depends on the problem topology, on the number of labels, on the structure of the set of pairwise constraints, and on the properties of the pairwise potential functions (submodularity) [14]. There exists approximate algorithms which are often usable in practice, e.g. [8, 15, 13, 25].

2 Comparison of grammars and labeling

In this section we compare the elementary properties of the grammar and labeling approaches. First, we will briefly explain both of them. An equivalence of (classical) one-dimensional stochastic grammars and hidden Markov models is

discussed in [11], but a generalization for two-dimensional case has never been studied in literature, as far as we know.

2.1 Schlesinger's two-dimensional grammars

Schlesinger [21, 22] defines two-dimensional context-free grammar as a direct generalization of a context-free grammar, so the grammar G is

$$G = \langle X, K, k_0, P_h, P_v, P_r \rangle, \tag{1}$$

where X is the set of terminal symbols, K is the set of non-terminal symbols, $k_0 \in K$ is the start symbol (axiom), $P_h \subset K \times K \times K$ is the set of horizontal concatenation rules of the form $\mathrm{A} \to \mathrm{B}|\mathrm{C}$, $P_v \subset K \times K \times K$ is the set of vertical concatenation rules of the form $\mathrm{A} \to \frac{\mathrm{B}}{\mathrm{C}}$, and $P_r \subset K \times (K \cup X)$ is the set of renaming (substitution) rules of the form $\mathrm{A} \to \mathrm{x}$ or $\mathrm{A} \to \mathrm{B}$. The symbols $\mathrm{A}, \mathrm{B}, \mathrm{C} \in K$ are non-terminals and $\mathrm{x} \in X$ is a terminal symbol. All the sets here are finite.

The terminal symbols may be either image pixels [22] or larger image regions [23]. All rules may be additionally associated with a penalty for its usage which creates a stochastic extension of the grammar. Then penalties associated with rules, where terminal symbols are renamed to their non-terminals, is determined based on data, i.e. a kind of distance of the observation (current image pixel or region) to the ideal etalon representing the non-terminal symbol. Penalties of other rules reflect an allowed derivation and the preference on their usage.

Schlesinger [22, 23] proposes to use a direct generalization of the Cocke-Younger-Kasami (CYK) algorithm [12] for both parsing and best matching task. The algorithm uses a bottom-up strategy to explain the input by the grammar. In 1D case, it subsequently considers, i.e. associates with possible rules, every substring of length 1, then of length 2 and so on, until the string of the full length is assigned to the axiom symbol k_0 or not. The basic idea of the 2D generalization is that the 2D image is processed in such a way that sub-images are processed in an order given by one-dimensional sequence determined by the *inclusion* relation. It means, first it considers sub-images (1×1), the images (2×2) are considered after all their sub-images, (2×1) and (1×2), were considered, and so on until the entire image of the full size $(m \times n)$ is assigned to the axiom symbol k_0 similarly as the original algorithm. The algorithm is described in [22], page 486 and its complexity is $\mathcal{O}\big((m^2 + n^2)(m + n)\big)$.

Example In a few following paragraphs, we will show an example of two-dimensional grammar generating simple facade images, like the one in Fig. 1. This is a binary image, since the grammar we will construct will be crisp, but a generalization to a stochastic grammar is possible, as discussed above.

The grammar construction is the following. The F is an starting non-

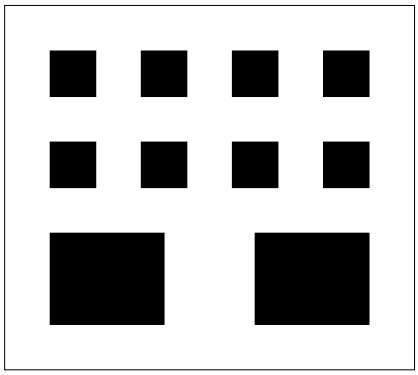


Figure 1: Simple facade image F.

terminal symbol axiom and

$$\begin{aligned} \mathrm{F} &\to \frac{\mathrm{N}}{\mathrm{F1}}, \\ \mathrm{F1} &\to \frac{\mathrm{R}}{\mathrm{F}}, \\ \mathrm{F} &\to \mathrm{N}, \end{aligned} \tag{2}$$

where F consists of blank rectangle N (representing a wall) vertically concatenated with sub-facade F1, Fig. 2(a). Then recursively, the sub-facade F1 is a row of windows R, Fig. 2(b), vertically concatenated with the facade F and to terminate we substitute F to N, as the blank wall is a facade too.

The blank rectangle is generated recursively by

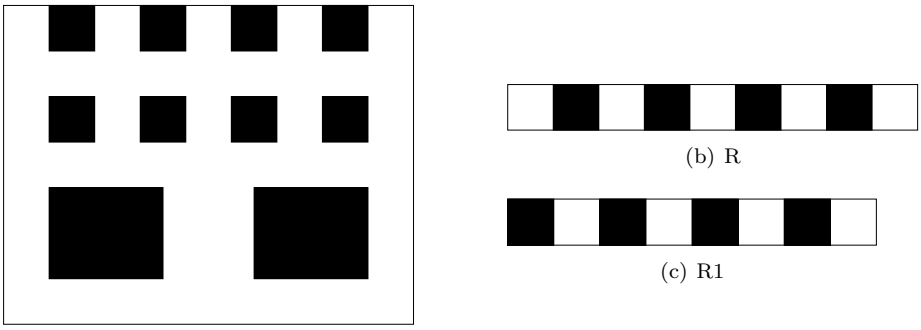
$$\begin{aligned} \mathrm{N} &\to \frac{\mathrm{N}}{\mathrm{N}}, \\ \mathrm{N} &\to \mathrm{N}|\mathrm{N}, \\ \mathrm{N} &\to \mathrm{n}, \end{aligned} \tag{3}$$

where n is a terminal symbol representing a white pixel, or in a stochastic version a likelihood that the pixel belongs to the wall (not to window).

Similarly to the facade F construction, the row of windows R is generated by

$$\begin{aligned} \mathrm{R} &\to \mathrm{N}|\mathrm{R1}, \\ \mathrm{R1} &\to \mathrm{W}|\mathrm{R}, \\ \mathrm{R} &\to \mathrm{N}, \end{aligned} \tag{4}$$

where R1 is a sub-row of windows, Fig. 2(c). Symbol W correspond to a window. We can have a simple window consisting of a single windowpane P, which is a



(a) F1

(b) R

(c) R1

Figure 2: Non-terminal symbols.

black rectangle

$$\begin{aligned} \mathrm{W} &\to \mathrm{P}, \\ \mathrm{P} &\to \frac{\mathrm{P}}{\mathrm{P}}, \\ \mathrm{P} &\to \mathrm{P}|\mathrm{P}, \\ \mathrm{P} &\to \mathrm{p}. \end{aligned} \tag{5}$$

The p is a terminal symbol representing a black pixel, or in a a stochastic version a likelihood that the pixel belongs to the windowpane.

The window W could be also more complex window consisting of several windowpanes and a grammar for this composition could also be constructed. Alternatively, the symbol W may be set as a terminal symbol and we could construct a classifier which would decide (or assign a likelihood in a stochastic version) whether a sub-image is a window or not. Such a classifier might also be based on the random field formulation.

The above example is not a solution of the facade interpretation problem in any case. It is intended to demonstrate a construction of the two-dimensional grammar.

2.2 Labeling problem

In this section we use the terminology from [9, 25]. The labeling problem is a triplet

$$L = \langle P, X, \mathbf{g} \rangle, \tag{6}$$

where $P = (T, E)$ is called the problem graph. It defines the topology of the problem. Its vertices T are called objects. An object $t' \in T$ is a neighbour of object $t \in T$ if there exist an edge $e = (t, t') \in E$. The objects may be either pixels (then a natural 4-neighbourhood topology is considered), or larger image regions with more complicated topology.

Each object $t \in T$ is associated with a set of labels X representing its possible state. The elements of $\mathbf{g}$ are node qualities $g_t(x)$ and edge qualities $g_{tt'}(x, x')$.

The node qualities $g_t(x)$, i.e. a quality of object $t \in T$ is in the state $x \in X$, are determined by the agreement of data (observation) with the object state. The qualities $g_{tt'}(x, x')$, i.e. a quality that object $t \in T$ is in the state $x \in X$ while its neighbour $t' \in T$ is in the state $x' \in X$, are determined by the prior model creating the structure.

The task of the max-sum labeling problem L is to find

$$\mathbf{x}^* = \arg \max_{\mathbf{x} \in X^{|T|}} \sum_t g_t(x_t) + \sum_{t,t'} g_{tt'}(x_t, x'_t), \tag{7}$$

which means to find a (hidden) state of all objects, i.e. to label all objects, such that this labeling is optimal with respect to (7).

The problem (7) is generally NP-complete. There are solvable subclasses, polynomial algorithm exists for cases: e.g. the binary labeling $|X| = 2$, this can be solved by max-flow/min-cut algorithm [2]; the problem graph P is a tree (has no cycles), this can be solved by dynamic programming or belief propagation algorithm [19], or edge qualities have a special form [9]. For complementary problems, there exist approximate algorithms [8, 15, 13, 25].

The grammar approach and the labeling approach have a lot in common. They both allow to model a structure, since both of them induce a language. The solution is then strongly restricted to fulfill the prescribed structure model. In case of grammars, this is due to rules P_h, P_v, P_r, and in the case of labeling, this is due to compatibility edges $g_{tt'}(x, x')$. Infinite value of $g_{tt'}(x, x') \in \{-\infty, \infty\}$ makes certain configuration of neighbouring labels to be {forbidden, obligatory} respectively, similarly to crisp (deterministic) grammars where all rules are allowed without a penalty.

When the grammar is stochastic, the rules have associated penalties for their usage in derivation. These penalties are analogical to compatibility edge qualities, where finite values represents the preferred rules or preferred neighbourhood configuration. This forms the prior model which expresses the preference on the structure of the result.

There is also a direct analogy to the penalty assigning a symbol to an observation in the lowest level in the grammar derivation and a node quality $g_t(x)$. This reflects an agreement of the data to the object state represented by the grammar symbol or the label.

We showed in [5] the max-sum labeling is equivalent to finding the most probable MAP labeling in the Bayesian formulation. From the considered analogies it seems that a similar Bayesian framework should be incorporated into grammatical derivation as well.

Obviously, the formulations based on 2D grammars and on labeling are not equivalent in general. In Fig. 3, there is an example of the image for which we cannot construct a 2D-grammar as above. But we can formulate a labeling problem for it which accepts a general set of non-overlapping axis-parallel

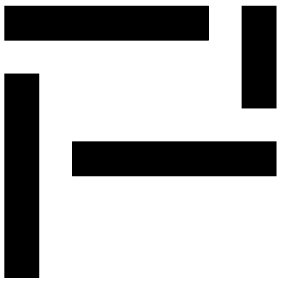


Figure 3: Example of an image which cannot be generated by the above 2D grammar, but can be solved in a labeling formulation.

rectangles, see Sec. 3. Presumambly, equivalence of two formulations holds for certain problems only.

However, there are also principal differences. Having the grammar, we can trivially generate (synthetize) images simply by applying rules from the distribution given by the penalties. Unfortunately, this is impossible in the case of the labeling formulation. Generating labeled objects subsequently from already labeled neighbours may lead to a conflict. This is due to a cyclic nature of the problem graph P.

On the other hand, when we have an input where all objects are labeled or all of them have assigned a symbol in the grammar, the task is to decide whether this is allowed. This problem is trivially solvable in case of labeling. We evaluate the energy term in (7). The value is proportional to a probability that the given labeling is a correct solution. If its value higher than $-\infty$, the labeling is allowed. In case of grammars, we have to run the parsing algorithm to decide.

The most important difference is in the complexity of the algorithms solving the best matching task (in case of grammars) and max-sum labeling task (7) (in case of labeling). The labeling problem (7) is generally NP-complete while the grammatical formulation is solvable. This is interesting, since it seems that some problems expressed in the labeling framework can be equivalently formulated using the 2D grammar. If it is really possible to construct equivalent problems, this would mean that there exists a subclass of labeling problems which is solvable by polynomial algorithms. The important open question is how to determine this class of labeling problems which are reduceable to an equivalent grammatical formulation and additionally what is the algorithm for this reduction.

3 Windowpane detection based on MAP labeling

We designed a detector of windowpanes based on the maximum aposteriori labeling. This is the paper [5], which is attached. We show there that modeling the structure is important and has a large impact on segmentation results in the comparison with a simple Potts model which is often used in segmentation to support the homogeneity of segmented regions.

The document [3] describes an image processing module for windowpane detection based on MAP labeling method. It shows how it is possible to incorporate the detector into the SCENIC system and reveals where there are points for a potential feedback loop. Closing the loop would cause a mutual influence of the higher level reasoning and the low level image processing, which should be beneficial for the entire system.

The performance of the windowpane image processing module is studied in [4].

4 Conclusion

In this paper, we showed the structure can be modeled by the two-dimensional grammars and by the labeling formulation. Both approaches are very similar in the sense they can formulate equivalent tasks for special problems. The fundamental difference is in the algorithm finding a solution of either problem. The labeling formulation is NP-complete, while the grammars possesses a polynomial algorithm. This is a theoretically interesting observation, which opens a lot of questions on the description of class of labeling problems solvable by the grammar and on the existence of a conversion algorithm. This requires further investigation.

A short-time goal is to implement a windowpane detector based on two-dimensional grammars and compare its properties to an implementation based on the labeling formulation which we developed so far. Studying the differences between the formulations in practice should bring more experience with the problem and should help finding answers to the theoretical questions mentioned above.

We believe that any results obtained as a result of this programme will have impact beyond facade interpretation.

References

[1] F. Alegre and F. Dallaert. A probabilistic approach to the semantic interpretation of building facades. In *Internatuional Workshop on Vision Techniques applied to the rehabilitation of city centers*, pages 1–12, 2004.

[2] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. *IEEE Trans. PAMI*, 26(9):1124–1137, 2004.

[3] J. Čech and R. Šára. Specification of windowpane IPM for use in SCENIC. Technical Report TN-eTRIMS-CMP-02-2006, 2006.

[4] J. Čech and R. Šára. Evaluation of the windowpane IPM. Technical Report TN-eTRIMS-CMP-02-2007, 2007.

[5] J. Čech and R. Šára. Windowpane detection based on maximum aposteriori labeling. Technical Report TR-CMP-2007-10, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, 2007.

[6] W. J. Christmas, J. Kittler, and M. Petrou. Structural matching in computer vision using probabilistic relaxation. *IEEE Trans. PAMI*, 17(8):749–764, 1995.

[7] F. Drewes, S. Ewert, R. Klempien-Hinrichs, and H.-J. Kreowski. Computing raster images from grid picture grammars. In *Proc. of Conference on Implementations and Applications of automata*, pages 113–121, 2001.

[8] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1), 2006.

[9] B. Flach and M. I. Schlesinger. A class of solvable consistent labeling problems. In *Proc. of IAPR International Workshops on Advances in Pattern Recognition*, pages 462–471, 2000.

[10] K. S. Fu. *Syntactic Pattern Recognition and Applications*. Prentice Hall, 1982.

[11] S. Geman and M. Johnson. Probability and statistics in computational linguistics, a brief review. In *Mathematical foundations of speech and language processing*, volume 138 of *IMA Volumes in Mathematics and its Applications*, pages 1–26. Springer-Verlag, New York, 2003.

[12] T. Kasami. An efficient recognition and syntax analysis algorithm for context-free languages. Technical Report AFCLR-65-758, Air Force Cambridge Research Laboratory, Bedford, Mass., USA, 1965.

[13] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *IEEE Trans. PAMI*, 28(10):1568–1583, 2006.

[14] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts. *IEEE Trans. PAMI*, 26(2):147–159, 2004.

[15] V. A. Kovalevsky and V. K. Koval. A diffusion algorithm for decreasing energy of max-sum labeling problem. Glushkov Institute of Cybernetics, Kiev, USSR, 1975. Unpublished manuscript.

[16] S. Kumar. *Models for Learning Spatial Interactions in Natural Images for Context-based Classification*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA, 2005.

[17] E. T. Lee, S.-Y. Zhu, and Chu P.-C. Generating rectangles using two-dimensional grammars with time and space complexity analyses. *International Journal of Pattern Recognition and Artificial Inteligence*, 3(4):321–332, 1989.

[18] P. Mueller, P. Wonka, S. Haegler, A. Ulmer, and L. Van Gool. Procedural modeling of buildings. In *Proc. of SIGGRAPH*, pages 614–623, 2006.

[19] Judea Pearl. *Probabilistic reasoning in intelligent systems : networks of plausible inference*. The Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann, San Francisco, 1988.

[20] D. Průša. *Two-dimensional languages*. PhD thesis, Faculty of Mathematics and Physics, Charles University, Prague, 2004.

[21] M. I. Schlesinger. *Mathematical Tools of Image Processing*. Naukova Dumka, Kiev, 1989. In Russian.

[22] M. I. Schlesinger and V. Hlaváč. *Ten Lectures on Statistical and Structural Pattern Recognition*. Kluwer Academic Publishers, Dordrecht, The Netherlands, 2002. chapter 10.

[23] M. I. Schlesinger, B. D. Savchynsky, and M. O. Anokina. Grammar approach to printed notes recognition. *Control systems and computers*, 4:30–38, 2003.

[24] R. Siromoney, K. G. Subramanian, V. R. Dare, and D. G. Thomas. Some results on picture languages. *Petern Recognition*, 32:295–304, 1999.

[25] T. Werner. A linear programming approach to max-sum problem: A review. Technical Report CTU–CMP–2005–25, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic, December 2005. http://cmp.felk.cvut.cz/cmp/software/maxsum/.

[26] J. W. Woods. Two-dimensional discrete Markovian fileds. *IEEE Trans. Information Theory*, 18:232–240, 1972.