Description of image content by means of graph grammars

Jiří Zuzaňák, Aleš Láník, Pavel Zemčík
Graph@FIT
Department of Computer Graphics and Multimedia
Faculty of Information Technology
Brno University of Technology, Brno 612 66, Czech Republic
{izuzanak,ilanik,zemcik}@fit.vutbr.cz

ABSTRACT

This paper presents an idea for partial bottom-up parse of image content by use of an attributed graph grammar, in order to achieve effective high-level representation of knowledge contained in image. Terminal nodes of the proposed grammar are formed by image objects (points, lines, and objects detected by classifiers) and areas detected in image by various image processing and segmentation methods. Based on attributes of terminal nodes, each production rule creates derived attributes for high-level representation of lower-level knowledge. Graph that is parsed by graph grammar is constructed in process of knowledge extraction by application of segmentation and image processing algorithms. Created graph is then processed by sequential application of graph grammar rules. Left side of rules is detected by isomorphism detector, and consequent rewrite is performed by rule with highest priority. A part of rewrite process is represented by processing of evaluations of vertices and edges, that describe various properties of objects and their relationships. Further in the paper we present example of attributed graph grammar application in order to describe image content.

Keywords: Graph grammar, Rewriting system, Bottom-up graph analysis, Knowledge representation

1 INTRODUCTION

Methods of region, object and edge detection forms basic approaches exploitable in process of image analysis. Mentioned methods produces set of objects and regions, with describable relations between them. For example, we can say that the detected edge *a* is parallel with edge *b* and has common end point with edges *c* and *d*.

Such information can be easily represented by structured data. In fact, in image analysis and pattern recognition, data structures are used for representation of objects topology and relations between these objects, which are not in basic properties different from mathematic graphs. Based on this fact it can be said that attributed graphs with unrestricted possibilities of vertex and edge evaluations can describe any of these structures.

Graphs created by the outlined approach are complicated structures, with dense set of vertices and edges, which cannot be easily interpreted without further modification. Principle of graph grammars based on graph rewriting systems (which will be further introduced) can be successfully applied in sense of reduction complexity of structured data represented by graph. Graph grammars thus enable creation of effective representation of knowledge contained in processed image. This paper targets extraction of knowledge contained in image (represented by graph of objects from which is image composed) by application of graph grammar rewriting system.

The paper is organized as follows. In section 2. papers and publications dealing with similar problem are mentioned. In section 3. the proposed approach to description of knowledge represented by image is introduced. Further in this section, an example of graph grammar and its application to image description is presented. Section 4. describes actual state of work on graph grammar rewriting system implementation and image processing tools. Finally section 5. concluding paper, contains the results achieved so far and proposed ideas for consequent work.

2 RELATED WORK

The idea of using graph grammars or some other formalism for recognition of image content described by structured data is not revolutionary. Many efforts based on use of attributed grammars, rule-based analyzers, and rewriting systems for image description have been made.

In [11], K. C. You and King-Sun Fu, describe object shape by attributed grammar, where its terminal symbols denote open curve segments and angle between two adjacent segments. Feng Han and Song-

Chun Zhu propose in [4] grammar composed from six rules, which enable (based on edges detected in image) description of rectangular objects and arrays of rectangular objects in image by use of top-down/bottom-up inference.

Strong image recognition and description system was introduced in [8] by Yuichi Ohta. The system is based on segmentation of input image by various methods, and sequential combined top-down and bottom-up analysis of retrieved information against model data. The introduced approach is not purely graph grammar rewriting system but more likely (as authors say) rule-based region analyser. Similar approach is proposed in [7] by Yuichi Ohta, Takeo Kanade, and Toshiyuki Sakai. Input image is segmented to basic areas by testing of intensity data, which are described by structured symbolic data. The knowledge represents set of rules by semantic nets.

A model called spatial random tree grammars is introduced in [10] by J. M. Siskind, Jr J. Sherman, I. Pollak, M. P. Harper, and C. A. Bouman. The approach is based on Bayesian methods. The authors developed algorithm for exact computation of likelihoods and maximum a posterior probabilities, and EM updates for model parameters estimation. The method is applied to the task of classifying of natural images and it is shown that use of described hierarchical structure significantly improves classifier effectivity.

In [3], an approach for image content description is proposed by T.-J. Fan, G. Medioni, and R. Nevatia, based on detection of image objects and finding relations defined between them. Introduced examples are based on description of 3D objects by their surfaces detected in input image by segmentation method based on curvature properties. Detected objects and their relations are in result represented by graph, whose vertices represent patches and edges represent geometric relations between them. Inference of objects in these graphs is then performed by reasoning of the type of connections between adjacent patches. The used graph processing is not based on graph grammars or some formal graph rewrite system.

The authors, Chungan Lin and Ramakant Nevatia, propose in [5] method for detecting of buildings and description of their 3D shape by use of geometric and projective constraints. These constraints are used to generate hypotheses for the presence of building roofs from low-level linear features, and their parallelograms. Generated hypothesis are then verified against created models.

3 PROPOSED APPROACH

In this section is on practical example demonstrated application of graph grammar rewriting system for detection of crosswalks in image. The proposed approach is based on detection of edges and, consequent creation of simple graph describing detected edges and their relations.

3.1 Image processing

Detection process is composed from the following steps: In source image, whose example is displayed in Figure 1, edges are detected by standard algorithms (Sobel, Laplace, Laplacian of Gaussians), displayed in Figure 2. Detected edges are then interpolated by lines described by geometric parameters (Hough transform). The result is displayed in Figure 3.

Figure 1: Crosswalk original

Figure 2: Crosswalk edges

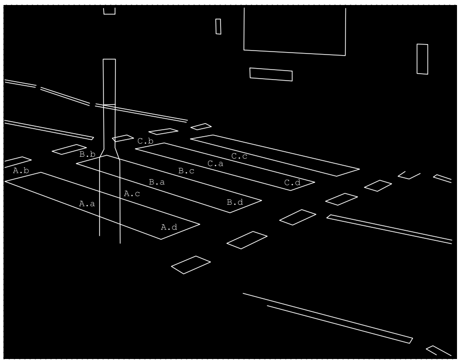

Figure 3: Crosswalk interpolated strong edges

The created set of edges is then preprocessed in order to retrieve their relations, such as common points, intersection points, orthogonality, shortest and longest distance, etc., depending on needs of defined graph grammar and its rules. Based on retrieved information is then created graph, whose vertices describe image edges, and edges of the graph describe relations between these vertices.

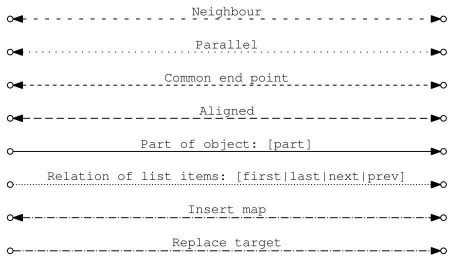

Figure 4: Edge colors legend

In image graph, are by edges expressed the following relations: Neighbour - connected objects are close each to other and there are no other objects between them. Parallel - connected edges are parallel. Common end point - connected edges have at least one common end point. Aligned - connected objects are aligned. Part - target object is part of source object. Tile list relation - description of relations between items in list object.

Graph created from input image (describing only labeled edges from Figure 3.) is displayed in Figure 5. Legend denoting color and style of edges used in example graphs is displayed in Figure 4.

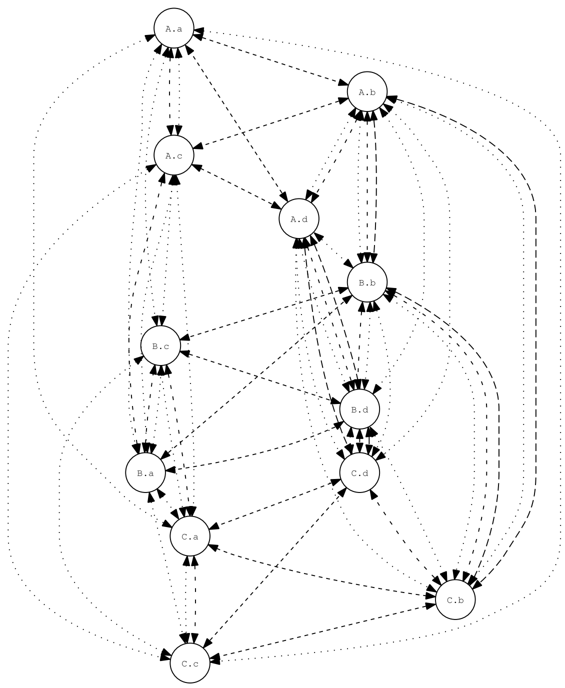

Figure 5: Graph of edges from image in Figure 3.

3.2 Example of graph grammar

Simple graph grammar designed for description of image content is composed from seven rules. Basic principle of the presented grammar is detection of tiles, and recursive merge of detected tiles to linear structure describing crosswalk. Short explanations of each grammar rule and their representation by graph follows. Number in brackets denote rule priority (rule with lower priority precede before rule with higher priority).

First rule (Figure 6.) serve for detection of tiles, based on description of tile by four edges connected in end points, where opposite edges of tile must be parallel. Second rule (Figure 7.) adds neighbour property to tiles determined by neighbour properties of edges from which are these tiles composed. Third rule (Figure 8.) mark tiles as aligned by detection of their mutual aligned edges. Rule determines alignment of tiles only when these tiles are already in neighbour relation. Fourth rule (Figure 11.) describes initialization of tile list. Tile list serves as linear list of aligned tiles where each two adjacent tiles are neighbour in processed image. Described structure serves as basic tool for description of crosswalk structure. Fifth and six rules (Figures 10. and 11.) describe insertion of new neighbour tiles to begin or end of already existing list of tiles. Last rule displayed in Figure 12. describes simple rewrite from list of tiles vertex to crosswalk vertex, which determine final detection of crosswalk. Recognize of incomplete tile lists is disabled by setup of rule priorities.

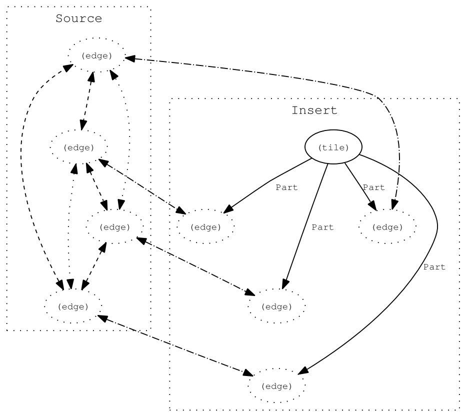

Figure 6: Rule 1: Detection of Tile (0)

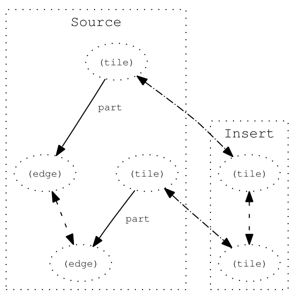

Figure 7: Rule 2: Neighbour tiles (1)

Important property of mentioned rules is their priority. Priority of rule determines order, in which the rules are applied when possibility exists to rewrite by more

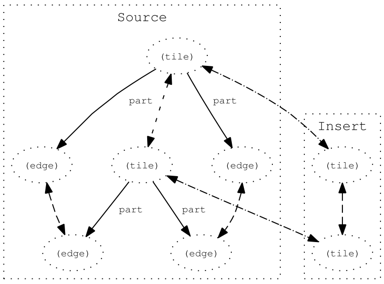

Figure 8: Rule 3: Aligned tiles (2)

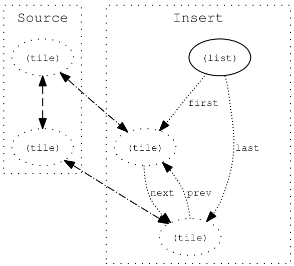

Figure 9: Rule 4: Initialization of tile list (4)

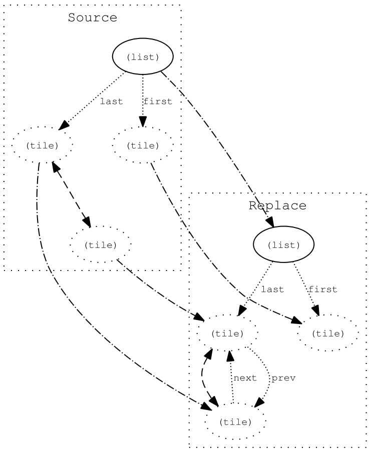

Figure 10: Rule 5: Insertion to tile list end (3)

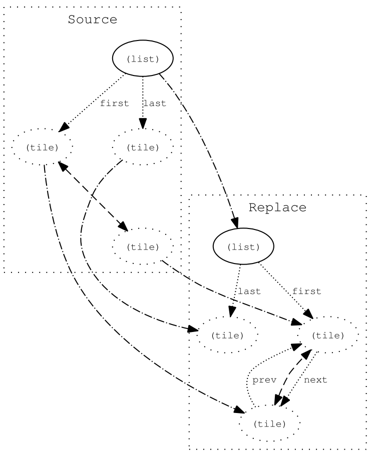

Figure 11: Rule 6: Insertion to tile list begin (3)

than one rule. Good example of rule priority importance are rules 6 and 5 vs. rule 7, ensuring that if there are some crosswalk tiles that can be inserted to tile list then rewrite by rule 7 is disabled.

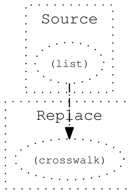

Figure 12: Rule 7: Detection of crosswalk (5)

Proposed graph grammar detects standard crosswalks composed from orthogonal tiles i.e. continental type of crosswalk displayed in Figure 13. By creation of special grammar for each type of crosswalk, we are able to generalize (by proper rules) different crosswalks to one general class.

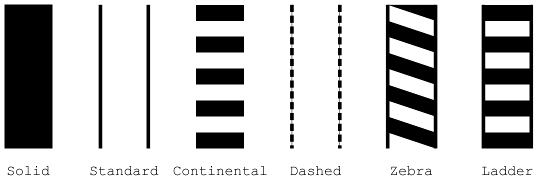

Figure 13: Differend types of crosswalks

4 ACTUAL STATE OF WORK

Graph grammar processing tool is in state of development. This tool consist from several components, which are: graph representation, graph isomorphism detector, graph rewriting system, and graph grammar representation.

Component for representation and manipulation with graphs is complete. This tool enables loading and saving of graphs with arbitrary structure and standard vertex and edge evaluations (standard data types, strings, arbitrary data types controlled through dynamic libraries).

Abstract data type set is implemented as red-black trees, enabling fastest possible searching for processed vertices and edges. Graph representation enables execution of various graph algorithms as are: detection of spanning tree, search for shortest path, detection of isomorphism between two graphs, decomposition to graph components, and more graph algorithms.

Second component of graph grammar processor is represented by graph isomorphism detector. Created detector of graph isomorphisms is specially designed for application in graph grammar processing system. Detector is based on automatized creation of so called graph parser (graph automata) which is intended for searching of set of subgraphs isomorphisms in one host graph. Isomorphism detector use described representation of graphs, which enables dynamic modification of graphs and thus is proper for graph rewriting.

Graph rewriting system which forms third part of graph grammar processing tool is in phase of design and partially in phase of implementation. The rewriting system so far enables loading of graph grammars (set of rules with defined priorities, where each rule is composed from two graphs, and transformation relations between these two graphs), creation of graph parser described previously (based on left side of grammar rules), detection of left rules and basic rewrite, so far without more complicated processing of vertex and edge evaluations.

Last part of graph grammar processor is formed by graph grammar representation, and it is in phase of design and implementation similarly as previous component. This component is closely bind to graph rewriting system.

In current state, graph grammar processing tool is capable of generating graphs based on set of given graph grammar rules, and parsing of these generated graphs by grammar reverse to generating grammar. Experimental grammars for processing graphs of program flow has been created and tested. From the results it can be implied that designed graph rewriting tool based on defined graph grammar is capable to significantly optimize graph of program flow.

5 CONCLUSION

In this paper was proposed an idea for description of image content by graph grammar rewriting system, along with example of graph grammar of such system applied for description of simple image content.

Content of an image is in the first steps of process transformed to graph, which describes objects of image by graph vertices, and relations between these objects by graph edges. Further information describing image content is represented by evaluation of graph vertices and edges. The created graph is then interpreted by graph grammar processor, where individual rewriting steps determine structure of data represented by graph. Applied graph grammar processor is based on rewrite system which is in phase of design. Rewrite system is based on graph isomorphism detector mentioned in Section 4.

Graph grammar processing tools has been so far tested on grammars generating graphs of program flow and reverse grammars parsing generated graphs. From the achieved results it can be seen that graph rewriting system is capable of described properties (parsing and interpreting complicated graph structures).

Further work will concentrate on finishing design and implementation of grammar representation and graph rewriting system. Next steps in design of graph isomorphism detector and consequently graph rewriting tool will be:

- Complex evaluation of vertices and edges, and mainly their processing and configurable modification by process of graph rewrite
- Allow use of stochastic graph grammar rules (designed if possible without need of change of graph isomorphism detector)
- Integration of graph rewriting system to graph grammar description, and completion of system allowing complex graph analyses based on given graph grammar
- Testing of designed system and evaluation on data extracted from real world images

Essential step of whole process will be testing of designed system on real world data, retrieved from videos and images provided by European project WeKnowIt, of which is this work part.

ACKNOWLEDGEMENTS

This work is supported by the European Commission under contract FP7-215453 - WeKnowIt.

This work was (also) supported by the project "Security Oriented Research in Information Technology" by Ministry of Eduction, Youth and Sports of Czech Republic no. MSM0021630528.

REFERENCES

[1] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. An improved algorithm for matching large graphs. In *In: 3rd IAPR-TC15 Workshop on Graph-based Representations in Pattern Recognition, Cuen*, pages 149–159, 2001.

[2] D. G. Corneil and C. C. Gotlieb. An efficient algorithm for graph isomorphism. *J. ACM*, 17(1):51–64, 1970.

[3] T.-J. Fan, G. Medioni, and R. Nevatia. Recognizing 3-d objects using surface descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 11(11):1140–1157, 1989.

[4] Feng Han and Song-Chun Zhu. Bottom-up/top-down image parsing with attribute grammar. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(1):59–73, 2009.

[5] Chungan Lin and Ramakant Nevatia. Building detection and description from a single intensity image, 1998.

[6] Brendan D. McKay. Practical graph isomorphism. In *Congressus Numerantium*, pages 45–87, 1981.

[7] Y. Ohta, Takeo Kanade, and T. Sakai. An analysis system for scenes containing objects with substructures. In *Proceedings of the Fourth International Joint Conference on Pattern Recognitions*, pages 752–754, 1978.

[8] Yuichi Ohta. *Knowledge-based interpretation of outdoor natural color scenes*. Pitman Publishing, Inc., Marshfield, MA, USA, 1985.

[9] A. Schfürr. Programmed graph replacement systems. pages 479–546, 1997.

[10] J. M. Siskind, Jr J. Sherman, I. Pollak, M. P. Harper, and C. A. Bouman. Spatial random tree grammars for modeling hierarchal structure in images with regions of arbitrary shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1504–1519, 2007.

[11] K. C. You and King-Sun Fu. A syntactic approach to shape recognition using attributed grammars. *Institute of Electrical and Electronic Engineers. Transactions on Systems, Man, and Cybernetics*, SMC-9(6):334–345, 1979.