MVA '96 IAPR Workshop on Machine Vision Applications, November. 12-14, 1996, Tokyo, Japan

Recognition of Composite Elements in Bar Graphs

Naoko YOKOKURA and Toyohide WATANABE *
Department of Information Engineering,
Graduate School of Engineering, Nagoya University

Abstract

The issue of graph recognition has been not always investigated until today though the subjects on the document image understanding are very interest and have proposed many methods/systems. This is partly because the graph reading is not yet sufficiently recognized in the practical applications, and partly because the graph recognition is more difficult than the ordinary document analysis/recognition.

In this paper, we address an analysis problem of graph structures: especially, we concentrate on the recognition method of composite elements on bar graphs. The knowledge about layout structures is called the layout knowledge. We propose the representation form of this layout knowledge clearly, and make it explicit that our approach based on this layout knowledge can analyze/recognize the graph structures effectively.

1 Introduction

The graphs are commonly used to represent the measurement among transitional states or/and the difference among individual items graphically. The issue on graph recognition is an objective to collect such information automatically from graph images. This issue has been not always investigated sufficiently until today though the subjects on the document image understanding are very interest and have proposed many methods/systems. This is partly because the importance of graph reading is not yet sufficiently recognized in the practical applications, and partly because the graph recognition is more difficult than the ordinary document analysis/recognition. Even if the documents contained some pictures/figures, in the document image understanding these components except character/symbol strings have been excluded in many cases from the main recognition objects [1, 2]. A few researchers treated graphs: for example, M.Koga et.al[3] and R.P.Futrelle[4], but they did not focus on the recognition of data represented by graph.

In this paper, we address an analysis problem of graph structures: especially, we concentrate on the recognition method of composite elements on bar graphs. M.H.Lee et al. attempted to recognize graph composite elements with only the image processing technique [5]. On the other hand, the main idea in our approach is based on the facts that the bar graph has its own layout structure, and that the layout structure can be alternatively specified under connective/neighboring relationships among composite elements.

In the following sections, we propose a method for recognizing composite elements on bar graphs, and show the recognition result which was experimented to evaluate this method. First, we address characteristics of the bar graph and its own layout structure in Section 2. In Section 3 we propose our recognition method of bar graph, and in Section 4, the experimental result is reported through some experiments. Finally, in Section 5 we discuss some problems and our future work in our method.

2 Structure of Bar Graphs

Our graph recognition process consists of two different processing phases as shown in Figure 1: the first is to extract composite elements, and the second is to classify these elements. This extraction phase uses constraint information for the geometrical features in graph structures, and the classification phase refers to the knowledge of logical relationships among composite elements and physical properties of individual elements in graph representations. In this section, these information and knowledge of bar graph are described.

Figure 2 shows an example of bar graph. In this figure, we can observe that the bar graph is composed mainly of graph area and index area, and they are separated by vertical and horizontal axes. The graph area is defined as a region which includes graphical picture segments, and the index area is defined as the region for accommodating explana-

*Address: Furo-cho, Chikusa-ku, Nagoya 464-01, JAPAN.
E-mail:
{yokokura,watanabe}@watanabe.nuie.nagoya-u.ac.jp

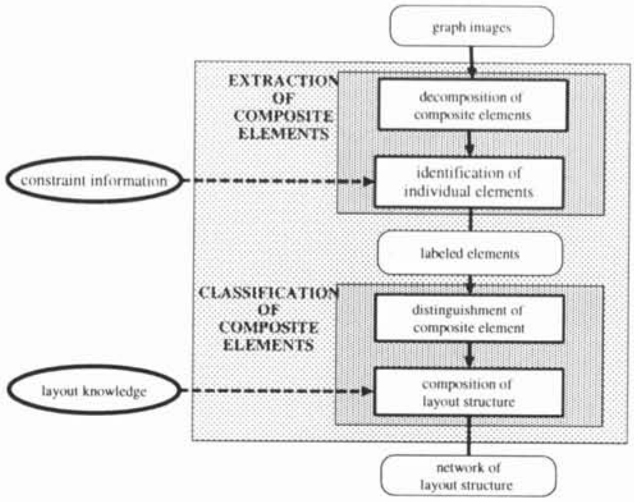

Figure 1: Processing flow

tion and indication data. Namely, in the physical view when the vertical axis is depicted in the left-side and the horizontal axis is so in the lower-side in Figure 2, the right-upper region is the graph area, and another region is the index area. Of course, some graphs may have not always these two axes. However, we can separate these two areas in many cases on the basis of graphical and non-graphical features of individual areas. The bar graph is composed of various composite graph primitives: for example, some of them are shown in Figure 3. Graph primitives can be divided into two types: graphical and string primitives.

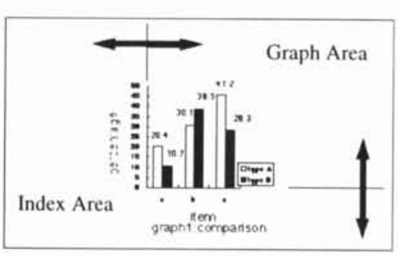

Figure 2: An example of bar graph

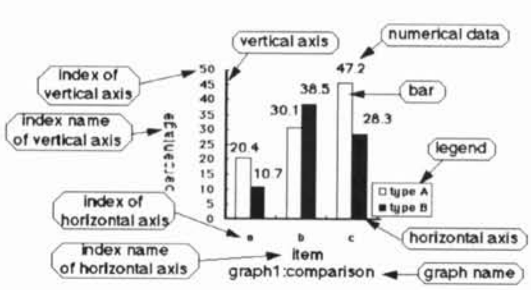

Figure 3: Composite graph elements

As described in Section 1, the bar graph has its own layout structure, and the layout structure can be alternatively specified under connective/neighboring relationships among composite elements. This knowledge of layout structure is represented as a network structure, as illustrated partly in Figure 4: we call this a layout network. In this layout network, nodes indicate individual elements in the bar graph and edges indicate the connective/neighboring relationships among nodes. The underlying nodes in the layout network are horizontal and vertical axes, and the network spreads toward the verge of the graph. Some composite elements also have logical layout relationships: for example, one numerical data always has a relationship corresponding to one bar. This relationship is also represented as an edge in the layout network(Figure 5).

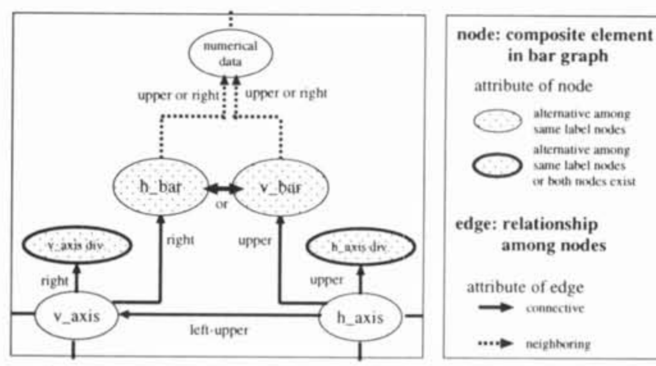

Figure 4: A layout network(part)

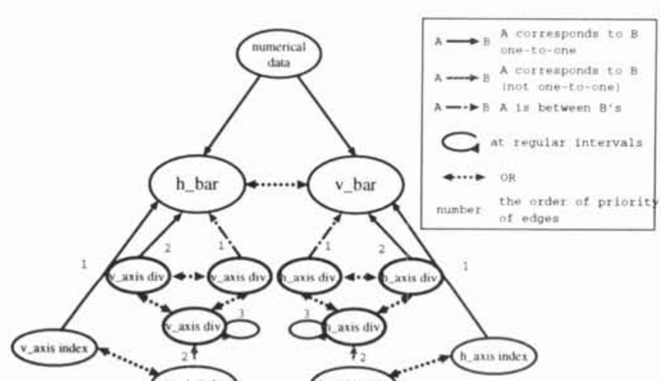

Figure 5: Logical relationships(part)

3 Recognition of Composite Elements

In this section, two phases in recognizing composite elements are explained. The first phase is done by a bottom-up manner with respect to an application of image processing techniques, while the second phase does so by a top-down manner on the basis of interpretation of layout network.

3.1 Extraction of Composite Elements

In this phase, various elements in graph drawings are separately decomposed. Composite elements in bar graph are divided into two types: graphical primitives and string primitives. Graphical primitives in bar graph, like bars and divisions, are always connected to axes, and exist in the graph area. On the other hand, string primitives, like index and numerical data, are not connected to axes, and can exist in both of two areas. So, if graph area and index area are decomposed and axes are decided, primitives in bar graph can be easily decomposed to graphical and string ones. First, vertical axis and horizontal axis are extracted as the most-left peak in the vertical histogram and the most-lower peak in horizontal histogram(Figure 6). Then, graph area and index area are decomposed with regard to these two extracted axes(Figure 7). Next, other elements are divided into graphical primitives which connect to axes and string primitives which do not connect to axes. In graphical primitives, decomposition processing using the histogram analysis techniques is repeatedly applied to stepwise extracted elements until the composite elements cannot be further divided into other elements. On the other hand, string primitives are extracted with respect to the previously identified string primitives.

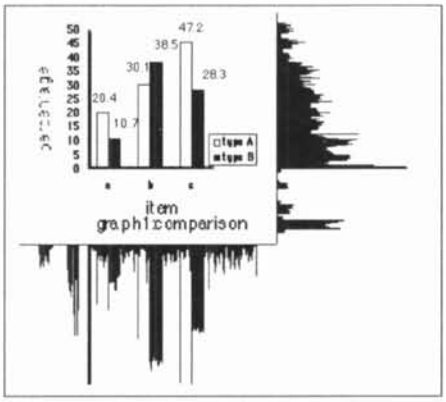

Figure 6: Histograms

3.2 Classification of Composite Elements

In this phase, the previously extracted elements are distinguished, using a layout network which represents the layout knowledge of bar graph structure. Namely, this classification procedure checks the properties of elements extracted in the first step, compares the extracted elements with nodes of layout network interpretatively, classifies the elements into appropriate graph primitives individually, and composes a layout structure as the recognized result, that is named a result network. The processing steps in this phase is explained in detail as follows:

1. Search each node in network by the depth ordering. The depth of node is defined as the number of edges along the longest path from the underlying nodes, which are axes;

2. Extract composite elements which correspond to the parent of searched node from the result network. If the element does not exist, then the elements which correspond to the grand-parent of searched node are extracted.

3. Select only elements which satisfy the constraint condition from composite elements extracted in Step2. Also, extract edges connected to searched node (Figure 8), and add them to the result network.

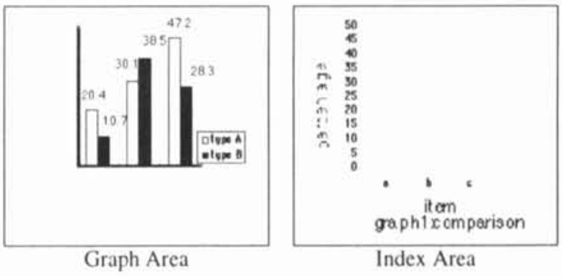

Figure 7: Graph area and index area

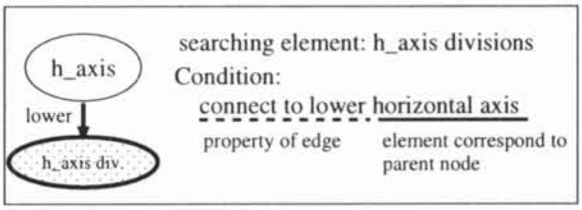

Figure 8: Condition

4 Experimental Result

In this section, we show the experimental result, extracted through our method(Table 1). Graph images are digitalized in 400 dpi. 10 sheets of bar graph images are recognized. Figure 9 shows an example of graph images. Figure 10 shows a result of extracted composite elements, and Figure 11 shows the result network whose nodes are classified and whose relationships among them are assigned.

Table 1: Experimental results

	(A)	(B)	(C)	ratio (C/A) (%)	ratio(C/B) (%)
extraction phase	371	351	327	88	93
classification phase	371	336	293	79	87

(A) ··· No. of primitives
(B) ··· No. of decomposed/classified primitives
(C) ··· No. of correctly decomposed/classified primitives

In the extraction phase, there were errors that some elements are looked upon as the same element because these different elements were connected mutually. Also, in some strings the strings were extracted separately because the characters in the strings were detached.

In the classification phase, there were some errors that are same composite elements are classified as different elements because the string is on over two lines. Figure 11 shows one of second line in horizontal indexes is recognized as names of horizontal indexes.

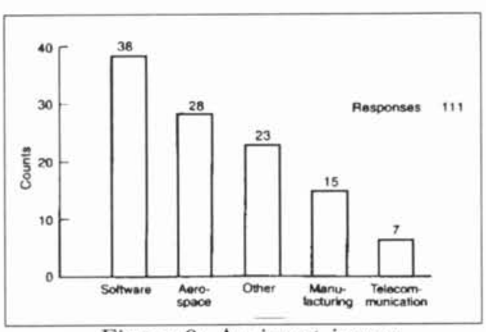

Figure 9: An input image

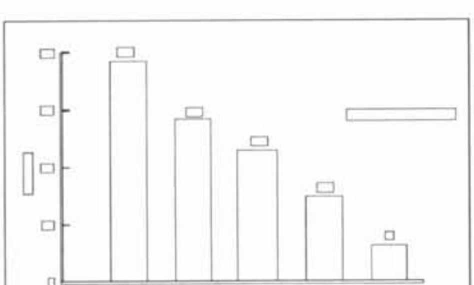
Figure 10: Extracted composite elements

5 Conclusion

In this paper, we addressed a recognition method of composite elements on bar graphs. As a result, our method was adaptable to recognize the compos-

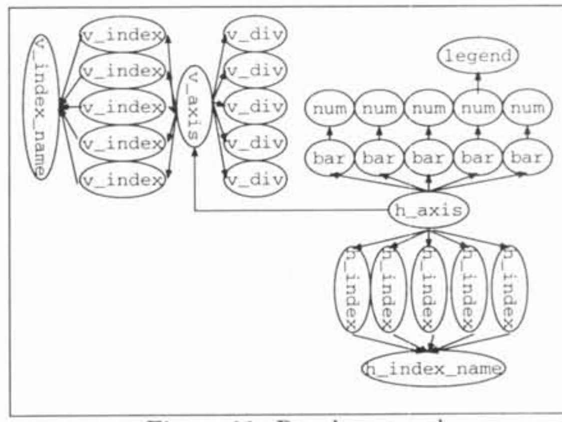

Figure 11: Result network

ite elements on bar graphs. However, there are some problems. First, it is necessary to reduce the decomposition failures, because they disturb from identifying composite elements sufficiently. Next, it is important to make the classification process flexible: for example, it should be applicable even in cases that string primitives are over two axes.

Acknowledgments

We are very grateful to Prof. T. FUKUMURA of Chukyo University, and Prof. Y. INAGAKI and Prof. J. TORIWAKI of Nagoya University for their perspective remarks, and also wish to thank Dr. Y. SAGAWA, Mr. K. ASAKURA and our research members for their many discussions and cooperations.

References

[1] G. Nagy, S. Seth and M. Viswanathan: "A Prototype Document Image Analysis System for Technical Journals", *IEEE Computer*, Vol. 25, No. 7, pp. 10–22 (1992).

[2] M. Doi, M. Fukui, K. Yamaguchi, Y. Takebayashi and I. Iwai: "Development of Document Architecture Extraction", *Trans.on IEICE*, Vol. J76-D-II, No. 9, pp. 2042–2052 (1993) [in Japanese].

[3] M. Koga, T. Murakami, Y. Shima and H. Fujisawa: "An Extraction Method of Search Indexes for Graph Image Retrieval", *Proc. of MVA '92*, pp. 163–166 (1992).

[4] R. P. Futrelle et al.: "Understanding Diagrams in Technical Documents", *IEEE Computer*, Vol. 25, No. 7, pp. 75–78 (1992).

[5] M. H. Lee, N. Babaguchi and T. Kitahashi: "Symbolization and Presentation of Graph Images for Intelligent Communication of Document Images", *Proc. ACCV'95*, Vol. 3, pp. 680–684 (1995).