

# Associating Text and Graphics for Scientific Chart Understanding

Weihua Huang, Chew Lim Tan and Wee Kheng Leow  
School of Computing, National University of Singapore  
{huangwh, tancl, [leowwk](mailto:leowwk@comp.nus.edu.sg)}@comp.nus.edu.sg

## Abstract

*This paper presents our recent work that aims at associating the recognition results of textual and graphical information contained in the scientific chart images. Text components are first located in the input image and then recognized using OCR. On the other hand, the graphical objects are segmented and form high level symbols. Both logical and semantic correspondence between text and graphical symbols are identified. The association of text and graphics allows us to capture the semantic meaning carried by scientific chart images in a more complete way. The result of scientific chart image understanding is presented using XML documents.*

## 1. Introduction

Research activities in the document image analysis field can be mainly classified into two categories: text processing that deals with the text components of a document image, and graphics processing that deals with the lines and symbol components that make up diagrams, maps and engineering drawings etc. Traditional research works in these two categories are usually independent of each other. OCR systems, as a good example of text processing application, tend to recognize the text in the document images without touching the graphics contained on the same page. On the other hand, most graphics recognition systems concentrate on the graphics segmentation, symbol construction and classification etc. without making use of the textual information contained in the images.

We believe that to achieve full document understanding, the textual information and graphical information should be combined to capture more complete semantic meaning of the document image and to enhance the performance of the recognition system. In fact, K. Tombre et al already pointed out that associating textual information with the graphics is an important step in the semantic labeling of graphics [1].

The work reported here does not aim to provide a general solution to all document image understanding problems. It is an attempt to combine the textual information with graphical information from both logical level and semantic level. We choose to focus on the diagrams that frequently present in the scientific papers and web pages. The main reason for choosing this kind of document is that the context of the graphical information is relatively easier to model and the role of the text components can be identified in a more standard way. As a beginning, we start with the commonly used scientific charts such as bar-chart, pie chart etc. The result of our work can be applied on content-based web image retrieval and auto-conversion of raster image into structured document etc.

The remaining sections of this paper are arranged in this way: section 2 surveys some previous works related to our study. Section 3 introduces the detailed design and implementation issues, focusing on textual information retrieval and association of text and graphics. Section 4 presents experimental results. Section 5 gives a conclusion to this paper.

## 2. Related works

In the past, although there were works dealing with various entities in graphics including text, very few of them actually attempted to extract textual information and associate it with the graphics. Kasturi et al developed a system to interpret various components in a line drawing, including text strings [2]. However the text components were not recognized. Joseph and Pridmore presented an experimental system for mechanical engineering drawing interpretation [3]. Like Kasturi's, the system had no provision for text recognition. B. Lamiroy et al conducted experiments to analyze the role of the text components in cutaway diagrams [4, 5]. The result reported was limited to identifying the relationship between drawings, their indices and the legends. However in their work the graphics layer is completely discarded so the text and graphics association was still preliminary.

In recent years, works on chart recognition have been continuously reported. Futrelle et al presented a diagram understanding system based on graphics constraint grammars to recognize x-y data graphs and gene diagrams [6], with the major assumption of proper graphics segmentation which is difficult to achieve. Yokokura et al proposed a schema-based framework to graphically describe the layout relationship information of the bar charts [7] based on vertical and horizontal projections. The bar chart styles that can be recognized are restricted due to the simplicity of the method. Zhou et al applied Hough-based techniques to achieve bar chart detection and segmentation [8]. Later they also proposed a learning-based chart recognition paradigm using Hidden Markov Models [9]. In both cases the main focus was only on the low-level features in the input image, without touching the semantic meanings in the chart. In all works mentioned above, textual information was not retrieved and thus was never combined with graphical information.

### 3. Design and implementation issues

The system proposed here handles both textual and graphical information. There are four main modules: text/graphics separation module, text recognition module, graphics recognition module and text/graphics association module. Figure 1 shows the flow of control in the system. The basic scheme here is to recognize the text and graphics in the input image separately, and then combine the two kinds of information to achieve full understanding of the input image. Text/graphics association is performed in the chart understanding module. The combined recognition result is captured using XML description for future interpretation. The details of the graphics recognition module can be found in our previous paper [10]. The outcome of the module includes major chart components and information about the chart type. In this paper, we will focus on the design and implementation of the text retrieval module and the text/graphics association module.

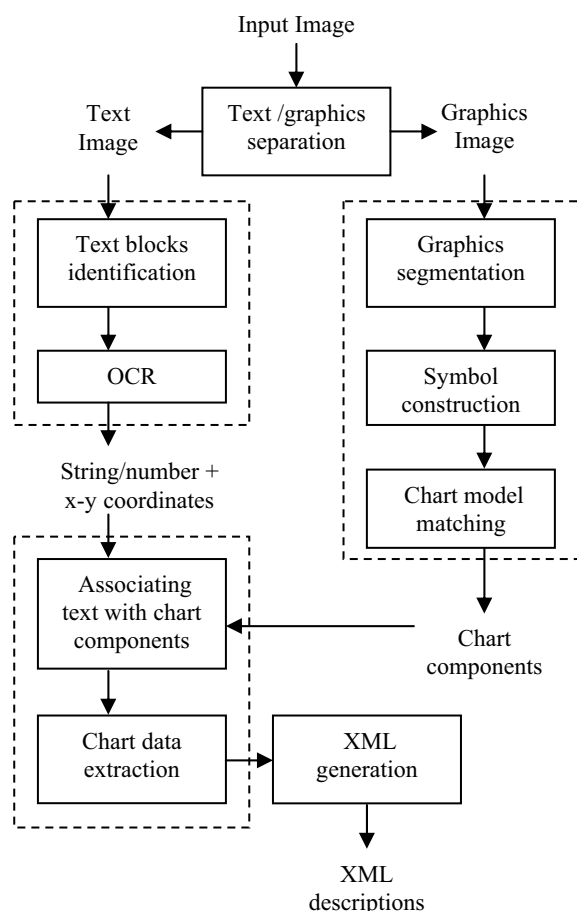


Figure 1. Overview of the proposed system

#### 3.1. Retrieving textual information

Text/graphics separation is done through connected component analysis. A series of filters are applied to all the components in the image to classify them into text and graphics. The text and graphics classified are stored as two separate images that will be processed by two different modules in the system later.

Text groups are formed through the calculation of “gravity”  $G$  based on the Newton’s formula [11].

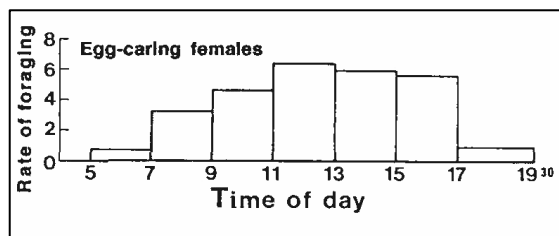
$$G = C_1 \cdot C_2 / r^2$$

where  $C_1$  and  $C_2$  are the sizes of the two components and  $r$  is the distance between the centers of the components. If  $G$  is greater than a threshold value, then the components belong to the same text group. The advantage of this method is that text groups in different orientations can all be handled efficiently.

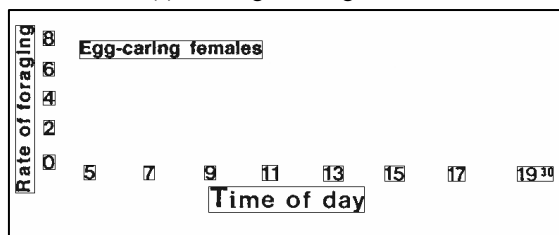
OCR is then applied to each text group to recognize its content. The result of text recognition is further classified as a string or a number (integer or floating point number). The value of the number is obtained through parsers. Text information obtained will be recorded as: string/number + x-y coordinates.

During textual information retrieval, there are mainly two sources of error: the error in text/graphics separation and the error in the character recognition process. Text/graphics separation errors come from text touching graphics which connected component analysis fails to handle. There are techniques proposed

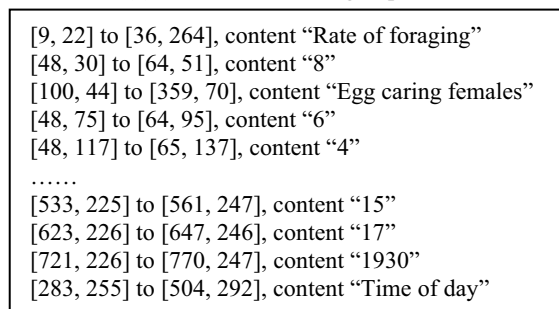
that deals with separating touching text and graphics, and they can be applied to reduce the errors [12]. OCR error rate heavily depends on the quality of the document. Since the OCR implementation is not the main focus of our project, we choose to manually correct the errors. To guarantee the performance of the whole system, the errors here should be minimized. Figure 2 gives an example of textual information retrieval.



(a) The original image



(b) Text extracted and grouped



(c) Text location and OCR result

**Figure 2. Obtaining the textual information**

### 3.2. Text-graphics association on logical level

There are two levels of text-graphics association: logical level and semantic level. On the logical level, we are interested in "what is the role of a text block in the chart?" In the case of scientific chart, logical association between text and graphics is obtained by examining the spatial relationship between the text blocks and the chart components.

In a chart image, text plays a number of logical roles. We summarize them into six main categories:

- *Caption*, including the title of the chart and sometimes additional descriptions.
- *Axis\_title*, the name of an axis.
- *Axis\_label*, defining the scope/range of an axis as well as the data type (string or number).
- *Legend*, distinguishing multiple data series. There is a small graphical object in front of the text in a legend.
- *Data\_value*. Sometimes the value of the data is directly shown inside or near the data components. *Data\_value* is required to be a number.
- *Others*. Any other supplementary description of the chart content.

If we treat these logical roles as a set of tags, then finding the logical association becomes a task of tagging the text blocks. Let's denote the tagging process as  $\phi(C)$ , where  $C$  is the input chart image. We define  $P(R, A, T)$  as a joint probability distribution associated with each chart type, and each joint probability  $P(r_i, t_i, a_k)$  shows the probability that the  $i_{th}$  text block of type  $t_i$  has logical role  $r_i$  in the  $k_{th}$  area  $a_k$  in  $C$ . The type  $t_i$  is either string or number. The available set of logic roles  $R$  depends on the type of the chart (which is already determined through graphics recognition module). To obtain  $P(R, A, T)$ , a set of training images for each chart type are used to calculate individual probability values. In the understanding phase, given  $A$  and  $T$ , our task is to find out the best way to tag the text such that:

$$\phi(C) = \arg \max_R P(R, A, T)$$

Now the remaining question is how to divide the chart image into a set of areas  $A$ . Firstly the area inside the data components and the area outside the data components are separated. For a chart type with x-y axes, the area outside the data components is further divided based on the plot area. Then we have:

- 1) Area above the plot area;
- 2) Area below the plot area;
- 3) Area on the left hand side of the plot area;
- 4) Area on the right hand side of the plot area;
- 5) Area within the plot area.

For chart types without x-y axes, such as a pie chart, the division is done similarly, but based on the position of the data components themselves instead of the plot area.

On the logical level, graphical information provides evidence about the chart type that further determines the available text roles. It also affects the detailed area division for text block tagging. Without graphical

information, the logical role of a text block can hardly be determined.

### 3.3. Text-graphics association on semantic level

The next step is to achieve semantic association between text and graphics which is more challenging. The goal here is to extract the absolute data values based on the chart components obtained from the graphics recognition module together with the text blocks whose logical role is determined. The steps involved are:

- (a) To estimate the absolute data value. Without losing generality, let's assume the x axis represents the index of the data component and y axis determines the value of the data component. Let  $l_m$  and  $l_n$  be two neighboring labels along the y axis, then define *step\_value* as |value of  $l_m$  - value of  $l_n$ | and *step\_dist* as *Distance*(center of  $l_m$ , center of  $l_n$ ). Absolute value of data component  $c_i$  is calculated as: *Distance*( $Ac_i$ , y-axis) \* *step\_value* / *step\_dist*, where  $Ac_i$  is the attribute of  $c_i$  that correlates with the data value, such as the height of a bar in the bar chart.
- (b) If *data\_value* associated with the data component is available, then there are two cases:
  - i. If *data\_value* agrees with the estimated data value obtained from part (a), then *data\_value* is picked.
  - ii. If the difference between *data\_value* and the estimated data value is too large, then *data\_value* is treated as a false value (which is caused by an error on the logical level) and the estimated data value is picked.

Note that step (a) shown above only works for chart with x-y axes. At the moment, pie chart is the only chart type without x-y axes. In a pie chart, the original data values are relative values, thus step (a) is not necessary. The detection of chart type is done in the graphics recognition module [10].

The association between text and graphics on the semantic level allows us to extract absolute data values, which is not achievable without textual information.

### 3.4. Generating XML description

Based on the text recognized and the data values extracted, we can generate an XML description for the chart image. The document `<chart>` contains the following parts:

- `<caption>`, the title of the chart and the description of the chart.

- `<x_axis>` and `<y_axis>`. The existence of x-y axes depends on the type of the chart. If they exist, `<x_axis_title>` and `<y_axis_title>` give titles to the axes and `<labels>` contains a set of `<label>` that defines the scope/range of each axis.
- `<data_set>`, the data values extracted from the chart image. Each data entry has an `<index>` which is automatically generated, and a `<value>`.
- `<legend>`, `<mark>` and `<description>`. Generally speaking, a chart with single data series does not need legends. `<mark>` is an attribute such as a small sample of color or texture etc. to represent a unique data series. `<description>` contains the text string that describes one legend.

```
<?xml version="1.0"?><!--chart_recognized.xml-->
<?xml-stylesheet type="text/xsl" href="chartXML.xsl"?>
<!DOCTYPE chart [
<!ELEMENT chart (caption, x_axis, y_axis, data_set) >
<!ELEMENT caption ( #PCDATA ) >
<!ELEMENT x_axis ( x_axis_title, labels ) >
<!ELEMENT y_axis ( y_axis_title, labels ) >
<!ELEMENT labels ( label+ ) >
<!ELEMENT label ( #PCDATA ) >
<!ELEMENT x_axis_title ( #PCDATA ) >
<!ELEMENT y_axis_title ( #PCDATA ) >
<!ELEMENT data_set ( data+ ) >
<!ELEMENT data ( index, value ) >
<!ELEMENT index ( #PCDATA ) >
<!ELEMENT value ( #PCDATA ) >
]>
<chart>
<x_axis><x_axis_title>Time of day</x_axis_title>
<labels><label>5</label>
.....
<label>1930</label></labels></x_axis>
<y_axis><y_axis_title>Rate of foraging</y_axis_title>
<labels><label>0</label>
.....
<label>8</label></labels></y_axis>
<data_set>
<data><index>1</index><value>0.819</value></data>
<data><index>2</index><value>3.273</value></data>
.....
<data><index>7</index><value>0.864</value></data>
</data_set></chart>
```

Figure 3. Example of an XML description

To transform an XML description file into something that can be viewed using the browser, an XML style sheet is needed. The current choice is to parse the XML descriptions into an HTML table. Due to space limitation, the XML style sheet is not discussed in detail here. Figure 3 shows the XML description of the chart image in Figure 2(a).

## 4. Experimental results

For testing purpose, we collected 53 chart images that were generated from scanner or downloaded from the internet. As we mentioned, OCR errors during text recognition are manually corrected. We count the number of text blocks correctly identified for each category and the result is presented in table 1. For graphics recognition, we count the number of symbols recognized by the system and calculate the recall and precision, which are shown in table 2. For chart understanding, it's hard to evaluate the data values extracted by the system mainly due to the lack of ground-truth for the testing data (except for those with known data values).

**Table 1. Result of textual information retrieval**

Category	Total Number	Extracted	Accuracy (%)
Caption	32	28	87.5
Axis title	39	30	76.92
Axis Label	641	616	96.1
Legend	46	30	65.22
Data value	37	30	81.08
Others	25	14	56
Total	820	748	91.22

**Table 2. Result of graphics recognition**

No. of components in the chart images	296
No. of symbols recognized	263
No. of symbols correct	245
Recall (%)	82.77
Precision (%)	93.15

## 5. Conclusion

This paper presents our work about associating textual and graphical information for the understanding of scientific chart images. Textual components are extracted from the input image and are recognized through OCR. Graphics are segmented and high-level chart components are obtained through a graphics recognition module. Understanding of the chart image is achieved by associating the textual and graphical information on both logical and semantic level. The overall recognition result is presented using XML document. In the future, more effort should be put to minimize the errors occurred in individual module in the system and to extend the system to handle more complex diagrams.

Acknowledgement: this research is supported in part by A\*STAR under grant R252-000-206-305 and NUS URC under grant R252-000-202-112.

## 6. References

- [1] K. Tombre and B. Lamiroy, "Graphics recognition - From re-engineering to retrieval", *Proc. of 7th ICDAR*, Edinburgh (Scotland, UK), pp. 148--155, August 2003.
- [2] R. Kasturi, S. T. Bow, W. El-Masri, Y. Shah, J. R. Gattiker and U. B. Mokate, "A System for Interpretation of Line Drawings", *IEEE Trans. PAMI*, vol. 2, no. 10, pp. 978-992, Oct. 1990.
- [3] S. H. Joseph and T. P. Pridmore, "Knowledge-Directed Interpretation of Mechanical Engineering Drawings", *IEEE Trans. PAMI*, vol. 14, no. 9, pp. 928-940, Sept. 1992.
- [4] B. Lamiroy, L. Najman, R. Ehrard, C. Louis, F. Quelain, N. Rouyer and N. Zeghache, "Scan-to-XML for Vector graphics: an Experimental Setup for Intelligent Browsable Document Generation", *Proc. 4th IAPR International Workshop on Graphics Recognition*, Kingston, Ontario (Canada), pp. 312-325, Sept. 2001.
- [5] E. Valveny and B. Lamiroy, "Scan-to-XML: Automatic Generation of Browsable Technical Documents", *Proc. of 16th ICPR*, Quebec (Canada), pp. 188-191, Aug. 2002.
- [6] R.P. Futrelle *et al.*, "Understanding diagrams in technical documents", *IEEE Computer*, Vol.25, NO.7, pp. 75-78, 1992.
- [7] N. Yokokura and T. Watanabe, "Layout-Based Approach for extracting constructive elements of bar-charts", *Graphics recognition: algorithms and systems, GREC'97*, pp. 163-174.
- [8] Y. Zhou and C L Tan, "Hough-based Model for Recognizing Bar Charts in Document Images", *SPIE conference on Document image and retrieval*, 2001.
- [9] Y. P. Zhou and C. L. Tan, "Learning-based scientific chart recognition", *4th IAPR International Workshop on Graphics Recognition, GREC2001*, pp. 482-492, 2001.
- [10] W. H. Huang, C. L. Tan and W. K. Leow, "Model based chart image recognition", *International Workshop on Graphics Recognition, GREC2003*, 30-31 July 2003, Barcelona, Spain.
- [11] C. L. Tan, B. Yuan and C. H. Ang, "Agent-based text extraction from pyramid images", *Int. Conf. on Advances in Pattern Recognition*, 1998, Plymouth, UK, pp. 344-352.
- [12] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, and P. Dosch, "Text/Graphics Separation Revisited", *5th International Workshop on Document Analysis Systems, DAS 2002*, pp. 200-211, 2002.