Model-based Chart Image Recognition

Weihua Huang, Chew Lim Tan and Wee Kheng Leow

SOC, National University of Singapore, 3 Science Drive 2, Singapore 117543
E-mail: {huangwh,tancl, leowwk@comp.nus.edu.sg}

Abstract

In this paper, we introduce a system that aims at recognizing chart images using a model-based approach. First of all, basic chart models are designed for four different chart types based on their characteristics. In a chart model, basic object features and constraints between objects are defined. During the chart recognition, there are two levels of matching: feature level matching to locate basic objects and object level matching to fit in an existing chart model. After the type of a chart is determined, the next step is to do data interpretation and recover the electronic form of the chart image by examining the object attributes. For testing purpose, a set of testing images are either downloaded from the internet or scanned from books and papers. By comparing the recovered data and the original chart, we are able to evaluate the current system and confirm its validity.

Keywords: Chart Recognition, Vectorization, Model Construction, Objects and Attributes.

1 Introduction

In recent years, people are developing various kinds of document image analysis techniques that cover a wide range of image types, from journal papers to name cards, from forms to engineering drawings. Although the major focus is still on the textual information contained in the images, there has been an increasing effort to extract information from graphs, diagrams and figures etc. Scientific charts, with very effective visual impact, are widely used in many documents to present statistical data, to analyse data trends and to compare different data series. Though the status of scientific charts cannot be ignored, through literature survey we found that there is very little work done and reported that focuses on scientific chart recognition. Thus when facing scientific charts in the process of converting imaged documents into electronic form, the information contained in the charts will be either incorrectly manipulated or simply lost.

In our previous attempts [1-2], we were working on recognition of several chart types using Hough transformation and learning-based approach. Though the results obtained were encouraging, there are still some limitations. First of all, there was no data interpretation yet so the information contained in the charts were not extracted. Secondly, through experiments we found that Hough transformation was computationally costly and may not work well when there are a large amount of line segments with various lengths in the chart image. As a new attempt, we are developing a system that is able to achieve both chart type recognition and chart data interpretation. In the new system, we performed raster-to-vector conversion using an algorithm similar to [3], which is simple and can detect lines and arcs that are precise enough for further processing. Based on the vectorized lines and arcs obtained, we were able to fit a given chart into one of the existing chart models and further recover the electronic form of a chart. Currently we concentrate on four commonly used chart types: bar chart, pie chart, line chart and high-low chart.

The core of our system is to use a model-based approach to achieve our goals. Model-based image recognition is very popular in research areas ranging from 2-D image analysis to 3-D object recognition. The basic idea is to model certain number of object classes and to match the models with views of new objects in the novel images. One typical model developed is the geon model first proposed in 1985 by Biederman [4], which contains total of 36 geons that are qualitative 3-D shapes containing viewpoint invariant features such as parallelism, symmetry and continuity etc. In our application, we adopt the idea of “geons” and define basic object shapes with distinguishable features for different chart models, so that the system is able to search for shapes in the given image that can match objects in existing models. Currently our system only detects 2D shapes to handle 2D bar charts, 2D line charts, 2D high-low charts and 2D pie charts. 2D shapes can also be used to handle 3D pie chart but some variations are needed, which will be discussed later.

The remaining sections of this paper will discuss the details of the proposed system. Section 2 talks about feature extraction from a given chart image. Section 3 introduces how the chart type is determined. Section 4 shows how data in the chart image is interpreted and the electronic form of the chart is recovered. Section 5 talks about experiments carried out followed by discussion of the results. Finally, section 6 concludes this paper with some future works mentioned.

2 Feature Extraction

There are two ways to do model based matching: top down and bottom up. For the former approach, a basic model is derived and is used to search for mappings in the given image. For the latter approach, some basic features are extracted and used for constructing an existing model. We adopted the second approach in our current system. The basic features extracted here are the straight lines and arcs in the chart image.

2.1 Text and Graphics Separation

A typical chart image contains two kinds of information: textual information and graphical information. Each kind of information indicates the data in the chart from a different way. Textual information mainly includes the title of the chart, the name of the data series, some descriptions of the chart, and the numerical values of the data. On the other hand, graphical information includes the actually drawing, such as reference lines, edges and arcs. The first stage in our system is to distinguish these two kinds of information and store them separately for further processing.

Karl Tombre etc. summarized a number of text/graphics separation techniques in their paper [5], which are very helpful for us to tackle our problem. We used the traditional connected component based analysis followed by a set of filters that examines various properties of connected components obtained. The filtering process is summarized in Figure 1.

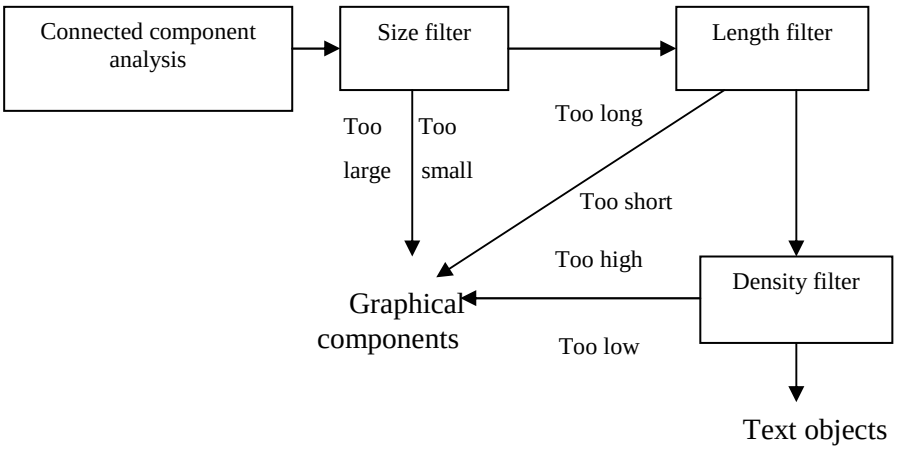
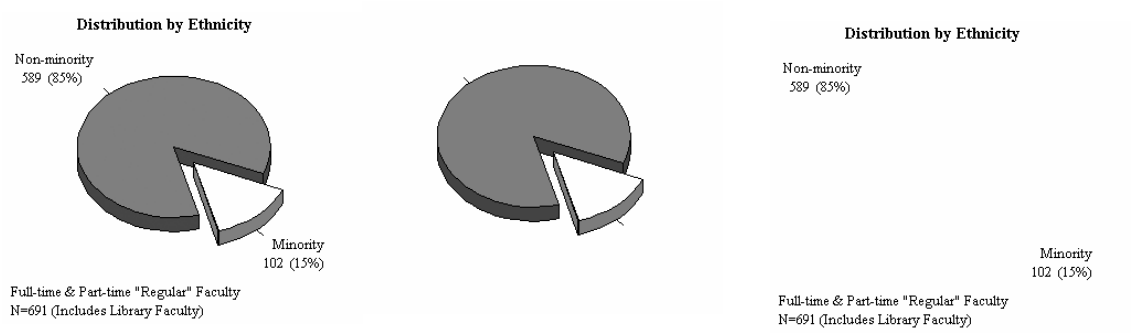


Figure 1. Text/graphics separation in the proposed system.

As we see from Figure 1, the filtering is based on thresholding. The threshold values are determined by going through some training samples. Using the set of filters above, most of the text and graphics can be separated. However we are aware that this is a very simple approach, thus a lot of issues are to be considered, such as: text touching graphics, text embedded in graphics, graphical components similar to text, etc.

The separated textual and graphical information are stored as two different images. One example is shown in Figure 2. For the image containing textual information, OCR technique can be applied to retrieve the text from it. The image containing graphical information will be passed to the next step of feature extract stage. Currently we focus on the graphical information only, but we believe with the help of a powerful OCR, the textual information can be retrieved and become a key to precise restoration of chart data.



(a) Original image (b) Graphics extracted (c) Text extracted

Figure 2. Example of text/Graphics separation.

2.2 Edge Detection

In a chart image, the colour or greyscale level within a graphical component is consistent. On the other hand, the colour difference or greyscale level difference between neighbouring graphical components is normally significant. Thus by examining these properties, we are able to find out the precise edges of graphical components. Even if the edges of graphical images become blurring due to scanning effect, setting a single threshold is powerful enough to find out the actual edges. After the edge detection step, only edge points are kept and passed to the next step for straight line detection, so that computational efficiency is maintained. An example of the edge map detected is shown in Figure 3.

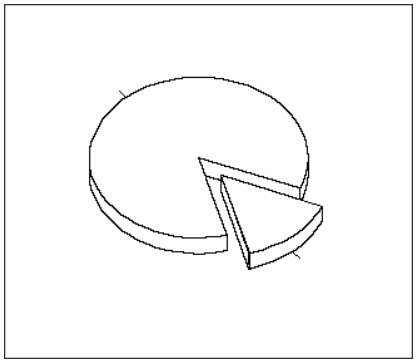
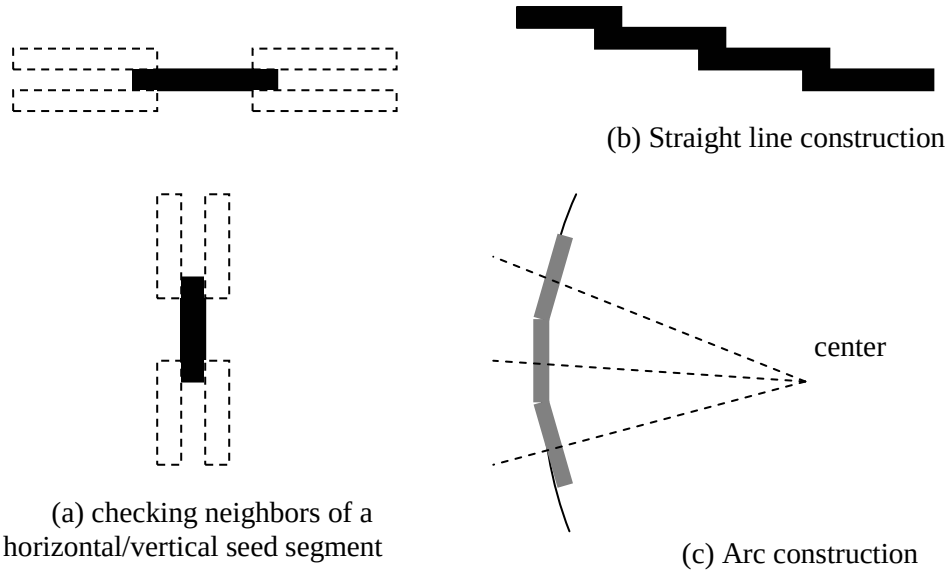


Figure 3. Example of edge map obtained.

2.3 Vectorization

Jiqiang Song, Feng Su etc. proposed the Line Net Global (LNG) Vectorization algorithm [3] that uses seed segments to perform directed vectorization. Here we use a similar approach: first of all, we find a straight line segment as a seed, and we try to extend it to find a complete line or arc by examining the neighboring segments. The seed segment indicates the direction of searching, thus the algorithm is

computationally powerful. The search direction can be horizontal or vertical, as illustrated in Figure 4(a), thus the searching process is executed twice for both directions.



(a) checking neighbors of a horizontal/vertical seed segment

(b) Straight line construction

(c) Arc construction

Figure 4. Vectorization of straight lines and arcs.

As shown in Figure 4(b), to construct a straight line, the line segments should satisfy the following conditions:

- All line segments should have the same orientation, either horizontal or vertical.
- All line segments should have similar length, except for the starting segment and end segment which can be shorter.
- Along the line, the x-y coordinates should change monotonically.

As shown in Figure 4(c), to construct an arc, we need to find three consecutive line segments and check the following condition:

- The bisector of the three line segments should converge to a common point, which will become the estimated center for the arc.

After the lines and arcs are vectorized, there is a need to refine them. The reason is that the image quality may be poor and as a result there may be broken lines and arcs. To handle this, we check through the lines and arcs obtained to see if any of them have the same orientation or share the same center, and the distance between their end points is smaller than certain threshold. If such case is found, two lines will merge to form a new line and two arcs will merge to become one arc.

3 Chart Type Recognition

After the vectorization, we have a set of vectors representing the straight lines and arcs. We can check the following relationships among the straight lines: parallelism, perpendicular and convergence. These relationships are important to us because they are the indications of the existence of certain chart objects, such as bars or pies etc.

First of all, we look for the existence of x-y axes. If we assume that the given image is well positioned without any skew angle, then the x axis should be the longest horizontal line while the y axis should be the longest vertical line. We also assume that the y axis is always at the left hand side of the image and the x axis is always at the bottom of the image. Further than that, the space within x-y axes should contain most other

lines. Our current program handles four types of charts: bar chart, pie chart, line chart and high-low chart. Once we determine the presence or absence of x-y axes, we can proceed to find out the actual chart type.

3.1 Bar Chart Recognition

For a bar chart, the x-y axes should be present. The most basic shape in the bar chart is a bar, which consists of two vertical lines and one horizontal line on the top. However there are some variations of the basic shape, as summarized in Figure 5.

It's easy to check for the basic shape and its variations, however we must check for other constraints before we treat a shape as a bar. These constraints are:

1. Both vertical lines must touch the x-axis.
2. There should be no other object on top of a bar.
3. For all the bars, the width of a bar or the distance between two vertical lines should be consistent.
4. In case of colour image or greyscale image, the colour of all bars should be the same (single data series is assumed here).

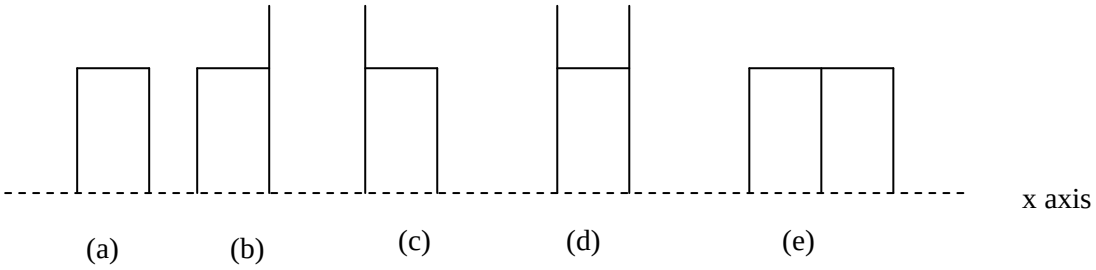


Figure 5. Basic bar shape and its variations.

There are two rounds of checking. In the first round, among all the lines obtained from the previous stage, we look for vertical lines touching the x-axis. Then for each neighbouring pair of such vertical lines, we look for horizontal line on top of them. If the top line is found, then a bar candidate is found. In this round, shape (a), (b) and (c) can be found, and the width and colour of bars are estimated. The second round is necessary to find shape (d) and (e). In this round, we revisit the vertical lines and horizontal lines and check the width and colour constraints. After both rounds finished, if the number of bars is greater than 2, then the given chart is considered as bar chart, otherwise it is passed to other chart models for checking.

3.2 Pie Chart Recognition

For a pie chart, there are no x-y axes. Thus all the images without x-y axes are passed to this process for pie chart checking. For a typical pie chart, no matter 2D or 3D, the basic shapes are shown in Figure 6.

There are some common features for these pie shapes:

1. The two lines converging to one point.
2. There is an arc connecting the other endpoint of the two lines.

Besides these features, there are also some constraints for the whole chart image:

1. The converging point of all pies should be close to each other.
2. The summation of angles of all pies should be 360 degree or at least close to that.

Figure 6. Basic pie shapes.

First of all, we can find all the converging pairs among the set of lines obtained. The next task is to test whether there is an arc connecting the non-converging endpoints of a pair of lines. Among the arcs obtained, we check if an arc has common endpoints as the two lines' and if the center of the arc is close to their converging point. If the arc test is passed, the converging pair is treated as a pie. After all pies are found and the global constraints are also satisfied, the given image is treated as a pie chart.

3.3 High-Low Chart Recognition

For a high-low chart, the x-y axes should be present. In a typical high-low chart, there are no objects but we can expect to see a number of vertical lines above the x-axis but do not touch the x-axis. Further than that, there should not be too many converging lines, which means the vertical lines should not be connected by other lines. Here we need to mention that telling whether a given chart is a high-low chart is not difficult, but sometimes we may face data loss because some high low line segments look like numerical "1" very much, and will be treated as text in the text/graphics separation stage. An example is shown in Figure 7.

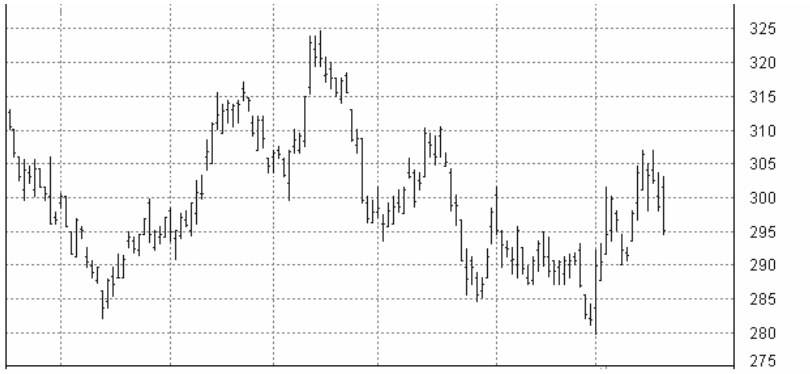


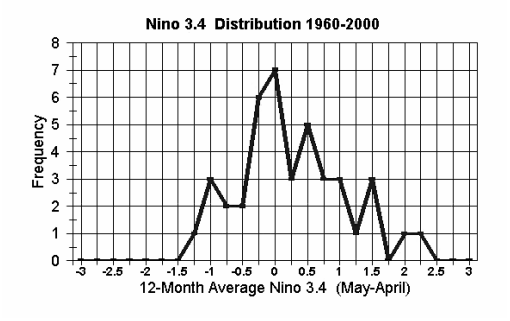
Figure 7. A high-low chart with many high-low bars similar to "1".

3.4 Line Chart Recognition

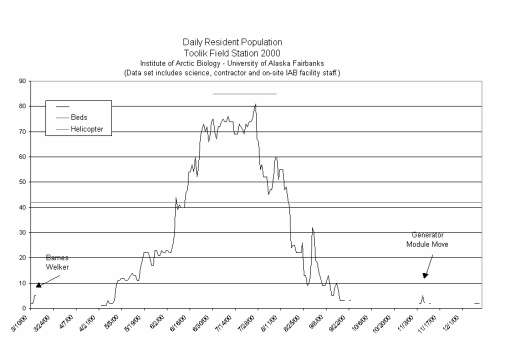
For a line chart, the x-y axes should be present. In a typical line chart, all data points are connected from left to right. So we can check for both convergence and continuity among lines. If there are just a few data points in the line chart, then there are fewer lines and checking for convergence is easier. If there are a lot of data points in the line, there is no way to locate the actual data points, so checking for continuity becomes the only choice. Figure 8 shows examples for both cases.

4 Data Interpretation

As we have mentioned in the first section, it is a new attempt to interpret data in the chart images. There are two different approaches to recover the data contained in a chart image. For bar chart, pie chart and high-low chart, our approach is to make use of the objects found during the recognition of chart type and check the object attributes to recover data. For line chart, as there may be too many data inside that we cannot



(a) Line chart with a few data points



(b) Line chart with a lot of data points

Figure 8. Example of line charts.

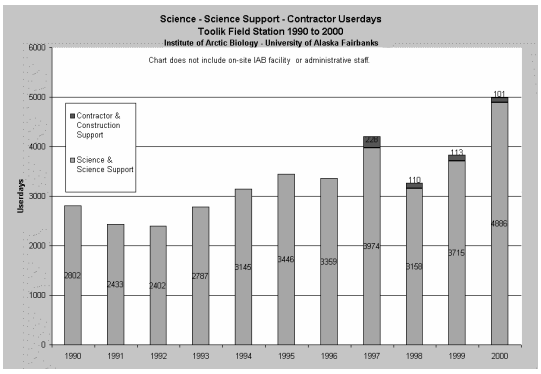
locate every data point, we adopt a different approach by sampling the space within the x-y axes and collect a series of data. The details of data interpretation are discusses in the subsections below.

4.1 Bar Chart Data Interpretation

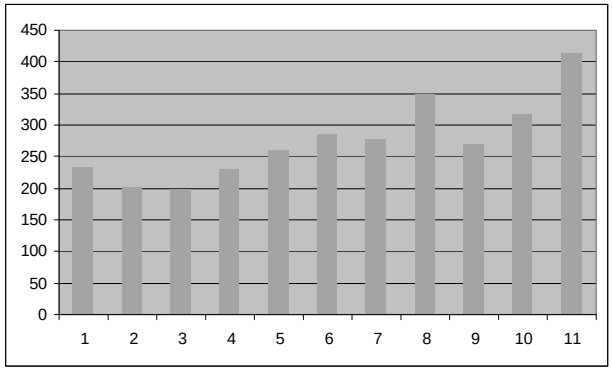
When recognising the chart type, we have already obtained a set of bars. Thus the task for restoring bar chart data can be done on the bars as follows:

1. To find out the value of a bar, we calculate its height by finding the difference between the top of the bar and the x-axis.
2. To find out the relative position of a bar, we calculate the absolute position of a bar on the x-axis, and then perform a sorting and arrange the data from left to right.

However the data recovered here can only reflect the relative positive and height of bar data. Since currently the textual information is not available yet, we are not able to recover the exact values of the data in the bar chart. An example of bar chart data interpretation is shown in Figure 9.



(a) Original bar chart image



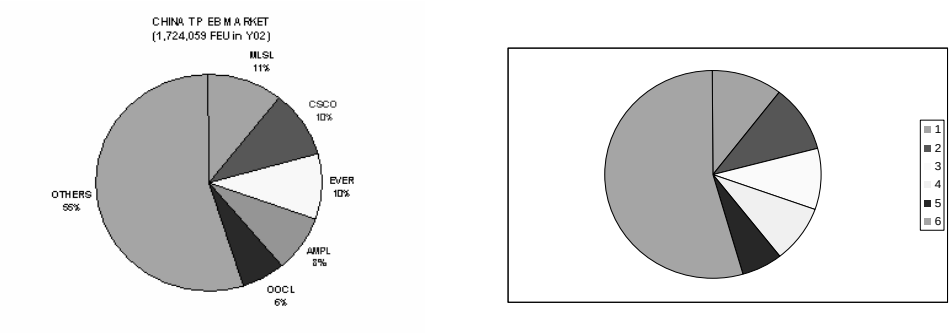
(b) Bar chart generated based on data recovered

Figure 9. Example of data interpretation for bar chart.

4.2 Pie Chart Data Interpretation

Similar to the bar chart data interpretation, we can use the pies obtained in the previous process to recover data. The angle for each pie is calculated, and the percentage of each angle versus 360 degree is treated as the data for each pie. An example is shown in Figure 10.

For 2D pie chart, the data recovered almost truly reflect the original data except for some minor error due to imperfect line detection. However, in a 3D pie chart we are viewing an image of an ellipse instead of a circle, thus the angle of a pie does not reflect the true percentage of data. Thus some further transformation needs to be done to convert the ellipse back to a standard circle, which may require more mathematical analysis.



(a) Original pie chart (b) Pie chart generated based on data recovered

Figure 10. Example of pie chart data interpretation.

4.3 Line Chart Data Interpretation

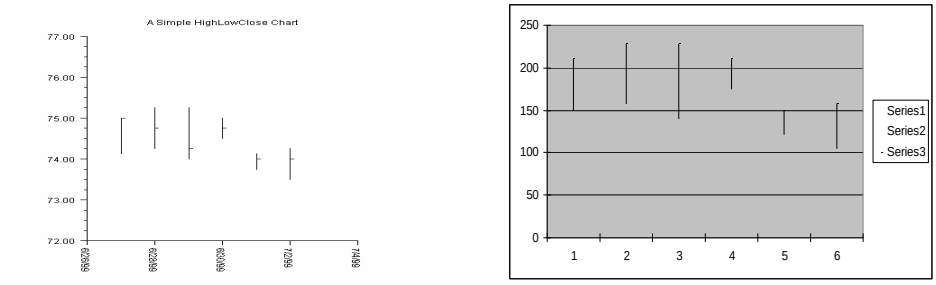
As we mentioned previously, we can locate the exact data points in some line charts but fail to do so in others. So our goal here is to simply recover the shape of the data line. One simple approach is to sample the space within the x-y axes from left to right and try to pick up one data point for each sample column. If we sample the space pixel by pixel, then we can expect to completely recover the shape of the line. The relationship between sample rate and precision of data recovery will be discussed in the next section, with some examples shown.

We found that some line charts will include either horizontal reference lines or vertical reference lines or both. As an extra processing step, we used a length filter to remove long horizontal lines and vertical lines, based on the assumption that the data line would not contain such long line segments.

4.4 High-Low Chart Data Interpretation

If all the high-low bars are found, the restoration of data is straightforward. We can find out the high value and low value by calculating the distance between the two end point of a bar and the x-axis. Again the data obtained here are only the relative values. If the original values are desired, text information must be obtained to help out.

Due to the imperfect text/graphics separation, many data in the high-low chart are lost. Currently we are still trying to improve our system on this issue. So not much testing is done to the data interpretation for the high-low charts. However we did manage to recover data for some high-low charts, as shown in Figure 11.



(a) Original high-low chart (b) High-low chart generated based on data

Figure 11. Example of high-low chart data interpretation.

5 Experimental Results

We have collected a number of testing images from various sources such as the internet or some scanned document pages. The entire test collection contains 8 bar chart images, 8 line chart images, 8 pie chart images and 3 high-low chart images.

To test the performance of our system, we run through these images and obtain two kinds of results: the type of the chart recognized by the system and the data recovered. For the data obtained, we convert the recovered data into Microsoft Excel files and used the drawing facilities to draw the same type of charts.

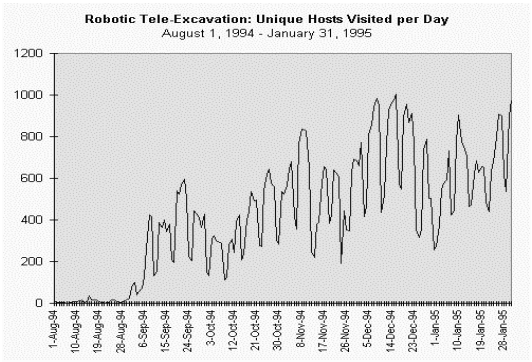
5.1 Comparing Recovered Data with Original Data

The most direct and best way to evaluate our system is to compare the recovered data with the original data. For the bar charts and pie charts with the original data known, we count the number of data that are correctly recovered. Since the data obtained from the bar chart are only relative values, we manually calculate the absolute values by multiplying corresponding ratio to the maximum value among the data. If the difference between the actual value and the absolute value obtained is smaller than 1% of the actual value, then the data is considered to be correctly recovered. For pie chart, since the actual percentages among the data are normally included in the image, we directly use them to compare with the percentages obtained from the angles. Again if the difference between the actual percentage and the percentage obtained is smaller than 1%, the pie is considered to be correctly recovered. The results are summarized in Table 1 below:

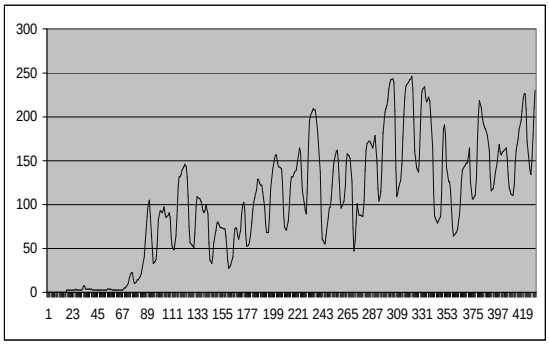
Bar chart	Chart type recognized?	No. of bars in the image	No. of bars recovered	Pie chart	Chart type recognized?	No. of pies in the image	No. of pies recovered
B1	Yes	5	5	P1(3D)	Yes	2	2
B2	Yes	15	15	P2(3D)	Yes	2	2
B3	Yes	11	11	P3(3D)	Yes	2	0
B4	Yes	3	3	P4(3D)	Yes	4	1
B5	Yes	6	6	P5(2D)	Yes	4	3
B6	Yes	7	7	P6(2D)	Yes	6	6
B7	Yes	11	11	P7(2D)	Yes	6	6
B8	Yes	6	6	P8(2D)	Yes	6	6

Table 1. Performance evaluation for the proposed system.

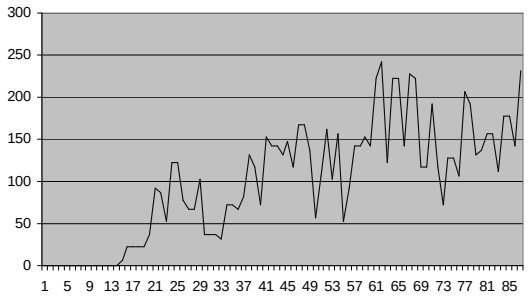
For most of the line charts used for testing, the original data is unknown and is difficult to tell from the chart itself. Thus the only way is to visually compare the original line chart with the chart generated by Microsoft Excel based on the data obtained. An example is shown in Figure 12.



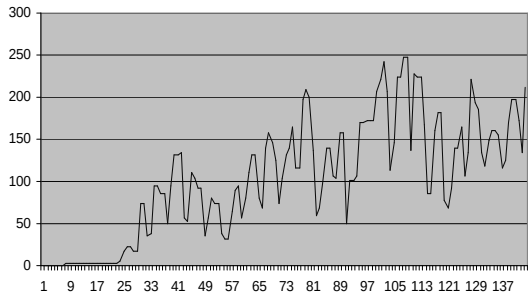
(a) Original line chart



(b) Sample rate = 1 pixel



(d) Sample rate = 5 pixels



(c) Sample rate = 3 pixels

Figure 12. Sampling line chart data with different sample rates.

5.2 Discussions

From the testing results obtained, we can see that the proposed system works very well for 2D bar charts. It also works well for 2D pie charts. However, when dealing with 3D pie charts, the angles from the pies may not reflect the true percentages among the data since the drawing is an ellipse instead of a circle. Thus the number of pies correctly recovered becomes very few for image P3 and P4. The results for P1 and P2 are still good since these two ellipses are similar to circles. An example is used to illustrate the difficulty with data recovery for 3D pie charts, as shown in Figure 13 and Table 2.

	Actual data percentage	Angle measured	Percentage of angle/360 degree
Pie 1	31%	141 degree	39%
Pie 2	5%	14 degree	4%
Pie 3	34%	87 degree	24%
Pie 4	30%	118 degree	33%

Table 2. Difference between actual data and data recovered
for the pie chart in Figure 12.

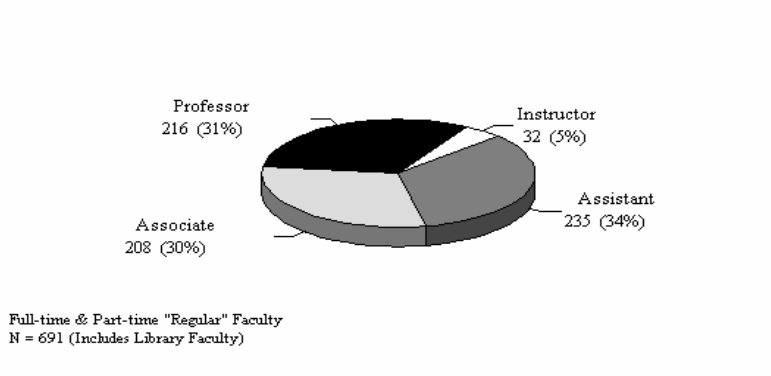


Figure 13. The testing 3D pie chart P4.

When recovering data in a line chart by sampling the space within x-y axes, we need to determine the sample rate. It can be seen from Figure 12 that a smaller sample rate returns higher precision but also means more data points to be stored. On the other hand, a larger sample rate will reduce the precision but saves space by skipping more data points in the chart image.

An obvious advantage for converting chart images into electronic forms is that much less space is required to store the data. The total size of the 25 testing images is about 6.58MB under windows XP system, while the total size of data recovered and stored as Excel Files is only about 518KB, saving more than 90% of the original space.

6 Conclusion and Future Works

In this paper, we introduce a model based system that aims at recognizing scientific chart images and recovering their electronic forms. The system separates text and graphics information, and then extracts lines and arcs from the graphics based on vectorization. Based on the lines and arcs obtained, various objects are constructed and are used to match with four kinds of chart models. After the type of a chart is determined, data interpretation is done to recover the data contained in the chart image. Experiments are done to a set of testing images and the results are encouraging.

However, as a new attempt, there are further works to be done, such as:

- Improving text/graphics separation, especially working towards separating text and graphics that touch each other.
- Recognizing text objects obtained and using text information for more precise data interpretation.
- Automatic locating chart images in document pages, separating them from normal text and other drawings.
- Further work on high-low charts, and possibly extend the work to handle other types of charts.
- Separating multiple data series in one chart and interpret each data series independently.

We believe that the importance of information contained in chart images will attract more and more attentions from researchers. Automatic and reliable conversion from chart image to equivalent electronic form will be realized in near future as more efforts are put in and major problems are solved.

References

[1] Y. P. Zhou and C. L. Tan, “Hough technique for bar charts detection and recognition in document images”, *International Conference on Image Processing, ICIP 2000*, page 494-497, 2000.

[2] Y. P. Zhou and C. L. Tan, "Learning-based scientific chart recognition", *4th IAPR International Workshop on Graphics Recognition, GREC2001*, page 482-492, 2001.

[3] J. Song, F. Su, J. Chen, C. L. Tai, and S. Cai, "Line net global vectorization: an algorithm and its performance analysis", *IEEE Conference on Computer Vision and Pattern Recognition*, page 383-388, South Carolina, 13-15 June, 2000.

[4] I. Biederman, "Human image understanding: Recent experiments and a theory", *Computer Vision, Graphics and Image Processing*, 32:29-73, 1985.

[5] K. Tombre, S. Tabbone, L. Pélissier, B. Lamiroy, and P. Dosch, "Text/Graphics Separation Revisited", *5th International Workshop, DAS 2002*, page 200-211, 2002.