

Extraction of Vectorized Graphical Information from Scientific Chart Images

Ruizhe Liu, Weihua Huang and Chew Lim Tan
School of Computing, National University of Singapore
{liurz, huangwh, tancl}@comp.nus.edu.sg

Abstract

Graphical components information extraction is a crucial step in the chart recognition and understanding process. However, existing methods of information extraction from chart images either are type-dependent or rely on certain assumptions. In this paper, we present a general method to extract vectorized graphical information from scientific chart images. Our algorithm firstly constructs a data structure called directional single-connected chains (DSCC). It then employs ellipse-specific fitting and orthogonal diagonalization to calculate the curvatures of the chains and classify the chains into either straight lines or arcs. Finally we combine all straight lines and all arcs accordingly and use linear regression to compute their attributes. The DSCC has a good property in that it is less susceptible to noise. The experiment results show that our algorithm is efficient, robust and accurate.

1. Introduction

Charts, as major visual aids of data analysis, are simple, clear and widely used in all kinds of scientific documents. It is stated that about one trillion statistical graphs are printed every year [1] and most of the statistical graphs appearing in scientific papers are scientific charts or diagrams. There are several commonly used chart types, such as bar chart, pie chart and line chart etc.

In recent years, several works have been reported in the area of chart recognition and understanding. For example, Zhou et al proposed two different approaches for chart classification, namely Hough transform and learning-based approach [2, 3]. Yokokura et al proposed a schema-based framework to graphically describe the layout relationship information of the bar charts [4]. Huang et al proposed a model-based approach for recognition of several commonly used types of chart image [5]. Knowing the attributes of graphical components in the charts, such as the starting and ending point of a straight line segment or an arc,

the center and radius of an arc, and the maximum and minimum radii of an elliptic arc, is a necessary step before high level chart interpretation and understanding can be achieved. However, there hasn't been a method that extracts general low-level graphical attributes without assumptions. This paper presents such a method that constructs a simple data structure called Directed Single-connected Chains and applies ellipse fitting to the chains to obtain attributes of straight lines, circular arcs as well as elliptic arcs.

The remaining sections of this paper will further discuss the details of our work. Section 2 surveys some previous works related to our study. Section 3 introduces the data structure of Directed Single-connected Chain. Section 4 talks about more detailed steps of the system. Section 5 presents experimental results together with discussions. Section 6 will give a conclusion to this paper.

2. Related works

The central aspect in graphics recognition is vectorization, which does the raster-to-vector conversion and extracts graphical primitives such as straight lines and arcs. Vectorization of straight lines and circular arcs is becoming stable and robust. Examples of straight line vectorization include [6, 7, 9] and examples of arc vectorization include [7-10]. Noise in the input image and line junctions are the common obstacles to achieving satisfactory vectorization performance, and another major concern is the computational complexity. In [11], a simple but efficient vectorization method based on a data structure called Directional Single-connected chains (DSCC) was originally proposed to extract frame lines in tables. It is easy to implement and computationally efficient. Using DSCC, noise handling can also be easily implemented. Our work adopts this data structure and further extends it with curve fitting to handle other graphical entities such as circular arcs and elliptic arcs. Unlike the work presented in [12], the elliptic arc fitting process does not rely on any assumptions.

3. The Directed Single-Connected Chain

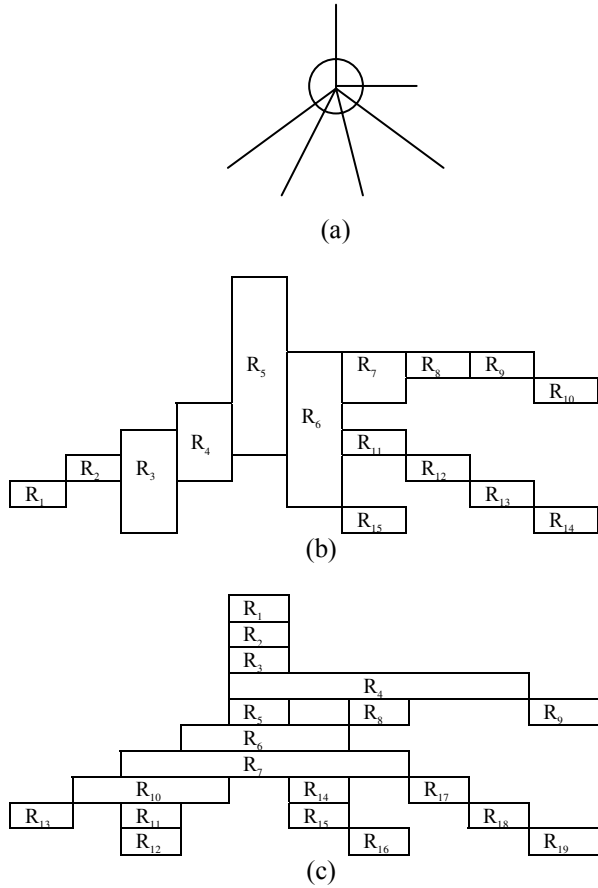


Figure 1 (a): real lines; (b): vertical run-lengths in the circled area in (a); horizontal chains: $C_1=\{R_1,R_2\}$; $C_2=\{R_3,R_4\}$; $C_3=\{R_5,R_6\}$; $C_4=\{R_7,R_8,R_9,R_{10}\}$; $C_5=\{R_{11},R_{12},R_{13},R_{14}\}$; $C_6=\{R_{15}\}$ (c): horizontal run-lengths in the circled area in (a); vertical chains: $C_1=\{R_1,R_2,R_3\}$; $C_2=\{R_4\}$; $C_3=\{R_5\}$; $C_4=\{R_8\}$; $C_5=\{R_9\}$; $C_6=\{R_6,R_7\}$; $C_7=\{R_{10}\}$; $C_8=\{R_{14},R_{15},R_{16}\}$; $C_9=\{R_{17},R_{18},R_{19}\}$; $C_{10}=\{R_{13}\}$; $C_{11}=\{R_{11},R_{12}\}$

The Directional Single-Connected Chain (DSCC) is constructed from a binary image [11]. A DSCC is composed by a set of short segments called run-lengths, and it can be horizontal or vertical. A horizontal (vertical) chain is formed by doing a linear regression of the middle points of its run-lengths, and the line has a degree $< (\geq) \pi/4$. A horizontal chain contains vertical run-lengths (Figure 1(b)) and a vertical chain contains horizontal run-lengths (Figure 1(c)).

A vertical run-length is defined as $R_i(x_p, y_s, y_e) = \{(x,y) \mid \forall p(x,y) = 1, x = x_p, y \in [y_s, y_e] \text{ and } p(x_p, y_s -$

$1) = p(x_p, y_{e_i+1}) = 0\}$, while $p(x,y)$ is the location of a pixel in the image, 1 is black pixel (foreground), 0 is white pixel (background). This run-length starts from (x_p, y_s) and ends at (x_p, y_e) . In a horizontal chain C_h , (Figure 1(b)) every run-length R_i is arranged in a horizontal sequence, and any two run-lengths R_i and R_{i+1} are connected horizontally. Except for the run-lengths at both ends of the chain, R_i and R_n , any R_i has one and only one run-length R_j connected on each side. For the left side of R_i and the right side of R_n , either there is no run-length or there are more than one run-lengths connected. The connection here refers to 8-neighbor connection. Also, the run-lengths in the same chain should have similar length within some range, which is set between half of the average length and twice of the average length. Otherwise, the chain is considered to be broken. The horizontal run-length is defined similarly as the vertical run-length described above. For a horizontal run-length $R_p, R_i(y_p, x_s, x_e) = \{(x,y) \mid \forall p(x,y) = 1, y = y_p, x \in [x_s, x_e] \text{ and } p(x_s-1, y) = p(x_e+1, y) = 0\}$. This run-length starts from (x_s, y_p) and ends at (x_e, y_p) . Similarly, a vertical chain C_v is formed by horizontal run lengths (Figure 1(c)).

4. Vectorization process

4.1. Preprocessing

The system accepts binary, grayscale or color images as input. We assume the image contains charts only. The first step is to perform text/graphics separation by connected components construction and classification proposed in [13]. Then edge detection is done based on calculating the intensity difference between neighboring pixels, and the binary edge map is obtained.

4.2. DSCC construction and post-processing

DSCCs are constructed using the definitions illustrated in section 3. For vertical chains, the pixels in the image are scanned from top to bottom, while for horizontal chains, the scanning process is from left to right. An example is shown in Figure 2. Since the input image may be noisy, there are several post-processing steps performed to refine the resulting chains:

- Filtering: the run-lengths with length 1 and no neighbors are treated as black dots and are removed. Run-lengths with length 1 and have only one neighbor are also removed. Such run-lengths are treated as protrusions into actual line segments. Tiny chains with the number of run-lengths being one or two are also treated as noise and removed.

- Smoothing: for vertical run-lengths, if two of them, R_i and R_j , are in the same column, i.e., $x_i = x_j$, and the blank area between them are less than 3 pixels, then they are combined to form a new run-length, R_k , where $x_k = x_i$, $y_{sk} = \min(y_{si}, y_{sj})$, and $y_{ek} = \max(y_{ei}, y_{ej})$. Similar process is carried out for horizontal run-lengths. With this step, the holes in the lines are filled. It also prevents broken line segments from appearing. An example of filtering and smoothing is shown in Figure 3.

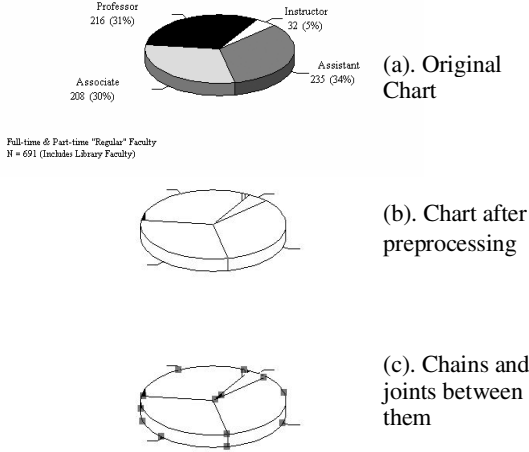


Figure 2. Example of processing a 3D pie chart

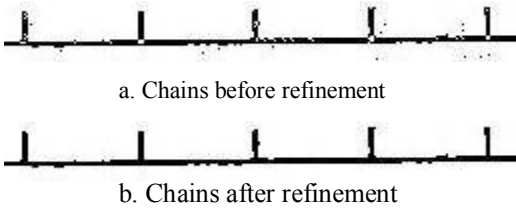


Figure 3. Smoothing run-lengths

- Splitting: a DSCC may be a straight line, or a curve or even a polyline. To distinguish between a curve and a polyline, we need to record down the turning points along the DSCC. The idea is to use a divide-and-conquer strategy. The starting point and ending point of the DSCC is linked to form a virtual line L . A point p on the DSCC with the largest distance to L is treated as a turning point if the distance is greater than a predefined threshold. The turning point is stored and the same search process is then applied to the two sub-chains of the DSCC on the two sides of the point. In the end, a set of turning points $\{p_1, p_2, \dots, p_n\}$ is obtained.

4.3. Obtaining the vectors

4.3.1. Ellipse-specific fitting theory using least square method

Fitzgibbon et al propose an ellipse-specific fitting [14]. This fitting method always constructs an ellipse from a given point set. In our case, the mid-points of the set of run-lengths stored with a DSCC are used as the point set for the method. By fitting the points to a hypothetical ellipse, and computing the ratio of the maximum radius versus the minimum radius, we are able to tell if a DSCC is a straight line, a circular arc or an elliptic arc.

The main idea of the method is to represent an arc as a second order polynomial:

$$F(A; X) = A \cdot X = ax^2 + bxy + cy^2 + dx + ey + f = 0 \quad (1)$$

where $A = [a \ b \ c \ d \ e \ f]^T$ and $X = [x^2 \ xy \ y^2 \ x \ y \ 1]^T$. $F(A; X_i)$ is called the "algebraic distance" of a point (x_i, y_i) to the conic $F(A; X) = 0$; Then the fitting of a general conic is done by minimizing the sum of squared algebraic distances:

$$D_A(A) = \sum_{i=1}^N F(A; X_i)^2 \quad (2)$$

of the curve to the N data points X_i . Enforcing quadratic constraint $4ac - b^2 = 1$ on the parameters helps to avoid the trivial solution to (2) and degenerate the problem to ellipse fitting. After a series of transformations, the minimization of (2) can be solved by solving a system of simultaneous equations:

$$SA = \lambda CA \quad (3)$$

$$A^T CA = 1 \quad (4)$$

where S is the *scatter* matrix $D^T D$. $D = [x_1 \ x_2 \ \dots \ x_n]^T$ is called the *design matrix* and C is the matrix that expresses the constraint. λ is the Lagrange multiplier. For the complete derivation of the equations, please refer to [14].

After the solution vector $A = [a \ b \ c \ d \ e \ f]^T$ (i.e. all parameters) is obtained, an affine transformation is performed to transform the ellipse from general quadratic form to an standard form:

$$\frac{(x-x_0)^2}{a^2} + \frac{(y-y_0)^2}{b^2} = 1 \quad (5)$$

where the center is (x_0, y_0) with maximum radius $\max(a, b)$ and minimum radius $\min(a, b)$. This is a basic operation in linear algebra. The ellipse fitted is shown in Figure 4(a).

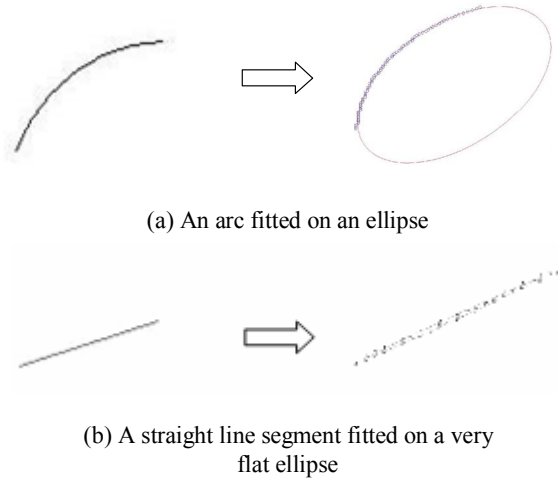


Figure 4. Example of ellipse fitting

4.3.2. Extracting straight lines and circular arcs

Because the method described in section 4.3.1 always returns an elliptic arc (or a complete ellipse) from the given set of points, we still need to come up with rules to extract straight lines and circular arcs. The basis of the rules is the ratio of the maximum radius versus the minimum radius of the extracted elliptic arc.

- Rule 1: if the ratio is greater than a predefined threshold value, then the original DSCC is treated as a straight line and the fitted elliptic arc is rejected. An example is shown in Figure 4(b).
- Rule 2: if the ratio is close to 1 (with a small allowed error range), then the fitted arc is treated as circular arc instead of elliptic arc.

As the DSCC may also be a polyline, one extra step is to calculate the least square error between the fitted arc and the original set of run-lengths stored with the DSCC. If the error is too significant, then the DSCC is considered not an arc but a polyline. The turning points obtained in the splitting step illustrated in section 4.2 are used as the turning points along the polyline.

4.3.3. Combining straight lines and arcs

The last step is to combine straight lines and arcs that were originally broken. Firstly, we define a connected area. That is, if the starting or ending points of two chains have Euclidean distance less than 8 pixels, they are considering in the same connected area.

To combine two arcs, we define the following rules:

1. The two arcs must be within the connected area.

2. The two tangent lines of the starting or ending points of the two arcs should be angled less than 10 degrees.

The first rule is straight forward. The second rule is implemented by computing the five continuous run-lengths counting from the starting or ending run-lengths and do the linear regression on the midpoints of the five continuous run-lengths. The angle of the two lines is then computed.

To combine two straight lines, we also define 2 rules:

1. The two lines must be within the connected area.
2. The two lines should be angled less than 10 degrees.

5. Experimental results and discussion

To test our system, we use the publicly available chart image dataset published in [15]. The dataset contains 200 chart images including 2D and 3D bar charts, 2D and 3D pie charts, and 2D line charts. It also provides multi-leveled ground truth information for performance evaluation, including vector level information of straight lines, circular and elliptic arcs.

The system is used to process the 200 images and extract straight line segments, circular arcs and elliptic arcs from them. For straight line segments, the attributes stored are the starting point, the ending point and the thickness of the line. For circular arc, the attributes stored are the starting point, the ending point, the center point, the radius of the circle and the thickness of the arc. For elliptic arc, the attributes stored are the starting point, the ending point, the center of the ellipse, the maximum and minimum radii of the ellipse and the thickness of the arc.

Table 1. Performance of vectorization

		Correct %	Broken %	Wrong %
Bar chart	Straight line	84.72	6.94	8.34
	Arc	-	-	-
Pie chart	Straight line	83.45	15.11	1.44
	Arc	82	13.72	4.28
Line chart	Straight line	93.06	3.57	3.37
	Arc	-	-	-

To evaluate the vector information obtained, we compare the extracted vectors with the vectors provided in the ground truth data. For comparison purpose, the overlapping segment s [16] is calculated between an extracted vector v_d and the corresponding vector in the ground truth v_g . $Coverage(s, v_i)$ is calculated as the length of s divided by the length of v_i , where v_i is either v_d or v_g . If both $Coverage(s, v_d)$ and $Coverage(s, v_g)$ are $\geq 90\%$, then the extracted vector is

correct. If $Coverage(s, v_d)$ is greater than 90% but $Coverage(s, v_g)$ is not, then v_d is treated as a broken subpart of v_g . If both $Coverage(s, v_d)$ and $Coverage(s, v_g)$ are below 90%, then v_d is considered to be wrong. The results are summarized in Table 1.

From the table, we can see that the proposed system works reasonably well. The relatively higher percentage of wrong segments for bar chart type is due to the existence of a set of scanned bar chart images that are noisier than others. For pie charts, more broken vectors occur due to the fact that arcs are more prone to being broken into small pieces than straight lines during DSCC construction. To reduce the number of broken vectors, more lenient rules for combination can be specified.

6. Conclusion

This paper presents our work of extracting graphical information from scientific chart images. The main approach is the construction of Directional Single-connected Chains (DSCC) followed by curve fitting. The approach does not rely on the type of input chart image and does not make any assumption on the graphical information in the image. The overall vectorization result is encouraging. The attributes of the vectors obtained can facilitate further interpretation and understanding of the chart image. In the future, texture detection and more robust edge detection methods can be added in to handle texture and gradient color that are often applied to color chart images. Also for several steps such as edge detection and chain combination, there are a series of predefined threshold used. Reducing the number of thresholds or automating the threshold setting is also one of our future works.

Acknowledgement: This research is supported by A*STAR grant 0421010085 and NUS URC grant R252-000-202-112.

7. References

- [1] E. R. Tufte, *The visual display of quantitative information*, Cheshire, CT, Graphics Press, 1985
- [2] Y. P. Zhou and C. L. Tan, "Hough technique for bar charts detection and recognition in document images", *Int. Conf. on Image Processing, ICIP 2000*, pp. 494-497, 2000
- [3] Y. P. Zhou and C. L. Tan, "Learning-based scientific chart recognition", *GREC 2001*, pp. 482-492, 2001
- [4] N. Yokokura and T. Watanabe, "Layout-Based Approach for extracting constructive elements of bar-charts", *GREC'97*, pp. 163-174, 1997
- [5] W. H. Huang, C. L. Tan and W. K. Leow, "Model based chart image recognition", *6th International Workshop on Graphics Recognition, GREC'03*, pp. 87-99, 2003
- [6] W. Liu and D. Dori, "Sparse Pixel Vectorization: An Algorithm and Its Performance Evaluation", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, pp. 202-215, 1999
- [7] P. L. Rosin and G. A. West, "Segmentation of Edges into Lines and Arcs", *Image and Vision Computing*, 7(2): pp. 109-114, May 1989
- [8] D. Dori and W. Liu, "Incremental Arc Segmentation Algorithm and Its Evaluation", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp. 424-431, 1998
- [9] J. Song, F. Su, J. Chen, C. Tai and S. Cai, "Line Net Global Vectorization: an Algorithm and Its Performance Evaluation", *CVPR 2000*, pp. 383-388, 2000
- [10] P. Dosch, G. Masini and K. Tombre, "Improving arc detection in graphics recognition", *Proc. of the 15th Int. Conf. on Pattern Recognition*, vol. 2, pp. 243-246, 2000
- [11] Y. F. Zheng, C. S. Liu, X. Q. Ding and S. Y. Pan, "A Form Frame-Line Detection Algorithm Based on Directional Single-Connected Chain", *Journal of Software*, Vol. 13, pp. 790-796, 2002
- [12] W. H. Huang, C. L. Tan and W. K. Leow, "Ellipse arc vectorization for 3D pie chart recognition", *Int. Conf. on Image Processing, ICIP2004*, pp. 24-27, 2004
- [13] K. Tombre, S. Tabbone, L. Pelissier, B. Lamiroy and P. Dosch, "Text/Graphics Separation Revisited", *DAS 2002*, pp. 200-211, 2002
- [14] A. Fitzgibbon, M. Pilu and R. B. Fisher, "Directed Least Square Fitting of Ellipses", *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 21, pp. 476-480, 1999
- [15] L. Yang, W. Huang and C. L. Tan, "Semi-automatic Ground Truth Generation for Chart Image Recognition", *Document Analysis Systems, DAS'06*, pp. 324-335, 2006
- [16] W. Liu and D. Dori, "A protocol for performance evaluation of line detection algorithms", *Machine Vision and Applications*, vol. 9, pp. 240-250, 1997