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

The Vectorization of Engineering Drawings Based on Statistics

Gang Wang Yijia Lin* Yueyue Song
Department of Computer and System Science
Nankai University P.R.China

Abstract

In this paper we present algorithms by using statistic/geometric features and simulating human vision mechanism to extract lines and arcs for vectorization of engineering drawings. Experimental results for complex line drawings confirm performances of these algorithms.

1 Introduction

The vectorization of engineering drawings is an important field of machine vision applications. Although it has already obtained a material progress and commercial products have been put into market , the usage is still limited because of the harsh terms to the quality of paper line drawings and a certain amount of interactive operations involved in controlling recognition and modifying mistakes. So we do have a lot of research work to do to improve the processing speed and recognition quality even at the low level vectorization of engineering drawings. Some representative work at this stage can be categorized as follows:

(a) The thinning algorithm has been widely accepted in a lot of work to extract skeletons of lines. One of the latest work has been presented in [1].

(b) Sparsely scan the image to find lines and search the trace of a line with the orthogonal zig-zag routine along the line[2].

(c) Divide the image into square meshes and encode the image with patterns of square meshes which will be used as the smallest units for recognition of lines[3-4].

(d) Use two sides of contours of line segments to extract skeletons of lines[5].

(e) Apply the run length code as a low level description of line drawings to extract the topological structure of lines[6-7].

(f) Search the longest straight lines along the radius and then approximately express the lines in broken lines [8].

In this paper we choose the run length code as a primary expression of line drawings for it may contain all information and is easy to extract skeletons of lines. We propose a statistic vectorization method which may eliminate noises, improve the precision and speed of recognition.

2 Image Coding

To reduce the loss in image information and economize the storage, a compressive version of image coding, called Modified Run Length Code (MRLC), has been accepted in this work. The run is defined to be a black pixel interval isolated by white pixels in a row. Simple adjacent runs on neighboring rows compose a primitive. Primitives keep all information of the image and are convenient for extracting the geometric features of drawings as well. Generally speaking, a primitive is one segment of a line or arc as shown in figure 1. However, if a line direction is close to horizontal, the coding may be hard for later processing. Therefore, it should be coded with vertical runs. The global horizontal scanning combining with conditional local vertical scanning is our basic strategy to generate line primitives in the image coding.

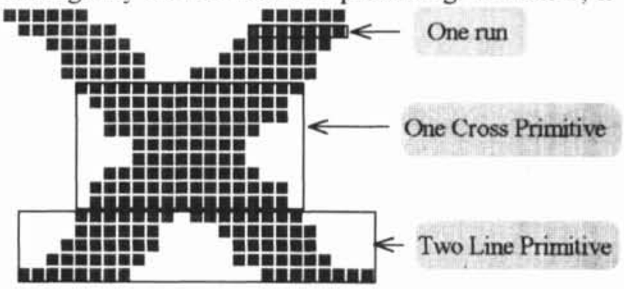


Figure 1. Modified Run Length Coding.

3 Broken Line Recognition with Chord Height Measurement

* Address: Department of Computer and System Science, Nankai University, Tianjin, P.R. China 300071.
E-mail: **linyj@centlab.nankai.edu.cn**

The line recognition is implemented on primitives. That a vector represents one line or one arc is expected at the initial time. However, a broken line, which is regarded as one primitive according to the definition of primitive, must be cut into several straight line primitives before vectorization. An algorithm using chord height measurement to identify a broken line is described as shown in figure 2.

(1) Find the middle point of each run as a skeleton point of this primitive.

(2) Draw a line *ab* between *a* and *b*, ends of the skeleton. L_{ab} denotes the length of line *ab*.

(3) Calculate the distance from each skeleton point of the primitive to line *ab*. Suppose that the maximum distance *cd* is D_{max}.

(4) The primitive should be cut at point c, if following conditions are satisfied:

$$D_{max} > T_D \quad \text{and} \quad L_{ab} > T_L$$

where T_D is a distance threshold and T_L is a length threshold.

(5) If the primitive is cut, then jump to (2) to measure if a further splitting for primitives *ca and cb* is needed; else exit.

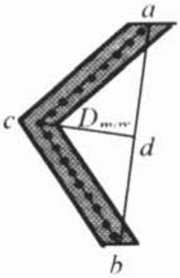


Figure 2.Broken line detection with chord height measurement .

4 Statistic Recognition of Straight Line with Triplets

4.1 Single Straight Line Recognition

To eliminate noises efficiently, a statistic algorithm has been accepted for extracting a straight line or arc from skeleton points of its primitive,

(1) Divide the skeleton into three sections evenly and then group corresponding skeleton points of each sections into *n* triplets $\{(\alpha_i, \beta_i, \gamma_i), i=1...n)\}$, as shown in figure 3. Suppose $(X(\alpha_i), Y(\alpha_i))$, $(X(\beta_i), Y(\beta_i))$, $((X(\gamma_i), Y(\gamma_i))$ to be coordinates of the triplet $(\alpha_i, \beta_i, \gamma_i)$. The distance on *y* axis from α_i to β_i is equal to that from β_i to γ_i in the triplet, that is $\Delta Y = Y(\alpha_i) - Y(\beta_i) = Y(\beta_i) - Y(\gamma_i)$.

(2) If the following condition is satisfied

$$2 * X(\beta_i) - (X(\alpha_i) + X(\gamma_i)) < T_d,$$

then points of the triplet are considered to be collinear, where T_d is a distance threshold. If more than $n/2$ triplets are collinear, then the primitive can be recognized as a straight line and go to (3); else exit.

(3) Suppose that there are *N* collinear triplets, the slope *k* of straight line can be calculated by the following equation

$$k = \sum_{i=1}^{N} (X(\gamma_{m_i}) - X(\alpha_{m_i})) / (2 * \Delta Y * N)$$

(4) The best collinear triplets are chosen as reference points to calculate ends of the straight line.

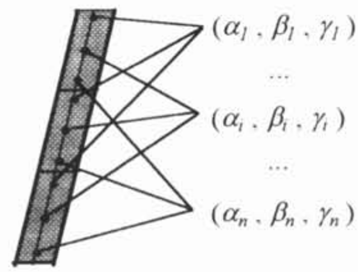


Figure 3.Single straight line recognition with triplets.

The conventional LMSE(Lest Mean Square Error) algorithm may extract the straight line as well by solving a linear equation group using all skeleton points without sieving noise points, so the performances of LMSE on precision and speed of recognition are not so good as algorithm above.

4.2 Merging Straight Lines on Maximum Extension Principle

A straight line crossing with other lines or arcs will be coded as several line primitives and crossing primitives at image coding phase. A long straight line will be cut into many short line segments after initial vectorization. In order to merge these line segments, maximum extension principle is adopted with consideration of the global heuristics of human vision. A straight line segment may be extended in both directions as the following steps:

Suppose the first straight line primitive has been recognized and is denoted as *CurPrimitive*. Try to extend it upwards.

(1) Get an upper adjacent primitive *AdjPrimitive of CurPrimitive*; if none, then exit; else go to (2);

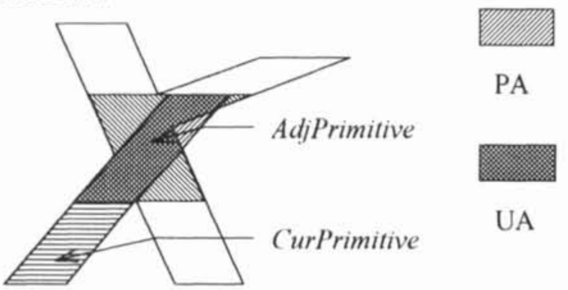
(2) If *AdjPrimitive* has been recognized as a straight line primitive then go to (3), else go to (4);

(3) As shown in figure 4(b), if the slope and the line width of *AdjPrimitive* are similar to those of *CurPrimitive* and the distance between their end points is less than the line width, then the line *CurPrimitive* is extended upwards, and set *CurPrimitive* = *AdjPrimitive* and go to (1) ; else go to (5) ;

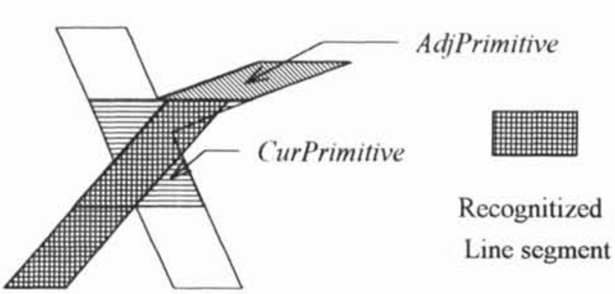
(4) As shown in figure 4(a),suppose the number of runs of *AdjPrimitive* is L_{Adj}. , the predicted area PA is line width * L_{Adj} and the union area UA is union of the extension of *CurPrimitive* with *AdjPrimitive*. If UA is lager than 2/3 PA , then extend *CurPrimitive* upwards and set *CurPrimitive* = *AdjPrimitive* , go to (1); else go to (5);

(5) Get next upper adjacent primitive of *Curprimitive* and set it to be *AdjPrimitive* , if none then exit; else go to (2);

The downward extension is similar to upward extension.



(a). Upword strenching through a corss primitive.



(b). No upword strenched line with different *slope* and *width*.

Figure 4. Upword strench line .

5 Statistic Recognition of Arc with Triplets

5.1 Single arc recognition

With the same consideration as for recognizing straight lines, the statistics has been used in this section for recognizing arcs.

(1) Get two end points and the middle point on the primitive skeleton. If these points are collinear, then exit; else calculate the center of the arc over the three points to determine the concave side of the arc.

(2) Get the points on the concave side of the arc and evenly divide the contour into three sections; corresponding points in each section compose *n* triplets $\{(\alpha_i, \beta_i, \gamma_i), i=1...n\}$ as shown in figure 5.

(3) Calculate the center c_i and radius r_i (i=1...n)of each triplet.

(4) Calculate mean radius $R=\sum_{i=1}^{n} r_i \Big/ n$, mean center $C=\sum_{i=1}^{n} c_i \Big/ n$ and deviation of centers $\Delta c_i = |c_i - C|$ (i=*1...n*).

(5) Set threshold T_R = $R/3$. If the number of Δc_i which are less than T_R is more than $n/2$, then the primitive can be recognized as an arc.

(6) Calculate C and R of the arc again by using selected triplets that satisfy threshold condition. Cut the arc into three pieces evenly and calculate the area S of the middle piece. The quotient of dividing S by 1/3 arc length approximately is the width of arc.

There are many arc detection algorithms, the most representative ones are Hough transform, Thomas algorithm and Landau iteration algorithm. The ability of anti-noise of Hough transform is stronger than the one of others. Although the Hough transform can get better result, the computing time limits its usage for huge data processing, as the case in our project.

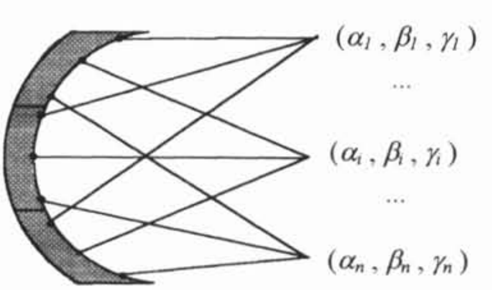


Figure 5.Single arc statistic recognition with triplets.

5.2 Merging Arcs

If the primitive of a circle or an arc intersects other primitives, it will be divided into many pieces (including crossing primitives). After short arcs have been recognized individually, they should be linked together to get a potential circle or arc. The algorithm can be described as follows:

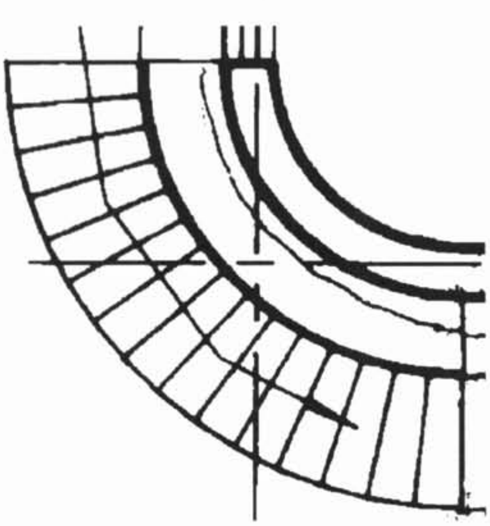
(1) Cluster individual arcs into k classes according to parameters (center C and radius R).
(2) Calculate the center C and radius R of each class by using data sampling from the concave side of arcs.
(3) Try to extend arcs. Suppose extended part to be denoted as EP and the union area of EP with unrecognized primitive to be denoted as UA. If UA is larger than $4/5$ EP, then the arc is extended.
(4) After extension, if the radian $\geq 2\pi$, then this group of arcs can be recognized as a circle; else they have been merged into several longer arcs.

6 Summary

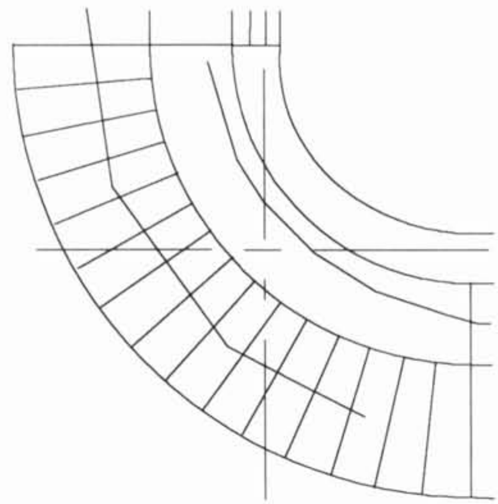
The merit of this work can be briefly presented as follows:

(a) The MRLC preserves all graphic information, and is convenient to extract geometric features of line drawings .
(b) Noises have been eliminated by statistics effectively; the precision is improved significantly.
(c) Computing time has been obviously saved by one pass operation and simple arithmetic.

An experimental result is shown in figure 6. Although it is better than thinning algorithm, it still have some difficulties: such as recognizing very short lines and locating intersection points precisely, specially locating multi-line intersections. The weakness should be improved in our further study.



(a). Original image.



(b). Result of vectorization.

Figure 6. An example of

References

[1] Nagasamy Vijia, Engineering Drawing Processing and Vectorization System , CVGIP, Vol. 49, 379-397, 1990.

[2] D.Dori, Sparse-pixel Recognition of primitives in engineering drawings, Machine Vision Applications, Vol. 6, 69-82, 1993.

[3] X.Lin, Efficient Diagram Understanding with Characteristic Pattern Detection, CVGIP, Vol. 30, 84-106, 1985.

[4] P.Vaxiviere, Celesstin: CAD Conversion of Mechanical Drawings, Computer, Vol. 25, No. 7, 46-54, 1992.

[5] C.C.Han and K.C.Fan, Skeleton Generation of Engineering Drawings via Contour Matching, Pattern Recognition, Vol. 27, No. 2, 261-275, 1994.

[6] L.Baotto, V.Consorti (IBM), An Interpretation System for Land Register Map, Vol. 25, No. 7, 25-32, 1992.

[7] Tan Jianrong and Peng Qunsheng, A Global Line Recognition to Scanned Image of Engineering Drawings Based on Graphics Constraints, Chinese J. Computers, Vol. 17, No. 8, 561-569, 1994.

[8] E. Masakazu etc., Automatic Recognition of Design Drawings and Maps, Int. Conf. On Pattern Recognition 7th, Montreal USA, 1296-1305, 1991.