Exact Representation of Glyph Contours and its Application in Text Vectorization

Tao He, Jie Feng, Bingfeng Zhou, Institute of Computer Science and Technology, Peking University, Beijing, China

Abstract

Contour extraction from binary image is important for many image processing applications, e.g., scanned document image vectorization, object segmentation, pattern recognition, automatic content interpretation of handwritten documents and AutoCAD drawings, hand-drawn cartoon animation etc. The proposed contour tracing algorithm generates exact representation for contours, which has the following features: Firstly, it can effectively deal with one-pixel width glyphs, as well as self-intersection glyphs. Secondly, during contour tracing the algorithm can associate the outer and inner (i.e., the letter 'B' has one outer and two inner contours) contours with the corresponding object regions, which favors subsequent manipulation of the object. Third, compared with traditional 8 or 4-neighbour connected contour tracing, our methods can exactly capture the glyph shape and size information, which is important in preserving the fidelity of objects. Lastly, the algorithm is concise and easy to implement.

I. Introduction

This paper relates to a computer-based system that scans documents into bitmaps and vectorizes them. More specifically, it serves as a required part of a raster-to-vector conversion system that transforms texts in a document image into vector format.

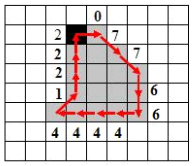
Scanned documents are usually in pixel-map format, a common A4 (8.27 by 11.69 inch) document scanned at 300 dpi can take up 8.7 million bits. Instead, a vector representation can significantly reduce the file size, as well as facilitate following manipulations. Thus it has a broad range of applications, such as displaying on portable devices with various resolutions, interactive image processing, online web graphics manipulations and so on. To vectorize text object (character) in a scanned document image, one must obtain an exact representation of the text object at the first place.

Objects can be represented in various forms depending on its later-on usage. A comprehensive survey of object representation and shape description can be found in [1]. Here we consider the contour-based methods for object representation, because it is usually a thinned point set that captures the extent and geometry of a given object, which could considerably economize the representation of object information without compromise. There have been a wide range of methods existing for contour extraction from binary images [2-8]. None of these generates exact representation of object contours except for Collins [5] and Skourikhine [7], which, however, need a two-pass process that marks up the edge pixels at the first run.

In this paper, we propose a new contour extraction algorithm improved on the chain-code representation, which was pioneered by Freeman [2, 3]. Chain-code is an image representation method based on region boundaries. It follows the boundary in counter-clockwise (or clockwise) manner and keeps track of the direction as we go from one contour pixel to the next. The chain-code of a region is determined by specifying a starting pixel and the sequence of unit vectors obtained from going either 4-connected or 8-connected paths along the boundary. Figure 1a illustrates a typical Freeman Chain code example, and Figure 1b shows a contour in Vertex chain code [4]. In many applications, this kind of representation is sufficient, for it captures the overall shape outline of the object and has a considerable size reduction compared with other representations. But, it falls short of the aims in the scenario of text vectorization on two points.

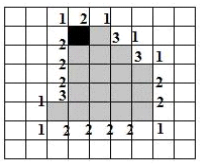
1. Character glyph of scanned text with one-pixel-width contour is a commonplace, where chain-code representation cannot properly handle the case. The outer-loop and inner-loop of one-pixel-width contour coincide with each other, and miss the ink-art of a character glyph.

2. Characters with self-intersections cannot be well captured by chain-code representation. The outer loop of such a glyph can interfere with itself.



FCC: 0776644441222

Figure 1a. Freeman Chain Code (FCC) example



VCC: 12131312212222113222

Figure 1b. Vertex Chain Code (VCC) example

Figure 2a gives an intuitive illustration of the aforementioned problems. With the pixel-based contour extraction algorithm, contours that should have an outer loop and an inner loop can only be detected with a single loop; while contours that should have only one outer loop could be associated with two loops because of the self-intersection property of the contour. Our proposed exact representation of contour glyphs can handle these cases and generate reasonable results. Figure 2b shows the theoretical results of applying our new contour

tracing algorithm to single-pixel-width contours and self-intersection contours.

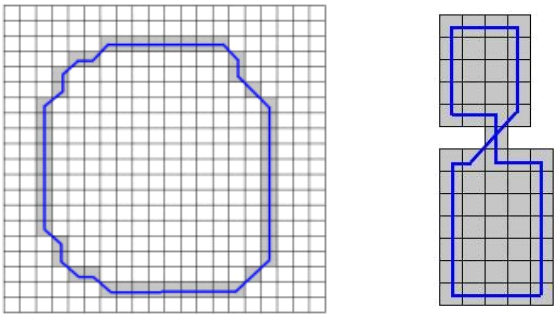


Figure 2a. Pixel-based chain-code representation of single-pixel width contour (Left) and self-intersection contour (Right)

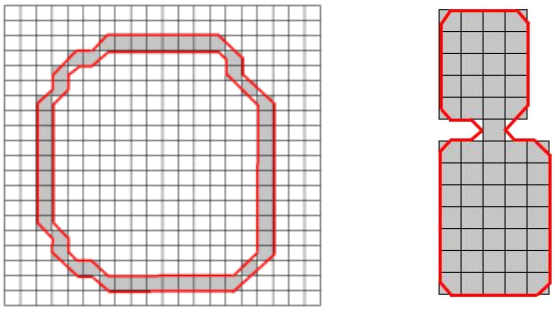


Figure 2b. Crack-based chain-code representation of single-pixel width contour (Left) and self-intersection contour (Right)

The effectiveness of the algorithm can be attributed to the way in which contours are traced. Instead of locating centers of boundary pixels along 8-connected or 4-connected paths as contour points like many freeman chain codes and its extensions do, our algorithm locates sequential contour points on mid-points of each crack along the tracing route. A crack is defined as the transitional edge between background and foreground pixels with direction information specified by the traverse route. This method of representation can avoid the self-intersection of contours resulting from revisiting pixels in thin regions. In fact, our algorithm can be applied to the extraction of object contours without ambiguities, even if the object has thin regions of just one pixel width.

The paper is organized as follows. Section 2 gives a detailed description about our proposed algorithm. Section 3 provides experimental results, and in section 4 we draw some conclusions.

II. The proposed Method

The proposed contour extraction algorithm is working on binarized document image. When the first foreground pixel of the object is identified, the algorithm traces the outer loop around objects contour and the inner loops around holes within the object. The algorithm does not make any assumption about the number of loops in an object. It will extract all the loops and associate them with the object appropriately (see figure 3).

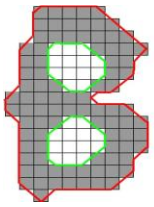


Figure 3. Letter B with one outer loop an d two inner loops

2.1 Conventions

To make it easier to describe the algorithm, we first give some illustrations about image processing conventions. An exemplary image coordinate system is depicted in figure 4a. Figure 5a and figure 5b illustrate the conventions of 8-connectedness and 4-connectedness pixel neighborhoods in chain code representation. The scanning order of an image is often from left to right, top to bottom, as illustrated in figure 4b. The naming conventions around the current pixel P is described in figure 6. Black square represents the current pixel. L and R denote pixels according to the current crack direction. When we traverse along the crack direction, the pixel lying on the left and right of the current crack direction is denoted by L and R respectively.

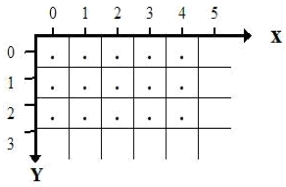


Figure 4a. Typical image coordinate system

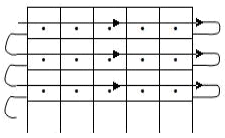


Figure 4b. Typical image data scanning order, from top to bottom, left to right

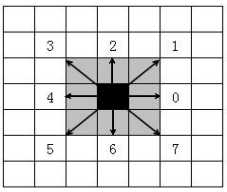


Figure 5a. Chain code convention: 8-connected pixel neighbourhoods

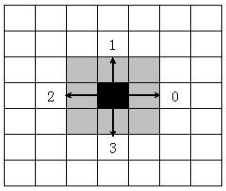


Figure 5b. Chain code convention: 4-connected pixel neighbourhoods

2.2 Details of the algorithm

Chain code contour tracing method often works as follows. It begins with finding a start pixel, as the black pixel depicted in figure 1. Then, an initialization step is performed. Different contour tracing algorithms have different initializations. It depends, for example, on whether the tracing is based on 4 or 8-neighborhoods, or on the coding scheme of the tracing algorithm. After that, the algorithm traces contour next point one by one along a traverse route. Tracing the next contour point is the key to the chain code method family.

The main difference between our proposed method and traditional freeman-chain-code-like methods lies in the "Tracing next contour point" step. In freeman chain code like contour extraction methods, the tracing of next contour point is simply based on the connectedness of 4-/8-connected neighborhoods, and the center of boundary pixel is marked as contour point.

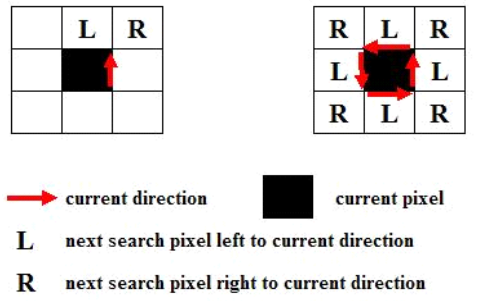


Figure 6. Naming conventions for neighboring pixels around the current pixel under the current crack direction

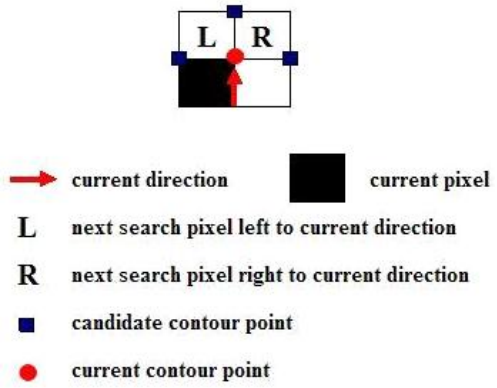


Figure 7. Illustration of the contour point tracing. Red circle and arrow depict current contour point and current crack direction. Next possible contour point can only appear in three places, as marked by blue rectangles.

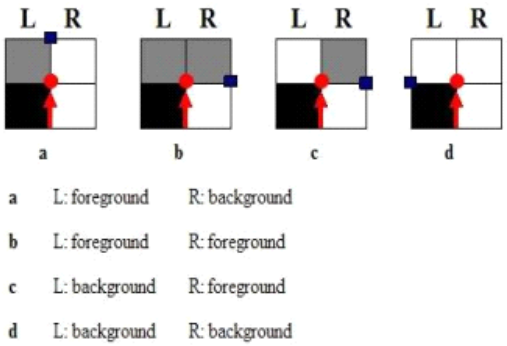


Figure 8. Determination of next contour point according to current pixel, current crack direction, L and R pixels. a, b, c, d show four cases respectively

Our method treats the boundary pixel as a rectangle, and the contour points lie on the mid-point of rectangle edges. The algorithm begins with finding the start pixel. Then in the initialization step, current pixel (CP) is set to the start pixel. Current crack (CC) is set to downward. Current contour point (CCP) is set to the mid-point of the transitional edge of the start pixel. The transitional edge is the crack that separates background pixel and the start pixel. For convenience, we depict at the figures the corner-point of a rectangle as contour points.

Tracing the next contour point is the core to our proposed algorithm. Walking along the CC direction, we first check the L and R pixels as mentioned in figure 6, and then extract next contour points according to the L and R pixels. After adding a contour point, CC direction and current pixel (CP) are modified accordingly.

As we can see in figure 7, the CC direction is upward. Current pixel (CP) is represented by Black Square. Current contour point (CCP) and next candidate contour point are denoted by red circle and blue rectangle respectively. Considering 8-connected neighborhoods, the determination of next contour point is shown in figure 8. The CC direction is then changed accordingly as a vector from CCP to next contour point. If we are considering 4-connected neighborhoods, the tracing route is a little different. Figure 9 shows the difference between 8-connectedness and 4-connectedness neighborhoods. Though it is a foreground pixel, R is also a diagonal pixel of CP that is not taken into consideration in 4-connectedness neighborhoods.

Figure 9. Determination of next contour point in case of L (background), R (foreground). Next contour point is different under 4-connected neighbourhoods(Left) and 8-connected neighbourhoods(Right)

III. Experimental Results

We have built a practical vectorization system that comprises our proposed contour tracing algorithm. The system takes a binary image as input, and then extracts text objects. After that, the boundaries of an object are exactly represented. A fitting process is then applied to these boundaries to generate a vectorized representation of the objects in the input image.

For characters, glyph is represented by an outer loop and several inner loops. Figure 3 shows an example of the letter 'B'. The ink-art for a character stroke is determined by the outer loop and inner loops. So, the fidelity of a vectorized character is determined by the precision of contour representation.

The proposed method can generate exact representation of contours even with only one-pixel width. Compared with traditional Freeman Chain-Code method, our exact representation method can lead to much better vectorization results. Figure 10 give some characters that are vectorized under freeman chain code representation and under the proposed crack-based exact contour representations. The first row shows the results of characters represented by Freeman chain code. Note

the abnormally "thin" stroke in 'a', 'e', and 'g'. The second row shows the results using our proposed algorithm and there is no such abnormal appearance. This improved ink-art reconstruction is due to the exact representation of glyph contours. Instead of marking up boundary pixels and locating centers of consecutive boundary pixels as contour points as in [5], the proposed method obtains exact representation of object contours by tracing the transitional edges between background and foreground pixels and locates the mid-point of the transitional edge as contour point.

Figure 10. Comparison of Freeman chain code representation and our exact representation applied in text vectorization.

IV. Conclusion

This paper proposed a new contour tracing algorithm. It can generate exact representations of object contours, even the contours are only one pixel wide. It is of great use in text image vectorization where such single-pixel width contours are commonplace. In contrast with other algorithms that extract pixel-based contours, and thus allow problematic situations such as contour self-intersection, our algorithm guarantees extraction of non-intersecting contours. This is achieved by treating pixels as rectangles and tracing mid-points of boundary pixel transition edges instead of tracing centers of boundary pixels. Therefore, the algorithm extracts sub-pixel contour. Our algorithm has several attractive properties.

(1). It avoids retracing of pixels in self-intersection contours;

(2). It can exactly represent object boundaries even in single-pixel wide cases;

(3). It maintains desired connectivity of object regions as specified by 8-neighbour or 4-neighbour connectivity of adjacent pixels;

(4). During contour tracing, the outer and inner loops of a text object can be associated appropriately, and the ink-art can then be reconstructed accordingly as shown in experiment result;

(5). The contour tracing algorithm is easy to understand and implement. It has a time complexity that is dominantly linear to the number of contour points.

Its low time complexity enables it a candidate for interactive image analysis and for on-line web graphics manipulation. The capability to precisely represent object contours even in case of single-pixel width boundary and self-intersection boundary makes the algorithm an effective alternative for traditional freeman chain code representations. In fact, it can serve as an in-place substitution for any applications that involve freeman chain code shape representation.

Acknowledgements

This work is supported by the National Natural Science Foundation of China. (Grant NO. 60973054)

References

1. D. Zhang, G. Lu, Review of Shape Representation and Description Techniques, Pattern Recognition, Vol. 37, No. 1., pp. 1-19(2004).
2. H. Freeman, on the Encoding of Arbitrary Geometric Configurations, in Proceedings of IRE Translation Electron Computer, pp.260-268, New York (1961).
3. H. Freeman, Computer processing of line drawing images, Computational Surveys, vol. 6, pp. 57-97(1974).
4. E.Bribiesca, a New Chain Code, Pattern Recognition, vol, 32, pp. 235-251(1999).
5. R.J. Collins, Drawing Pixmap to Vector Conversion, US Patent 6173075 (2001).
6. M. Ren, J. Yang, and H. Sun, Tracing boundary contours in a binary image, Image and Vision Computing, Vol. 20, pp.125-131(2002).
7. A.N. Skourikhine, Dilated contour extraction and component labeling algorithm for object vector representation, Proc. of SPIE vol. 5916(2005).
8. L.A. Wulandhari, H. Haron, the Evolution and Trend of Chain Code Scheme, ICGST-GVIP 8(III), 17–23 (2008).

Biography

Tao He received his B.E. degree in Software Engineer from Harbin Institute of Technology, China in 2007. He is currently pursuing a M.S. degree in Computer Applications at Peking University, China. His research interest is in document image processing, including binarization, object segmentation, corner detection and image vectorization.

Jie Feng is an assistant researcher of the Institute of Computer Science & Technology at Peking University. She earned her PH.D. from the Center for Information Science at Peking University in 2005. Her research interests include 3D mesh model compressing, multi-resolution modeling, image-based modeling and relighting.

Bingfeng Zhou, Ph.D., professor, Ph.D. supervisor. His research interests include graphics simulation of robot kinematics, geometry models and CAD/CAM, color image processing, multimedia system and image special effects, digital image halftone, image based rendering and modeling, virtual reality, etc.