

Improving Arc Detection in Graphics Recognition

Philippe Dosch
LORIA-Université de Nancy 2*
dosch@loria.fr

Gérald Masini
LORIA-CNRS*
masini@loria.fr

Karl Tombre
LORIA-INPL*
tombre@loria.fr

Abstract

In the context of graphics recognition, arc detection consists in the extraction of circles and arcs from the image of a graphics document or from the segments yielded by its vectorization. Several methods have been proposed for this purpose, and we briefly survey them in this paper. Then, we describe an improved algorithm inspired by two existing methods, and including a fitting step for a better precision.

1. Introduction

Graphics recognition techniques are slowly maturing—at least the low-level image processing, segmentation and vectorization steps—and emphasis has been put on *robust* and *stable* methods, which can be implemented as a set of stable software components, reusable from one application to the other [3, 7]. A central aspect in graphics recognition is *vectorization*, *i.e.* the raster-to-graphics conversion process [8]. To be complete and useful for higher-level recognition and analysis phases, vectorization should not be limited to the recognition of straight line primitives, but should at least include a reliable circular arc detection process. It may actually cover even more than that, as higher-level processes often need to work on an extended set of graphical primitives, such as dashed lines, cross-hatched areas, etc., to provide useful results.

After having designed with great care a stable and robust vectorization process [8], we therefore turned to the task of reliably recognizing arcs. As we have done in the past [7], our aim was not necessarily to design a new and “flashy” method, but to reuse as often as possible the best approaches from the field. Therefore, we started our work by combining the best of two approaches, Rosin & West’s edge segmentation method [6] and Dori’s vector-based arc segmentation [1]. However, we found it necessary to add a fitting process to better adjust a detected arc with the pixels of its skeleton in the image.

In this paper, after a brief overview of the methods on which we base our approach, and a description of their limitations (§ 2), we present the method we have designed and the guidelines we followed (§ 3). We conclude with some results and perspectives (§ 4).

2. The Base of our Approach

According to Dori [1], there are two main families of arc detection methods. The methods of the first family are based on the Hough transform and directly work on the original pixels of the graphics image. Such a technique is well-known and proves to be quite robust in the presence of noise. However, it is computationally expensive and does not provide enough accuracy in the localization of the center and end points of the detected arcs. This stems from the very low level at which the information is processed.

The second family of methods works on chains of points, or on segments yielded by the polygonal approximation of such chains. The basic idea is to compute an estimation of the curvature for these chains. This approach typically is what we are looking for, as our vectorization process is based on the computation of a distance skeleton. The pixels of the skeleton are then linked together to form chains, and a polygonal approximation converts the chains into straight line segments. Instead of discarding the chains after that, we have taken the option to keep them, so that they can be used by the fitting process.

Rosin & West [6] propose a method based on recursive splitting, for segmenting a curve into a set of arcs and segments. It is an extension of a previous work by Lowe [4] to find points of maximum curvature (Fig. 1). Such a point is computed using a ratio between the maximum deviation and the length of the approximating segments. Whereas Lowe uses this feature to approximate a chain by a set of segments, Rosin & West add the recognition of arcs, when an arc is a better approximation of the original chain.

As an arc requires more parameters than a straight segment, the simple ratio measure mentioned above is not enough to characterize an arc. Therefore, Rosin & West take connectivity (arcs are supposed to start and end at the

* Common address: LORIA, B.P. 239, 54506 Vandœuvre-lès-Nancy Cedex, France

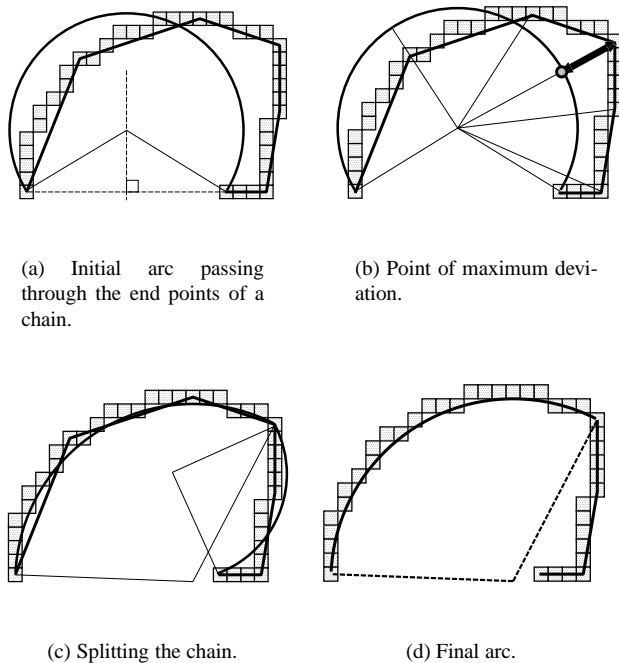


Figure 1. Principle of Rosin & West's method.

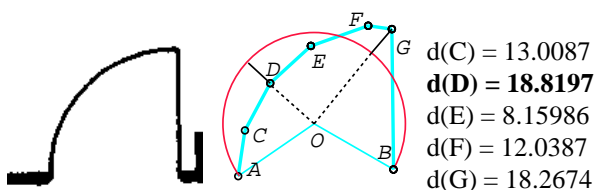


Figure 2. Example of inadequate splitting at a maximum deviation point (D).

extremities of the segments of the polygonal approximation) and geometry (the center of the arc must be equidistant to both extremities, which strongly constrains its position) into account. In this way, the position of an arc can be computed through simple least squares minimization.

This method is very interesting, as it does not require any explicit threshold. We actually use it for our polygonal approximation because of this very reason. Moreover, although the significance measure, *i.e.* the ratio $deviation / length$, may be considered to be too simplistic, Rosin has proposed other significance measures which may further improve the method [5].

Nevertheless, the method has also its limitations. The initial list of points is split at the point of maximum deviation, and arc detection is performed again on each sublist. In some cases, as illustrated by figure 2, the maximum deviation point is not the most relevant one and the subsequent

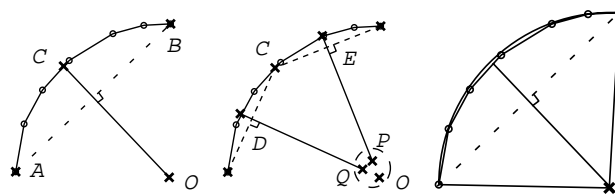


Figure 3. Dori's method for computing the center of a potential arc. P (resp. Q) is the center of sub-arc CB (resp. AC).

splitting does not lead to a correct recognition.

Dov Dori and his team have designed another method, called SRAS¹ [1], which works iteratively. A sparse-pixel vectorization of the document [2] extracts so-called *bars*, that are grouped to form polylines. These polylines in turn are used as *keys* in the arc recognition process. Thresholds are applied to select bars which are not too short and not too long, and which form pairs with the right angular orientation. A first position of the center of the arc can then be computed. This position is refined by using all the vertices of the polyline involved in the arc hypothesis. A potential arc center area is thus determined, and each pixel of the area is tested, using an averaged square distance to optimize the position of the center (Fig. 3). After the validation of this initial arc hypothesis, the algorithm tries to stepwise extend the arc at its extremities, by searching in potential extension areas and testing the arc hypothesis again, in terms of width, angular measures and polyline continuity. For each possible extension, the new center is computed, according to the principles described previously.

The method yields good results, but is limited by the use of a certain number of thresholds, in particular to determine the center of the arc. Dori's and Rosin & West's methods both share another limitation: The arc hypotheses and the computation of the error are based on the polygonal approximation of the graphic image. Although the approximation is very useful to find the right hypotheses in an efficient way and without losing the connectivity, it leads to uncontrolled location errors when referring back to the original image. This explains our perceived need for a fitting step, to make arc hypotheses match their pixel representations.

3. Fitting Arc Hypotheses to the Skeleton

The method we designed is basically inspired by that of Rosin & West, but we included two ideas from Dori's method: The way to compute the center of the arc, and the use of polylines instead of simple segments. We also added some improvements. The most important of them concerns the computation of the error associated with an arc hypothe-

¹Stepwise Recovery Arc Segmentation Algorithm.

esis: It is no longer performed with respect to the polygonal approximation, but with respect to the original chain of skeleton pixels. In fact, each set of segments delivered by the polygonal approximation step of our vectorization process is associated with the pixel chain that the segments approximate. Segments are grouped into polylines, each polyline being the approximation of a complete chain. The original linked chain corresponding to a polyline can then be retrieved using a simple index.

Our arc detection algorithm works in two phases: Arc hypotheses generation and validation of the hypotheses. The hypotheses are built from the polygonal approximation. Let (S_1, \dots, S_n) be a chain of connected segments, described by their extremities (P_1, \dots, P_{n+1}) , such that:

- It contains at least four points (as there is always a possible arc passing through three points),
- the successive angles $\widehat{S_i S_{i+1}}$ and $\widehat{S_{i+1} S_{i+2}}$ are quite the same.

Such a chain is retained as an hypothesis to be examined by the arc detection process. If the chain as a whole cannot be considered as representing an arc, a segment is removed at one of the extremities of the chain, and the chain is tested again, until a valid arc is found or until there are too few points for a pertinent hypothesis. The test phase is performed using Rosin & West's least squares minimization approach. The error is not estimated using the segments of the polygonal approximation, but using the subchains of points, which can be retrieved thanks to our indexing structure, as previously mentioned.

The method also detects full circles. When working on a closed loop of successive segments, one of the segments is eliminated before applying arc detection. If a unique arc, including all the segments, is detected, the presence of a circle is tested by checking the validity of the last segment.

4. Results and Conclusion

In the fall of September, the method participated in the Third IAPR Graphics Recognition contest, where complete vectorization methods are run on ground-truthed data. At the time of writing this paper, we are still awaiting the performance evaluation results on these data.

Figure 4 illustrates results obtained from a rather simple architectural drawing. With Rosin & West's raw method, false arcs are detected due to chains of short segments provided by the vectorization, in particular around junction points (Fig. 4b). These arcs disappear when using our improved method, and full circles are correctly extracted (Fig. 4c). Arc location is also more accurate (Fig. 4d), although it is not plainly emphasized by the figure (drawings should be displayed at a larger scale).

There are still several possible improvements to the method. One of them is to test arc hypotheses on more than one polyline, as the skeleton linking algorithm starts new chains at each junction. This would lead to the possibility of recognizing a single arc, even when it is crossed by another line, or to recognize two full arcs whenever they share short segments like those pointed by dotted arrows on figure 4d. The main difficulty here does not concern the method, but the computational complexity of the implementation.

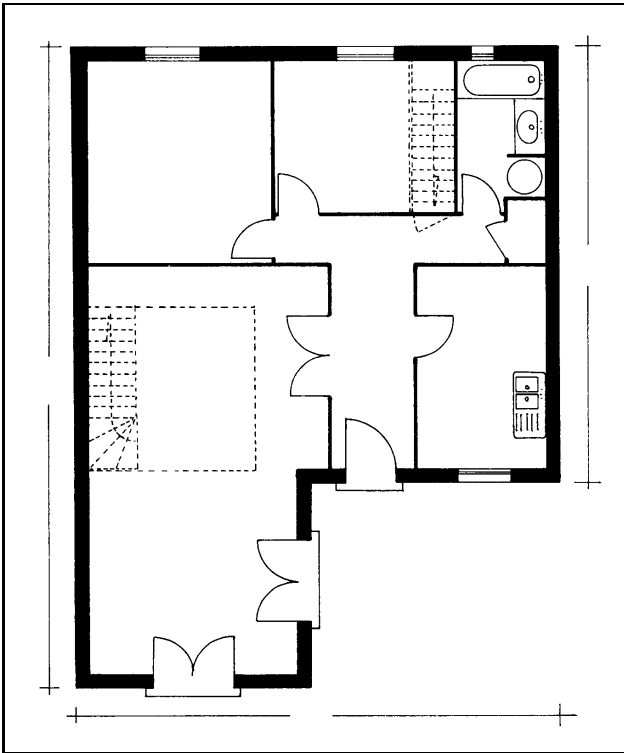
We also still have thresholds in the method, especially for the similarity between two angular measures. A possible improvement would be to extend Rosin & West's work to define relevant significance measures for arcs.

Acknowledgments

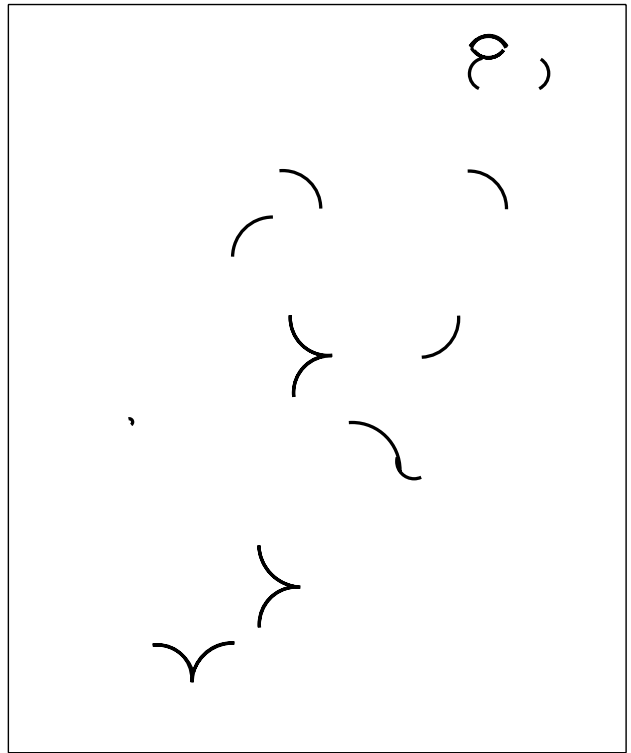
We would like to thank Wissam Dagher, Nicolas Lieber, Antoine Sorba and Séverin Voisin, who participated in the implementation work for a large part.

References

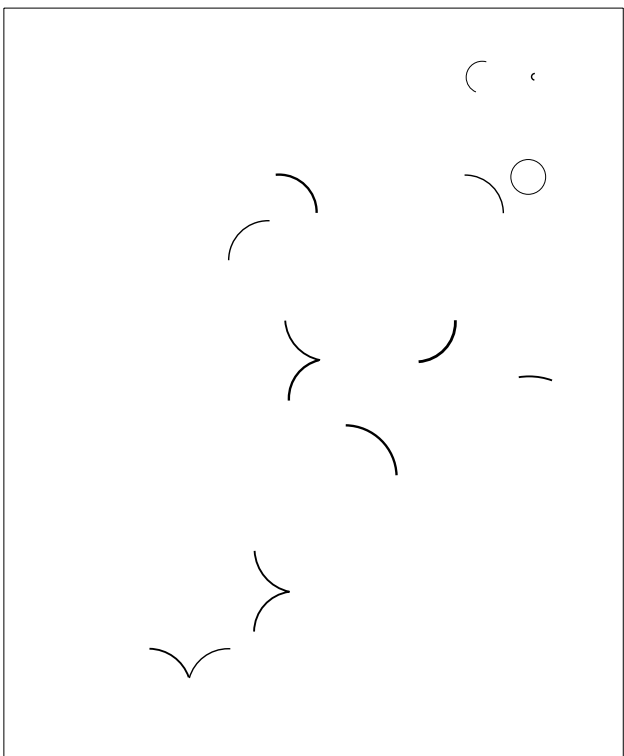
- [1] D. Dori and W. Liu. Stepwise recovery of arc segmentation in complex line environments. *International Journal on Document Analysis and Recognition*, 1(1):62–71, Feb. 1998.
- [2] D. Dori and W. Liu. Sparse Pixel Vectorization: An Algorithm and Its Performance Evaluation. *IEEE Transactions on PAMI*, 21(3):202–215, Mar. 1999.
- [3] P. Dosch, C. Ah-Soon, G. Masini, G. Sánchez, and K. Tombre. Design of an Integrated Environment for the Automated Analysis of Architectural Drawings. In S.-W. Lee and Y. Nakano, editors, *Document Analysis Systems: Theory and Practice. Selected papers from Third IAPR Workshop, DAS'98, Nagano, Japan, November 4–6, 1998, in revised version*, Lecture Notes in Computer Science 1655, pages 295–309. Springer-Verlag, Berlin, 1999.
- [4] D. Lowe. Three-Dimensional Object Recognition from Single Two-Dimensional Images. *Artificial Intelligence*, 31:355–395, 1987.
- [5] P. L. Rosin. Techniques for Assessing Polygonal Approximation of Curves. *IEEE Transactions on PAMI*, 19(6):659–666, June 1997.
- [6] P. L. Rosin and G. A. West. Segmentation of Edges into Lines and Arcs. *Image and Vision Computing*, 7(2):109–114, May 1989.
- [7] K. Tombre, C. Ah-Soon, P. Dosch, A. Hated, and G. Masini. Stable, Robust and Off-the-Shelf Methods for Graphics Recognition. In *Proceedings of the 14th International Conference on Pattern Recognition, Brisbane (Australia)*, pages 406–408, Aug. 1998.
- [8] K. Tombre, C. Ah-Soon, P. Dosch, G. Masini, and S. Tabbone. Stable and Robust Vectorization: How to Make the Right Choices. In *Proceedings of 3rd International Workshop on Graphics Recognition, Jaipur (India)*, pages 3–16, Sept. 1999. Revised version to appear in a forthcoming LNCS volume.



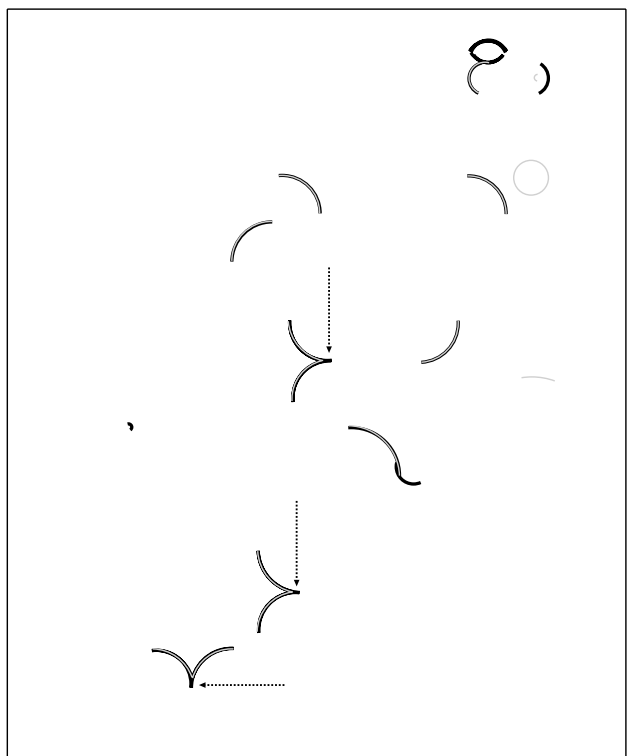
(a) Original image.



(b) Arc detection using Rosin & West's raw method.



(c) Arc detection using our method.



(d) Superposition of both results.

Figure 4. Results of arc detection.