Robust Circle Detection

Bart Lamiroy, Olivier Gaucher, Laurent Fritz

Robust Circle Detection

Bart Lamiroy, Olivier Gaucher and Laurent Fritz
Nancy Université – INPL – LORIA
Équipe Qgar – Bât. B
Campus Scientifique – BP 239 – 54506 Vandoeuvre-lès-Nancy Cedex – France
Bart.Lamiroy@loria.fr

Abstract

In this paper we present a method of robustly detect circles in a line drawing image. The method is fast, robust and very reliable, and is capable of assessing the quality of its detection. It is based on Random Sample Consensus minimization, and uses techniques that are inspired from object tracking in image sequences.

Note : some details of the illustrations in this paper are in colour. Reading them on a greylevel printout will reduce their intelligibility.

1. Introduction

Graphical document interpretation or symbol recognition often requires for segmenting or locating symbols, parts of symbols or forms in complex images. Finding circles is one of those issues [6, 1, 13, 8, 10]. The main difficulty with the existing approaches is that they often are of considerable complexity (e.g. Hough-like [7] or feature grouping approaches [3]) sensible to image quality, line thickness, or rely on a number of user defined parameters or thresholds that make them extremely difficult to apply to generic problems or on heterogeneous document sets.

The approach developed in this paper reduces the set of needed parameters to a minimal set of very elementary and visually significant values and can be applied without prior knowledge of the document set, regardless of line widths, connectedness or complexity. It relies on elementary (3,4)-distance transform skeletonization [12] and circular arc detection [9].

The following section establishes how to determine if a single circle is present, provided we have a rough initial guess of its position, how to robustly detect and locate it using RANSAC (Random Sample Consensus [4]). Section 3 then explains how to generalize to detecting and localizing any number of circles, without *a priori* knowledge of their position. The last two sections conclude by eliminating spurious detections and by establishing the limits of the approach.

2. How To Determine the Presence of a Circle

In this section we address the problem of detecting a circle, given an initial estimate of its center (x_c, y_c) and its radius σ. This estimate, as we shall see further, can be very approximate. The main goal, in this first stage, is to detect whether or not, a circle is present in the image, near the vicinity of the given center (x_c, y_c).

2.1 General Algorithm

The general approach we develop consists in taking the set $\mathcal{P} = \{p_i\}$ of all pixels p_i lying on the discrete circle $\mathcal{C}^0$ defined by (x_c, y_c) and σ. For each of these pixels p_i we define the discrete line Δ_i, starting at (x_c, y_c), and passing through p_i. Let q_i be the pixel on Δ_i that is the closest to p_i and that is black. Let $\mathcal{Q}^0 = \{q_i\}$. $\mathcal{Q}^0$ therefore is the set of all black pixels closest to the initial estimate $\mathcal{C}^0$ in the direction of the circle radius.

Now, let $\mathcal{C}^1$ be the best fitting circle over $\mathcal{Q}^0$ (any criterion can be used, but we are using the Least Median of Squares – LMedS [11], global Least Squares are too error prone [2]), and let us generalize the previous step, such that $\mathcal{Q}^t$ contains the set of all black pixels closest to the theoretical circle $\mathcal{C}^t$ in the direction of the circle ray, and that $\mathcal{C}^{t+1}$ is the best fitting circle over $\mathcal{Q}^t$

Continuing this iteration until $\mathcal{C}^t = \mathcal{C}^{t+1}$ will yield the best estimate of the circle (if any) closest to the initial $\mathcal{C}^0$.

In the following sections we are going to detail the different steps of this general approach.

0-7695-2822-8/07 $25.00

2.2 Using RANSAC and LMedS

Since there is no guarantee that any $\mathcal{Q}^t$ may effectively contain points that form a circle, it may be extremely hazardous to use global minimization approaches (like Least Squares, for instance). It is known that these estimators are very sensible to outliers or spurious data that does not conform to the required model. Using these functions would invariably lead to degenerate convergence.

RANSAC is much better suited for fitting very noisy data – especially data containing measures that do not belong to the model that is to be estimated – The approach consists in selecting the strict minimum of data points required for estimating an instance of the model (*e.g.* three points for estimating a circle) and then computing the residual error of the other data points to this model. This is done a number of times, and the final model is the one with the lowest residual error.

More formally: let $\mathcal{Q}^t$ the set of model points. $\mathcal{Q}^t$ supposedly, and in the worst case, contains a ratio of τ outliers. Let q_n, q'_n and q''_n be three random points belonging to $\mathcal{Q}^t$, and let $\mathcal{C}_n$ be the circle defined by and passing through q_n, q'_n and q''_n. Let $\delta(\mathcal{C}, p)$ be the distance of a point p to a circle $\mathcal{C}$, and let $\text{Med}_\tau(\mathcal{S})$ be the τ-quantile median value of the set $\mathcal{S}$. We then define the residual error of a set of model points $\mathcal{Q}^t$ to a circle $\mathcal{C}_n$ as

$$\text{RsdErr}\left(\mathcal{Q}^t, \mathcal{C}_n\right) = \text{Med}_\tau\left(\left\{\delta\left(\mathcal{C}_n, p\right) | p \in \mathcal{Q}^t\right\}\right)$$

RsdErr gives the maximum distance of a set of points to a circle, discarding a proportion of τ outliers.

With RANSAC we choose R random subsets of 3 points within $\mathcal{Q}^t$, each giving rise to the computation of a circle $\mathcal{C}_n$. For each subset, we compute the corresponding $\text{RsdErr}(\mathcal{Q}^t, \mathcal{C}_n)$, thus obtaining

$$\mathcal{C}^{t+1} = \underset{\mathcal{C}_n, n\in[1\ldots R]}{\text{argmin}} \left(\text{RsdErr}\left(\mathcal{Q}^t, \mathcal{C}_n\right)\right)$$

The number of required subsets can be formally deduced from both the quality of the data (expected rate of outliers τ) the dimensionality of the problem (here 6, since we need three points for estimating a circle, each point having two dimensions) and the required confidence in the result [4].

2.3 Limitations and Dependency on Initial Estimate

Using only one run for estimating the position of the best circle, corresponding to an initial estimate defined by (x_c, y_c) and σ may not be a good choice if the estimate is too far off. Figure 1 shows some of these situations. In this example we gave as initial estimate (x_c, y_c) the center of the image and σ the width of the image. The figure shows

Figure 1. Examples of RANSAC estimation with erroneous initialization

in blue the estimation of the best circle, in green, the points correctly corresponding to the circle, and in red, the points closest to the estimated circle. On the left is the initial estimate, and on the right the best fitting circle.

As can be seen, the circles are not found. We partially solve this problem by restarting the estimation process until stability of the found circle. This allows for finding the circle in the first case, but fails to find the circle in the second example, as shown in Figure 2.

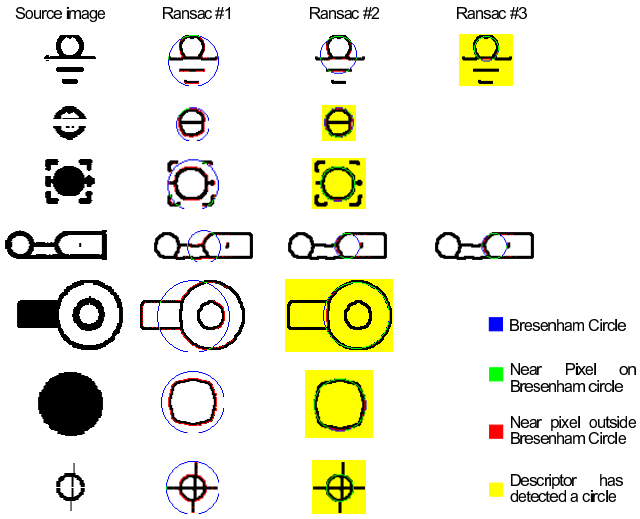

Figure 2. Examples of circle detection results with initial guess at center of image

2.4 Data Points and Selection Criteria

In our current approach we do not use the full pixel image for the circle estimation, but we rely on the image skeleton [12]. This is for two main reasons. The first and most essential one concerns the convergence in degenerate situations where the algorithm will naturally have a tendency to converge to very small circles that lie within the thickness of the drawn lines, especially in configurations where lots of intersections occur. This is specially true if the draw-

ing contains filled forms. On the other hand, using skeleton images exclusively, completely eliminates (or at best, heavily distorts) filled shapes. The data used in our algorithm is mainly coming from a skeleton image, in which we added some supplementary treatment in order not to loose information related to filled shapes. The image first undergoes a thin/thick separation using a simple histogram splitting algorithm. The thick drawing parts (corresponding to the filled shapes) are converted to their contours, and reinjected into the thin part. Only then we proceed to computing the skeleton image.

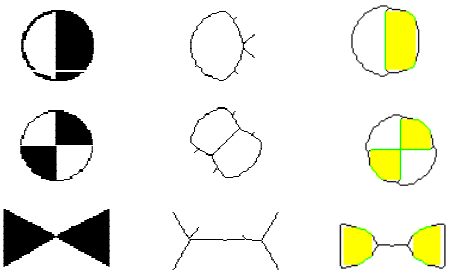

Figure 3. Intelligent skeleton computation

Figure 3 illustrates what happens. Images containing filled shapes are compared with both their raw skeleton (middle) and with the thin/thick separated skeleton (right). The right column highlights both the contour that is reinserted into the image, and the thick part that has been removed.

This approach allows us to correctly identify both filled and non-filled circles, without falling into local minima produced by the infinity of circles that can be drawn within the thick parts of the drawing. This has a drawback, compared to [5], for instance, since we cannot account for the exact positioning of the estimated circle within the line thickness. This is further analyzed in section 6.

However, once the best fitting circle (with respect to the skeleton) is found, we add a last selection criterion that will determine if it really corresponds to a circle in the original image. We therefore project it onto the original pixels and retain only those circles for which the number of black pixels exceeds a predetermined ratio (in our case 96%).

3. Generating Hypotheses

The previously presented method does a very good job in robustly determining whether there is a circle close to a given center and radius (x_c, y_c) and σ. This is not sufficient for identifying all the circles in an image. We automatically segment the image and extract all circular arcs using a basic Rosin & West vectorization [9]. Here again, we stress the fact that our method is sufficiently robust to be able to cope with approximate initializations. We initialize the previously described circle detection with all circular arcs that have been extracted. Some examples of results are shown in Figures 4 and 5.

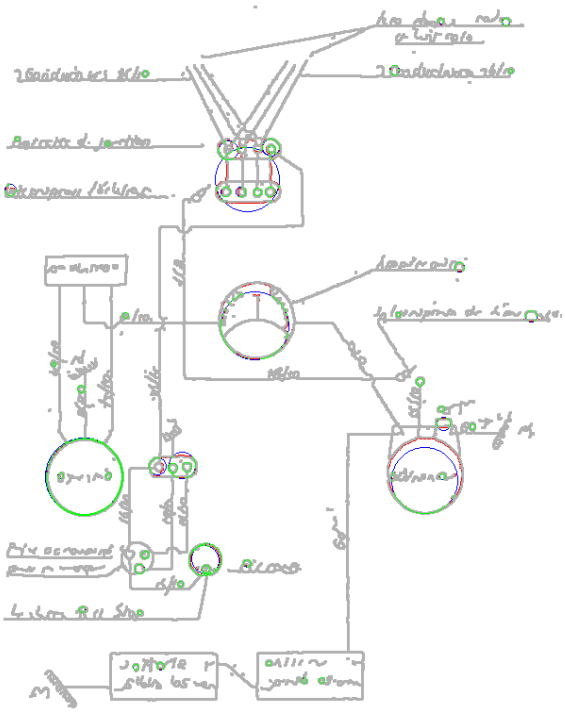

Figure 4. Expression of hypothetical circles after elementary arc extraction

4. Filtering Spurious or Multiple Detections

It is clear from Figures 4 and 5 that the hypothetical circles found in the previous step need filtering. Indeed, numerous circular arcs in the image do not necessarily correspond to full circles in the image, and clutter, intersections and complex symbols may account for multiple circular arcs originating from the same circle in the image. Furthermore, when a circle is split up in several arc segments, it is very likely that various numerical and visual side effects contribute to quite erroneous estimations of the arc center and radius. This inevitably results in poor initial circle estimates that may be quite distinct one from another, although they originally come from the same object.

The previously described method will quite elegantly deal with most of the cases: as shown in section 2.3 and Figure 2 poor initial estimates will generally converge to the correct circle. Moreover, our criterion that consists of retaining only those circles that are sufficiently covered by black pixels in the image will efficiently filter out arcs that

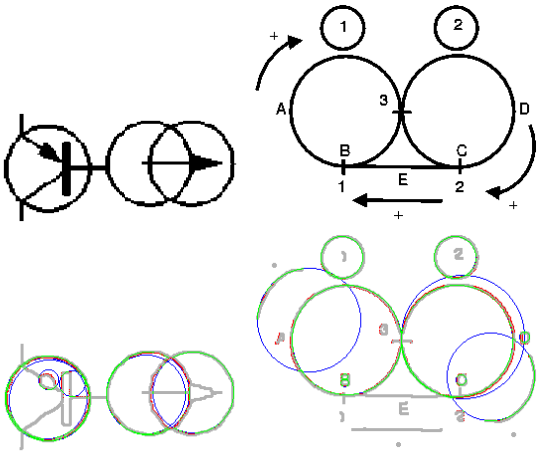

Figure 5. Expression of hypothetical circles after elementary arc extraction

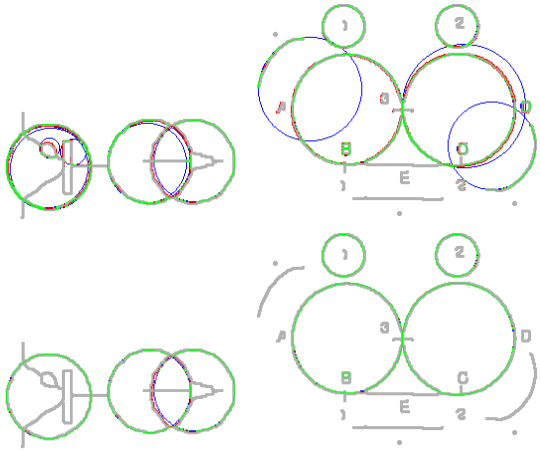

Figure 6. Filtering of hypothetical circles and selection of final results

do not account for full circles. The only remaining issue is the one mentioned in section 2.4: since the RANSAC algorithm only takes into account skeleton data points, and that global verification is only done by substraction of the initial image, any circle correctly covering the image circle will be considered correct. If the original circle is made of sufficiently thick strokes, the method can converge to several circles for the same image circle. This is, however the only situation in which multiple detections may occur.

We currently solve this problem by introducing the MaxRadiusError threshhold (in our experiments its value is 15%). Two circles $\mathcal{C}_1 = (c_1, \sigma_1)$ and $\mathcal{C}_2 = (c_2, \sigma_2)$ are considered identical *iff*

$$|\sigma_1 - \sigma_2| < \min(\sigma_1, \sigma_2) \times \text{MaxRadiusError}$$

and

$$d(c_1, c_2) < \min(\sigma_1, \sigma_2) \times \text{MaxRadiusError}$$

Figure 6 shows the filtering results obtained on the hypothesis of possible circles shown in Figure 5. The drawing on the left hand side initially has 8 possible circles, the drawing on the right hand side 11. After filtering, as described above, three circles remain for the left hand side drawing ; 5 for the right hand side (the fifth detected circle consists of the upper part of the letter B).

5. Computational Complexity and Performance

Our approach is in appearance yet another way of dealing with the same problem, like those develloped in [3, 7]. There are a number of fundamental differences that make it far less expensive where computational cost is concerned, and make it more robust and precise. On the other hand, we make the assumption that the images we are working on can be segmented into reasonable correct circular arcs. This is not the case in work line Randomized Sampling or Constrained Hough Transforms [3, 7], where no assumption whatsoever is made on the quality of the image. This difference set aside, we are considering all these methods on the same kind of image quality.

Complexity of our method is inherently close to the one presented in [7] with a major difference that the preliminary segmentation and selection process, given by the circular arcs drastically reduces the search space, and that furthermore, the error model is far simpler, without any loss of quality or precision on the one hand (since the underlying minimization goals remain very similar) but with a far higher robustness factor because of the LMeds minimization.

Indeed, one needs to proceed to $\frac{log(\bar{P})}{log(1-\varepsilon^N)}$ trials with all of the methods [3, 4, 7]. With $\bar{P}$ the probability of picking a erroneous sample set, ε the proportion of spurious trial points within the global point set, and N the number of points per trial. Our method, however, because of its initial guess based on arc detecion, reduces the global set of data points, and increases their liability, thus minimizing the number of trials. This selection process does not interfere with global complexitiy since it is done in linear time with respect to the number of points in the image.

6. Conclusions

We have presented in this paper a highly robust method for detecting circles in a line drawing images. It is very fast, extremely robust and reliable, and it is capable of assessing

the quality of its detection. It is based on Random Sample Consensus minimization, and uses techniques that are inspired from object tracking in image sequences.

The method is quite difficult trick into misdetecting circles. Figure 7 shows two drawings, one in which no circle is found, and a second where filled shape managed to get it to detect a circle where there was none.

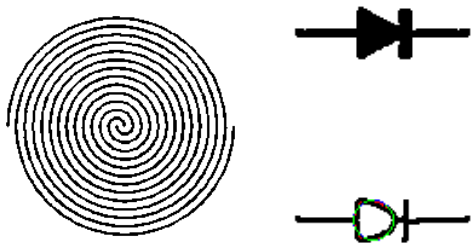

Figure 7. Trying to trick the algorithm

It is, however possible to find configurations where our method fails to find circles. Figure 8 shows some of them. It represents the initial circle hypotheses, showing, for the first symbol that no hypotheses exist for the smaller circles. This is due to the fact that not circular arcs are detected at that level. For the second symbol, the central circle is too cluttered for the circular arcs to be sufficiently robust. This results in the initial circle hypotheses being far off the correct guess. Combined with the characters and strokes within the circle, the RANSAC algorithm gets stuck in a local minimum, failing to detect the correct circle.

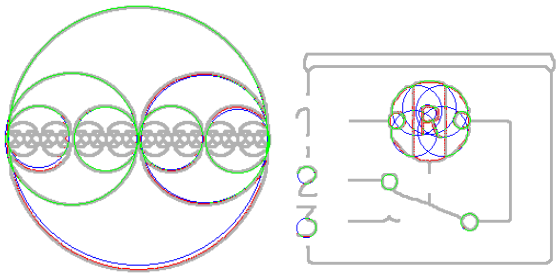

Figure 8. Failures in detecting the correct circles

The main origin of this defect is due to the circular arc detection. Further work will try to establish in what sense this can be improved. Other extensions concern the quite *ad hoc* MaxRadiusError and MaxCenterError thresholds (which can easily be removed by expressing in function of the line thickness) and the consideration of other forms that mere circles.

Acknowledgments

Authors acknowledge funding from the EC FP6 Strep Project “Fresh” FP6-516059.

References

[1] I. Amir. Algorithm for Finding the Center of Circular Fiducials. *Computer Vision, Graphics and Image Processing*, 49(3):398–406, 1990.

[2] M. Berman. Large Sample Bias in Least Squares Estimators of a Circular Arc Center and Its Radius. *Computer Vision, Graphics and Image Processing*, 45:126–128, 1989.

[3] T.-C. Chen and K.-L. Chung. An Efficient Randomized Algorithm for Detecting Circles. *Computer Vision and Image Understanding*, 83(2):172–191, Aug. 2001.

[4] M. A. Fischler and R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[5] X. Hilaire and K. Tombre. Robust and Accurate Vectorization of Line Drawings. *IEEE Transactions on PAMI*, 28(6):890–904, June 2006.

[6] U. B. Mokate and R. Kasturi. An Algorithm for Recognition of Circles in Graphics. Computer Engineering Technical Report TR-88-061, The Pennsylvania State University, University Park, Pennsylvania 16802, 1988.

[7] C. F. Olson. Constrained Hough Transforms for Curve Detection. *Computer Vision and Image Understanding*, 73(1):329–345, Mar. 1999.

[8] D. Pao, H. F. Li, and R. Jayakumar. Graphic Features Extraction for Automatic Conversion of Engineering Line Drawings. In *Proceedings of 1st International Conference on Document Analysis and Recognition (Saint-Malo, France)*, volume 2, pages 533–541, 1991.

[9] P. L. Rosin and G. A. West. Segmentation of Edges into Lines and Arcs. *Image and Vision Computing*, 7(2):109–114, May 1989.

[10] G. Roth. Primitive Extraction Using Incremental Curve Generation. In *Proceedings of IAPR Workshop on Machine Vision Applications, Tokyo (Japan)*, pages 411–414, Dec. 1992.

[11] P. J. Rousseeuw and A. M. Leroy. *Robust Regression and Outlier Detection*. Wiley Series in Probability and Mathematical Statistics. John Wiley and Sons, 1987.

[12] G. Sanniti di Baja. Well-Shaped, Stable, and Reversible Skeletons from the (3,4)-Distance Transform. *Journal of Visual Communication and Image Representation*, 5(1):107–115, 1994.

[13] B. Shahraray, A. T. Schmidt, and J. M. Palmquist. Defect Detection, Classification and Quantification in Optical Fiber Connectors. In *Proceedings of IAPR Workshop on Machine Vision Applications, Tokyo (Japan)*, pages 15–22, 1990.