Robust and Precise Circular Arc Detection

Bart Lamiroy, Yassine Guebbas

Robust and Precise Circular Arc Detection*

Bart Lamiroy and Yassine Guebbas

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 to robustly detect circular arcs 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. It is based on simple initial guesses, either based on connected line segments, or on elementary mainstream arc detection algorithms. Our method consists of gradually deforming these circular arc candidates as to precisely fit onto the image strokes, or to reject them if the fitting is not possible, this virtually eliminates spurious detections on the one hand, and avoiding non-detections on the other hand.

1 Introduction

Finding circular arcs is one the recurring problems in graphical document interpretation or symbol recognition. The main difficulty with the existing approaches is that they often are of considerable complexity (e.g. Hough-like [1] or feature grouping approaches [2]) sensitive 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 [3] and segment detection [4]. Unlike extremely efficient methods like [5], ours does not require reasonable segmentation of arcs. This work is tightly related to [6].

The following section establishes how to determine if a single circular arc is present, provided we have a rough initial guess of its position, and how to robustly detect and locate it using RANSAC (Random Sample Consensus [7]).

* The original version of this paper was published in "Graphics Recognition. Achievements, Challenges, and Evolution", Lecture Notes in Computer Science, 2010, Volume 6020/2010, 49-60, DOI: 10.1007/978-3-642-13728-0_5, and is available at www.springerlink.com

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 Determining the Presence of a Circular Arc

In this section we address the problem of detecting a circular arc, given an initial estimate of its center (x_c, y_c), its radius σ, and its two endpoints p_l and p_r[1]. This estimate, as we shall see further, can be very approximate. The main goal, in this first stage, is to detect whether or not, an arc is present in the image, near the vicinity of the given parameters.

2.1 General Algorithm

We are mainly exploiting the algorithm described in [6], with one major adjunction. The cited method has been developed to identify and locate full circles, and therefore only needs to consider adapting to two variables: the center (x_c, y_c), and the radius σ. This is not the case anymore for detecting arcs, since two parameters are added: p_l and p_r, the left and right endpoints.

The general approach we develop consists of taking the set $\mathcal{P} = \{p_i\}$ of all pixels p_i lying on the discrete circular arc $\mathcal{A}^0$ defined by (x_c, y_c), σ, p_l and p_r. As in, [6], we define, for each of these pixels p_i, 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 black pixel to p_i. Let $\mathcal{Q}_a^0 = \{q_i\}$. $\mathcal{Q}_a^0$ therefore is the set of all black pixels closest to the initial estimate $\mathcal{A}^0$ in the direction of the circle radius.

Figure 1 gives an illustration of this estimation. Initial guesses are drawn in blue. For each conjectured circle, green pixels are those found at the correct distance from the center, while red ones lie on the radius and are closest to the circle.

Now, let $\mathcal{C}^1$ be the best fitting circle over $\mathcal{Q}_a^0$ (any criterion can be used, but we are using the Least Median of Squares – *cf.* section 2.2), and let us generalize the previous step, such that $\mathcal{Q}_c^t$ contains the set of all black pixels closest to the theoretical circle $\mathcal{C}^t$ in the direction of the circle radius (and similarly for $\mathcal{Q}_a^t$).

By construction, $\mathcal{Q}_a^t \subset \mathcal{Q}_c^t$, and while this new set of points allows for a re-estimation of (x_c, y_c), σ, the other parameters p_l and p_r need to be re-evaluated as well. The approach is the following:

Let τ^t be the error measure between $\mathcal{A}^t$ and $\mathcal{Q}_a^t$. *i.e.* τ^t represents the fitness between the model $\mathcal{A}^t$ and its corresponding data $\mathcal{Q}_a^t$. Let $\mathcal{A}_<^t \subset \mathcal{A}^t$ a smaller

[1] In this document we shall conveniently ignore the fact that there is a small ambiguity with defining an arc by the center of its corresponding circle, the radius and the endpoints: one also has at least the orientation of the arc to consider as to know what part of the circle between the two endpoints is belonging to the arc, and which part isn't.

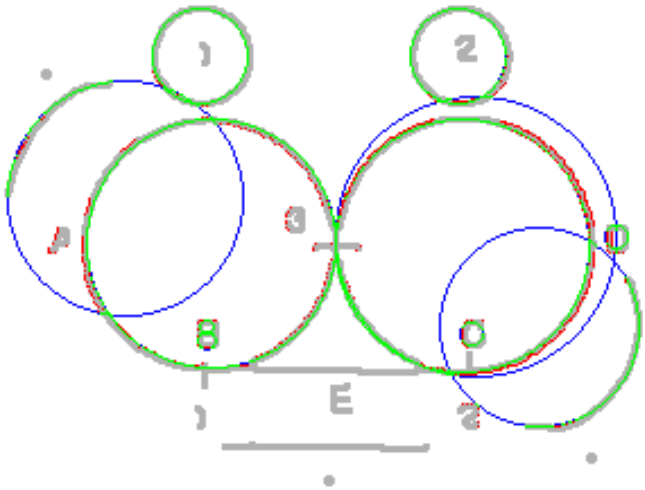

Fig. 1. Example of circle hypotheses: in blue, the initial guess; in green, points correctly lying on the conjectured circle; in red, point closest to the circle.

circular arc[2] than $\mathcal{A}^t$ such that $\tau^t_< > \tau^t$ and that

$$\forall \mathcal{A}^{t\star} | \mathcal{A}^t_< \subset \mathcal{A}^{t\star} \subset \mathcal{A}^t : \tau^{t\star} < \tau^t. \tag{1}$$

In other terms, $\mathcal{A}^t_<$ is a smaller arc that fits the dataset better than $\mathcal{A}^t$ and all intermediate arcs fit less. This means that $\mathcal{A}^t_<$ is the largest sub-arc fitting the data better than $\mathcal{A}^t$.

We do a similar search by increasing the arc size thus obtaining $\mathcal{A}^t \subset \mathcal{A}^t_>$ a larger circular arc than $\mathcal{A}^t$ such that $\tau^t_> > \tau^t$ and that

$$\forall \mathcal{A}^{t\star} | \mathcal{A}^t \subset \mathcal{A}^{t\star} \subset \mathcal{A}^t_< : \tau^{t\star} < \tau^t, \tag{2}$$

$\mathcal{A}^t_>$ thus being the smallest super-arc fitting the data better than $\mathcal{A}^t$.

We can then define $\mathcal{A}^{t+1}$ as being the $\text{argmax}_\tau \{\mathcal{A}^t_<, \mathcal{A}^t, \mathcal{A}^t_>\}$. Continuing this iteration until $\mathcal{A}^t = \mathcal{A}^{t+1}$ will yield the best estimate of the arc (if any) closest to the initial $\mathcal{A}^0$.

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

2.2 Using RANSAC and LMedS

Since there is no guarantee that any $\mathcal{A}^t$ or $\mathcal{Q}^t$ may effectively contain points that form a circle, it may be extremely hazardous to use global minimization

[2] "smaller" meaning having the same center and radius, but having a smaller aperture while being fully included in the "larger" one.

approaches (like Least Squares, for instance) [8]. It is known that these estimators are very sensitive to outliers or spurious data that does not conform to the required model [9]. Using these functions would invariably lead to degenerate convergence.

RANSAC [7] 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 of 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 $\mathcal{Q}^t$ be 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 $\mathrm{Med}_\tau(\mathcal{S})$ 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

$$\mathrm{RsdErr}\left(\mathcal{Q}^t, \mathcal{C}_n\right) = \mathrm{Med}_\tau\left(\left\{\delta\left(\mathcal{C}_n, p\right) | p \in \mathcal{Q}^t\right\}\right). \tag{3}$$

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 $\mathrm{RsdErr}(\mathcal{Q}^t, \mathcal{C}_n)$, thus obtaining

$$\mathcal{C}^{t+1} = \underset{\mathcal{C}_n, n \in [1 \ldots R]}{\operatorname{argmin}} \left(\mathrm{RsdErr}\left(\mathcal{Q}^t, \mathcal{C}_n\right)\right). \tag{4}$$

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 [7].

3 Robust Arc Detection

The previously presented method does a very good job of robustly determining whether there is a circular arc close to a given center and radius (x_c, y_c) and σ. However, it needs some initial guess on where to search. The method we are developing here proceeds in three main phases:

1. Generate a high number of possible arc candidates, without consideration of uniqueness, overlapping or exact localization.
2. Verify the quality of each candidate using the approach described in section 2. The output of this verification is a list $\mathcal{A}$ of genuine arcs, correctly fitted on the image data.
3. Detect and merge multiple and/or partial detections of the same curves as to obtain a set of unique, disjoint arcs.

3.1 Arc Candidate Generation

In order to obtain the largest possible set of arc candidates, we automatically segment the image using a basic Rosin & West line segment vectorization [4]. We then simply enumerate all connected pairs of segments. Each pair gives us three points, which is exactly the amount of data that allows for getting an initial guess for a circular arc: (p_1, p_2, p_3). These points define a unique circle on the one hand, and furthermore, since they are ordered – p_2 being in the middle – they define the left and right extrema for the definition of an arc.

This approach is combined with direct arc detection from [4] as to produce the largest possible set of arc candidates to bootstrap our localization method (*cf.* section 2).

3.2 Merging of multiple detections

Since the method is based on unfiltered hypothesis generation, it has a clear tendency toward over-segmentation, as shown in Figure 2. The main idea behind being tolerant towards this over-segmentation is to be confident that (almost) all image pixels belonging to an arc are covered by at least one initial arc candidate. Merging arcs should therefore result in a full coverage of each arc of the image by one unique, genuine arc. Merging arc candidates representing the same circular arc in the image requires two distinct operations: merging estimates covering the same pixels and merging arc candidates not sharing the same pixels but being partial estimates of a same wider arc. These two operations can be performed by first increasing the aperture of the arcs (*cf.* section 3.2.1), thus making hypothetical arcs share pixels, and, secondly, merging the arc candidates sharing pixels (*cf.* section 3.2.3). For merging arcs, we do not use the full circle image, but the image skeleton [6].

3.2.1 Increasing Aperture To increase the aperture of an arc, we first set a threshold to the maximum distance between a point from the discrete hypothetical arc and the closest pixel, as shown in Figure 4, where the distance is measured on the line going through the center of the hypothetical arc and a point on the hypothetical arc.

The increase of the aperture of an arc is done by starting from the endpoints of the candidate arc p_l and p_r and then increasing the aperture pixel-wise, as long as the distance to the closest pixel remains below the threshold.

3.2.2 Finding Which Arcs to Merge Once the set of maximal arc candidates obtained, overlapping ones or those lying on the same image curve need to respond to the following criteria in order to be merged:

1. A non-empty intersection between two arcs means that these two arcs are likely part of a same covering arc. However two arcs having common closest pixels are not necessarily sub-arcs of a same arc as shown in Figure 5.

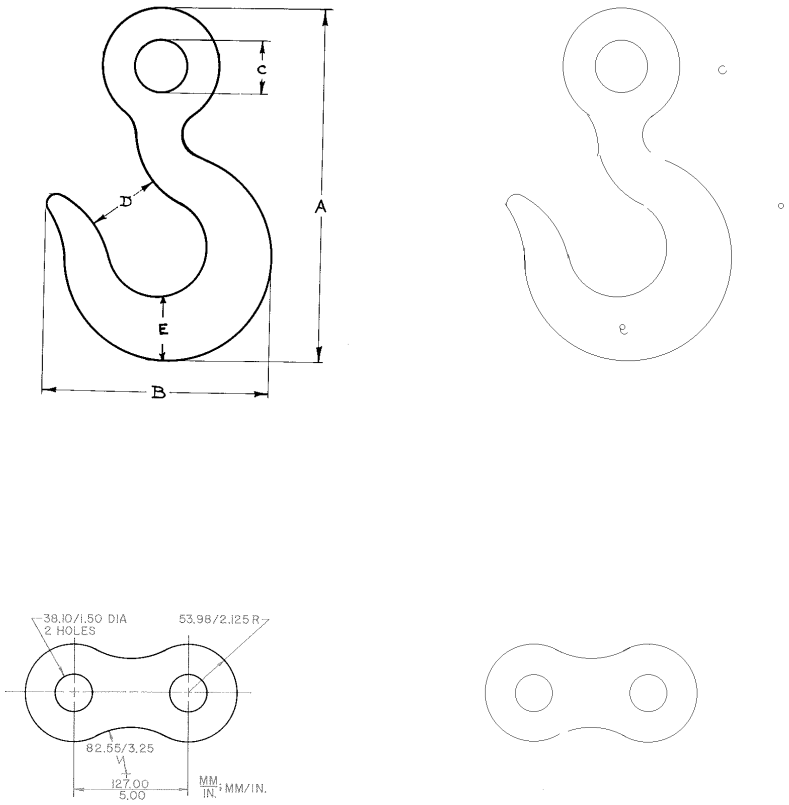

Fig. 2. GREC 2007 contest images: original image (left) – final segmentation (right)

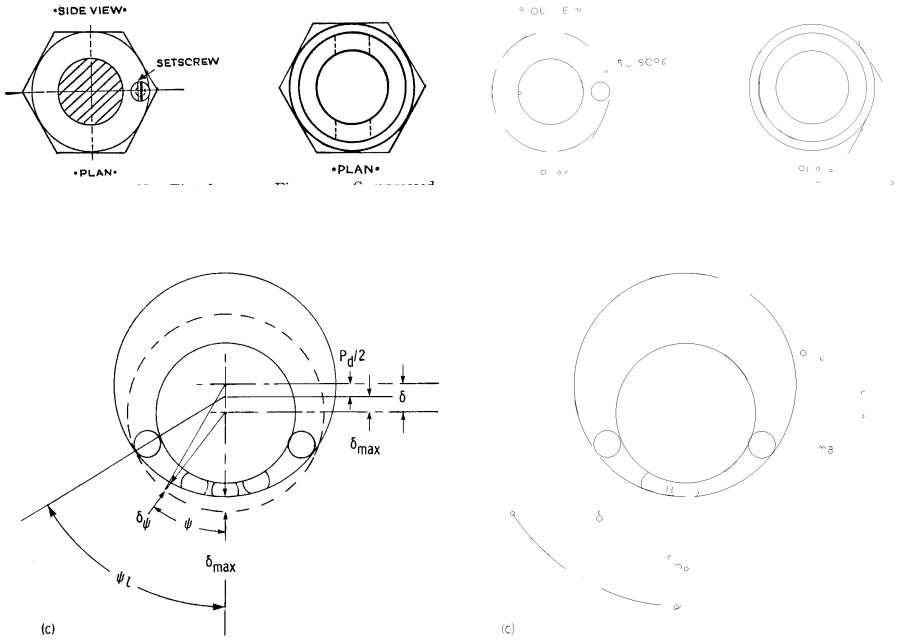

Fig. 3. GREC 2009 contest images: original image (left) – final segmentation (right)

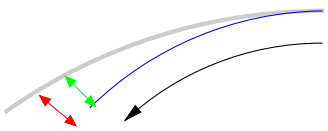

Fig. 4. Distance between a point of the hypothetical arc and the closest pixel: in blue the hypothetical arc; in Grey, image pixels; in green, distance between a point on the hypothetical arc and the closest pixel; in red, threshold

2. Arcs having comparable radii are merge candidates. This is checked through a radius ratio with the formula:

$$\frac{|r_1 - r_2|}{\max(r_1, r_2)} < \texttt{RatioRadiusError}. \tag{5}$$

Checking the center of the circle is less robust since small changes in curvature may be visually insignificant, but generate large differences in the center position.

3. Arcs having opposed normal vectors (*cf.* Figure 5) are not eligible for merging, even though they may overlap. This criterion is verified by choosing a point I from the overlapping part of two arcs, and constructing a vector $\overrightarrow{IO_1}$ that originates from I and finishes at O_1 the center of the first arc, and defining similarly a vector $\overrightarrow{IO_2}$. We then compute the scalar product of these vectors:

$$\overrightarrow{IO_1} \cdot \overrightarrow{IO_2} = |\overrightarrow{IO_1}||\overrightarrow{IO_2}| \cos\theta \tag{6}$$

If the sign of the product is positive, we consider that the two arcs stem from the same covering one.

Fig. 5. Intersection of arcs with opposed curvature signs or with significantly different radii

Once these criteria are verified, three different configurations may occur for merging. They are depicted in Figure 6. In configuration A the two arcs are "adjacent" sharing some pixels, in configuration B one arc includes another and in configuration C the two arcs are "explementary". The covering arc is formed by considering that the two arcs belong to the same circle. Therefore the resulting arc is the union of the two arcs as if they had the same center and the same radius, in other words, the computation of the arc's angle and aperture is based on the angles and apertures of the two arcs.

3.2.3 Merging Arcs The last phase consists of creating the final, genuine arcs by merging the selected candidates corresponding to the previously described criteria. The method developed here tries to find the three points of the equilateral triangle which is circumscribed by the merged arc circle. This increases the odds of having the best fitting circle as with this method we avoid choosing either noisy or numerically instable points. This procedure begins by

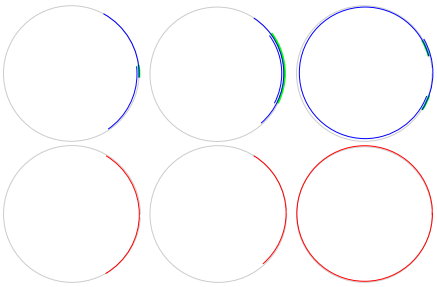

Fig. 6. Intersection configurations (top) and corresponding covering arcs (bottom). Configurations are labeled from left to right: A,B and C.

choosing either the arc 1 or 2, and then computes the edge length of the circumscribed equilateral triangle

$$\text{edgeLength} = 2r_i \sin\frac{\pi}{3}, \tag{7}$$

where r_i is the radius of the circle.

Now, let $\mathcal{R}_i$ be the subset of image curve points ($\mathcal{Q}$) which are close to the arc candidate $\mathcal{C}_i$

$$\mathcal{R} = \{p \in \mathcal{Q} | \delta(\mathcal{C}, p) < \text{RsdErr}(\mathcal{Q}, \mathcal{C})\}. \tag{8}$$

We can then define a partitioning of the points ($\mathcal{D}_1$ and $\mathcal{D}_2$) belonging to the two arc candidates, as well as their intersection $\mathcal{I}$

$$\mathcal{I} = \mathcal{R}_1 \cap \mathcal{R}_2, \mathcal{D}_1 = \mathcal{R}_1 \backslash \mathcal{R}_2, \mathcal{D}_2 = \mathcal{R}_2 \backslash \mathcal{R}_1. \tag{9}$$

We then define p_3 and p_1 such that

$$p_3 p_1 = \min_{p_i \in \mathcal{I}, p_j \in \mathcal{D}_2} |\text{edgeLength} - p_i p_j| \tag{10}$$

and find p_2 such that

$$p_2 p_3 + p_2 p_1 = \max_{p_i \in \mathcal{D}_2} (p_i p_3 + p_i p_1). \tag{11}$$

If we consider that the initial arcs belong effectively to the same circle, this method constructs the equilateral triangle and gives three points of a same circle. The more the three points are distant from each other, the more accurate the construction of the new circle is.

In fact, we are likely to find a better distributed set of three points over a circle if instead of using $\mathcal{D}_1$ and $\mathcal{D}_2$ we use $\mathcal{R}_1$ and $\mathcal{R}_2$.

4 Experiments

In collaboration with E. Barney Smith [10] we have been conducting an exhaustive survey of the influence of all possible internal parameters and external noise variations on the quality of the arc detection. Full analysis and report of this work is beyond the scope of this paper and will be published separately. Fig 7 shows some samples of used synthetic data for assessing the quality and precision of our approach.

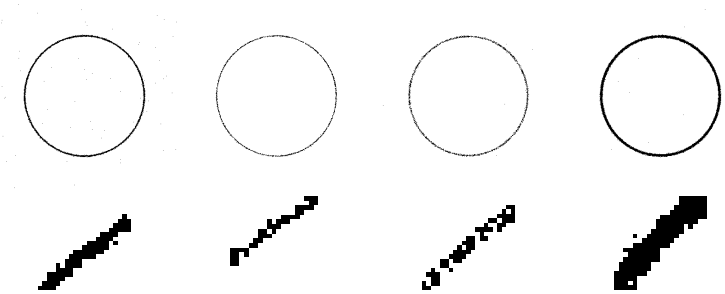

Fig. 7. Various Degraded Synthetic Circles and Selected Zooms

The overall precision in circle detection is extremely high. Precision was measured using three different metrics:

Circle Center Precision is obtained by measuring the euclidian distance of the detected arc circle to the theoretical circle center.

Circle Radius Precision is obtained by measuring the absolute difference between the detected radius an the theoretical radius.

Overlapping is a metric correlating the two previous ones, and expresses the overlapping surface ratio of both circles. It is normalized to $[0, 1]$, where 1 signifies perfectly identical circles, and 0 perfectly disjoint circles.

Tested over a wide range of parameters (τ outlier quantile, required coverage rates – *cf.* next section – ...) our method gives the following results:

	Worst Case	Best Case
Avg. Center Precision Error (pixels)	0.67	0.38
St.Dev. Center Precision Error	0.71	0.52
Avg. Radius Precision Error (pixels)	0.11	0.01
St.Dev. Radius Precision Error	0.37	0.12
Avg. Overlapping (%)	98.2	99.2
St.Dev. Overlapping (%)	6.0	2.3

This translates into estimation errors upto a pixel for center and radius. Coverage standard deviation might seem high for the announced detection precisions, but comes from situations where we tested on small circles (radius 8

pixels) where a single pixel shift accounts for a significant proportion of non overlapping.

Figures 2 to 3 show results on the GREC 2007 and 2009 contest images. The initial images are in black, while detected arcs are in Grey (right column).

4.1 Parameters and their Influence

All parameters mentioned here are either direct transpositions of the algorithm described in this paper, or are direct call parameters of the software available for download (*cf.* note below).

One parameter that has an influence on determining whether two arcs are partial estimates of a same global arc is `RatioRadiusError` (*cf.* section 3.2.2). To compare the radii of two arcs, experiments show that a value of 64% for `RatioRadiusError` makes the merge possible for most arcs having common black pixels and avoids merging arcs with significantly different radii as shown in Figure 8. For the image in Figure 8 65% was too high and resulted in a loss of precision as shown within the rectangle.

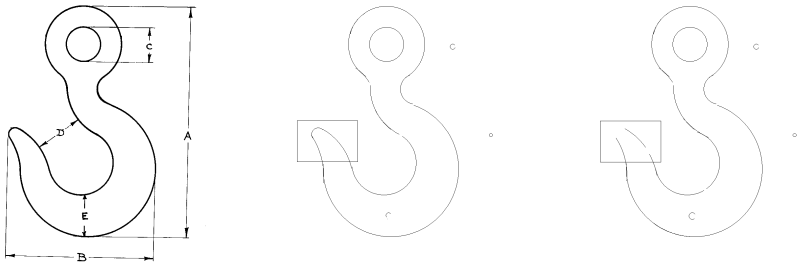

Fig. 8. RatioRadiusError tuning;the original image;64% filter; 65% filter

The `filterCoverage` parameter is used to keep only those arc candidates that have a sufficient percentage of pixels effectively lying on pixels of the image. This process uses the original image instead of the skeleton image. The coverage percentage of `filterCoverage` is set to 89% to ensure keeping only accurate estimates.

`preFilterCoverage` checks if the percentage of pixels of the discrete estimate of the given arc lying effectively on black pixels of the original image, oversteps a cover percentage and returns a boolean. This process uses the original image instead of the skeleton image. The cover percentage of `preFilterCoverage` can be lower than the cover percentage defined for `filterCoverage`, as some intermediate arcs that do not withstand the cover percentage of `filterCoverage` might be merged with another arc resulting in an arc that does withstand the

cover percentage of `filterCoverage`. The percentage 89% proved to be a good value for the cover percentage of `preFilterCoverage`, while 90% removed some good circles. This is often due to the fact that in reality, the images have slight deformations, an percieved circles are actually ellipses.

The merging algorithm can be improved by increasing the aperture of the arcs "virtually'. In other words, each arc has two angles and two apertures. The "virtual" angle and aperture are those which are increased and used to check if two arcs have to be merged. The actual angle and aperture are not changed. In fact, while increasing the actual aperture, the discrete arc of an estimate might no longer withstand the `preFilterCoverage` processing, as it is likely to have more pixels that are not black. Thus by keeping the initial angle and aperture unchanged we maintain arcs that were not merged. Moreover when performing the merge, the points used to construct the new arc are from the closest points of the actual arc, which is more accurate especially if we choose the points from those who belong as well to the black pixels of the image. The "virtual" increase uses the threshold defined in 3.2.1, namely a value of 3.

5 Conclusion and Further Work

In this paper we have presented a highly efficient and complete arc detection algorithm that needs extremely few parameters or contextual knowledge to operate. We have validated it on quite difficult images, coming from the GREC 2007 and 2009 contest. Further work will include stroke width integration in order to obtain a more precise localization of the arcs, as well as a more quantitative assessment of the positioning and localization of the detected arcs.

Acknowledgments

Authors acknowledge funding from the PROCORE-FRANCE/HONG KONG JOINT RESEARCH SCHEME (F-HK04/05T, 9050187): Knowledge representation issues for the performance evaluation of graphic symbol recognition methods. B. Lamiroy more particularly thanks Prof. Liu Wenyin for having received him at the City University of Hong Kong for finalizing this work. Authors aknowledge reporting snippet of yet unpublished work in collaboration with E. Barney Smith on evaluation of noise influence in section 4. Furthermore, the circle overlapping metric described in section 4, is the result of fruitful, informal discussions with E. Magagnin. B. Lamiroy was a visiting scientist at Lehigh University at the time of publication of this article.

Access to Source Code

The source code of this work is available for download and evaluation under LGPL at `http://gforge.inria.fr/projects/visuvocab/`.

References

1. Olson, C.F.: Constrained Hough Transforms for Curve Detection. Computer Vision and Image Understanding **73**(1) (March 1999) 329–345
2. Chen, T.C., Chung, K.L.: An Efficient Randomized Algorithm for Detecting Circles. Computer Vision and Image Understanding **83**(2) (August 2001) 172–191
3. Sanniti di Baja, G.: Well-Shaped, Stable, and Reversible Skeletons from the (3,4)-Distance Transform. Journal of Visual Communication and Image Representation **5**(1) (1994) 107–115
4. Rosin, P.L., West, G.A.: Segmentation of Edges into Lines and Arcs. Image and Vision Computing **7**(2) (May 1989) 109–114
5. Hilaire, X., Tombre, K.: Robust and Accurate Vectorization of Line Drawings. IEEE Transactions on PAMI **28**(6) (June 2006) 890–904
6. Lamiroy, B., Gaucher, O., Fritz, L.: Robust Circle Detection. In: Proceedings of 9th International Conference on Document Analysis and Recognition, Curitiba (Brazil). (2007) 526–530
7. Fischler, M.A., Bolles, R.C.: Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM **24**(6) (1981) 381–395
8. Rousseeuw, P.J., Leroy, A.M.: Robust Regression and Outlier Detection. Wiley Series in Probability and Mathematical Statistics. John Wiley and Sons (1987)
9. Berman, M.: Large Sample Bias in Least Squares Estimators of a Circular Arc Center and Its Radius. Computer Vision, Graphics and Image Processing **45** (1989) 126–128
10. Smith, E.H.B.: Characterization of image degradation caused by scanning. Pattern Recognition Letters **19**(13) (1998) 1191–1197