Unsupervised, Fast and Precise Recognition of Digital Arcs in Noisy Images

Thanh Phuong Nguyen, Bertrand Kerautret, Isabelle Debled-Rennesson, Jaques Oliver Lachaud

Unsupervised, Fast and Precise Recognition of Digital Arcs in Noisy Images *

Thanh Phuong Nguyen[1], Bertrand Kerautret[1], Isabelle Debled-Rennesson[1], and Jacques-Olivier Lachaud[2]

[1] ADAGIo team, LORIA, France
{nguyentp,kerautret,debled}@loria.fr
[2] LAMA, University of Savoie, France
jacques-olivier.lachaud@univ-savoie.fr

Abstract. In image processing and pattern recognition, the accuracy of most algorithms is dependent on a good parameterization, generally a computation scale or an estimation of the amount of noise, which may be global or variable within the input image. Recently, a simple and linear time algorithm for arc detection in images was proposed [1]. Its accuracy is dependent on the correct evaluation of the amount of noise, which was set by the user in this former version. In the present work we integrate a promising unsupervised noise detection method [2] in this arc recognition method, in order to process images with or without noise, uniformly distributed or variable within the picture. We evaluate the performance of this algorithm and we compare it with standard arc and circle detection methods based on extensions of the Hough transform.

1 Introduction

Since the last three decades, the problem of arc and circle detection is intensively studied in the computer vision and pattern recognition community. Even if important progresses have been made from the classical generalized Hough Transform, current methods are always dependent on a good choice of parameters. As shown in Fig. 1, a correct choice of parameter is necessary to get good quality in the arc detection (method of [1]). Furthermore, the complexity of several detection methods is very dependent on their parameterization, which may forbid their use in real time applications (see example (f) of Fig. 8).

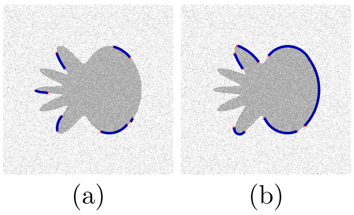


Fig. 1. Detected arcs at different widths 2 (a) and 3 (b).

The arc and circle detection gives important information for detecting specific objects in digital images or for shape recognition. We can classify the existing methods into two main groups: image and contour-based approaches. The first group contains the methods that detect directly arcs and circles through some well known techniques of transformation in images (Hough or Radon transform). The generalized Hough transform is widely used as a basic tool for solving this problem [3–5]. These methods try to transform the input image into the parameter space and the circle candidates are detected as peaks in this space. However, these methods require a lot of computation and memory space,

* This work is supported by ANR in the framework of the GeoDIB project (ANR-06-BLAN-0225).

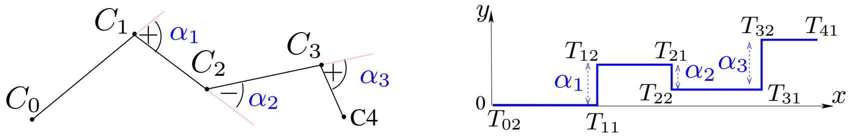
which hinders their efficiency when detecting large arcs or circles. In addition, they depend on a threshold for both accepting short arcs and eliminating noises. The second group of methods contains the ones that recognize arcs and circles on extracted contours [1, 6]. Although some of them can process damaged contours [1], all require user parameterization indicating the expected amount of noise. The approach of Roussillon *et al.* allows to perform arc recognition in linear time however it requires one of three constraints (see [7] for more detail).

The main contribution of this work is to propose a simple, efficient and unsupervised contour based algorithm for arc and circle detection. The method is based on the linear time supervised approach [1] and exploits an original and simple automatic noise detector based on discrete geometry [2]. It can process indifferently perfect images, or damaged by a uniformly distributed or variable noise. In addition, it is more efficient than traditional and improved methods based on Hough transform. The paper is organized as follows. Section 2 introduces a linear method for arc recognition. The unsupervised noise detection is presented in section 3. In section 4, we present a framework for arc segmentation along noisy curves. Several experiments and comparisons are given in section 5.

2 Linear method for arc recognition and segmentation

The arc recognition algorithm proposed in [1] is based on some properties of arcs in the tangent space representation introduced by Latecki *et al.* [8]. The discrete circles are seen through the arithmetic definition of Andres [9] that defines a digital circle as a sequence of points included in a ring.

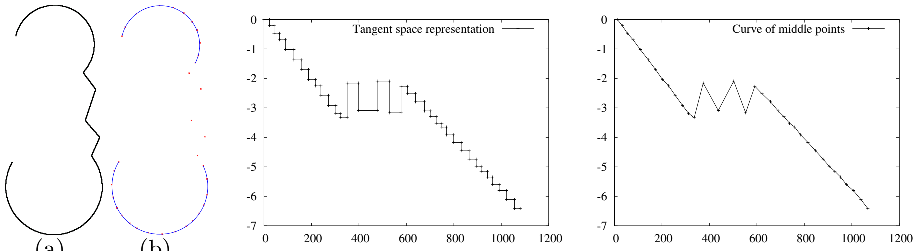
Tangent space representation for arc recognition. The main idea of this representation is to transform the polygon $C = \{C_i\}_{i=0}^{n}$ of $\mathbb{Z}^2$ by using lengths and angles between the successive segments of this polygon (see Fig. 2). We denote by α_i the angle $\angle(\overrightarrow{C_{i-1}C_i}, \overrightarrow{C_iC_{i+1}})$. If C_{i+1} is on the right hand side of $\overrightarrow{C_{i-1}C_i}$ then $\alpha_i > 0$, otherwise $\alpha_i < 0$. Some properties of a set of sequential chords of a circle in the tangent space have been established in [1]. They are summarized in the following property 1.



(a) Input polygonal curve (b) Tangent space representation

Fig. 2. Tangent space representation

Property 1. Let $C = \{C_i\}_{i=0}^{n}$ be a polygon, $\alpha_i = \angle(\overrightarrow{C_{i-1}C_i}, \overrightarrow{C_iC_{i+1}})$ such that $\sin\alpha_i \simeq \alpha_i$, $0 < i < n$. The length of C_iC_{i+1} is l_i, for $i \in \{1, \ldots, n\}$. We consider the polygon $T(C)$, that corresponds to its representation in the modified tangent space, constituted by the segments $T_{i2}T_{(i+1)1}$, $T_{(i+1)1}T_{(i+1)2}$ for i from 0 to $n-1$. $MpC = \{M_i\}_{i=0}^{n-1}$ is the midpoint set of $\{T_{i2}T_{(i+1)1}\}_{i=0}^{n-1}$. So, C is a polygon that approximates a circle or an arc of circle if and only if $MpC = \{M_i\}_{i=0}^{n-1}$ is a set of collinear points.



(a) (b) (c) Tangent space representation (d) Curve of midpoints (MpC)

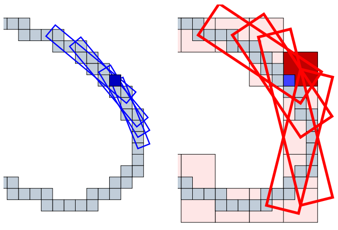
Fig. 3. Arc segmentation results on a digital curve: (a) input curve, (b) vertex of the approximated polygon and result of arc segmentation.

Blurred Segment of width ν [10]. It is a set of points that is included in a band of width ν. It can thus represent a discrete straight segment while taking into account the amount of noise contained in the initial data (according to ν). This notion is used for both the polygonalisation step (ν_1) and for the collinearity evaluation (ν_2).

The main steps of the algorithm are illustrated on Fig. 3: (a,b) polygonalization step, (d) associated tangent space representation (d). The midPoints curve (e) highlights arc circles in its straight parts (c). The algorithm is detailed in [1].

3 Unsupervised noise detection

In the work presented in [2], the authors propose a simple and efficient approach for detecting noise along discrete contours. We present here briefly the main ideas and refer the reader to the preceding reference for more details.



(a) (b)

Fig. 4. Illustration of maximal segments on (a) an initial contour C, and (b) on its subsampled contour $\phi_3(C)$.

The noise estimation is linked to the asymptotic properties of the *maximal segment* primitive. We recall first that a standard digital straight line (DSL) is a 4-connected digital set $\{(x, y) \in \mathbb{Z}^2, \mu \leq ax - by < \mu + |a| + |b|\}$, all parameters being integers. A *digital straight segment (DSS)* is a 4-connected piece of DSL. If we consider a 4-connected path C, a *maximal segment M* is a subset of C that is a DSS and which is no more a DSS when adding any other point of $C \backslash M$. Fig. 4 (a) displays the set of all the maximal segments covering the dark pixels.

Asymptotic properties of maximal segments. The presence of noise is related to the discrete lengths of maximal segments. Let us denote by $\mathrm{Dig}_h(X)$ the Gauss digitization with step h of a shape X. Let p be some point on the boundary of

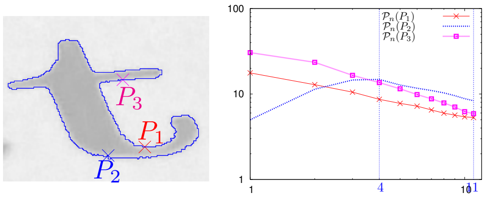
X. We denote by (L^h_j) the discrete lengths of the maximal segments, defined around $\mathrm{Dig}_h(X)$, and which cover p. It can be shown that if X is a simply connected shape in $\mathbb{R}^2$ with a piecewise C^3 boundary ∂X and U an open connected neighborhood of p, then we have:

$$\text{if } U \text{ is strictly convex or concave, then } \Omega(1/h^{1/3}) \leq L^h_j \leq O(1/h^{1/2}) \quad (1)$$

$$\text{if } U \text{ has null curvature everywhere, then } \Omega(1/h) \leq L^h_j \leq O(1/h) \quad (2)$$

Since in practice, it is not possible to obtain the asymptotic digitizations of the initial shape O with finer and finer grid steps h, a solution is to consider the subsampling $\phi_i(O)$ with increasing covering pixel sizes $i \times i$ for $i = 2, .., n$. Then, we can consider the discrete lengths $(L^{h_i}_j)$ of the maximal segments on the subsampled shapes with the increasing sequence of digitization grid steps $h_i = ih$ (see Fig. 4(b)). For a given subsampling size i, the average discrete length of all the maximal segments containing the subsampled pixel is denoted as $\overline{L}^{h_i}$.

Multiscale profile and noise detection. The multiscale profile $\mathcal{P}_n(P)$ of a point P is defined as the sequence of samples $(X_i, Y_i) = (log(i), log(\overline{L}^{h_i}))_{i..n}$. Some multiscale profiles are illustrated on the real contour of the following floating figure. According to inequality (1) (resp. (2)), if P is located on curved (resp. flat) part, the slope of an affine approximation of the multiscale profile should be in $[-\frac{1}{2}, -\frac{1}{3}]$ (resp. $[-1, -\frac{1}{2}]$). Since for noisy contour parts the preceding properties are not valid, it is used to detect such a part. A threshold t_m is given to determine the *meaningful scale* defined as a pair (i_1, i_2), $1 \leq i_1 < i_2 \leq n$, such that for all i, $i_1 \leq i < i_2$, $\frac{Y_{i+1} - Y_i}{X_{i+1} - X_i} \leq t_m$. For the example of Fig. 5, the meaningful scales of the points P_1, P_2 and P_3 are respectively equal to $(1, 11)$, $(4, 11)$ and $(1, 11)$.



(a) (b)

From the previous meaningful scale (i_1, i_2) , $i_1 - 1$ defines the *noise level* of a point P (denoted as $\eta(P)$). Experimentally the threshold value $t_m = 0$ gives best results both on curved or flat noisy parts. Fig. 5 shows some results obtained on various shapes (a,b). The noise level is well detected on both synthetic and real images. The curved/flat part areas of a shape can also be detected by thresholding the slope of the multiscale profile on the meaningful scale (see the multiscale of the point P_3 of the previous floating figure). Fig. 5 (c) shows such a detection (curved area represented by a thick contour).

4 A framework for arc recognition along noisy curves

As mentioned in Section 2, the arc detection algorithm uses a parameter ν_1 to take into account the amount of noise in the polygonalization step. This parameter is adjusted manually to obtain a representative polygon that allows a good arc recognition. As mentioned in [1], an interesting question is raised: *"For each noisy curve, how can we choose the value of ν_1 to obtain the best result?"*. This section proposes a framework to obtain a parameter-free method.

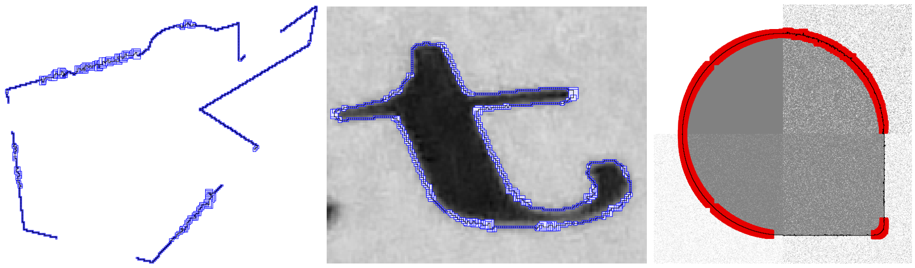


Fig. 5. Noise level detection (a,b). For each point P the noise level is represented by a centered box of size $\eta(P)+1$. (c) Curved/flat areas detection from meaningful scales (curved areas represented by a thick contour).

4.1 Polygonalization adapted to noisy curves

We propose here two solutions for the arc recognition on noisy curves. Both solutions construct an approximating polygon by taking into account the amount of noise of the discrete contour. We have considered different hypotheses on the noise distribution. The first one considers the hypothesis of uniform distribution while the second one is an adaptive solution which take into accounts locally the amount of noise of the discrete contour.

Algorithm 1: Polygonalization based on unsupervised noise detection.

```
Data: $C = \{C_i\}_{i=0}^n$ digital curve, $\eta = \{\eta_i\}_{i=0}^n$ noise information,
      $uniformNoise$- $true$ if uniform noise distribution, $false$ otherwise
Result: P-approximated polygon
begin
    b ← 0; Add $C_b$ to $P$;
    if $!uniformNoise$ then
        while $b < n$ do
            Use [10] to recognize $\{C_b, \ldots, C_e\}$ as blurred segment of width $\eta_b$;
            Add $C_b$ to $P$; b ← e;
    else
        $\bar{\eta}$ ← mean value of $\eta = \{\eta_i\}_{i=0}^n$;
        while $b < n$ do
            Use [10] to recognize $\{C_b, \ldots, C_e\}$ as blurred segment of width $\bar{\eta}$;
            Add $C_b$ to $P$; b ← e;
end
```

These solutions are based on the unsupervised noise detector presented in Section 3. This method gives noise information at each point of the noisy input curve. The local noise estimation is used to construct a polygonal approximation of the curve, adapted to the local amount of perturbation. These two processes of polygonalization are described by Algorithm 1. It considers two possible options for the noise distribution:

Uniform noise distribution. In this case the approximated polygon is constructed by using a constant width value in the blurred segment based polygonalization. This width value is defined as the mean noise level $\bar{\eta}$ obtained on the whole input curve.

Non-uniform noise distribution. Here the construction of the approximated polygon is obtained by modifying the width of the blurred segments from each new segment recognition. The width is chosen according to the noise level η of the first point added to the blurred segment.

Independence with respect to starting point. To do so, the information related to flat/curved areas determines the better way to start the polygonalization. Indeed we compute the largest interval (f_b, f_e) $b, e < n$ of the contour that is a flat area. Then, if the contour is closed, we apply Algorithm 1 with a circular permutation of size e of the initial contour. Otherwise Algorithm 1 is called twice with the partial contours $C'_{e,n}$ and $C'_{0,b}$.

4.2 Arc recognition along noisy curves

Algorithm 2 presents a parameter-free approach for arc segmentation along noisy curves. As it is shown in [1], the error bound of arc recognition depends on the width ν_2 of the blurred segment recognition and on the maximal admissible variation angle α_{max} in the tangent space. To guarantee the quality of segmentation, we fix these parameters to $\nu_2 = 0.25$ and $\alpha_{max} = \frac{\pi}{4}$ in Algorithm 2 (see also section 7.4 of [1] for more details and for the α_{max} setting). Therefore, the proposed method is in practice parameter-free when we work with noisy curves, thanks to the unsupervised noise detection. It uses no thresholding and automatically provides detected arcs in noisy curves.

Algorithm 2: Arc segmentation along a noisy digital curve

```
Data: $C = \{C_i\}_{i=0}^{n}$ noisy digital curve
Result: ARC- sequence of extracted arcs
begin
    $N \leftarrow \{N_i\}_{i=0}^{n}$ noise information determined by [2] (see Section 3);
    $ARC \leftarrow \emptyset$; Use Algorithm 1 to polygonalize $C$ in $P = \{P\}_{i=0}^{m}$;
    Represent $P$ in the tangent space by $T(P)$ (see Section 2);
    Determine the midpoint set $MpC = \{M_i\}_{i=1}^{n}$ (see Section 2);
    Use [10] to polygonalize $MpC$ into a sequence $S = \{S\}_{i=0}^{k}$ of blurred
    segments of width 0.25;
    for $i$ from 0 to $k-1$ do
        $\{M_j\}_{j=b}^{e}$: sequence of points of $MpC$ that corresponds to $S_iS_{i+1}$;
        $C^{'}$: part of $C$ that corresponds to $S_iS_{i+1}$;
        $isArc \leftarrow true$;
        for $i$ from $b$ to $e-1$ do
            if $M_{i+1}.y - M_i.y > \frac{\pi}{4}$ then $isArc \leftarrow false$
        if $isArc$ then Add $C^{'}$ to $ARC$
end
```

5 Experimental evaluation

The first experiments were performed on the noisy synthetic shapes of Fig. 6. Gaussian noise was added globally in images (a,b) and specifically to each quadrant in images (c,d). The user has only to choose between a uniform or a variable noise distribution to obtain a more precise arc detection (boolean parameter $uniformNoise$). Another experiment on a real image from a 10 million pixels digital camera is presented on Fig. 7. All the experiments presented here were obtained on a *Mac OS X*, $2.8GHz$, $4GB$ *DDR*3 *Intel Core Duo*. The contours were extracted from a simple threshold set to 128. Fig. 8 presents some others experiments on real pictures, with also a comparison with three methods based on the Hough transform: CHT [3], MHT [4] and FHT [5]. For all of these experiments the source image was thresholded by a value defined in the HSB color space. The numerous parameters μ_C, μ_M, and μ_F were respectively associated to $(radiusMin, radiusStep, nbCircles)$, $(minRadius, maxRadius)$, $(gradTreshold, radius, radiusVariation)$. Note that other parameters were not mentioned here, like for FHT and can have some influence in the result (α and β see [5]).

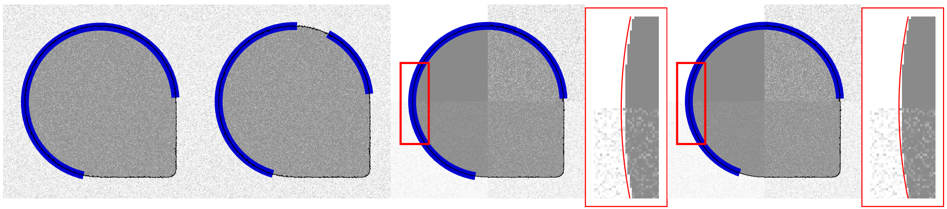

(a) global 1.07s 2228 points (b) adaptive 0.98s 2228 points (c) global 0.96s 1926 points (d) (e) adaptive 0.88s 1926 points (f)

Fig. 6. Arcs detection from the global noise based approach (a,d) and the adaptive approach (b,e) (image size 512x512). (d,f) close-up view of (c,e).

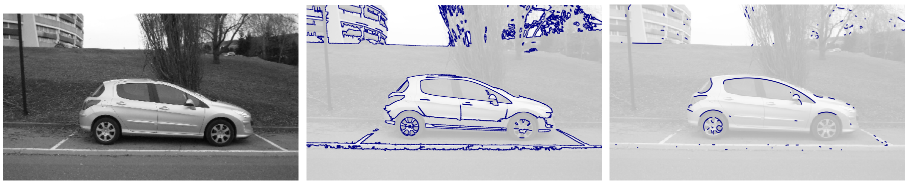
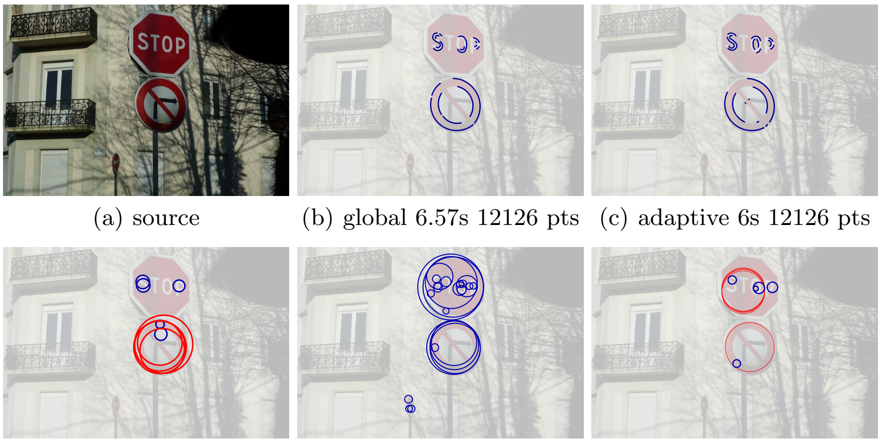

(a) source image (b) contours (c) result (uniform noise)

Fig. 7. Arc detection with our method on an image of a car (size 4000x2672 pixels).

6 Conclusion and perspectives

We have presented a new approach for arc segmentation of digital curves in noisy images by combining it with an unsupervised noise detector. The proposed methods lead to an efficient arc detector in images, especially in comparison with methods based on the Hough transform, which require both large memory and execution time. Moreover the proposed method is not dependent to the need



(a) source (b) global 6.57s 12126 pts (c) adaptive 6s 12126 pts

(d) CHT [3](1024x684)
$\mu_C = (70, 2, 25)$ 1m19s
$\mu_C = (5, 1, 20)$ 1m0s

(e) MHT [5](1024x684)
$\mu_M = (10, 190)$ 2.0s

(f) FHT [4] (4000x2672)
$\mu_F = (200, 330, 100)$1m27s
$\mu_F = (170, 50, 100)$ 4m26s

Fig. 8. Application of our method on a real picture (size 4000x2672 pixels) with the possible values for $uniformNoise$ (a-c), and comparison with three methods based on Hough transform (d-f).

to set a specific parameter. In the future, we plan to integrate the detection of curved zone in noisy curves of [2] (see also figure 5(c)) as a preprocessing step to enhance the robustness of the arc detector.

References

1. Nguyen, T.P., Debled-Rennesson, I.: A linear method for segmentation of digital arcs. Technical report, LORIA (2010) `http://www.loria.fr/~nguyentp/pubs/techreport_arcsegmentation.pdf`.
2. Kerautret, B., Lachaud, J.O.: Multi-scale analysis of discrete contours for unsupervised noise detection. In: IWCIA. Volume 5852 of LNCS. (2009) 187–200
3. Kimme, C., Ballard, D., Sklansky, J.: Finding circles by an array of accumulators. Short Communications Graphics and Image Processing **18** (1975) 120–122
4. Davies, E.: A modified hough scheme for general circle location. Pattern Recognition Letters **7** (1984) 37 –43
5. Rad, A.A., Faez, K., Qaragozlou, N.: Fast circle detection using gradient pair vectors. In: Digital Image Comp.: Techniques and Applications. (2003) 879–887
6. Coeurjolly, D., Gérard, Y., Reveillès, J.P., Tougne, L.: An elementary algorithm for digital arc segmentation. Discrete Applied Mathematics **139** (2004) 31–50
7. Roussillon, T., Tougne, L., Sivignon, I.: On three constrained versions of the digital circular arc recognition problem. In: DGCI. Volume 5810 of LNCS. (2009) 34–45
8. Latecki, L., Lakamper, R.: Shape similarity measure based on correspondence of visual parts. PAMI, IEEE Transactions on **22** (2000) 1185–1190
9. Andres, E.: Discrete circles, rings and spheres. Comp.&Graphics **18** (1994) 695–706
10. Debled-Rennesson, I., Feschet, F., Rouyer-Degli, J.: Optimal blurred segments decomposition of noisy shapes in linear time. Computers & Graphics **30** (2006)