Ellipse detection through decomposition of circular arcs and line segments

Thanh Phuong Nguyen, Bertrand Kerautret

To cite this version:

Thanh Phuong Nguyen, Bertrand Kerautret. Ellipse detection through decomposition of circular arcs and line segments. International Conference on Image Analysis and Processing, Sep 2011, Ravenna, Italy. hal-00601877

HAL Id: hal-00601877
https://hal.science/hal-00601877

Submitted on 21 Jun 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Ellipse detection through decomposition of circular arcs and line segments

Thanh Phuong Nguyen[1] , Bertrand Kerautret[1,2]

[1] ADAGIo team, LORIA, Nancy University, 54506 Vandoeuvre, France
[2] LAMA (UMR CNRS 5127), University of Savoie, France
{nguyentp,kerautre}@loria.fr

Abstract. In this work we propose an efficient and original method for ellipse detection which relies on a recent contour representation based on arcs and line segments [1]. The first step of such a detection is to locate ellipse candidate with a grouping process exploiting geometric properties of adjacent arcs and lines. Then, for each ellipse candidate we extract a compact and significant representation defined from the segment and arc extremities together with the arc middle points. This representation allows then a fast ellipse detection by using a simple least square technique. Finally some first comparisons with other robust approaches are proposed.

1 Introduction

Shape identification is an important task in image analysis. Ellipse is a basic shape that can appear naturally in images from 3D environment. Therefore, ellipse detection is a key problem in many applications in the field of computer vision or pattern recognition.

In general, we can group existing methods into three main approaches. The first one relies on the Hough transform [2–6]. These methods transform image into parametric space and then take the peaks in this space as candidate of ellipse. Generally, it requires a parameter space that has five dimensions - contrariwise to two for straight line detection and three for circle detection, so it needs more execution time and memory space than the two last approaches. Some modifications [4–6] of Hough transform have been proposed to minimize storage space and computation complexity. Daul et al. [4] reduce the problem to two dimensional parametric space. Later, Chia et al. [5] introduced a method based on Hough transform in one dimensional parametric space. Lu et al. [6] proposed an iterative randomized Hough transform (IRHT) for ellipse detection with strong noise.

The second one uses least square fitting technique [7–10] that minimizes the sum of square error. There are two main types of least square fitting (see [10]) that are characterized by the definition of error distances: algebraic fitting and geometric fitting. Concerning the first type, the error distance is defined by considering the deviation at each point to the expected ellipse described by implicit equation $F(x, a) = 0$ where a is vector of parameters. Contrariwise, for the second type, the error distance is defined as orthogonal distance from each point to the fitting ellipse.

The third group of approach detects ellipse candidates by using their moment [11–15].

We propose a new method for ellipse detection based on the decomposition of an edge image into arc and line primitives. The main contribution of this paper is to propose a pre-processing step that allows to speed up the detection of ellipse based on a linear scanning process on the sequence of arc and line primitives. The rest of this paper is organized as follows. The following section recalls a method for the representation of a digital curve by arcs and line segments. Section 3 presents the proposed method for ellipse detection before experimentation.

2 Descriptor based on arc and line primitives

In this section, we recall a linear method [1] for the decomposition of a digital curve into circular arcs and line segments.

2.1 Tangent space representation and properties of arc in the tangent space

Nguyen and Debled-Rennesson proposed in [16] some properties of arcs in tangent space representation that are inspired from Latecki [17]. Let $C = \{C_i\}_{i=0}^{n}$ be a polygon, l_i - length of segment C_iC_{i+1} and $\alpha_i = \angle(\overrightarrow{C_{i-1}C_i}, \overrightarrow{C_iC_{i+1}})$. If C_{i+1} is on the right of $\overrightarrow{C_{i-1}C_i}$ then $\alpha_i > 0$, otherwise $\alpha_i < 0$ (see illustration of Fig. 1(a)).

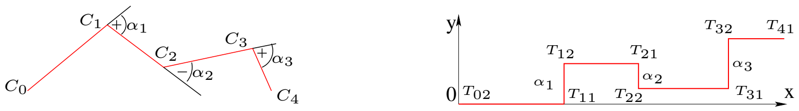
Let us consider the transformation that associates a polygon C of $\mathbb{Z}^2$ to a polygon of $\mathbb{R}^2$ constituted by segments $T_{i2}T_{(i+1)1}, T_{(i+1)1}T_{(i+1)2}$, $0 \leq i < n$ (see Fig. 1(b)) with:

$T_{02} = (0, 0)$,

$T_{i1} = (T_{(i-1)2}.x + l_{i-1}, T_{(i-1)2}.y)$, i from 1 to n,

$T_{i2} = (T_{i1}.x, T_{i1}.y + \alpha_i)$, i from 1 to $n-1$.

Nguyen et al. also proposed in [16] some properties of a set of sequential chords of a circle in the tangent space. They are resumed by proposition 1 (see also Fig. 2).



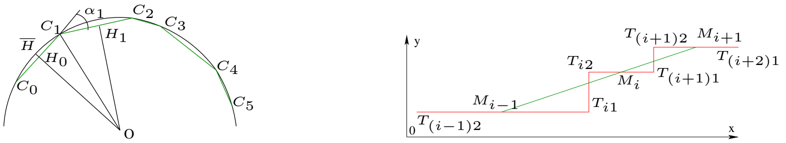
(a) Input polygonal curve

(b) Tangent space representation

Fig. 1. Tangent space representation

Proposition 1. *[16] 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 $\alpha_i \leq \alpha_{max} \leq \frac{\pi}{4}$. 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 whose vertices are on a real arc only if $MpC = \{M_i\}_{i=0}^{n-1}$ is a set of quasi collinear points.*

From now on, MpC is called the midpoint curve.



(a) A set of sequential chords of an arc.

(b) Its property in tangent space representation.

Fig. 2. The chords in tangent space.

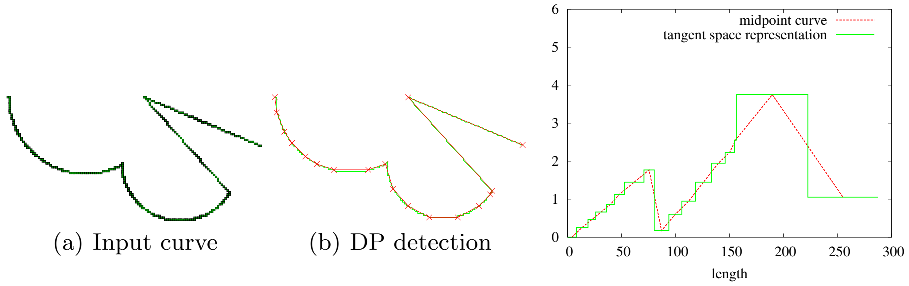
2.2 Arc line decomposition

Proposition 1 can be used to decide if a digital curve is a circular circle by detecting straight line segment in the tangent space. Moreover, it is also used for the decomposition of a curve into arcs and line segments. Nguyen introduced the definition below.

Definition 1. *In the curve of midpoints in the tangent space,* **an isolated point** *is a midpoint satisfying that the differences of ordinate values between it and one of its 2 neighboring midpoints on this curve is higher than the threshold α_{max}. If this condition is satisfied with all 2 neighboring midpoints, it is called* **a full isolated point**

Let us consider Fig. 3. In this example, there are all basic configurations among the primitive arc and line: arc-arc, arc-line and line-line. Fig. 4 presents these configurations in detail in the tangent space. Concerning the midpoint curve (MpC) in the tangent space, Nguyen et al. [1] introduced several remarks below.

- An isolated point in MpC corresponds to an extremity among two adjacent primitives in C.
- A full isolated point in MpC corresponds to an line segment in C.
- An isolated point in MpC can be co-linear with a set of co-linear points that corresponds to an arc.

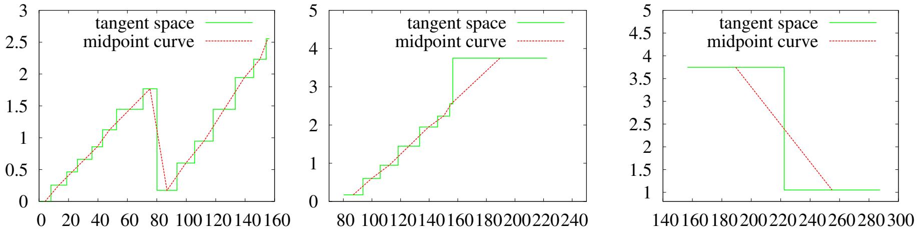


(a) Input curve

(b) DP detection

(c) Representation in the tangent space

Fig. 3. An example of curve



arc-arc

arc-line

line-line

Fig. 4. Configurations on tangent space.

In [1], Nguyen et al. proposed an algorithm (see[3] algo. 1) to decompose a curve C into arcs and straight line segments. First, the sequence of dominant points (DpC) of C is computed by using an algorithm presented in [18]. DpC is then transform in the tangent space and the MpC curve is constructed. An incremental process is then used and each point of MpC is tested: if it is not an isolated point (in this case, it corresponds to an arc segment in C), the blurred segment recognition algorithm [19] permits to test if it can be added to the current blurred segment (which corresponds to an arc in C). If it is not possible, a new blurred segment starts with this point.

3 Ellipse detection

We present hereafter a new method for ellipse extraction from edge map of an image. It is based on three steps:

- Construction of a representation based on arc and line primitive of the edge map of input image.

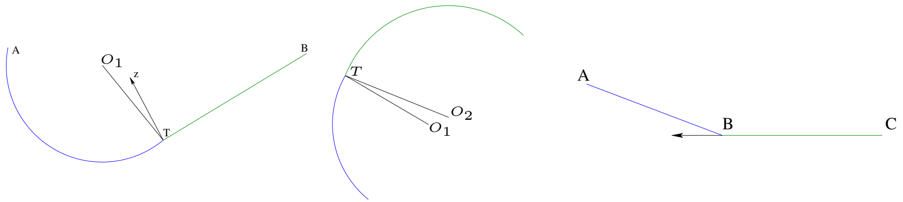
[3] Note this algorithm includes some corrections of algo. 3 of [1].

Algorithm 1: [1] Curve decomposition into arcs and lines [3].

```
Data: $C = \{C_1, \dots, C_n\}$-a digital curve, $\alpha_{max}$- maximal angle, $\nu$-width of
      blurred segments
Result: $ARCs$- set of arcs, $LINEs$- set of lines
begin
    Use [18] to detect the set of dominant points: $DpC = \{D_0, \dots, D_m\}$;
    $BS = \emptyset$;
    Transform $DpC$ in the tangent space as $T(DpC)$;
    Construct the midpoint curve $MpC = \{M_i\}_{i=0}^{m-1}$ of horizontal segments of
    $T(DpC)$;
    for $i$=0 to $m$-1 do
        $\{C_i\}_{b_i}^{e_i}$- part of $C$ wich corresponds to $M_i$;
        if ($BS \cup M_i$ is a blurred segment of width $\nu$ [19]) and
        $(|M_i.y - M_{i-1}.y| < \alpha_{max})$ then
            $BS = BS \cup M_i$;
        else
            $C^{'}$- part of C corresponding to $BS$;
            Push $C^{'}$ to $ARCs$;
            $BS = \{M_i\}$;
            if $(|M_i.y - M_{i+1}.y| > \alpha_{max})$ then
                Push $C_{b_i}C_{e_i}$ to $LINEs$;
                $BS = \{\emptyset\}$ ;
end
```

- Grouping of arcs and lines for detection of ellipse candidate based on geometric properties.
- Fitting of ellipse candidate based on least square fitting.

The first step is done by applying the decomposition of a curve into arcs and lines presented in the above section. We construct the corresponding edge image from input image by using Canny filter. This edge image is considered as a list of digital curves. Thanks to above technique [1], we can obtain a compact representation of this edge image based on arc and line primitives.



(a) Arc-line (b) Arc-arc (c) Line-line

Fig. 5. Arc and line grouping based on geometric property.

Grouping of arcs and lines based on geometric property. In this step, ellipse candidate is detected by grouping adjacent arcs and lines based on geometric property. Its main idea is to group the adjacent primitives that have a same tangent vector at the common extremity. In practice it is done by verifying the angle between two adjacent primitives.

Let us consider two adjacent primitives. There are three possible configurations: arc-arc, arc-line and line-line. We define the angle among two primitives that depends on its configuration as follows.

- *Arc-line:* Let us see Fig. 5.a. The arc has its center O_1. Two primitives share a common point T. Tz is perpendicular with TB. The angle among two primitives is define as $\angle O_1Tz$.
- *Arc-arc:* Let us see figure 5.b. Two adjacent arcs whose centers are O_1 and O_2 share common point T. The angle among these arcs is define as $\angle O_1TO_2$.
- *Line-line:* Let us see figure 5.c. The angle between two line segments AB and BC is $\pi - \angle ABC$.

Thanks to the notion of angle among two adjacent primitives and sequential property of the representation based on arcs and line segments, a linear scanning process is used for grouping adjacent primitives satisfying that the angle among two adjacent primitives does not exceed a fixed threshold. A such group of primitives that contains at least one primitive of arc is called an *ellipse candidate*. To avoid false positive with small detected ellipse, we use two constraints about the arcs in each group of primitives: the maximal radius of arc must be greater than 5 and the total subtending angle of arcs must be greater than $\frac{\pi}{5}$.

Fitting of ellipse candidate. For each ellipse candidate constructed from the extracted arcs and lines, we try to fit it by using least square fitting. Contrariwise to existing techniques based on least square fitting, we need a very small set of extracted points for fitting.

This good property is given by the representation of the curve by arc and line primitives. The set of extracted points for ellipse fitting is constituted from extremities of arcs and lines. Moreover, it contains also midpoints of arcs to reduce the approximated error between the fitting ellipse and the curve reconstructed by arcs and lines. Fig. 6 illustrates this strategy. The ellipse candidate is composed of segments AB, BC and arcs $\widehat{CD}$, $\widehat{DE}$. The data set contains A, B, C, D, E the extremities of arc and line primitives, M_1, M_2 the midpoints of arcs $\widehat{CD}$ and $\widehat{DE}$.

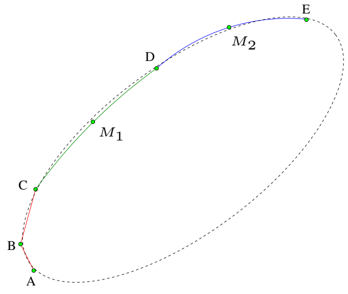


Fig. 6. Ellipse fitting based on least square fitting.

Concerning least square fitting, there are two main categories: algebraic fitting and geometric fitting (see [10]). Let us consider a general conic described by this implicit function: $F(A, X) = A \cdot X = ax^2 + bxy + cy^2 + dx + ey + f = 0$ where

Algorithm 2: Ellipse detection

```
Data: Img- a digital image, $\alpha_{max}$- maximal angle, $\nu$-width of blurred segments,
      $\beta_{max}$ threshold of angle [4]
Result: $E$ - set of ellipses
begin
    Use Canny to detect $edgeImg$ as edge image;
    Consider $edgeImg$ as a list of digital curves;
    Use algorithm 1 to represent $edgeImg$ by $Primitives$ - a sequence of arcs
    and lines; i=-1; $E = \emptyset$;
    while $i < sizeof(Primitives)$ do
        i++; id=i; numArc=0;
        while determineAngle(Primitives[i],Primitive[i+1])< $\beta_{max}$ do
            if Primitives[i] is an arc then numArc++; id++;
        if ($numArc > 0$) & ($id > i$) & ($maximal\_radius > 5$) &
        ($total\_substending\_angle > \frac{\pi}{5}$) then
            Collect $\{Primitives[j]\}_{j=i}^{id}$ as an ellipse candidate; $ES = \emptyset$;
            for j=i to id do
                Add the first point of Primitives[j] to ES;
                if Primitive[i] is an arc then Add the middle point of
                Primitives[j] to ES;
            Add the last point of Primitives[id] to $ES$;
            Use least square technique to fit $ES$ by a conic curve $\zeta$ [20];
            if $\zeta$ is an ellipse then Add $\zeta$ to $E$;
end
```

$A = [a, b, c, d, e, f]^t$, $X = [x^2, xy, y^2, x, y]^t$. $F(A, X_i)$ is defined as algebraic (resp. geometric) distance to the conic $F(A, X) = 0$. The least square fitting is used to minimize the sum of squared error distance: $\sum_{i=0}^{m} F(A, X_i)^2$ with none trivial solution $A \neq [0, 0, 0, 0, 0, 0]^t$. Many works have been proposed for minimizing this sum of square error. In our work, the ellipse is fitted from the extracted set of points by using the code of ellipse fitting proposed by Ohad Gal [20].

Proposed Algorithm. Algorithm 2 describes the ellipse detection based on the representation of an edge image by arc and line primitives. The scanning process is applied for the detection of ellipse candidates by grouping adjacent arcs and lines which have their adjacent angles smaller than a threshold (β_{max}). For each ellipse candidate, the algorithm tries to construct an approximated conic by using least square fitting on a small set of points constructed from extremities of arcs and lines and middle points of arcs. If the conic is an ellipse, it is considered as a detected ellipse. In practice, the threshold of angle is set from $\frac{\pi}{8}$ to $\frac{\pi}{5}$.

[4] By default, $\alpha_{max} = \frac{\pi}{4}$, $\nu = 1$, $\beta_{max} = \frac{\pi}{5}$ (see algo. 2).

4 Experimentations

We have experimented the proposed method on a 2.26 GHz CPU linux computer, with a 4Go of RAM. The results are illustrated in Fig. 7 and Fig. 8. From the input image (a), the edge image (b) is computed. Then, the detected arcs and lines from algorithm 1 are represented in (c). Afterwards, a scanning process is applied to group arc and line primitives to detect the ellipse candidate (d) by verifying the angle between adjacent primitives. Finally, for each ellipse candidate, the technique of least square fitting is applied to construct the fitting conic on the small set of extracted points (marked points in detected ellipses in figures Fig. 7.e and Fig. 8.e). Fig. 7.f and Fig. 8.f present the results obtained by other methods [21], [22].

Table 1 shows some information about processing of ellipse detection from the previous experiments. Thanks to the representation based on arcs and lines, the number of primitives for processing is reduced from 13699 (resp. 7778) to 308 (resp. 261) for image in Fig. 7 (resp. Fig. 8). In addition, the scanning process on the sequence of primitives is done in linear time due to its sequential property. After applying this process, the number of ellipse candidates is reduced to 27 (resp. 6) for image in Fig. 7 (resp. Fig. 8). Moreover, the average number of extracted points for ellipse fitting of each candidate is only 11.18 and 8.17.

The proposed method has two main advantages that guarantees its fastness:

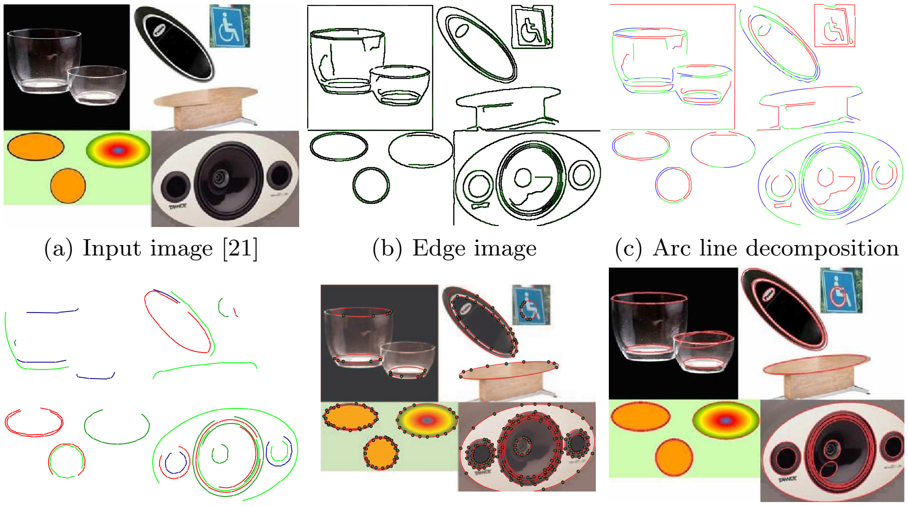
- *An efficient pre-processing step for the detection of ellipse candidates.* It is based on a fast method to represent the edge image by arcs and line segments and a linear scanning process for the detection of ellipse candidates.
- *A small set of extracted points for ellipse fitting.* Thanks to the representation based on arcs and lines, we don't need all of points corresponding to ellipse candidate for ellipse fitting. We only use extremity points of the primitives (arcs, lines) and midpoints of arcs for this task.

Table 1. Ellipse detection in images in figures 7, 8.

Figure	Image size	N^o of points in edge image	N^o of primitives	N^o of candidates	N^o of ellipses	N^o of points per candidate	Times (ms)
7	584x440	13699	308	27	23	11.18	880
8	508x417	7778	261	6	5	8.17	520

5 Conclusions

We have presented a promising new method for ellipse detection in images. The scanning process for detection of ellipse candidate is efficient because the representation based on arcs and lines allows us to work with a small number of primitives in relation with the number of points in edge image. In addition,



(a) Input image [21] (b) Edge image (c) Arc line decomposition

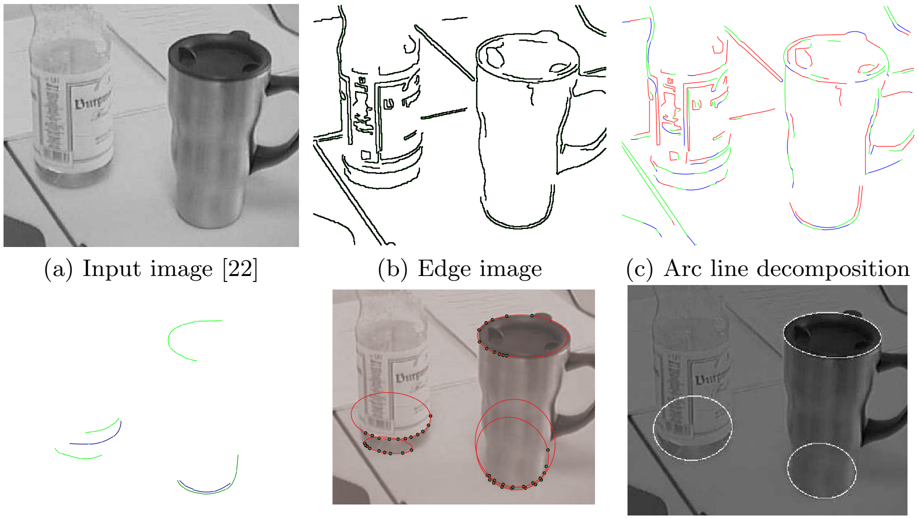
(d) Arc line grouping (e) Proposed result (f) Wu's method

Fig. 7. Comparison with Wu's method [21].

this process is done in linear time. Moreover, for each ellipse candidate, the least square fitting is not directly applied on all the points of the candidate. In future work we plane to perform other comparisons with different approaches and include a recent noise detection defined through the concept of meaningful scale [23].

References

1. Nguyen, T.P., Debled-Rennesson, I.: Decomposition of a curve into arcs and line segments based on dominant point detection. In: SCIA. Volume 6688 of LNCS. (2011) 794–805
2. Huang, C.L.: Elliptical feature extraction via an improved hough transform. PRL **10** (1989) 93–100
3. Aguado, A., Montiel, M., Nixon, M.: On using directional information for parameter space decomposition in ellipse detection. PR **29** (1996) 369–381
4. Daul, C., Graebling, P., Hirsch, E.: From the Hough Transform to a New Approach for the Detection and Approximation of Elliptical Arcs. CVIU **72** (1998) 215–236
5. Chia, A., Leung, M., Eng, H., Rahardja, S.: Ellipse detection with hough transform in one dimensional parametric space. In: ICIP. (2007) 333–336
6. Lu, W., Tan, J.: Detection of incomplete ellipse in images with strong noise by iterative randomized Hough transform (IRHT). PR **41** (2008) 1268–1279
7. Ahn, S., Rauh, W.: Geometric least squares fitting of circle and ellipse. PRAI **13** (1999) 987
8. Ciobanu, A., Shahbazkia, H., du Buf, H.: Contour profiling by dynamic ellipse fitting. In: ICPR. Volume 3. (2000) 750–753
9. Fitzgibbon, A.W., Pilu, M., Fisher, R.B.: Direct least square fitting of ellipses. PAMI **21** (1999) 476–480



(a) Input image [22] (b) Edge image (c) Arc line decomposition

(d) Arc line grouping (e) Proposed result (f) Libuda's method

Fig. 8. Comparison with Libuda's method [22].

10. Ahn, S., Rauh, W., Warnecke, H.: Least-squares orthogonal distances fitting of circle, sphere, ellipse, hyperbola, and parabola. PR **34** (2001) 2283–2303
11. Voss, K., Suesse, H.: Invariant fitting of planar objects by primitives. PAMI **19** (1997) 80–84
12. Lee, R., Lu, P., Tsai, W.: Moment preserving detection of elliptical shapes in gray-scale images. PRL **11** (1990) 405–414
13. Heikkila, J.: Moment and curvature preserving technique for accurate ellipse boundary detection. In: ICPR. Volume 1. (1998) 734–737
14. Zunic, J., Sladoje, N.: Efficiency of characterizing ellipses and ellipsoids by discrete moments. PAMI **22** (2000) 407–414
15. Rosin, P.L.: Measuring shape: ellipticity, rectangularity, and triangularity. Mach. Vis. Appl. **14** (2003) 172–184
16. Nguyen, T.P., Debled-Rennesson, I.: A linear method for segmentation of digital arcs. In: Computer Analysis of Images and Patterns. LNCS (2011) to appear.
17. Latecki, L., Lakamper, R.: Shape similarity measure based on correspondence of visual parts. PAMI **22** (2000) 1185–1190
18. Nguyen, T.P., Debled-Rennesson, I.: A discrete geometry approach for dominant point detection. Pattern Recognition **44** (2011) 32–44
19. Debled-Rennesson, I., Feschet, F., Rouyer-Degli, J.: Optimal blurred segments decomposition of noisy shapes in linear time. Comp. & Graphics **30** (2006) 30–36
20. Gal, O.: Matlab code for ellipse fitting (2003) http://www.mathworks.com-/matlabcentral/fileexchange/3215-fitellipse.
21. Wu, J.: Robust real-time ellipse detection by direct least-square-fitting. In: CSSE (1). (2008) 923–927
22. Libuda, L., Grothues, I., Kraiss, K.: Ellipse detection in digital image data using geometric features. (2006) 229–239
23. Kerautret, B., Lachaud, J.O.: Multi-scale analysis of discrete contours for unsupervised noise detection. In: IWCIA. Volume 5852 of LNCS. (2009) 187–200