An algorithm to decompose noisy digital contours

Phuc Ngo, Hayat Nasser, Isabelle Debled-Rennesson, Bertrand Kerautret

An algorithm to decompose noisy digital contours

Phuc Ngo[1,2], Hayat Nasser[1,2], Isabelle Debled-Rennesson[1,2],
Bertrand Kerautret[1,2]

[1] Université de Lorraine, LORIA, UMR 7503, Vandoeuvre-ls-Nancy, F-54506, France
[2] CNRS, LORIA, UMR 7503, Vandoeuvre-lès-Nancy, F-54506, France
{hoai-diem-phuc.ngo,hayat.nasser,isabelle.debled-rennesson,
bertrand.kerautret}@loria.fr

Abstract. From the previous digital contour decomposition algorithm, this paper focuses on the implementation and on the reproduction of the method linking to an online demonstration. This paper also gives improvement of the previous method with details on the intern parameter choice and shows how to use the C++ source code in other context.

Keywords: Digital Contours Representation, Image Vectorisation, Discrete Geometry

1 Introduction

The representation of a digital contour is a key point for the quality of image vectorisation algorithms. Such algorithms are mainly exploited in commercial or open source softwares such as *Adobe Illustrator* or *Inskape* when importing a bipmap image and converting it into a vectorized format like *postscrip* or *svg*. The quality of the resulting vectorisation depends on the contour geometry which is mostly related to the choice of geometric primitive types (segments, arcs or others). It also depends on the presence of noise which can degrade the quality of the resulting image in terms of primitive number or from visual quality point of view. Furthermore, the potential noise presence and the setting of user parameter can as well influence the result quality.

In this work, we focus on the contour representation based on two types of primitive which are arcs and segments by taking into account locally the potential noise presence in digital contours. A special attention is given on the implementation and reproduction of the proposed algorithm which was presented in previous work [12] with some new improvements.

In the following section we recall the main decomposition algorithm with its prerequisite notions before describing how to reproduce results (3). The experiments and details are given in section 4.

2 Decomposition algorithm [12, 14]

2.1 Prerequisite

Maximal blurred segment The notion of blurred segment is introduced in [3] as an extension of arithmetical discrete line [18] with a width parameter ν for noisy or disconnected digital contours. This notion is defined as follows.

Definition 1. *A sequence of points S_f is a* blurred segment of width ν *iff*

- $\forall (x, y) \in S_f$, $\mu \leq ax - by < \mu + \omega$ *where* $a, b, \mu, \omega \in \mathbb{Z}$ *and* $gcd(a, b) = 1$, *and*
- *the vertical (or horizontal) distance* $d = \frac{\omega - 1}{\max(|a|, |b|)}$ *equals to the vertical (or horizontal) thickness of the convex hull of S_f, and*
- $d \leq \nu$.

Given a discrete curve C, let $C_{i,j}$ be the sequence of points indexed from i to j in C. We denote the predicate "$C_{i,j}$ is a blurred segment of width ν" as $BS(i, j, \nu)$. Then, C_{ij} is said to be *maximal* if it can not be extended (i.e no point added) on the left nor on the right, and noted by $MBS(i, j, \nu)$. Illustration of blurred segment and maximal blurred segment are given in *Fig.* 1.
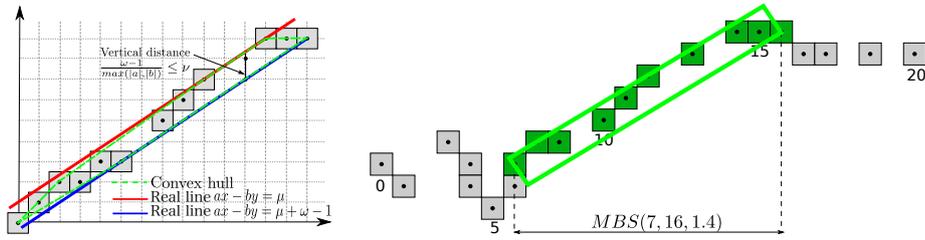

Fig. 1. Example of (left) blurred segment (greys points) and (right) maximal blurred segment (green points).

The sequence of the maximal blurred segments of width ν along a curve is called *width ν tangential cover* (see *Fig.* 2(a)). This structure is used in numerous discrete geometric estimators such as length, tangent, curvature estimators . . . (see [10] for a state of the art). In [14], an extension of width ν tangential cover, namely *adaptive tangential cover (ATC)*, is proposed to handle with noisy curves. In particular, ATC is composed of MSB of different width values deduced from the noise estimation of *meaningful thickness* [9] in which two parameters are set: (1) sampling step size –namely, **samplingStep**– and (2) maximal thickness –namely, **maxScale**– for which the shape is analysed for the meaningful thickness (see [9] for more details). Examples of ATC are given in *Fig.* 2(b) and *Fig.* 3. Still in [14], an algorithm is proposed to compute the ATC of discrete noisy curves.

Tangent space representation This notion is presented in [1, 11] as a tool for shape characterization and comparing polygonal shapes.

Definition 2. *Let $P = \{P_i\}_{i=0}^m$ be a polygon, l_i length of segment $P_i P_{i+1}$ and $\alpha_i = \angle(\overrightarrow{P_{i-1}P_i}, \overrightarrow{P_i P_{i+1}})$ such that $\alpha_i > 0$ if P_{i+1} is on the right side of $\overrightarrow{P_{i-1}P_i}$ and*
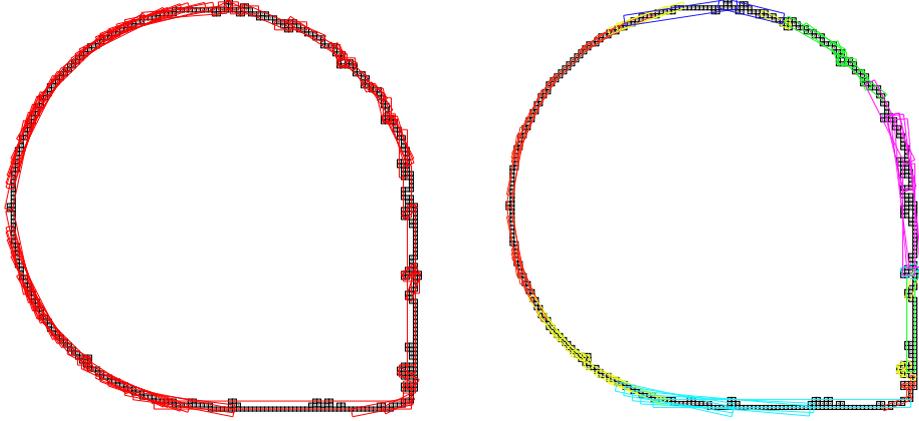
Fig. 2. Examples of (left) width ν tangential cover with $\nu = 2$ and (right) adaptive tangential cover (ATC) with different widths transmitted from meaningful thickness detection.
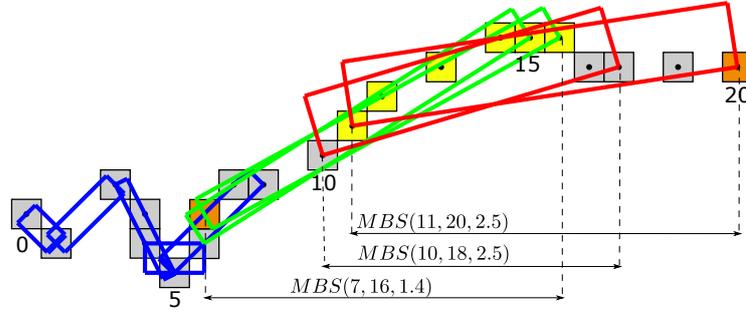

Fig. 3. Example of adaptive tangential cover (ATC) with three width values $\nu = 1, 1.4$ and 2.5 (in blue, green and red respectively). Considering the last three MBS of the ATC, in yellow are points in the common zone, and in orange are its two endpoints.

$\alpha_i < 0$ *otherwise. A tangent space representation $T(P)$ of P is a step function which is constituted of segments $T_{i2}T_{(i+1)1}$ and $T_{(i+1)1}T_{(i+1)2}$ for $0 \le i < m$ with*

- $T_{02} = (0,0)$,
- $T_{i1} = (T_{(i-1)2}.x + l_{i-1}, T_{(i-1)2}.y)$ *for* $1 \le i \le m$,
- $T_{i2} = (T_{i1}.x, T_{i1}.y + \alpha_i)$, $1 \le i \le (m-1)$.

Roughly speaking, the tangent space of a polygon is the representation of exterior angles versus segment lengths of the polygon as illustrated in *Fig.* 4.

2.2 Decomposing algorithm using tangent space representation

The proposed algorithm for decomposing digital curves into arcs and segments is composed of three steps. Firstly, the curve is simplified and represented by a polygon. Secondly, the the polygon is transformed into the tangent space in

(a) Input polygon   (b) Tangent space of (a)
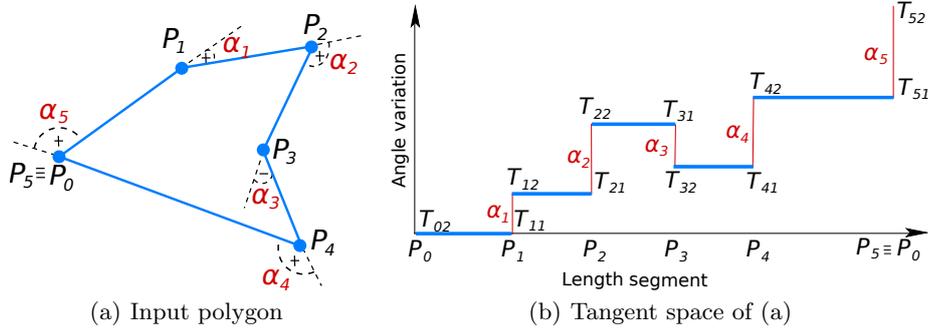
Fig. 4. Tangent space representation of a polygon.

which the analysis is performed to determined which parts of the polygon belong to arcs and which ones are segments. Finally, the fitting step is performed on arc parts to calculate the best fitting arcs that are approximated the input curve. These steps are detailed in the following.

Polygonal simplification This step consists in finding the characteristic points, namely *dominant points*, to form a polygon approximating/representing the given discrete curve. This step, called *polygonal simplification*, allows to use only the characteristic points, instead of all points of the curve, for the decomposition into arcs and segments using the tangent space.

Issued from the dominant point detection method proposed in [13, 17], an algorithm is presented in [14] to determine the characteristic points of noisy curves using ATC notion. The idea is that the candidates of dominant point are localized in common zone of successive MBS of the ATC of the given curve, and such common zone can be easily found by checking the starting and ending indices of MBS (see *Fig.* 3). Then, dominant point in each common zone is the point having the smallest curvature which is estimated as the angle between the point and the left and right endpoints of the left and right MBS constituting in the common zone, as illustrated in *Fig.* 5.

Due to the nature of ATC, common zones detected are mostly over-much and stay very near. As a consequence, the obtained dominant points in common zones are sometimes redundant which is presumably undesirable for polygonal simplification and in particular for curve decomposition algorithm. Therefore, we propose an optimization process to eliminate certain dominant points to achieve a high compression of the approximated polygon while preserving the principle angular deviation of the input curve. More precisely, each detected dominant point is associated to a weight describing its importance w.r.t. the approximating polygon of the curve. This weight is computed as the ratio of
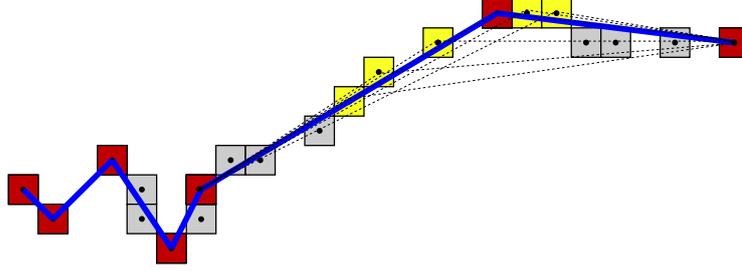
Fig. 5. Dominant points are detected as point having the smallest angle measure w.r.t. the two end points. Dominant points of the curve are depicted in red, and the common zone in *Fig.* 3 is depicted yellow.

Algorithm 1: Dominant point detection

Input: $C = (C_i)_{0 \leq i \leq n-1}$ discrete curve of n points
Output: D set of dominant points

1 $ATC(C) = \{MBS(B_j, E_j, .)\}_{j=0}^{m-1}$ adaptive tangential cover of C

   `/* Detection of dominant points in common zones              */`

2 $q = 0$; $p = 1$; $D = \emptyset$

3 **while** $p < m$ **do**

4       $D = D \cup \{min_{i \in [\![B_{p-1}, E_q]\!]} Angle(C_{B_q}, C_i, C_{E_{p-1}})\}$

5       $q = p - 1$

   `/* Optimization process to get significant dominant points    */`

6 $F = FOM_2(D)$

7 **repeat**

8       $\widehat{F} = F$

9       $D = D \setminus \{\min_{p \in D} Weight(p)\}$

10      $F = FOM_2(D)$

   until $F < \widehat{F}$;

ISSE[3] and angle with the two dominant point neighbours; $ISSE/angle$ [13]. Then, the optimization process removes one by one the dominant points of small weight until reaching the maximum FOM_2[4] criterion [19, 20]. The algorithm is given in *Alg.* 1.

The following auxiliary functions are used in *Alg.* 1:

---

[3] Integral sum of square errors is calculated by $ISSE = \sum_{i=1}^{n} d_i^2$, where d_i is distance from i-th curve point to approximating polygon. The smaller ISSE, the better description of the shape by the approximating polygon.

[4] Figure of merit 2 is calculated by $FOM_2 = \frac{CR^2}{ISSE}$, where CR is the compression ratio between number of curve points and number of detected dominant points. FOM_2 compromises between the low approximation error and the benefit of high data reduction.

- $Angle(C_{B_q}, C_i, C_{E_{p-1}})$ (Line 3) calculates the angle between three points C_{B_q}, C_i and $C_{E_{p-1}}$
- $FOM_2(D)$ (Line 5, 9) calculates $FOM_2 = CR^2/ISSE$ of the set D
- $Weight(p)$ (Line 8) calculates the $weight = ISSE/angle$ associated to p

Tangent space analysis In [15, 16], the following result is obtained for the tangent space representation of a set of sequential chords are proposed.

Proposition 1. *Let $P = \{P_i\}_{i=0}^m$ be a polygon, $l_i =| \overrightarrow{P_iP_{i+1}} |$, $\alpha_i = \angle(\overrightarrow{P_{i-1}P_i}, \overrightarrow{P_iP_{i+1}})$ such that $\alpha_i \leqslant \alpha \leqslant \frac{\pi}{4}$ for $0 \leq i < n$. Let $T(P)$ be the tangent space representation of P and $T(P)$ constitutes of segments $T_{i2}T_{(i+1)1}$, $T_{(i+1)1}T_{(i+1)2}$ for $0 \leq i < m$, $M = \{M_i\}_{i=0}^{m-1}$ the midpoint set of $\{T_{i2}T_{(i+1)1}\}_{i=0}^{m-1}$. P is a polygon whose vertices are on a real arc only if the set M belongs to a small width strip bounded by two real parallel lines, namely quasi collinear points (see Fig. 6).*


(a) Set of sequential chords of an arc   (b) Midpoint curve of (a)
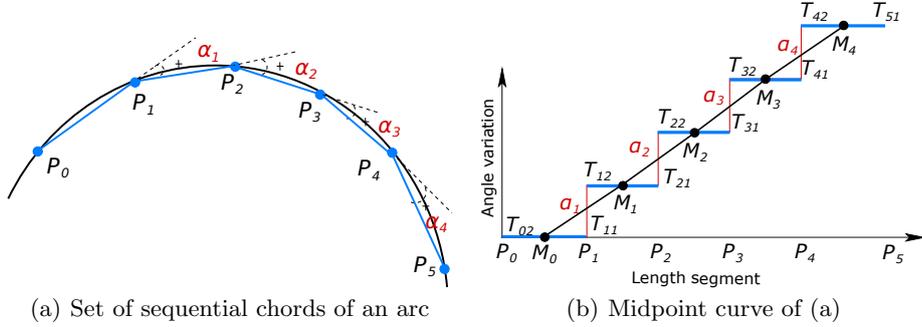
Fig. 6. Tangent space and the curve of midpoints of a set of sequential chords. If the midpoints is *quasi collinear*, then the initial points belong to a circular arc [15, 16].

Therefore, the arc detection of P becomes the problem of verifying the quasi collinearity of midpoints M in tangent space representation of P. In particular, this can be handled with the algorithm of recognition of MBS of width ν with midpoint curve in the tangent space.

It should be mentioned for quasi collinear midpoints in the tangent space that the more points are recognized in MBS of width ν –namely, **thickness**– the more *surely* arc is estimated. Therefore, to enhance the arc detection using the tangent space, we define a threshold for the number of elements in a set of quasi collinear midpoints –namely, **nbPointCircle**. Then, the set of quasi collinear midpoints is associated to an arc if its cardinality is greater than or equal to the threshold. Moreover, any two points are always collinear, then nbPointCircle≥ 3.

Still in [15, 16], it is observed for non linear midpoints that

- a midpoint is an *isolated point* if it has the difference of ordinate values between it and *one of the two* neighboring midpoints is higher than a threshold α –namely, **alphaMax**– and it corresponds to a junction of two primitives,
- a midpoint is a *full isolated point* if it is isolated with *all two neighbors* and it corresponds to a segment primitive (see *Fig. 7*).

(a) Input polygon   (b) Midpoint curve of (a)
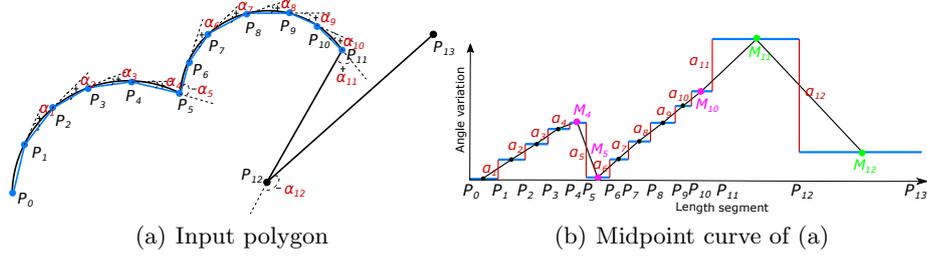
Fig. 7. Classification of midpoints in tangent space. Pink (resp. green) points are isolated points (resp. full isolated points) and correspond to junctions of two primitives (resp. segments), while black points are points of arcs.

Fitting of arcs *Prop.* 1 allows to determine the sequence of points belonging to an arc. A fitting process is performed to find the most appropriate arc in least square sense. It should be mentioned that for the continuity of the decomposition in between the primitives, we consider the junctions as dominant points. More precisely, let M_i to M_j be quasi-collinear midpoints in the tangent space representation. Let C_{b_i} and C_{e_i} (resp. C_{b_j} and C_{e_j}) be starting and ending dominant points that correspond to the midpoint M_i (resp. M_j). Then, junction points are C_{b_i} and C_{e_j} which are also ending points of the arc. In order to determine an arc for points from C_{b_i} to C_{e_j}, we need at least three points. Due to a high angular deviation near the endpoints of an arc, the fitting arc is performed using least square distance with one point in the central one-third portion of C_{b_i} and C_{e_j}. Such the best fitting arc associated to C_{b_i} and C_{e_j} can be denoted by $Arc(O, R, \beta_b, \beta_e)$, where O is the arc center, and R is the radius with angles go from β_b to β_e (1.e from C_{b_i} to C_{e_j}) which is calculated as follows

$$Arc(O, R, \beta_b, \beta_e) = min_{k \in [\![\frac{C_{b_i} + 2C_{e_j}}{3}, \frac{2C_{b_i} + C_{e_j}}{3}]\!]} d^2(C_k, \mathcal{C}_{(C_{b_i}, C_k, C_{e_j})}(O, R)) \quad (1)$$

where $\mathcal{C}_{(C_{b_i}, C_k, C_{e_j})}(O, R)$ is the circle passing through three points C_{b_i}, C_k, C_{e_j} and has the center O with radius R, and $d^2(C_k, \mathcal{C}_{(C_{b_i}, C_k, C_{e_j})}(O, R))$ is the square distance of the point C_k to the circle $\mathcal{C}_{(C_{b_i}, C_k, C_{e_j})}(O, R)$.

Furthermore, in some cases the approximation of a curve part by an arc may not be the optimal solution in particular when the part is quasi flat. Therefore, we propose a threshold of error approximation –namely **isseTol**– of the curve part using an arc and segments. More precisely, the curve part is an arc approximation if the ISSE by the arc is isseTol times smaller than this by segments.

Proposed algorithm *Alg.* 2 puts all steps together for decomposing a noisy discrete curve into arcs and segments. The algorithm scheme is given in *Fig.* 8.

Complexity analysis In *Alg.* 2, Line 1 to detect the dominant points of C using *Alg.* 1 is performed in $O(n \log n)$ [14] where n is the number of points of C. Lines 2–3 to transform the dominant points detected into the tangent space
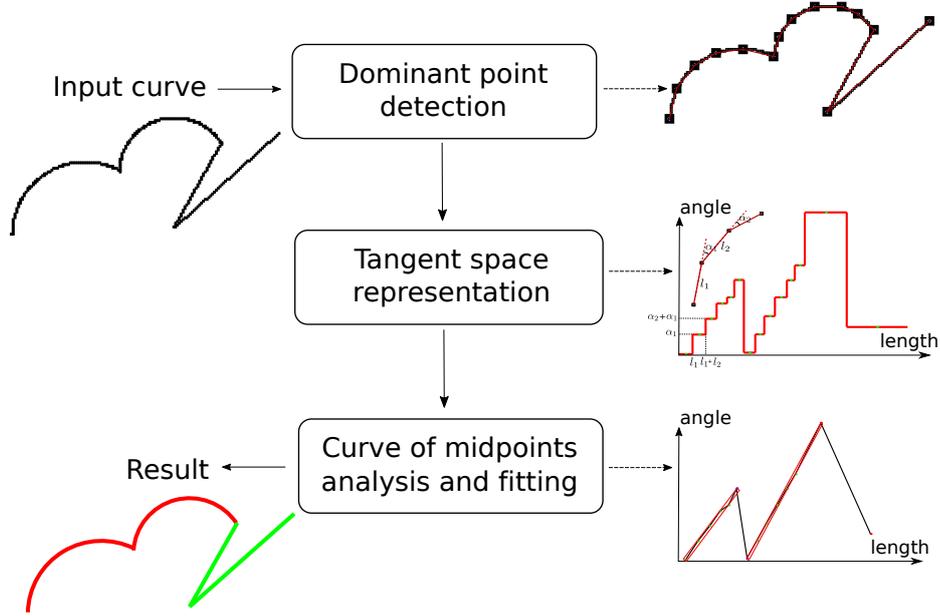
Fig. 8. Flowchart of the proposed algorithm.

and to compute the midpoint curve is in $O(m)$ with $m << n$ is the number of dominant points. The loop iterates over the midpoints to find the corresponding segments and arcs (Lines 4–23) is performed in $O(nm)$. More precisely, Lines 6 for verifying admissible angle in the tangent space is done in $O(1)$. Then, Lines 10 for the recognition of MBS of the midpoint curve can be done in $O(m)$ [4]. For finding a segment associated to a part of the curve, Lines 7, 20, 22 are executed in $O(1)$ since the segment is determined by the extremities $C_{b_i}C_{e_i}$. For finding a best-fitting arc associated to $pArc$ of the curve, Lines 14–18 are executed in $O(|pArc|)$ to compute the fitting error, and $|pArc| = n/3$ in the worst case. Finally, the complexity of the algorithm is $O(n\log n + nm)$.

3 Source code

3.1 Download and installation

The algorithm is implemented in C++ using the open source libraries DGtal [5] (Digital Geometry Tools and Algorithms) and ImaGene[6]. It is available at the github repository: `https://github.com/ngophuc/CurveDecomposition`. The installation is done through classical cmake procedure [7] (version ≥ 2.8) (see *INSTALLATION.txt* file).

---

[5] `http://dgtal.org`

[6] `https://gforge.liris.cnrs.fr/projects/imagene`

[7] `http://www.cmake.org`

Algorithm 2: Curve Decomposition into Arcs and Segments

Input: $C = (C_i)_{0 \leq i \leq n-1}$ a discrete curve of n points

       thickness width of MBS for collinear test in the tangent space

       alphaMax threshold of admissible angle in the tangent space

       nbCirclePoint threshold of midpoint number belonging to an arc

       isseTol threshold of approximation by an arc or segments

Output: $ARCs$ and $SEGs$ sets of arcs and segments of C

1 $ARCs \leftarrow \emptyset, SEGs \leftarrow \emptyset, pARCs \leftarrow \emptyset, MBS_\nu \leftarrow \emptyset$

2   Detect the dominant point D of C (see *Alg.* 1)

3   Transform D into the tangent space $T(D)$ (see *Def.* 2)

4   Construct the midpoint curve $\{M_i\}_{i=0}^{m-1}$ of $T(D)$ (see *Prop.* 1)

5   **for** $i \leftarrow 1$ *to* $m-2$ **do**

6 $C_{b_i}C_{e_i}$ is the part of C corresponding to M_i

7   **if** $(\mid M_i.y - M_{i-1}.y \mid > alphaMax) \& (\mid M_i.y - M_{i+1}.y \mid > alphaMax)$ **then**

8 $SEGs \leftarrow SEGs \cup \{C_{b_i}C_{e_i}\}$

9 $MBS_\nu \leftarrow \emptyset$

     else

11   **if** $MBS_\nu \cup \{M_i\}$ *is a MBS of thickness* **then**

12 $MBS_\nu \leftarrow MBS_\nu \cup \{M_i\}$

13 $pARC \leftarrow pARC \cup \{C_{b_i}C_{e_i}\}$

        else

15   **if** $\mid MBS_\nu \mid \geq nbCirclePoint$ **then**

16 $C_{b_i}C_{e_j}$ is the part of C corresponding to $pArc$

17 $Arc(O, R, \beta_b, \beta_e)$ is the best fitting arc for $C_{b_i}C_{e_j}$ (see *Eq.* 1)

18   **if** $ISSE(C_{b_i}, C_{e_j}, Arc(O, R, \beta_b, \beta_e)) <$

                $isseTol \sum_{C_{b_i}C_{e_i} \mid C_{b_i}C_{e_i} \in pARC} ISSE(C_{b_i}, C_{e_i}, \overline{C_{b_i}C_{e_i}})$ **then**

19 $ARCs \leftarrow ARCs \cup \{Arc(O_i, R_i, \beta_b, \beta_e)\}$

              else

21 $SEGs \leftarrow SEGs \cup \{C_{b_i}C_{e_i} \mid C_{b_i}C_{e_i} \in pARC\}$

           else

23 $SEGs \leftarrow SEGs \cup \{C_{b_i}C_{e_i} \mid C_{b_i}C_{e_i} \in pARC\}$

24 $pARC \leftarrow \emptyset$

3.2   Description and usage

In the package of source code, there are:

– **decompositionAlgorithm.h,cpp** contain the proposed algorithm
– **functions.h,cpp** contain the auxiliary functions used in the algorithm

     The proposed method (contained in **decompositionAlgorithm.h,cpp** files) is defined from the following functions:

– **adaptiveTangentCoverDecomposition** for computing of ATC of a curve
– **dominantPointDetection** for detecting dominant points (see *Alg.* 1)

– **dominantPointSimplification** for reducing the detected dominant points to obtain an optimal representation of the shape (see *Alg.* 1)
– **tangentSpaceTransform** for transforming the dominant point into tangent space representation
– **arcSegmentDecomposition** contains the implementation of *Alg.* 2
– **drawDecomposition** for drawing the decomposed arcs and segments

The executable file is generated in the **build** directory and named **testContourDecom**.

Input: A sdp file contains the several contours as lists of points:

```
x0 y0 x1 y1 ... xn yn # Points of contour 1
xn+1 yn+1 xn+2 yn+2 ... xm ym # Points of contour 2
```

Such lists of contour points are obtained after a contour extraction from image. Note that there is a new line for the last contour.

Command line from the CODESOURCES/build for running the decomposition algorithm on **contour.sdp** file with *samplingStep*=1.0, *maxScale*=10, *alphaMax*=0.78, *thickness*=0.2, *nbPointCircle*=3 and *isseTol*=4.0 is

```
./testContourDecom -i contour.sdp -d IMAGENEDIRECTORY
--samplingStep 1.0 -{}-maxScale 10  -a 0.78  -t 0.2 -n 3 -s 4.0
```

More details about the options are given in the command line helper.
Output: Several files are generated as output (in svg or eps format)

OutPts.svg	File of input points
ATC.svg	Result of ATC computation
DPnew.svg	Result of dominant point simplification
OnlyArcSeg.svg	Arc and segment decomposition result

4   Experimental results

We now present some experiments using *Alg.* 2 to decompose discrete curves into arcs and segments. Firstly, we show the decomposition results with noisy data. Secondly, the effects of the results to different sets of input parameters. Finally, some borderline cases are given.

In all experimental results, the arcs and segments are respectively colored in red and green. The contour points are extracted from images using the tool named *img2freeman* from DGtalTools [8] of DGtal library. It should be mentioned that the input of the algorithm is constituted of several sequences of points, thus different extraction methods can be used such as classical Canny contour detection [2] or smooth contour detection [5].

---

[8] http://dgtal.org/tools

4.1   Experiments on various noisy shapes

We present in this section the decomposition results obtained with various noisy shapes. More precisely, noise is added uniformly in the input images in order to test the robustness of the proposed algorithm towards noise. Two noise models are considered: Gaussian noise and a statistical noise model similar to the Kanungo noise [6], in which the probability P_d of changing the pixel located at a distance d from the shape boundary is defined as $P_d = \beta^d$ with $0 < \beta < 1$.

Fig. 9 (left) shows curves generated with different Gaussian noise of standard deviation $\sigma = 0, 10, 20, 30$, and *Fig.* 10 (left) shows curves obtained with different Kanungo noise levels for $\beta = 0.3, 0.5, 0.7$. We observe in *Figs.* 9–10 that the ATC allows a better model of tangential cover and thus more relevant to noise, and it can be seen that good reconstructions with arcs and segments are obtained even with an important and non-uniform noise on digital contours. More details regarding the parameter setting to obtain these results is given in *Sec.* 4.3.
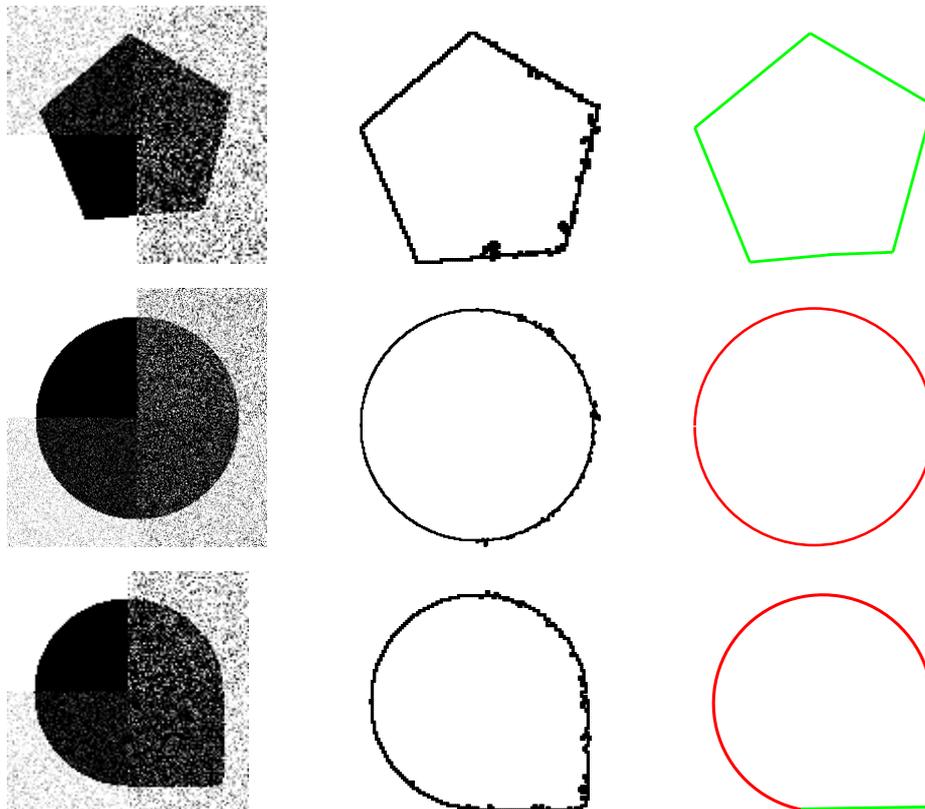

Fig. 9. Arcs and segments reconstruction of noisy curves. Left: input with Gaussian noise, middle: extracted curves, right: decomposition results.
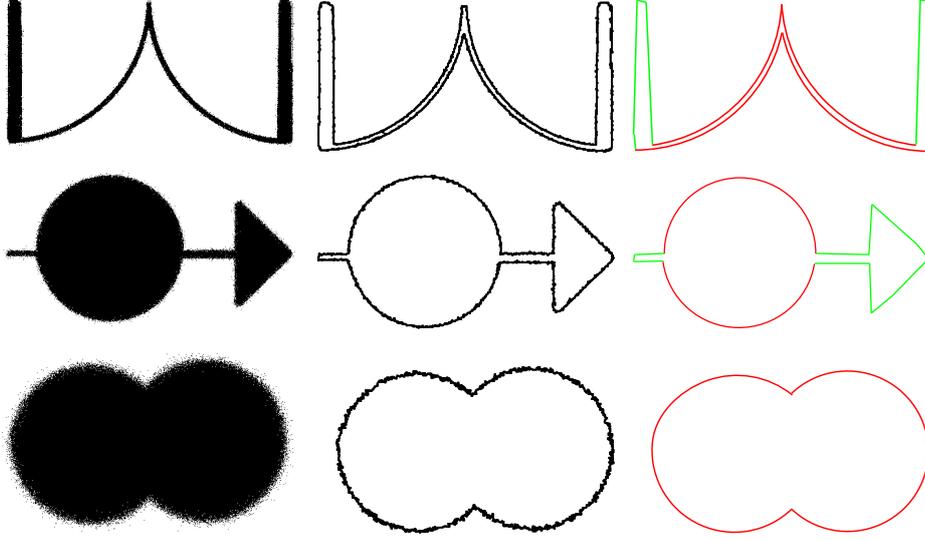
Fig. 10. Arcs and segments reconstruction of noisy curves. Left: input with Kanungo noise for $\beta = 0.3$ (first image), 0.5 (second image) and 0.5 (third image), middle: extracted curves, right: decomposition results.

4.2 Experiments on real images

Experiments are carried out as well on technical and real images. The results in *Figs.* 11–12 are obtained using the default parameters; *i.e.,*

- sampling step of meaningful thickness detection is $samplingStep = 1$
- maximal thickness of meaningful thickness detection is $maxScale = 15$
- admissible angle in the tangent space is $\alpha = \frac{\pi}{4} = 0.785$
- width for the quasi collinear test of MBS is $\nu = 0.2$
- minimum number of midpoints associated to an arc $nbCirclePoint = 3$
- ISSE tolerance between arc and segments approximation $isseTol = 4$

We observe a good reconstruction of the shapes by arcs and segments. It should be mentioned that the results of the decomposition algorithm depend a lot on the extracted curves from images. With the input images in *Figs.* 11–12 (left) a good threshold needs to be chosen to get such results for the decomposition.

4.3 Effects of parameter changes

The decomposition of digital contours into arcs and segments depends on the result of dominant point detected which is changed regarding the values of *maxScale* and particularly *samplingStep*. More precisely, the bigger value of *maxScale*, a smoother dominant point detection is obtained. For examples, *samplingStep*=0.2 or 0.5, we obtain a number of dominant points much greater than *samplingStep*=1 or 2 (see *Figs.* 13-14). The other parameters such as *alphaMax*,
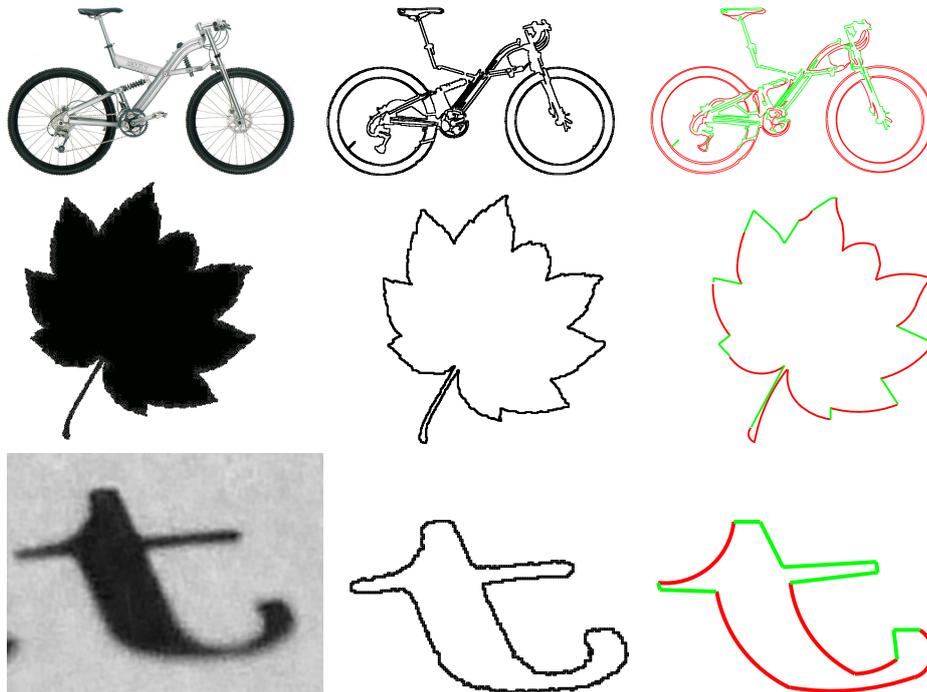
Fig. 11. Arcs and segments reconstruction on real images using default parameter setting. Left: input images, middle: extracted curves, right: decomposition results.

thickness, *isseTol* and *nbPointCircle* control the arc approximation of the input curve. *Fig.* 15 presents some borderline examples of the proposed algorithm in which a non-expected result is obtained for the decomposition of curves into arcs and segments using default parameter values. From *Figs.* 13–15, it can be clearly seen the sensibility of the decomposition w.r.t. to the parameter setting.

4.4 Image credits

 Gaussian noise sectors [7] generated *by shapeGenerator* of DGtalTools

 Kanungo noise added by *imgAddNoise* of ImaGene

 From [15]

 From [8]
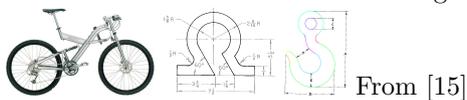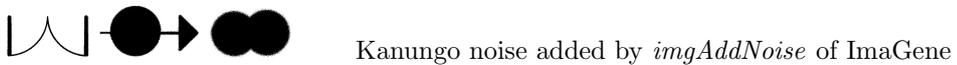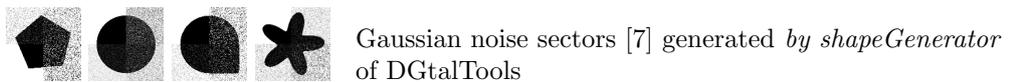
All other images by the authors.
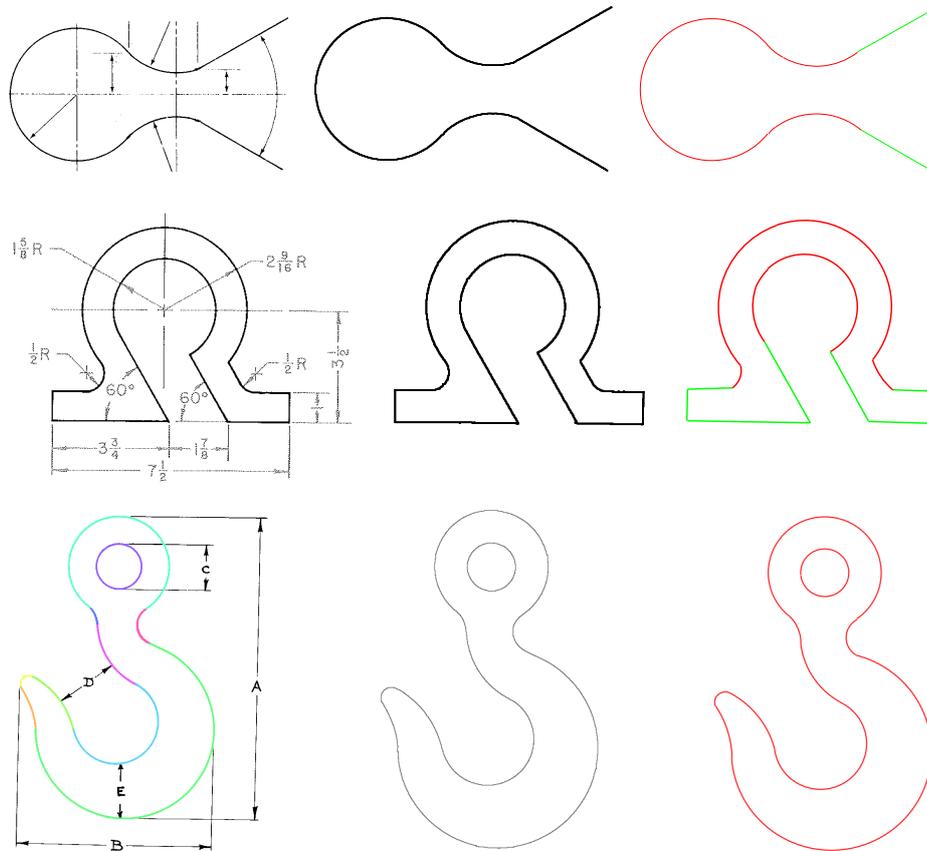
Fig. 12. Arcs and segments reconstruction on technical images using default parameter setting. Left: input images, middle: extracted curves, right: decomposition results.

5   Conclusion and perspectives

This work presents a synthesis of several papers [9, 12–17] about decomposition of noisy digital contours. Moreover, an online demonstration[9] is proposed as well as the details of the implementation. The role of the different parameters is studied and permits a better understanding of the presented method. Based on this study, a perspective is to reduce the number of input parameters and further an automatic approach to determine the best parameters adapted to a given contour. Another possible consideration is to integrate the topological information into the decomposition. More precisely, we would like to compute a arcs-and-segments decomposition which has the same topology as the input curve. We also want to extend to the 3D curves the proposed method. A first step could be the extension of the notion of adaptive tangential cover.

---

[9] `http://ipol-geometry.loria.fr/~phuc/ipol_demo/RRPR_demo`
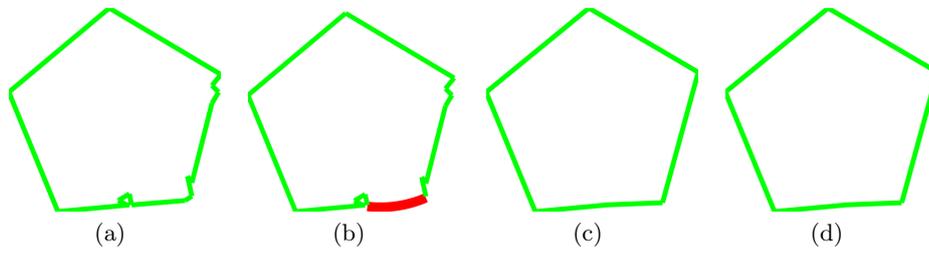
Decomposition	maxScale	samplingStep	alphaMax	thickness	isseTol	nbPointCircle
(a)	10	0.2	$\pi/8$	0.4	2.0	3
(b)	10	0.2	$\pi/4$	0.2	4.0	3
(c)	15	1.0	$\pi/4$	0.2	4.0	3
(d)	15	2.0	$\pi/8$	0.2	4.0	3
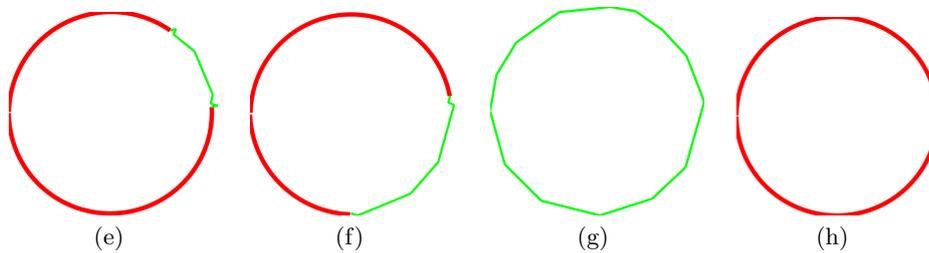
Decomposition	maxScale	samplingStep	alphaMax	thickness	isseTol	nbPointCircle
(e)	15	0.5	$\pi/8$	0.4	2.0	4
(f)	10	1.0	$\pi/4$	0.2	4.0	5
(g)	15	1.5	$\pi/8$	0.2	1.0	5
(h)	10	2.0	$\pi/4$	0.2	4.0	3
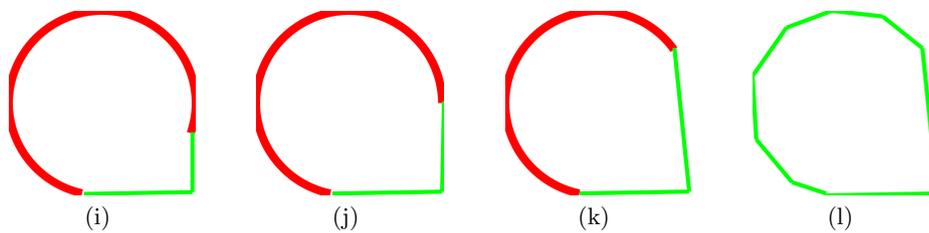
Decomposition	maxScale	samplingStep	alphaMax	thickness	isseTol	nbPointCircle
(i)	10	0.2	$\pi/4$	0.2	4.0	3
(j)	5	1.0	$\pi/4$	0.2	4.0	5
(k)	15	1.0	$\pi/4$	0.2	4.0	4
(l)	10	2.0	$\pi/8$	0.4	2.0	3

Fig. 13. Experiments on the sensibility to parameters of *Fig.* 9.

References

1. Arkin, E.M., Chew, L.P., Huttenlocher, D.P., Kedem, K., Mitchell, J.S.B.: An efficiently computable metric for comparing polygonal shapes. In: Proceedings of

Decomposition	maxScale	samplingStep	alphaMax	thickness	isseTol	nbPointCircle
(a)	10	0.5	$\pi/8$	0.4	2.0	3
(b)	10	1.0	$\pi/8$	0.2	4.0	3
(c)	15	2.0	$\pi/8$	0.4	2.0	4
(d)	10	2.0	$\pi/4$	0.2	4.0	3

Decomposition	maxScale	samplingStep	alphaMax	thickness	isseTol	nbPointCircle
(e)	10	0.5	$\pi/8$	0.2	4.0	3
(f)	10	1.0	$\pi/4$	0.2	2.0	3
(g)	10	1.0	$\pi/8$	0.2	4.0	4
(h)	15	2.0	$\pi/4$	0.4	1.0	4

Decomposition	maxScale	samplingStep	alphaMax	thickness	isseTol	nbPointCircle
(i)	15	0.5	$\pi/4$	0.2	4.0	3
(j)	10	1.0	$\pi/8$	0.5	4.0	3
(k)	10	1.0	$\pi/4$	0.5	6.0	4
(l)	15	2.0	$\pi/8$	0.4	4.0	4

Fig. 14. Experiments on the sensibility to parameters of *Fig.* 10.

the First Annual ACM-SIAM Symposium on Discrete Algorithms. pp. 129–137. SODA '90 (1990)

2. Canny, J.: A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence 8(6), 679–698 (1986), `http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4767851`

3. Debled-Rennesson, I., Feschet, F., Rouyer-Degli, J.: Optimal blurred segments decomposition of noisy shapes in linear time. Computers & Graphics 30(1), 30–36 (2006)

4. Feschet, F., Tougne, L.: Optimal time computation of the tangent of a discrete curve: Application to the curvature. In: DGCI. LNCS, vol. 1568, pp. 31–40 (1999)
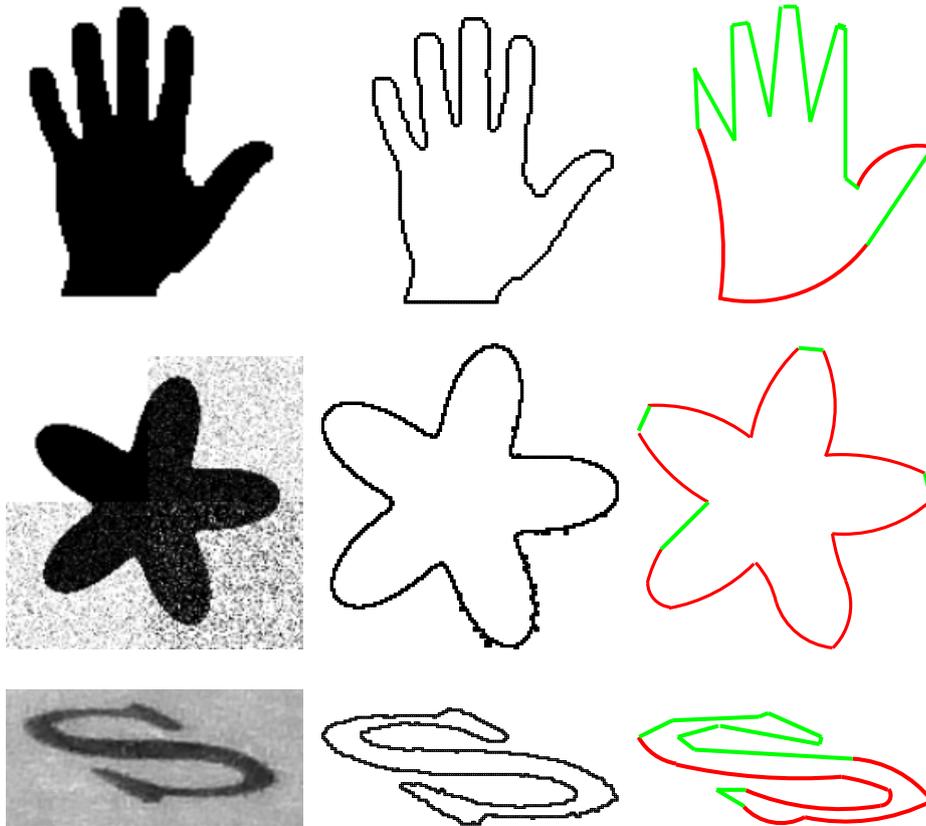
Fig. 15. Borderline cases of decomposition using default parameter setting. Left: input images, middle: extracted curves, right: decomposition results.

5. von Gioi, R.G., Randall, G.: Unsupervised smooth contour detection. Image Processing On Line (2016), preprint: `http://www.ipol.im/pub/pre/175/`

6. Kanungo, T., Haralick, R.M., Stuezle, W., Baird, H.S., Madigan, D.: A statistical, nonparametric methodology for document degradation model validation. IEEE Trans. Pattern Anal. Mach. Intell. 22(11), 1209–1223 (2000)

7. Kerautret, B., Lachaud, J.O.: Meaningful Scales Detection along Digital Contours for Unsupervised Local Noise Estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(12), 2379–2392 (Dec 2012), `http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6138862`

8. Kerautret, B., Lachaud, J.O.: Meaningful scales detection: an unsupervised noise detection algorithm for digital contours. Image Processing On Line 4, 98–115 (2014), 10.5201/ipol.2014.75

9. Kerautret, B., Lachaud, J.O., Said, M.: Meaningful thickness detection on polygonal curve. In: Pattern Recognition Applications and Methods. pp. 372–379 (2012)

10. Lachaud, J.: Digital shape analysis with maximal segments. In: Applications of Discrete Geometry and Mathematical Morphology - First International Workshop, WADGMM 2010, Istanbul, Turkey. pp. 14–27 (2010)

11. Latecki, L.J., Lakämper, R.: Shape similarity measure based on correspondence of visual parts. IEEE Trans. Pattern Anal. Mach. Intell. 22(10), 1185–1190 (2000)
12. Ngo, P., Nasser, H., Debled Rennesson, I.: A discrete approach for decomposing noisy digital contours into arcs and segments. In: Workshop on Discrete Geometry and Mathematical Morphology for Computer Vision. Tapei, Taiwan (2016)
13. Ngo, P., Nasser, H., Debled-Rennesson, I.: Efficient dominant point detection based on discrete curve structure. In: IWCIA '15. LNCS, vol. 9448, pp. 143–156 (2015)
14. Ngo, P., Nasser, H., Debled-Rennesson, I., Kerautret, B.: Adaptive tangential cover for noisy digital contours. In: DGCI '16. LNCS, vol. 9647, pp. 439–451 (2016)
15. Nguyen, T.P., Debled-Rennesson, I.: Arc segmentation in linear time. In: CAIP '10. LNCS, vol. 6854, pp. 84–92 (2010)
16. Nguyen, T.P., Debled-Rennesson, I.: Decomposition of a curve into arcs and line segments based on dominant point detection. In: SCIA '11. pp. 794–805 (2011)
17. Nguyen, T.P., Debled-Rennesson, I.: A discrete geometry approach for dominant point detection. Pattern Recognition 44(1), 32–44 (2011)
18. Reveillès, J.P.: Géométrie discrète, calculs en nombre entiers et algorithmique (1991), thèse d'état. Université Louis Pasteur, Strasbourg
19. Rosin, P.L.: Techniques for assessing polygonal approximations of curves. IEEE Trans. Pattern Anal. Mach. Intell. 19(6), 659–666 (1997)
20. Sarkar, D.: A simple algorithm for detection of significant vertices for polygonal approximation of chain-coded curves. Pattern Recogn. Lett. 14(12), 959–964 (1993)