An algorithm to decompose noisy digital contours

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

{hoai-diem-phuc.ngo, hayat.nasser, isabelle.debled-rennesson, bertrand.kerautret}@loria.fr

1. Motivations

Arcs and segments are the most appearing primitives in images

- Detection of shapes
  - medical imaging
  - technical images
  - manual drawings
- Character recognition
  - manual sketches
  - scanned documents

Fig 1: Arc and segment detection of an image

This decomposition allows a simple descriptor of object contours and facilitates their manipulation in image processing applications.

2. Concepts & method

The proposed method consists in decomposing the contours extracted from images, called *discrete curves* into arcs and segments. In particular, the decomposing algorithm aggregates mainly three following concepts:

1. *Adaptive tangent cover* (ATC) [1]: a discrete geometry tool allowing to study the geometrical characteristics of a curve
2. *Dominant points* (DP) [2]: the characteristic points of a curve allowing an approximation of the curve by a polygon
3. *Tangent space* (TS) [3]: a representation of the polygon issued by the dominant points of the curve allowing to detect the points on arcs and segments of the polygon

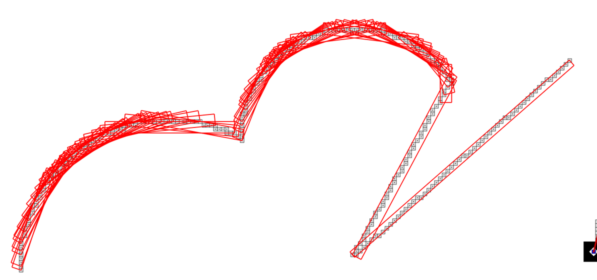


Fig 2: ATC

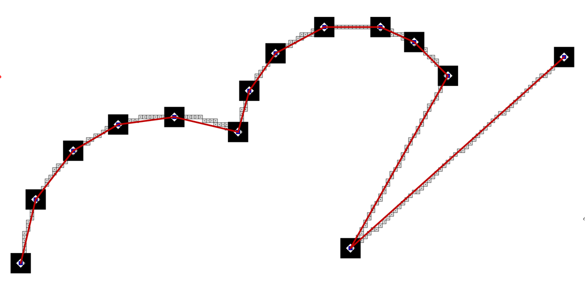


Fig 3: DP

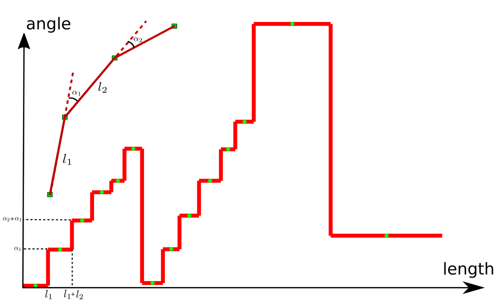


Fig 4: TS

References

[1] P. Ngo, H. Nasser, I. Rennesson-Debled, and B. Kerautret. Adaptive tangential cover for noisy digital contours. In *Proceeding of DGCI*, volume LNCS 9647, 2016.

[2] P. Ngo, H. Nasser, and I. Rennesson-Debled. Efficient dominant point detection based on discrete curve structure. In *Proceeding of IWCIA*, volume LNCS 9448, pages 143–156, 2015.

[3] T. P. Nguyen and I. Rennesson-Debled. Arc segmentation in linear time. In *Proceeding of CAIP*, volume LNCS 6854, pages 84–92, 2011.

3. Proposed algorithm

Algorithm 1: 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 midpoints belonging to an arc
$isseTol$ threshold of approximation by an arc or segments
Output: $ARCs$ and $SEGs$ sets of arcs and segments of C

$ARCs \leftarrow \emptyset, SEGs \leftarrow \emptyset, pARCs \leftarrow \emptyset, MBS_\nu \leftarrow \emptyset$
Detect the dominant point D of C
Transform D into the tangent space $T(D)$
Construct the midpoint curve $\{M_i\}_{i=0}^{m-1}$ of $T(D)$
for $i \leftarrow 1$ *to* $m-2$ **do**
    $C_{b_i}C_{e_i}$ is the part of C corresponding to M_i
    if $(\mid M_i.y - M_{i-1}.y \mid > alphaMax) \& (\mid M_i.y - M_{i+1}.y \mid > alphaMax)$ **then**
        $SEGs \leftarrow SEGs \cup \{C_{b_i}C_{e_i}\}$
        $MBS_\nu \leftarrow \emptyset$
    else
        if $MBS_\nu \cup \{M_i\}$ *is a MBS of* $thickness$ **then**
            $MBS_\nu \leftarrow MBS_\nu \cup \{M_i\}$
            $pARC \leftarrow pARC \cup \{C_{b_i}C_{e_i}\}$
        else
            if $\mid MBS_\nu \mid \geq nbCirclePoint$ **then**
                $C_{b_i}C_{e_j}$ is the part of C corresponding to $pArc$
                $Arc(O, R, \beta_b, \beta_e)$ is the best fitting arc for $C_{b_i}C_{e_j}$
                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**
                    $ARCs \leftarrow ARCs \cup \{Arc(O_i, R_i, \beta_b, \beta_e)\}$
                else
                    $SEGs \leftarrow SEGs \cup \{C_{b_i}C_{e_i} \mid C_{b_i}C_{e_i} \in pARC\}$
            else
                $SEGs \leftarrow SEGs \cup \{C_{b_i}C_{e_i} \mid C_{b_i}C_{e_i} \in pARC\}$
            $pARC \leftarrow \emptyset$

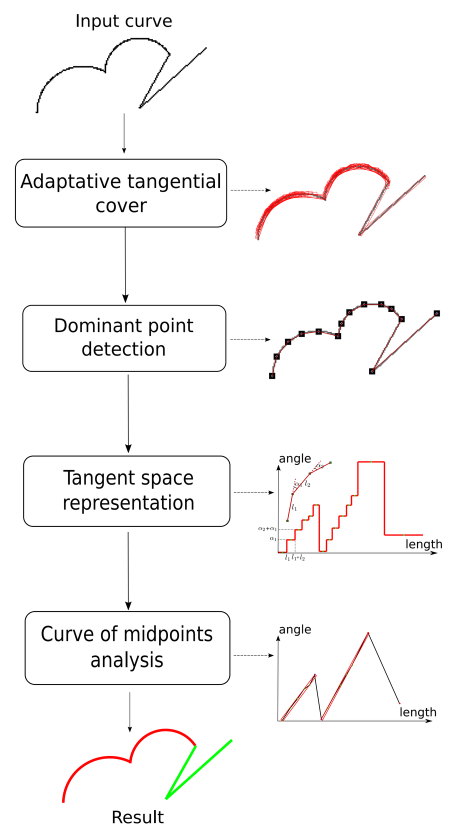


Fig 5: Algorithm's flowchart

Parameter (default value)	Signification
samplingStep (1)	Sampling step of noise detection
maxScale (15)	Maximal thickness of noise detection
thickness (0.2)	Width for the quasi collinear test
alphaMax (0.78)	Admissible angle in the tangent space
nbCirclePoint (3)	Threshold of midpoints belonging to an arc
isseTol (4)	Threshold of approximation by an arc or segments

4. Source code

Download and installation:

`https://github.com/ngophuc/CurveDecomposition`

Online demonstration:

`http://ipol-geometry.loria.fr/~phuc/ipol_demo/RRPR_demo`

Input: A file in sdp format containing several contours

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

Output: Several files in svg (or eps) format

__OutPts.svg	File of input points
__ATC.svg	Result of ATC computation
__DP.svg	Result of dominant point simplification
__OnlyArcSeg.svg	Arc and segment decomposition result

Code: Execution is named *testContourDecom*, to run the program on *contour.sdp* with *samplingStep=1.0, maxScale=10, alphaMax=0.78, thickness=0.2, nbPointCircle=3 and isseTol=4.0*

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

4. Experimental results

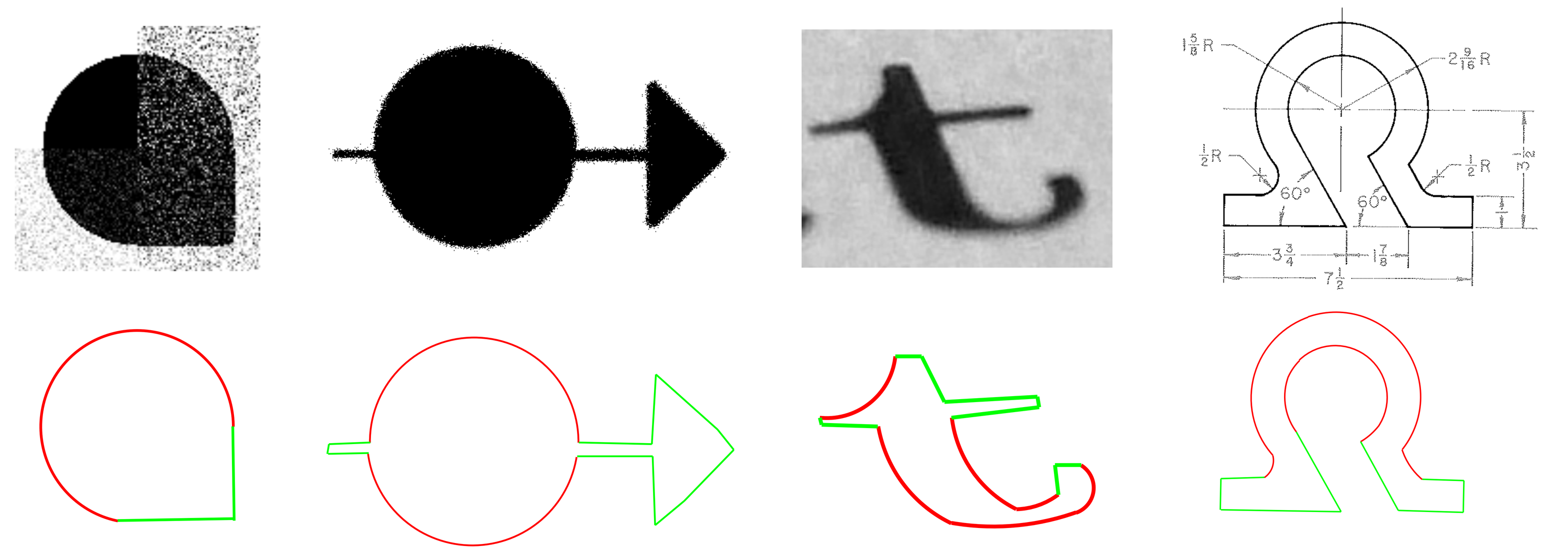
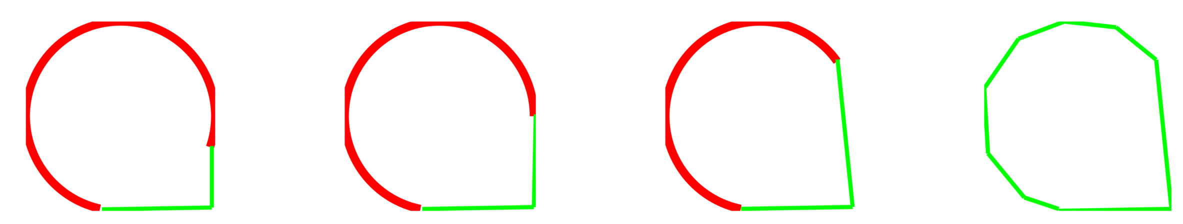
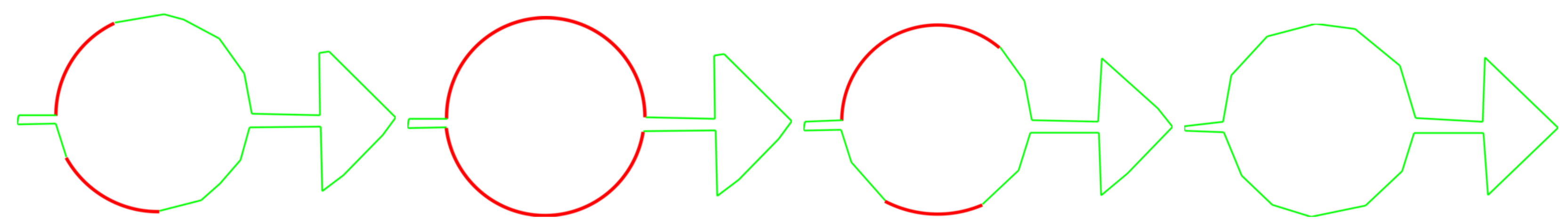


Fig 6: Experiments on different types of noise and images



Decomposition	*maxScale*	*samplingStep*	*alphaMax*	*thickness*	*isseTol*	*nbPointCircle*
left	10	0.2	$\pi/4$	0.2	4.0	3
middle left	5	1.0	$\pi/4$	0.2	4.0	5
middle right	15	1.0	$\pi/4$	0.2	4.0	4
right	10	2.0	$\pi/8$	0.4	2.0	3



Decomposition	*maxScale*	*samplingStep*	*alphaMax*	*thickness*	*isseTol*	*nbPointCircle*
left	10	0.5	$\pi/8$	0.2	4.0	3
middle left	10	1.0	$\pi/4$	0.2	2.0	3
middle right	10	1.0	$\pi/8$	0.2	4.0	4
right	15	2.0	$\pi/4$	0.4	1.0	4

Fig 7: Experiments on the parameter sensibility

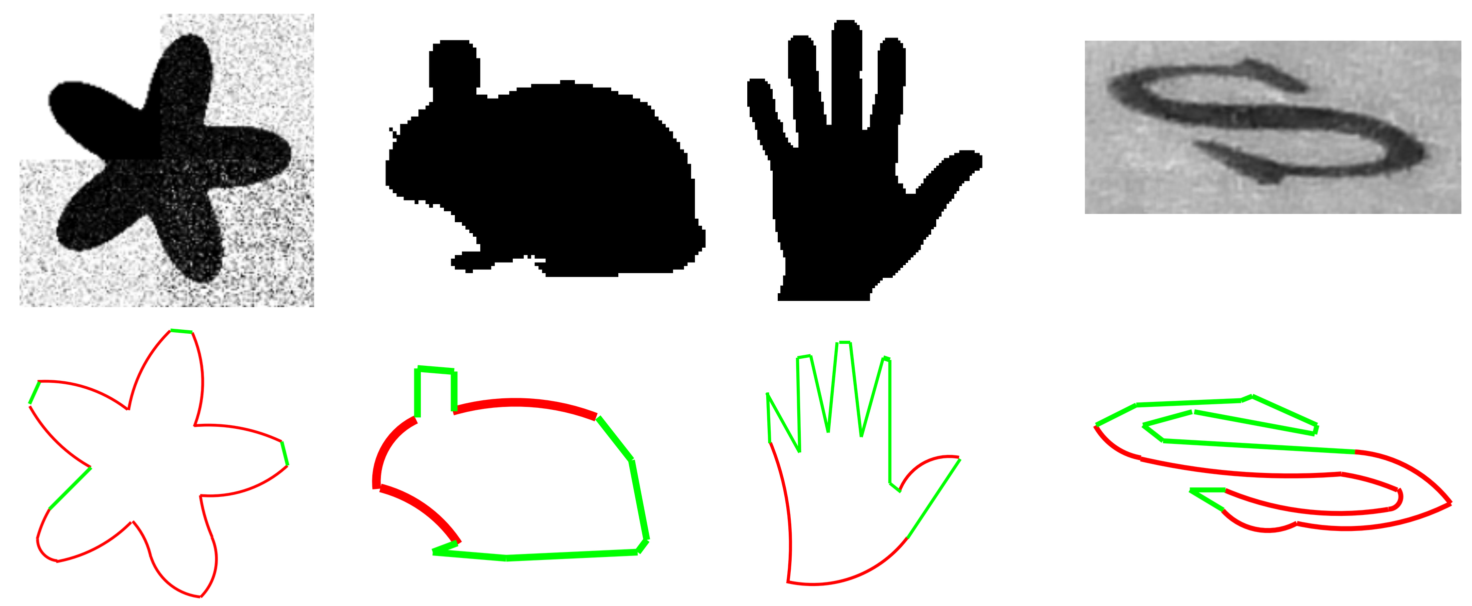


Fig 8: Borderline cases using default parameters

5. Conclusion & Perspectives

- Algorithm to decompose a noisy curve into arcs and segments
- A better descriptor of the curve comparing to dominant points

- Extension to other types of primitive: ellipse, parabola, ect.
- Extension to 3D curves
- Integration of topological information into the decomposition
- Application to image processing tasks: shape reconstruction, character recognition, ect.