Robust decomposition of a digital curve into convex and concave parts

Tristan Roussillon, Isabelle Sivignon, Laure Tougne

Laboratoire d'InfoRmatique en Image et Systmes d'information

UMR5205 CNRS/Universit de Lyon/Universit Lyon2

Universit Lyon2 - 5 avenue Pierre Mends France - 69676 Bron cedex, France

tristan.roussillon@liris.cnrs.fr

Abstract

We propose a linear in time and easy-to-implement algorithm that robustly decomposes a digital curve into convex and concave parts. This algorithm is based on classical tools in discrete and computational geometry: convex hull computation and Pick's formula.

Data

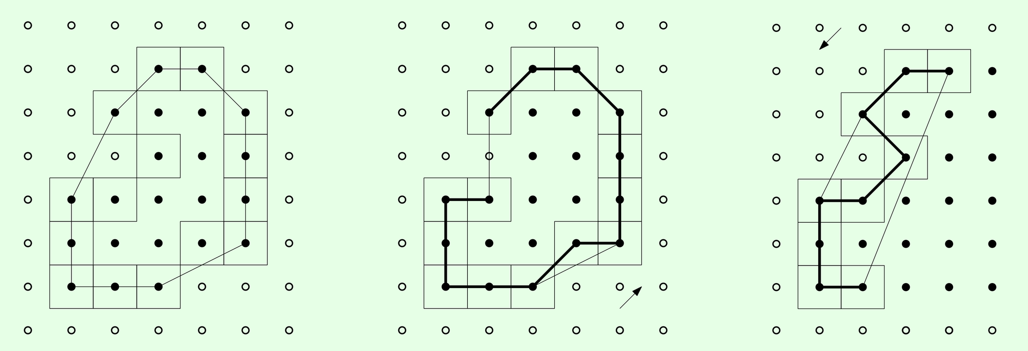


Fig.1: O (black disks) is bounded by C (squares). (left) The solid line that encloses $CH(O)$ depicts $\mathcal{CH}(O)$. (middle) $P \in C$ is 0-convex because $A(L(P)) = 0$. (right) $P \in C$ is neither 0-convex nor 0-concave because $A(L(P)) = 1$ and $A(R(P)) = 2$.

Measures

$$\text{convexity}(O) = \frac{A(CH(O)) - A(O)}{A(CH(O))} .$$

$$\text{convexity}(P) = A(L(P))/A(CH(O))$$
$$\text{concavity}(P) = A(R(P))/A(CH(O)) .$$

Function A returns the digital area $A(O)$ of a digital object O (*i.e.* the number of digital points belonging to O). The digital area is computed from the Euclidean area thanks to the Pick's formula.

Pick's formula

$$InAndOn(\mathcal{S}) = \mathcal{A}(\mathcal{S}) + On(\mathcal{S})/2 + 1 .$$

Numerical examples

In Fig.1 (left), $A_O = 14.5 + 15/2 + 1$, $A_{CH(O)} = 16.5 + 13/2 + 1$. Then, $convexity(O) = \frac{(24-23)}{24} = \frac{1}{24}$.
In Fig.1 (right), $A(L(P)) = 2 + 7/2 - 9/2 = 1$, $A(R(P)) = 5.5 + 2/2 - 9/2 = 2$. Then, $convexity(P) = \frac{1}{24}$ and $concavity(P) = \frac{2}{24}$.

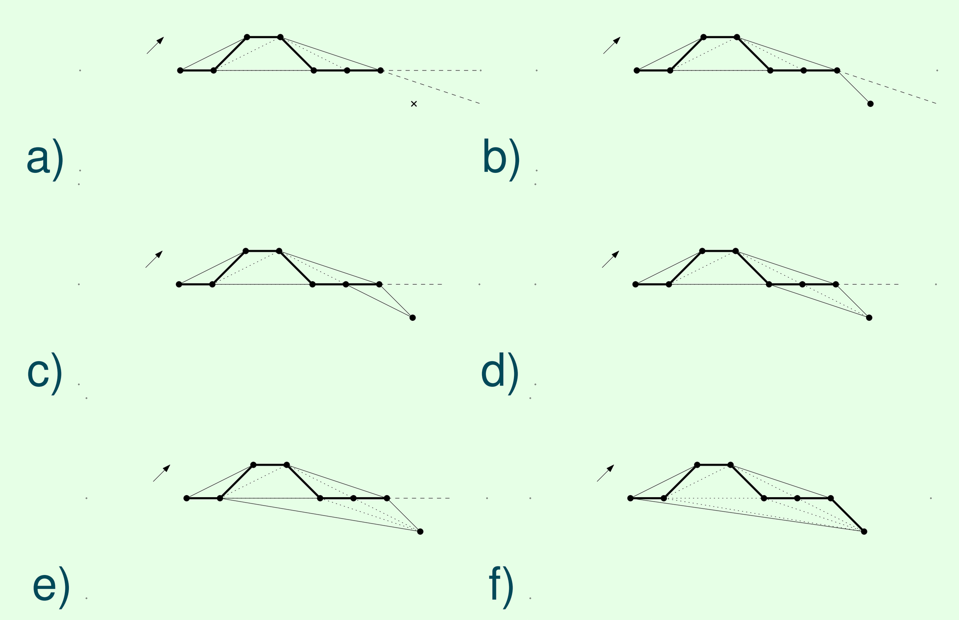
Update of $A(L(P))$ (resp. $A(R(P))$)

Algorithm 1: addPoint(LDeque, leftArea, n, p)
Input: p, the last of the n points of P
Output: $A(L(P))$

$last$ = LDeque.back();
LDeque.pop back();
$prev$ = LDeque.back();
$a = \mathcal{A}(prev, last, p)$;
while $(a < 0)$ **do**
  leftArea += $|a|$;
  $last = prev$;
  LDeque.pop back();
  $prev$ = LDeque.back();
  $a = \mathcal{A}(prev, last, p)$;
LDeque.push back(p);
return leftArea - ((n-LDeque.size())/2);

Running of the update algorithm

$A(L(P)) = 2 + 5/2 - 7/2 = 1$
$A(R(P)) = 1.5 + 4/2 - 7/2 = 0$



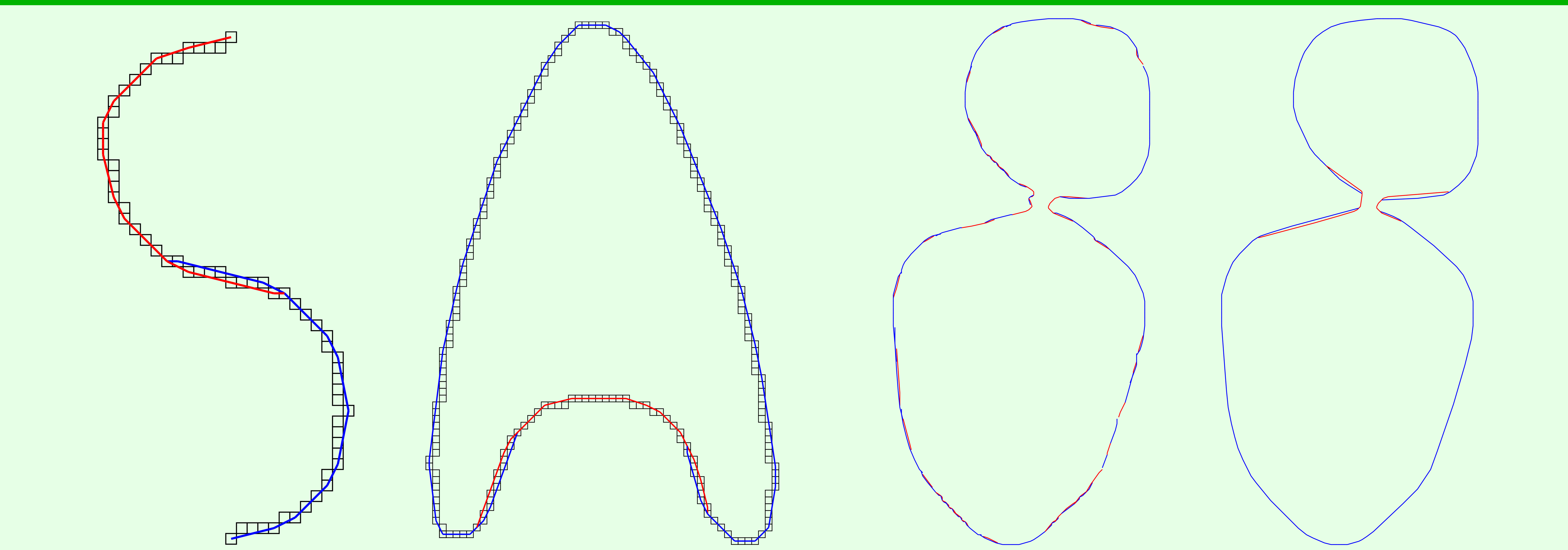
$A(L(P)) = 5 + 2/2 - 8/2 = 2$
$A(R(P)) = 1.5 + 5/2 - 8/2 = 0$

Shape decomposition

Algorithm 2: AdHocSegmentation(C, k)
Input: A curve C of n points and a threshold k

i = 0;
while $i < n$ **do**
  $P = C_0$; j = 0; i++;
  while $(A(L(P \cup C_i)) \leq k)$ *and (i<n)* **do**
    P += C_i; i++;
  while $(A(L(P \cup C_j)) \leq k)$ *and (j>0)* **do**
    P += C_j ; j- -;
  $P = C_i$; j = i; i++;
  while $A(R(P \cup C_i)) \leq k$ *and (i<n)* **do**
    P += C_i; i++;
  while $A(R(P \cup C_j)) \leq k$ *and (j>0)* **do**
    P += C_j ; j- -;

Results



Conclusion and Perspectives

- Linear in time and easy-to-implement algorithm.
- Can be used to robustly detect digital straight line of any thickness.
- Can be used into a multiresolution framework.