Robust decomposition of a digital curve into convex and concave parts

Tristan Roussillon, Isabelle Sivignon, Laure Tougne
Laboratoire d’InfoRmatique en Image et Systèmes d’information
UMR5205 CNRS/Université de Lyon/Université Lyon2
Université Lyon2 - 5 avenue Pierre Mendès 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 CH(O); (middle) P ∈ C is 0-convex because A(L(P)) = 0; (right) P ∈ C is neither 0-convex nor 0-concave because A(L(P)) = 1 and A(R(P)) = 2.

Measures
\[ \text{convexity}(O) = \frac{A(CH(O)) - A(O)}{A(CH(O))} . \]
\[ \text{convexity}(P) = \frac{A(L(P))}{A(CH(O))} . \]
\[ \text{concavity}(P) = \frac{A(R(P))}{A(CH(O))} . \]

Function A returns the digital area A(0) of a digital object 0 (i.e. the number of digital points belonging to 0). The digital area is computed from the Euclidean area thanks to the Pick’s formula.

Numerical examples
In Fig.1 (left), \( A_0 = 14.5 + 15/2 + 1 \). Then, \( \text{convexity}(O) = \frac{24-25}{2} = -0.5 \).
In Fig.1 (right), \( A(L(P)) = 2 + 7/2 - 9/2 = 1 \), \( A(R(P)) = 5.5 + 2/2 - 9/2 = 2 \). Then, \( \text{convexity}(P) = \frac{1}{2} \) and \( \text{concavity}(P) = \frac{2}{7} \).

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))
1 last = LDeque.back();
2 LDeque.pop_back();
3 prev = LDeque.back();
4 a = A(prev, last, p);
5 while (a < 0) do
6 leftArea += a;
7 last = prev;
8 LDeque.pop_back();
9 prev = LDeque.back();
10 a = A(prev, last, p);
11 LDeque.push_back(p);
12 return leftArea - (n-LDeque.size())/2;

Running of the update algorithm
\[ A(L(P)) = 2 + 5/2 - 7/2 = 1 \]
\[ A(R(P)) = 1.5 + 4/2 - 7/2 = 0 \]

Shape decomposition
Algorithm 2: AdHocSegmentation(C, k)
Input: A curve C of n points and a threshold k
1 i = 0;
2 while i < n do
3 \[ P = C_i; j = 0; i+++ \];
4 while \( A(L(P \cup C)) \leq k \) and \( (i < n) \) do
5 \[ P = C_i; i+++ \];
6 while \( A(R(P \cup C)) \leq k \) and \( (j < 0) \) do
7 \[ P = C_j; j++ \];
8 \[ P = C_i; j = 1; i+++ \];
9 while \( A(L(P \cup C)) \leq k \) and \( (i < n) \) do
10 \[ P = C_i; i+++ \];
11 while \( A(R(P \cup C)) \leq k \) and \( (j < 0) \) do
12 \[ P = C_j; j++ \];

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.