Robust Decomposition of a Digital Curve into Convex and Concave Parts*

Tristan Roussillon†, Laure Tougne
Laboratoire LIRIS, Université de Lyon, 5 Av Pierre-Mendès France 69676 Bron
{tristan.roussillon, laure.tougne}@liris.cnrs.fr

Isabelle Sivignon
Laboratoire LIRIS, CNRS, 8 Bd Niels Bohr 69622 Villeurbanne
isabelle.sivignon@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.

1 Introduction

This paper focuses on the problem of decomposition of a digital curve into convex and concave parts. The main motivation is that convex and concave parts of objects coarsely determine their meaningful parts [8, 2].

Even though the concept of digital convexity has been thoroughly studied the forty past years (see [11] for a deep bibliography on the topic), such decompositions into convex and concave parts have not been studied as much. The few previously proposed methods for decomposition into convex and concave parts are either sensitive to noise [1] or approximative [4].

The main objective of the paper is to design a fast algorithm that is able to robustly perform such a decomposition. In [8], robustness is reached by applying the discrete contour evolution on the digital curve viewed as a polygonal line. This kind of structural methods are powerful for shape matching but cannot return simple indices such as the number of convex and concave parts.

We propose an original measure-based approach. A measure that decreases with deviations from convexity is assigned to a digital arc. By thresholding the measure, the arc is said convex or concave within a certain tolerance depending on the noise.

2 Digital convexity

A binary image $I$ is viewed as a subset of points of $\mathbb{Z}^2$ called digital points, located inside a rectangle of size $M \times N$. A digital object $O \subseteq I$ is a 4-connected subset of $\mathbb{Z}^2$. Its complementary set $\bar{O} = I \setminus O$ is the so-called background. The boundary $C$ of $O$, defined as the 8-connected circular list of digital points having at least one 4-neighbor in $\bar{O}$, is a digital curve. A connected subset $P$ of $C$ is a digital arc.

In the Euclidean plane, convexity is well defined. However several definitions of digital convexity exist (see [11] or [3]). Our definition is equivalent to the ones given, for instance, in [12]:

**Definition 1** (Fig. 1.a) Let $CH(O)$ be the set of digital points located inside the Euclidean convex hull of $O$, $CH(O)$. $O$ is digitally convex if and only if $CH(O)$ only contains digital points belonging to $O$.

Given a digital object $O$, the measure of digital convexity is defined as the number of points in $CH(O)$ but not in $O$, normalized by the size of $CH(O)$ (a missing pixel

---

*Work supported by the GEODIB ANR project.
†Author supported by a grant from the DGA.
does not have the same influence on the convexity for a small and a large object).

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

where function \( A \) is defined as follows:

**Definition 2** The digital area \( A(O) \) of a digital object \( O \) is the number of digital points belonging to \( O \).

Thanks to these definitions of digital convexity (Eq. 1) and of digital area (Def. 2), it is clear that, (i) \( \text{convexity}(O) \) depends on the area of the concavities, (ii) \( \text{convexity}(O) = 0 \) if and only if \( O \) is digitally convex and \( 0 < \text{convexity}(O) < 1 \) otherwise, (iii) the measure is convergent while the resolution increases (since \( A(O) \) and \( A(CH(O)) \) are convergent [7]). (iv) the measure is invariant under rigid transformations at infinite resolution and quasi-invariant otherwise.

Def. 3 is the analog of Def. 1 for digital arcs and defines convex and concave arcs.

**Definition 3** (Fig. 1c) Let \( P \) be an oriented digital arc. The polygonal line linking all digital points of \( P \) is denoted by \( \mathcal{P} \) and the shortest polygonal line linking the first and last digital point of \( P \), such that \( \mathcal{P} \) is located on its left (resp. right) side is denoted by \( \mathcal{L}(P) \) (resp. \( \mathcal{R}(P) \)). \( P \) is digitally convex (resp. concave) if there is no digital point between \( \mathcal{P} \) and \( \mathcal{L}(P) \) (resp. \( \mathcal{R}(P) \)).

Suppose that \( P \) is part of the boundary \( C \) of a digital object \( O \). Moreover, suppose that \( P \) is oriented such that \( O \) is on the left of \( P \). Let \( \mathcal{P}_{\text{left}} \) (resp. \( \mathcal{P}_{\text{right}} \)) be the set of digital points located between \( \mathcal{P} \) and \( \mathcal{L}(P) \) (resp. \( \mathcal{R}(P) \)). In Fig. 1b, \( P \) is digitally convex because no digital point is between \( \mathcal{P} \) (bold) and \( \mathcal{L}(P) \) (superimposed to \( \mathcal{P} \)): \( \mathcal{P}_{\text{left}} = \emptyset \). In Fig. 1c, \( P \) is neither digitally convex (\( \mathcal{P}_{\text{left}} = 1 \)) nor concave (\( \mathcal{P}_{\text{right}} = 2 \)).

Eq. 2 is the analog of Eq. 1 for digital arcs.

\[ \text{convexity}(P) = \frac{A(P_{\text{left}})}{A(CH(O))} \]

\[ \text{concavity}(P) = \frac{A(P_{\text{right}})}{A(CH(O))} . \]  

Notice that the normalization by \( A(CH(O)) \) ensures the scale-invariance of this measure. Definition 4 relaxes this definition introducing a parameter \( \alpha \). We will see in section 4 that this definition leads to a robust decomposition algorithm.

**Definition 4** \( P \) is \( \alpha \)-digitally convex (resp. concave) if \( \text{convexity}(P) \leq \alpha \) (resp. \( \text{concavity}(P) \leq \alpha \)).

For \( \alpha = 0 \), Def. 4 is equivalent to Def. 3.

### 3 Efficient computation of Eq. 1

The input is a digital curve \( C \) that bounds a digital object \( O \). The algorithm runs in two steps before using Eq. 1: (i) computation of \( CH(O) \), (ii) computation of \( A(O) \) and \( A(CH(O)) \).

Since \( CH(O) \) is the digitization of \( CH(O) \), the problem is to compute such convex hull. In classical computational geometry, the convex hull computation of \( n \) points requires a \( \mathcal{O}(n \log n) \) time. However, in the digital grid, we may take profit of the intrinsic order of the digital points [5] or of its arithmetical properties [1] in order to compute such a hull in a \( \mathcal{O}(n) \) time.

As \( C \) and \( CH(O) \) may be represented by a polygon whose vertices are digital points, Pick’s formula [10] is used to efficiently compute \( A(O) \) and \( A(CH(O)) \):

\[ \text{InAndOn}(S) = A(S) + \text{On}(S)/2 + 1 . \]  

\( S \) is a polygon the vertices of which are digital points. Functions \( \text{InAndOn()} \) returns the number of digital points located on or strictly inside \( S \) and \( \text{On()} \) returns the number of digital points located on \( S \). Function \( A(S) \) returns the Euclidean area of \( S \). In Fig. 1a, \( A_O = 14.5 + 15/2 + 1, A_{CH(O)} = 16.5 + 13/2 + 1 \). Then, \( \text{convexity}(O) = \frac{(24-23)}{21} = \frac{1}{21} \).

The whole process is linear in time, because computing \( CH(O) \) is linear in time, computing \( A(O) \) and \( A(CH(O)) \) as well as the number of vertices of the corresponding polygons is linear in time, and finally, applying Eq. 1 and Eq. 3 is constant in time.

### 4 On-line algorithm

From a digital arc \( P \), we incrementally compute three polygonal lines: \( \mathcal{P}, \mathcal{L}(P) \) and \( \mathcal{R}(P) \). In Fig. 2, these three polygonal lines are depicted with solid lines.
Algorithm 1: addPoint2LDeque(p, n)

Input: p, end point of a digital arc P of n points

Output: A(P_{left})

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

we apply Pick’s formula (Eq. 3) on each non degenerate polygon. After a simple calculus, the last line of algorithm 1 is derived. Doing this, we assume that method size() of LDeque (resp. RDeque) returns all the vertices of L(P) (resp. R(P)), which is not always true: all the digital points not belonging to P and located on an edge of L(P) or R(P) are not taken into account. The measure may be made exact with some computations of greatest common divisor, which increases the complexity. But in practice, this approximated measure is used, because it is both very close and slightly less sensitive to small concavities (Fig. 3).

5 Shape Decomposition Process

The digital points are processed one by one along C in a counter-clockwise orientation. The core of the shape decomposition process runs in two steps: (i) the digital points that follow a starting point p, are pushed to the back of LDeque and RDeque until convexity(P) > α. Let us denote by q the last point of P, such that P is α-digitally convex. In order to make P maximal (P cannot remain digitally convex when a point is added in front or behind P), we perform the following second step: (ii) the digital points that precede p are pushed to the front of LDeque and RDeque until convexity(P) > α. When the latter expression is true, the growth of P as an α-digitally convex part stops. This algorithm is reinitialized at q and begins the detection of an α-digitally concave part P′, updating concavity(P′) as long as an insertion to the back, then to the front of LDeque and RDeque is possible, and so on.

Since a digital point is processed twice at most, the whole decomposition is done in O(n), where n denotes...
the number of digital points belonging to $C$.

Figures 4 and 5 illustrate this shape decomposition process respectively with synthetical digital curves and real-world images. For visualization purpose, $\forall P, L(P)$ and $R(P)$ are drawn. For “A” [3, 2] and “S” (Fig. 4) idealized digital shapes, the natural convex and concave parts are perfectly retrieved with $\alpha = 0$. In real-world images, a lot of meaningless convex and concave parts are retrieved with $\alpha = 0$ (Fig. 5.a). Slightly increasing $\alpha$ enables to get the expected convex and concave parts (Fig. 5.b).

6 Conclusion and Perspectives

An original, linear in time and easy-to-implement algorithm that robustly decomposes a digital curve into convex and concave parts is presented. The two main perspectives are the following: on the one hand, we think that our algorithm may be used to robustly detect digital straight line of any thickness, since a digital straight line is expected to be both digitally convex and concave, and on the other hand, we think that the multiresolution approach allowed by running our procedure for various values of $\alpha$ in a given range would be of great interest for shape representation.

References


