

Digital Convexity and Cavity Trees

Gisela Klette

AUT University, Private Bag 92006
Auckland 1142, New Zealand

Abstract. The notion *convexity* has a long history in mathematics. It is a useful concept to describe shapes, functions, smoothness of curves or boundaries, and it has applications in many fields. Researchers apply different definitions for *digital convexity* to adapt known concepts from the continuous space, and to make use of proven theories and results. We review different approaches and we propose cavity trees for (a) analyzing the convexity of digital objects, (b) to decompose those objects into meaningful parts, and (c) to show an easy way to find convex and concave parts of a boundary of a digital region.

Keywords: shape analysis, feature extraction, minimum-perimeter polygon, minimum-length polygon, cavity tree, digital convexity, geometric estimators.

1 Introduction

Convexity is an important notion in many areas of mathematics, physics and related disciplines and applications. Ancient Greeks already knew that there are only five regular convex polyhedra where the surface consists of congruent regular polygons. The first proper definition of a *convex curve* in 2D space and of a *convex surface* in 3D space goes back to Archimedes [17]. According to Archimedes, a curve in the plane is convex “if any two curve points are taken, then either all the straight lines connecting the points fall on the same side of the curve, or some fall on one and the same side while others fall on the line itself but none on the other side”. This definition is valid for simple curves or arcs in the Euclidean plane.

A common definition for a convex set in \mathbb{R}^n is the following:

Definition 1. A set $S \subseteq \mathbb{R}^n$ is convex iff¹ with any two points p and q belonging to S , the entire line segment joining p and q also lies in S .

Often, also the following equivalent version is used:

Definition 2. A subset $S \subseteq \mathbb{R}^n$ of points is convex iff S is equal to the intersection of all half planes containing S .

¹ Read: if and only if.

The convex hull $CH(S)$ of a set S of points is the smallest (by area) convex polygon P that contains S .

Since Archimedes, research on convexity in continuous space continued and defined at the end of the 19th century finally its own field in mathematics, the *convexity theory*. There are fundamental differences between concepts and applications in the Euclidean space and the digital space. It is not trivial to make use of results of convexity theory for digital image analysis purposes.

We distinguish between “digital convexity of polygonal lines”, lines which represent parts of a boundary of a 2D digital object, and “digital convexity of a region”, defined with the intention to analyze digital shapes of objects in an image.

This paper reviews different approaches for defining digital convexity and proposes a characterization based on cavity trees which are kind of by-products of a recursive computation of minimum-length polygons (MLP).

2 Preliminaries

Digital imaging uses models of different digitization methods for discussing concepts for property measurements. An object can be modeled as an inner grid polygon and an outer grid polygon, normally defined in a way like simulating an inner and outer Jordan digitization [7].

An *object* is considered to be a digitization of a measurable set $S \subset \mathbb{R}^2$ into a regular grid. We assume that a 2D picture P is composed of equally sized

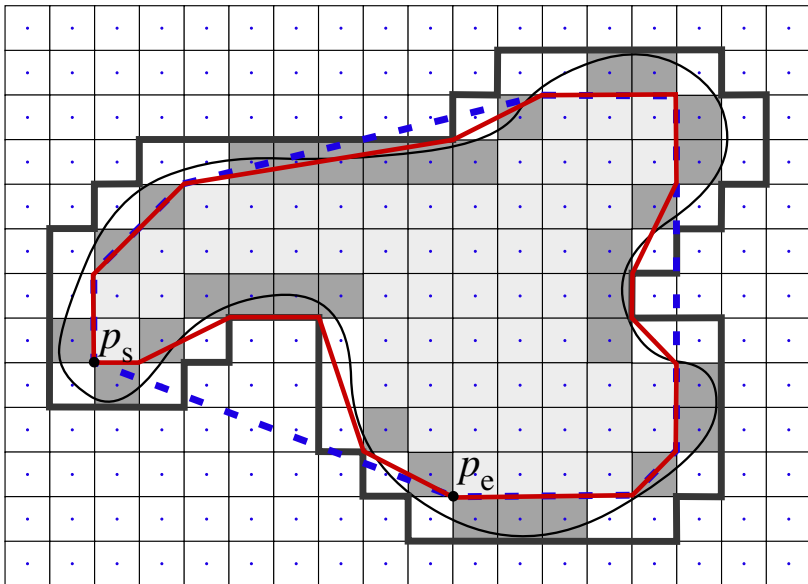


Fig. 1. Example for basic definitions

squares (i.e. grid cells), where edges have length 1 and centres of those squares have integer coordinates.

The *inner polygon* A is the union of all grid squares completely contained in the topological interior of a given object $S \subseteq \mathbb{R}^2$ (squares are light grey in Figure 1). The *outer polygon* B is the union of all grid squares having a nonempty intersection with the set S and the inner polygon A (the frontier of B is a thick black line in Figure 1). The unknown frontier of the set S is assumed to be a Jordan curve γ located between the frontiers of polygons A and B (drawn as a thin black line in Figure 1). We assume that those polygons are simple polygons, also known as Jordan polygons. We consider only simple digital contours which are called *grid continua* in [24].

The Gauss digitization of sets $S \subseteq \mathbb{R}^2$ defines all grid squares with their centroids in S as the resulting digital object (the union of all light grey squares and dark grey squares in Figure 1). A set of connected grid squares is also called a polyomino. To be precise, a digital object O is a polyomino iff it is 4-connected, and the complement is also 4-connected (see for example: [7]). The frontier of such a digitized object is a Jordan curve consisting of grid edges that separate the interior of the object from the exterior; these isothetic edges can be encoded by a Freeman chain code. The ordered list of grid squares $C = \langle p_1, p_2, \dots, p_n \rangle$ that have at least one 4-neighbor in \overline{O} is the *digital boundary* of O .

The *relative convex hull* of two simple polygons belongs to the class of *nested polygons*. Let A be a simple polygon with n vertices, $A = \langle p_1, p_2, \dots, p_n \rangle$, and let B be a simple polygon with m vertices, $B = \langle q_1, q_2, \dots, q_m \rangle$, with $A \subseteq B \subset \mathbb{R}^2$.

Definition 3. *A polygon A is B -convex (or relative-convex to B) iff any straight line segment in B that has both end points in A , is also contained in A . The convex hull of A relatively to B [in short, the B -convex hull of A ; formally $\text{CH}_B(A)$] is the intersection of all B -convex polygons containing A .²*

The *minimum length polygon* (MLP) of a 2D *digital object* (i.e. a connected region in a digital picture) coincides with the relative convex hull of the inner grid polygon A relatively to the outer grid polygon B as generated by a Jordan digitization [9]. The MLP for the object in Figure 1 is drawn as a red line, the convex hull of A is a blue dotted line.

We recall a few more basic definitions, to be used later in this paper.

Definition 4. *A cavity of a polygon A is the topological closure of any connected component of $\text{CH}(A) \setminus A$.*

Let m be the number of cavities in B and $\text{Int}(\text{CAV}(A))$ the interior of a cavity.

Definition 5. *A cavity $\text{CAV}(A)$ of a given inner polygon A and a given outer polygon B is called terminal if*

$$\text{Int}(\text{CAV}(A)) \cap \bigcap \text{Int}(\text{CAV}_i(B)) = \emptyset$$

for all $1 \leq i \leq m$.

² This definition can be generalized to higher dimensions than just $n = 2$.

Definition 6. A cover is a real straight-line segment in the frontier of $CH(A)$ that is not contained in the frontier of A .

A cover splits the complement of the digital region (i.e., the background) into two parts.

One example for a cavity is given in Figure 1 by its cover marked with the end points p_s and p_e .

We also recall definitions of *digital straight lines* and *digital straight segments*; see, for example, [7,18].

Definition 7. A *digital straight line* (DSL) with slope a/b , $b \neq 0$, and lower bound μ , where a, b, μ, ω are integers with $\gcd(a, b) = 1$, is the set of digital points $p = (x, y)$ satisfying $\mu \leq ax - by < \mu + \omega$.

The DSL is a 4-connected ordered set of points if $\omega = |a| + |b|$ and it is an 8-connected ordered set of points if $\omega = \max(|a|, |b|)$. A digital straight segment is an ordered subset of a digital boundary between p and q , $S(p, q)$, if there is a digital straight line that includes this subset. A digital straight segment $S(p, q)$ is called *maximal* iff $S(p-1, q)$ is not a digital straight segment and $S(p, q+1)$ is not a digital straight segment. The computation of all maximal digital straight segments of a digital boundary can be done in linear time [12].

Digital straight segments and MLP are basic tools to describe convexity of digital objects.

3 Digital Convexity

Digital convexity together with digital straightness are topics of research since digital geometry started as a new discipline as a result of the development of computer and image technologies. For example, see [15,20,21]. Numerous publications have contributed to this area; we only recall a few results.

3.1 Convexity of Digital Regions

Kim and Rosenfeld provided several equivalent definitions [10,11]. They have shown, for example, that a digital region R is convex iff its convex hull does not contain any discrete point of the complement \overline{R} of R . They also proved the analogous to the definition of convexity in the continuous space: A digital region R is convex iff any two points of this region can be connected by a digital straight line containing only points of R .

Sklansky [21] studied already convexity of digital regions when computer technology had just started to process images. He called digitized images of sets $S \subseteq \mathbb{R}^2$, composed of equally sized squares, “cellular blobs” and defined convexity for those as follows:

Definition 8. A *cellular blob* is convex iff there exists at least one convex set $S \subseteq \mathbb{R}^2$ of which the given cellular blob is an image.

The question arises how can we find such a set S based on this definition.

In a more recent journal article [5], a precise equivalent and more general definition for convexity is given that covers both convexity in the n -dimensional Euclidean space and in the digital space as follows:

Definition 9. *Given a subset $W \subseteq \mathbb{R}^n$. A subset R of W is W -convex if there exists a convex subset C of \mathbb{R}^n such that $R = C \cap W$.*

In 2D digital image analysis, we are interested to describe convexity for subsets R of W where W represents a regular grid, $W = \mathbb{Z}^2$. Such a convex subset C of \mathbb{R}^n is not uniquely defined in general. For showing convexity, we only need to compute one subset C of \mathbb{R}^n such that $R = C \cap W$. The minimum length polygon (MLP) is known to be such a subset C and it is uniquely defined by an inner polygon A which is the result of a Jordan digitization of a given measurable set in the Euclidean plane.

Proposition 1. *A digital region defined by its inner polygon A is digitally convex iff its MLP is convex.*

The MLP of A is different to $CH(A)$ if there exists at least one cavity in A and one cavity in B such that the intersection $Int(CAV_i(A)) \cap Int(CAV_j(B))$ is not empty.

All vertices of the convex hull of the inner polygon A are vertices of the MLP, start and end vertices of cavities of $CH(A)$ are always vertices of the MLP.

We will apply those properties in our computation of an MLP.

It is common to use digital straight segments to compute convexity of digital regions. For each given digital region, we can compute a set of digital straight segments, and this set is unique. A digital region R is convex iff every pair of consecutive maximal digital straight segments (MS) takes a convex turn [3,18,19,16].

The problem of deciding whether a polyomino coded by its contour word, also called Freeman chain code, is convex has been studied in a few publications based on the combinatorics on words [1,7]. The authors of [1,16] propose a linear time algorithm to solve this problem.

The journal paper [5] gives a very detailed overview about straightness of digital curves and the relationship to digital convexity. The author characterizes straightness of digital curves using difference operators, and he proves that definitions for digital straightness based on the chord property, word combinatorics, double Diophantine inequalities, and hyperplanes are equivalent.

3.2 Convexity of Subsets of Digital Boundaries

We consider a simple digital object as a result of a Gauss digitization (see [7]) and a part of its digital boundary $B(p_i, p_j)$ between boundary points p_i and p_j . One option to define convexity for those subsets is given in [3,19]:

Definition 10. *A part of a digital boundary $B(p_i, p_j)$ is convex iff every pair of its consecutive maximal digital straight segments MS_k and MS_{k+1} takes a convex turn.*

Two consecutive segments $MS_k(a, b, \mu, \omega)$ and $MS_{k+1}(a', b', \mu', \omega')$ describe a convex turn iff $ab' - a'b > 0$, but make a concave turn iff $ab' - a'b < 0$.

In [18,19], the boundary points are traversed clockwise and convexity for parts of a digital boundary is defined as follows:

Definition 11. Let $B(p_i, p_j)$ be a part of the digital boundary B between points p_i and p_j of a region R . The shortest polygonal line linking p_i and p_j located on the left of $B(p_i, p_j)$ is called the hull $H(p_i, p_j)$. $B(p_i, p_j)$ is convex iff there is no digital point $q \in \mathbb{Z}^2$ located between $B(p_i, p_j)$ and its hull $H(p_i, p_j)$.

Both papers [18,19] introduce *maximal convex* or *maximal concave parts* analogous to the notion of maximal digital straight segments. A part of a digital boundary $B(p_i, p_j)$ is maximal convex if $B(p_i, p_j)$ is convex and $B(p_{i-1}, p_j)$ and $B(p_i, p_{j+1})$ are not convex. A digital curve can be decomposed into maximal convex and maximal concave parts in linear time. The authors define a faithfully polygon (FP) and refer to an online algorithm that uses well known routines of adding a point to the front of a DSS and removing a point on the back of a DSS. They translate each vertex of FP to compute MLP.

We recall that a local operator calculates a result based only on inputs within a circular neighborhood of a given fixed radius [7]. Some papers ([4,19]) discuss the question whether we can find a local operator to decide about convexity for a part of a digital boundary. The authors of [19] give the following answer: "The smallest part required for checking convexity is given by a MS, plus at least one of the two points located just before and after this segment". It follows that an operator using MS for deciding about convexity is not local. Applying the definition for local operators, we consider a neighborhood with a fixed radius r and a part of a digital boundary which includes three MS then the first and the second turn of MSs can be more than r apart.

Contrary to the Euclidean space, the decisions about digital convexity for regions or parts of digital regions cannot be done locally.

We describe a recursive algorithm in the following section and we like to highlight the fact that the decision whether a single vertex p_i of a frontier is convex, concave or collinear is a local operation. A vertex is convex if the frontier takes a positive turn, that means the determinant $t(p_{i-1}, p_i, p_{i+1}) > 0$, and analog a vertex is concave if the value of the determinant is negative where p_{i-1} and p_{i+1} are 4-neighbors of p_i and they belong to the frontier of A . A vertex is collinear if the value of the determinant is zero. Only inputs belonging to the fixed neighborhood of a vertex p :

$$N(p) = \{q \in \mathbb{Z}^2 : |p_x - q_x| + |p_y - q_y| \leq 1\}$$

are required.

4 Convexity and Cavity Trees

4.1 Recursive Computation of MLP and Cavity Trees

We compute the minimum length polygon (MLP) with a recursive algorithm and generate cavity trees. This approach supports the decomposition of naturally

shaped objects into significant parts and it decomposes a digital boundary into convex and concave parts at the same time.

Cavity trees have been introduced in [22] as "concavity trees" with the intention to describe the shape of simple digital objects.

In [9], the definition for cavity trees is slightly different, they are created by a recursive algorithm for the computation of MLP for an inner polygon A and an outer polygon B . The ordered set of convex or concave vertices (collinear vertices are not candidates for MLP) of the frontier of A (generated by a Jordan digitization, [7]) and their mappings into B are the inputs for a recursive algorithm. Only convex vertices of the inner polygon and concave vertices of the outer polygon are candidates for the relative convex hull (here MLP) [23]. The algorithm identifies those candidates in a first run and it computes the convex hull of the inner polygon A . We apply the definition of convexity for any finite set of points as known in Euclidean geometry using the Melkman algorithm [14], it works efficiently for any simple polyline by using a deque (i.e., a double-ended queue). The frontier of polygons A in our study are simple polylines which form loops.

Figure 2 shows the inner and the outer polygon of a digital contour. The input for this example is an ordered list of coordinates of those vertices:

$$A = [p_1(x, y)^+, \dots, p_3(x + 1, y - 1)^-, \dots, p_n(x - 1, y - 1)^-]$$

The positive vertex q_1 of B in Fig. 2 for example has no influence for the computation of MLP. Only positive marked vertices A_{pos} of A are required for the computation of the convex hull of A . The result of the first recursion step $CH(A_{pos})$ is stored in a deque

$$D = [p_1(x, y)^+, \dots, p_s(x, y)^+, p_e(x, y)^+, \dots, p_1(x, y)^+]$$

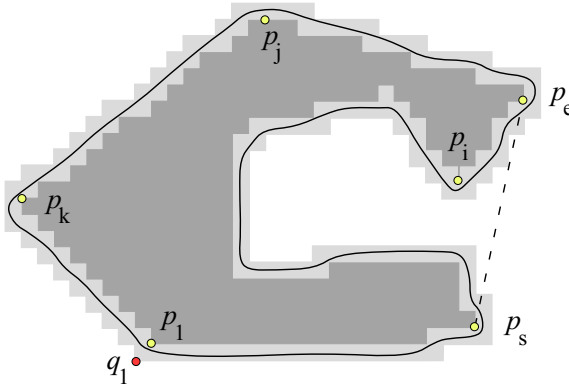


Fig. 2. Inner and outer polygon of a digital contour

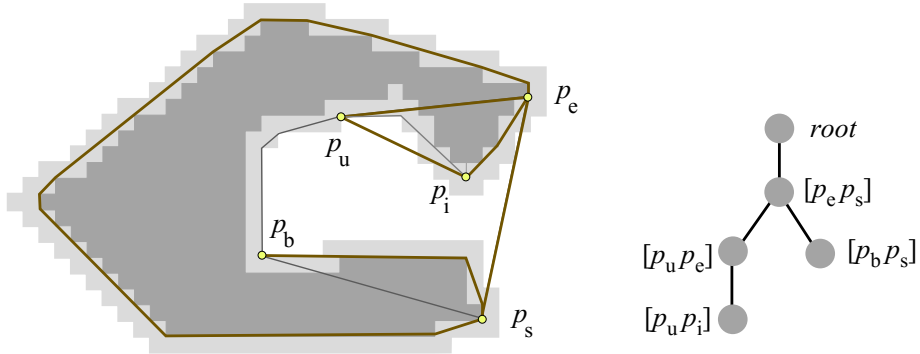


Fig. 3. Left: Created cavities. Right: Cavity tree. Vertices in this tree are named by both end vertices of the defining cover.

A child node has been found if a non-terminal cavity with a cover starting at a convex vertex p_s of A and ending with a convex vertex p_e of A has been detected. This is equivalent to the fact that the intersection between this cavity and a cavity of the outer polygon are not empty:

$$Int(CAV(A)) \cap Int(CAV_i(B)) \neq \emptyset$$

for all $1 \leq i \leq m$. Applying one of the definitions for digital convexity of regions, we can follow that the given digital region R contains at least one discrete point of the complement \overline{R} of R and it is not convex. The algorithm could stop here if we only need to decide about convexity of the original digital object.

We construct a new input for the next recursion step by changing the marks from "-" to "+" such that all former concave vertices of the outer polygon become convex vertices inside the cavity:

$$A_1 = [p_s(x, y)^+, \dots, p_{s+1}(x + 1, y + 1)^+, \dots, p_i(x, y)^-, p_e(x, y)^+]$$

The second recursion step with input A_1 computes the convex hull of the cavity between p_s and p_e which is a simple polygon where concave vertices of the original outer polygon, marked with "-" in the original list, are convex vertices inside the cavity and only those are candidates for MLP.

All vertices of $CH(CAV(A))$ are vertices of MLP [8]. But if there is a non-terminal cavity in $CAV(A)$, then the set of vertices for MLP is still not complete and the next recursion step follows analog.

The final results of the algorithm are a cavity tree as illustrated in Figure 3 and a list of marked vertices that represents MLP.

The computation of cavity trees leads to an easy characterization of digital convexity.

Definition 12. *A digital region of a 2D object is digitally convex iff the root of its cavity tree has no leaves.*

That means, if a digital region is convex then all cavities are terminal cavities and the MLP coincides with the convex hull of A . In this case, any part of the boundary is convex. And analogous, a digital region is digitally concave iff the root of its cavity tree has at least one child.

4.2 Convexity and Decompositions

The result of the first recursion step is the frontier of a new simple polygon which consists of convex parts of the inner polygon and covers of non-terminal cavities. Those convex parts are not maximal in all cases.

In each recursive step, we find new vertices of MLP located in cavities that we identified in the previous step (see Figure 3):

$$\begin{aligned} R_1 &= [p_1^+, \dots, p_s^+, p_e^+, \dots, p_1^+] \\ R_2 &= [p_1^+, \dots, p_s^+, p_b^-, \dots, p_u^-, p_e^+, \dots, p_1^+] \\ R_3 &= [p_1^+, \dots, p_s^+, p_{s+1}^+, p_b^-, \dots, p_u^-, p_i^+, p_e^+, \dots, p_1^+] \\ R_4 &= [p_1^+, \dots, p_s^+, p_{s+1}^+, p_b^-, \dots, p_u^-, p_{u+1}^-, p_i^+, p_e^+, \dots, p_1^+] \end{aligned}$$

Vertices of the final list R_4 are the vertices of the minimum length polygon. Ordered lists of positive marked vertices in a final list R , in our example

$$T_1^+ = [p_1^+, \dots, p_s^+, p_{s+1}^+]$$

and

$$T_3^+ = [p_i^+, p_e^+, \dots, p_1^+]$$

are convex parts of MLP. Ordered lists of negative marked vertices in R , in our example

$$T_2^- = [p_b^-, \dots, p_u^-, p_{u+1}^-]$$

are concave parts of MLP.

Because T_1^+ starts with p_1^+ and T_3^+ ends with p_1^+ , we can join both parts by replacing p_1^+ in T_1^+ with T_3^+ and we get a decomposition of the MLP into:

$$T_1^+ = [p_i^+, p_e^+, \dots, p_1^+, \dots, p_s^+, p_{s+1}^+]$$

and

$$T_2^- = [p_b^-, \dots, p_u^-, p_{u+1}^-].$$

The part of MLP starting at p_i^+ and ending at vertex p_{s+1}^+ is a convex part of MLP and the part starting at p_b^- and ending at p_{u+1}^- is concave.

More general, the output R is presenting all vertices of MLP for a given digital region. We consider sequences T_i^+ of vertices in R marked with "+" where we join the two sequences: the one that starts with p_1^+ and the one that ends with p_1^+ . We replace p_1^+ in the first list with the second list. Each list T_i^+ represents a convex part of MLP, each list T_j^- represents a concave part of MLP.

The decomposition of the final list of MLP-vertices leads to a decomposition into convex and concave parts of a digital boundary.

Proposition 2. *The first vertex in T_i^+ is the first vertex and the last vertex in T_i^+ is the last vertex of a maximal convex part of the digital boundary.*

The first vertex in T_i^- is the first vertex and the last vertex in T_i^- is the last vertex of a maximal concave part of the digital boundary.

Proof: We assume that a given digital region (represented by A and B) is convex. Then $MLP(R) = T_1^+ = CH(A)$.

Now we assume that a given digital region is concave. Then there exist at least one non-terminal cavity $CAV1$ with positive marked vertices p_s^+ and p_e^+ on both ends and at least one concave vertex q_i^- of B (or more) is inside this cavity. The part between p_e^+ and p_s^+ (without the cover) is digitally convex and the cavity must include at least one concave part.

Inside $CAV1$ we can have three types of new covers for new cavities: the first one starts at p_s and ends with a negative vertex, the second type starts and ends with a negative vertex, the third type starts with a negative vertex and ends with p_e . Assume the first type has no new cavities then the algorithm computes the convex hull of a set of positive vertices. The MLP would only take convex turns between p_s^+ and q_i^- and the convex part would grow to the last vertex before q_i^- (see Figure 3). If the second type has no new cavities then we discovered a digitally convex part of the boundary. If the third type has no new cavities then new positive marked vertices extend the convex part in front of p_e .

The next recursion step is analog. □

5 Discussion

The recursive computation of MLP applies convexity definitions for a finite set of points (here only convex vertices of an inner polygon) in Euclidean space using the Melkman algorithm for the computation of convex hulls in linear time. It uses well known properties of the relative convex hull and the specific characteristics of MLP as a result of a Jordan digitization. In a preliminary step, a local operator decides whether a single vertex in the frontier of a polygon takes a convex (is positive), or a concave turn (is negative) or it is in a collinear position to its neighbors. Convexity of a digital region or for parts of boundaries cannot be decided locally.

Properties about digital straightness, word combinatorics, Diophantine inequalities or difference operators have no direct influence into the design of this algorithm. A deeper analysis would be interesting to show correlations between those different concepts to characterize convexity. It could be a continuation of the work in [5] and it could be used to prove the efficiency of the recursion.

6 Conclusions

The paper recalls a few important notions, properties and algorithms with respect to convexity of digital regions and parts of digital boundaries. It explains a recursive algorithm for the computation of MLP that provides two easy options

to decompose a digital boundary into meaningful parts at the same time. The leafs of a cavity tree present a decomposition of a digital object into cavities of digital shapes. This could be a very useful tool for boundary smoothing where the algorithm could stop at a given smoothing parameter. A small adjustment could include the computation of the length of the perimeter per cavity which could deliver interesting features for image analysis. Even more, the resulting list of MLP vertices provides a decomposition into convex and concave parts of the frontier of a digitized object. The algorithm runs in linear time if the maximum height of the cavity tree is limited by a constant.

References

1. Brlek, S., Lachaud, J.-O., Provençal, X.: Combinatorial view of digital convexity. In: Coeurjolly, D., Sivignon, I., Tougne, L., Dupont, F. (eds.) DGCI 2008. LNCS, vol. 4992, pp. 57–68. Springer, Heidelberg (2008)
2. Debled-Rennesson, I., Remy, J.-L., Rouyer-Degli, J.: Detection of discrete convexity of polyominoes. *Discrete Applied Mathematics* 125, 115–133 (2003)
3. Doerksen-Reiter, H., Debled-Rennesson, I.: Convex and concave parts of digital curves. *Computational Imaging and Vision* 31, 145–160 (2006)
4. Eckhardt, U., Doerksen-Reiter, H.: Polygonal representations of digital sets. *Algorithmica* 38(1), 5–23 (2004)
5. Kiselman, C.O.: Characterizing digital straightness and digital convexity by means of difference operators. *Mathematika* 57, 355–380 (2011)
6. Klette, R., Zunic, J.: Multigrid convergence of calculated features in image analysis. *J. Mathematical Imaging Vision* 13, 173–191 (2000)
7. Klette, R., Rosenfeld, A.: *Digital Geometry – Geometric Methods for Digital Picture Analysis*. Morgan Kaufmann, San Francisco (2004)
8. Klette, G.: Recursive calculation of relative convex hulls. In: Debled-Rennesson, I., Domenjoud, E., Kerautret, B., Even, P. (eds.) DGCI 2011. LNCS, vol. 6607, pp. 260–271. Springer, Heidelberg (2011)
9. Klette, G.: Recursive Computation of Minimum-Length Polygons. *Computer Vision and Image Understanding* 117, 386–392 (2012)
10. Kim, C.E.: Digital convexity, straightness, and convex polygons. *PAMI* 4, 618–626 (1982)
11. Kim, C.E., Rosenfeld, A.: Digital straight lines and convexity of digital regions. *PAMI* 4, 149–153 (1982)
12. Lachaud, J.O., Vialard, A., de Vieilleville, F.: Fast, accurate and convergent tangent estimation on digital contours. *Image and Vision Computing* 25, 1572–1587 (2007)
13. Li, F., Klette, R.: *Euclidean Shortest Paths*. Springer, London (2011)
14. Melkman, A.: On-line construction of the convex hull of a simple polygon. *Information Processing Letters* 25, 11–12 (1987)
15. Minsky, M., Papert, S.: *Perceptrons*. MIT Press, Reading (1969)
16. Provençal, X., Lachaud, J.-O.: Two linear-time algorithms for computing the minimum length polygon of a digital contour. In: Brlek, S., Reutenauer, C., Provençal, X. (eds.) DGCI 2009. LNCS, vol. 5810, pp. 104–117. Springer, Heidelberg (2009)
17. Papadopoulos, A.: *Metric Spaces, Convexity and Nonpositive Curvature*. European Mathematical Society (2005)

18. Roussillon, T., Tougne, L., Sivignon, I.: What does digital straightness tell about digital convexity? In: Wiederhold, P., Barneva, R.P. (eds.) IWCIA 2009. LNCS, vol. 5852, pp. 43–55. Springer, Heidelberg (2009)
19. Roussillon, T., Sivignon, I.: Reversible polygon that faithfully represents the convex and concave parts of a digital curve. *Pattern Recognition* 44, 2693–2700 (2011)
20. Rosenfeld, A.: *Picture Processing by Computer*. Academic Press, New York (1969)
21. Sklansky, J.: Recognition of convex blobs. *Pattern Recognition* 2, 3–10 (1970)
22. Sklansky, J.: Measuring cavity on a rectangular mosaic. *IEEE Trans. Computing* 21, 1355–1364 (1972)
23. Sloboda, F., Stoer, J.: On piecewise linear approximation of planar Jordan curves. *J. Computational and Applied Mathematics* 55, 369–383 (1994)
24. Sloboda, F., Zatko, B., Stoer, J.: On approximation of planar one dimensional continua. In: Klette, R., Rosenfeld, A., Sloboda, F. (eds.) *Advances in Digital and Computational Geometry*, pp. 113–160 (1998)