

Two linear-time algorithms for computing the minimum length polygon of a digital contour

Xavier Provençal^{*1,2} and Jacques-Olivier Lachaud^{**1}

¹ Laboratoire de Mathématiques, UMR 5127 CNRS, Université de Savoie,
73376 Le Bourget du Lac, France.

² LIRMM, UMR 5506 CNRS, Université Montpellier II, 34392 Montpellier, France.
provençal@lirmm.fr, jacques-olivier.lachaud@univ-savoie.fr

Abstract. The Minimum Length Polygon (MLP) is an interesting first order approximation of a digital contour. For instance, the convexity of the MLP is characteristic of the digital convexity of the shape, its perimeter is a good estimate of the perimeter of the digitized shape. We present here two novel equivalent definitions of MLP, one arithmetic, one combinatorial, and both definitions lead to two different linear time algorithms to compute them.

1 Introduction

The minimum length polygon (MLP) or minimum perimeter polygon has been proposed long ago for approaching the geometry of a digital contour [1, 2]. One of its definitions is to be the polygon of minimum perimeter which stays in the band of 1 pixel-wide centered on the digital contour. It has many interesting properties such as: (i) it is reversible [1]; (ii) it is characteristic of the convexity of the digitized shape and it minimizes the number of inflexion points to represent the contour [2, 3]; (iii) it is a good digital length estimator [4, 5] and is proven to be multigrid convergent in $O(h)$ for digitization of convex shapes, where h is the grid step (reported in [6–8]); (iv) it is also a good tangent estimator; (v) it is the relative convex hull of the digital contour with respect to the outer pixels [2, 9] and is therefore exactly the convex hull when the contour is digitally convex.

Several algorithms for computing the MLP have been published. We have already presented the variational definition of the MLP (length minimizer). It can thus be solved by a nonlinear programming method. The initial computation method of [1] was indeed an iterative Newton-Raphson algorithm. Computational complexity is clearly not linear and the solution is not exact. We have also mentioned its set theoretic definition (intersection of relative convex sets). However, except for digital convex shapes, this definition does not lead to a specific algorithms. The MLP may also be seen as a solution to a shortest path query in some well chosen polygon. An adaptation of [10] to digital contour could be implemented in time linear with the size of the contour. It should however be

* Author supported by a scholarship from FQRNT (Québec).

** Partially funded by ANR project FOGRIMMI (ANR-06-MDCA-008-06).

noted that data-structures and algorithms involved are complex and difficult to implement. Klette *et al.* [11] (see also [4, 6]) have also proposed an arithmetic algorithm to compute it, but as it is presented, it does not seem to compute the MLP in all cases.³

The MLP is in some sense characteristic of a digital contour. One may expect to find strong related arithmetic and combinatorial properties. This is precisely the purpose of this paper. Furthermore, we show that each of these definitions induces an optimal time integer-only algorithm for computing it. The combinatorial algorithm is particularly simple and elegant, while the arithmetic definition is essential for proving it defines the MLP. These two new definitions give a better understanding of what is the MLP in the digital world. Although other linear-time algorithms exist, the two proposed algorithms are simpler than existing ones. They are thus easier to implement and their constants are better.

The paper is organized as follows. First Section 2 recalls standard definitions. Section 3 presents how to split uniquely a digital contour into convex, concave and inflexion zones, the arithmetic definition of MLP follows then naturally. Section 4 is devoted to the combinatorial version of MLP. Section 5 illustrates our results and concludes. Due to space limitations, we are not able to provide the proofs that these definitions induce the MLP, we only give hints and discuss the algorithms to extract them.

2 Preliminaries

This section presents the generally standard definitions that we will use throughout the paper, in order to avoid any ambiguity. A *polyomino* is a set of digital squares in the plane such that its topological boundary is a Jordan curve. It is thus bounded. It is convenient to represent a polyomino as a subset of the digital plane \mathbb{Z}^2 , which code the integer coordinates of the centroids of its squares, instead of representing it as a subset of the Euclidean plane \mathbb{R}^2 . When seeing a polyomino as a subset of \mathbb{R}^2 , we will say *the body of the polyomino*. For instance, the Gauss digitization of a convex subset of the plane is a polyomino iff it is 4-connected. A subset of \mathbb{Z}^2 , or *digital shape*, is a polyomino iff it is 4-connected and its complement is 4-connected.

In the following, a *digital contour* is the boundary of some polyomino, represented as a sequence of horizontal and vertical steps in the half-integer plane $(\mathbb{Z} + \frac{1}{2}) \times (\mathbb{Z} + \frac{1}{2})$. One can use for instance a Freeman chain to code it. Again, the *body* of a digital contour is its embedding in \mathbb{R}^2 as a polygonal curve. Now, since the body of a digital contour is a Jordan curve, it has one well-defined inner component in \mathbb{R}^2 , whose closure is exactly the polyomino whose boundary is the digital contour. There is thus a one-to-one map from digital contours to polyominoes, denoted by I .

Given a digital contour C , its *inner polygon* $L_1(C)$ is the erosion of the body of $I(C)$ by the open unit square centered on $(0, 0)$. Its *outer polygon* $L_2(C)$ is

³ Its edges seem restricted to digital straight segments such that the continued fraction of their slope has a complexity no greater than two.

the dilation of the body of $I(C)$ by the closed unit square centered on $(0,0)$. In the following, the inner and outer polygons play the role of inner and outer constraints for constructing the MLP. We only deal in this paper with simple digital contours (or grid continua in the terminology of [8]), therefore both $L_1(C)$ and $L_2(C)$ are assumed to be simple polygons.

The relative convex hull leads to a simple and elegant set-theoretic definition of the characteristic polygon, which is valid in arbitrary dimension. In the following, the notation \overline{xy} stands for the straight line segment joining x and y , i.e. their convex hull.

Definition 1. [9]. Let $U \subseteq \mathbb{R}^n$ be an arbitrary set. A set $C \subseteq U$ is said to be U -convex iff for every $x, y \in C$ with $\overline{xy} \subseteq U$ it holds that $\overline{xy} \subseteq C$.

Let $V \subseteq U \subseteq \mathbb{R}^n$ be given. The intersection of all U -convex sets containing V will be termed convex hull of V relative to U , and denoted by $\text{Conv}_U(V)$.

The *set-theoretic MLP* of a digital contour C is the convex hull of $L_1(C)$ relative to $L_2(C)$. It is easily proven that the MLP of a digitally convex contour is exactly the convex hull of its digital points.

The standard definition of the MLP of C [1, 8, 4] is the shortest Jordan curve whose digitization is (very close to) the polyomino of C . More precisely, letting \mathcal{A} be the family of simply connected compact sets of \mathbb{R}^2 , we define:

Definition 2. The minimum perimeter polygon, *MPP*, of two polygons V, U with $V \subset U^\circ \subset \mathbb{R}^2$ is a subset P of \mathbb{R}^2 such that

$$\text{Per}(P) = \min_{A \in \mathcal{A}, V \subseteq A, \partial A \subset U \setminus V^\circ} \text{Per}(A), \quad (1)$$

where $\text{Per}(A)$ stands for the perimeter of A , more precisely the 1-dimensional Hausdorff measure of the boundary of A .

In [7] (Theorem 3), and in [9] it is shown that Equation (1) has a unique solution, which is a polygonal Jordan curve whose convex vertices (resp. concave) belong to the vertices of the inner polygon (resp. the vertices of the outer polygon), and which is also the convex hull of V relative to U . The *variational MLP* of C is thus the MPP of $L_1(C), L_2(C)$.

3 Arithmetic MLP

3.1 Tangential cover and arithmetic properties

We recall that the *tangential cover* of a digital contour is the sequence of its Maximal Digital Straight Segments (MDSS). In the following, the tangential cover is denoted by $(M_l)_{l=1..N}$, where M_l is the l -th MDSS of the contour. Let us denote by θ_l the slope direction (angle wrt x-axis) of M_l . All indices are taken modulo the number of MDSS N . Since the directions of two consecutive MDSS can differ of no greater than π , their variation of direction can always be casted in $]-\pi, \pi[$ without ambiguity. The angle variation $(\theta_l - \theta_{l+1}) \bmod [-\pi, \pi[$ is denoted by

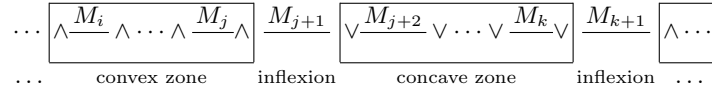
$\Delta(\theta_l, \theta_{l+1})$. For clarity, we will also write $\theta_l > \theta_{l+1}$ when $\Delta(\theta_l, \theta_{l+1}) > 0$. We always consider the digital contour to turn clockwise around the polyomino. A couple of consecutive MDSS (M_l, M_{l+1}) is thus said to be a \wedge -turn (resp. \vee -turn) when $\Delta(\theta_l, \theta_{l+1})$ is negative (resp. positive). The symbol \wedge stands for “convex” while the symbol \vee stands for “concave”.

We have the following theorem from [12], which relates convexity to MDSS directions. It also induces a linear time algorithm to check convexity.

Theorem 1. [12] *A digital contour is digitally convex iff every couple of consecutive MDSS of its tangential cover is made of \wedge -turns.*

For a given DSS M , its first and last upper leaning points are respectively denoted by $U_f(M)$ and $U_l(M)$, while its first and last lower leaning points are respectively denoted by $L_f(M)$ and $L_l(M)$. In the same paper, it is proven that the point $U_l(M_l)$ is no further away than $U_f(M_{l+1})$. The same property holds for lower leaning points.

We may now consider the succession of turns along a digital contour to cut it into parts. The above-mentioned ordering on leaning points between successive MDSS guarantees the consistency of the following definition.



Definition 3. *A digital contour C is uniquely split by its tangential cover into a sequence of closed connected sets with a single point overlap as follows:*

1. A convex zone or (\wedge, \wedge) -zone is defined by an inextensible sequence of consecutive \wedge -turns from (M_{l_1}, M_{l_1+1}) to (M_{l_2-1}, M_{l_2}) . If it is a proper zone of C , it starts at $U_l(M_{l_1})$ and ends at $U_f(M_{l_2})$, otherwise it starts and ends on the first point.
2. A concave zone or (\vee, \vee) -zone is defined by an inextensible sequence of consecutive \vee -turns from $(M_{l'_1}, M_{l'_1+1})$ to $(M_{l'_2-1}, M_{l'_2})$. It starts at $L_l(M_{l'_1})$ and ends at $L_f(M_{l'_2})$.
3. A convex inflexion zone or (\wedge, \vee) -zone is defined by a \wedge -turn followed by a \vee -turn around MDSS M_i . It starts at $U_f(M_i)$ and ends at $L_l(M_i)$.
4. A concave inflexion zone or (\vee, \wedge) -zone is defined by a \vee -turn followed by a \wedge -turn around MDSS $M_{i'}$. It starts at $L_f(M_{i'})$ and ends at $U_l(M_{i'})$.

Note that a convex or concave zone may be reduced to a single turn between two successive inflexions. In this case, the zone may or may not be a single contour point (point A in Fig. 1).

Since a MDSS is contained in a digital straight line, it is formed of exactly two kind of steps, with Freeman codes c and $(c+1) \bmod 4$. These codes defines the *quadrant* of the MDSS. The *quadrant vector* is then the diagonal vector that is the sum of the two unit steps coded by the Freeman codes of the quadrant, rotated by $+\frac{\pi}{2}$.

We eventually associate pixels to contour points (C_i) as follows:

- the *inside pixel* $\text{in}(C_i)$ of C_i is the pixel $C_i - \frac{\vec{v}}{2}$, where \vec{v} is the quadrant vector of any MDSS containing it (or the last MDSS strictly containing it at a quadrant change).
- the *outside pixel* $\text{out}(C_i)$ of C_i is the pixel $C_i + \frac{\vec{v}}{2}$, where \vec{v} is the quadrant vector of any MDSS containing it (or the last MDSS strictly containing it at a quadrant change).

Fig. 1 illustrates these definitions. It is clear that inside pixels belong to $\partial L_1(C)$ and outside pixels to $\partial L_2(C)$.

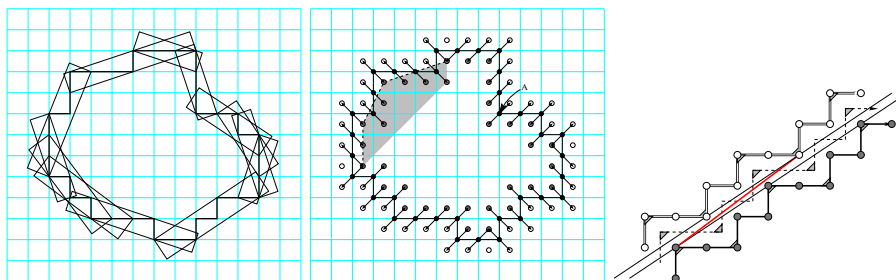


Fig. 1. Left: digital contour, tangential cover. Center: inside and outside pixels, edges of $\text{AML}_P(C)$ in a convex part of C . Right : Geometry of a DSS of slope $2/3$ along a digital contour. The digital contour is drawn in-between the corresponding inside and outside pixels. They clearly draw the same contour up to a translation. Upper and lower leaning points are denoted by grey right triangles. The thick red line connects the first upper leaning point on the inside contour to the last lower leaning point on the outside contour. The two straight lines define a straight band which separates inside from outside pixels. The thick red line is in this band, thus in $L_2(C)$.

3.2 Definition of the arithmetic MLP of C

The boundary word of any hv -convex polyomino admits a unique factorization as $Q_{0<1}0Q_{3<0}3Q_{2<3}2Q_{1<2}1$, where $Q_{a<b}$ is written over the two letters $\{a,b\}$ and ends by b . The words $Q_{a<b}$ are called its *quadrant words*. As a corollary, convex zones of a polyomino admits a unique factorization into a prefix of a conjugate of $Q_{0<1}0Q_{3<0}3Q_{2<3}2Q_{1<2}1$.

Definition 4. Assume $C_{i,j}$ is a connected part of a contour, with only two kinds of steps. The left envelope of $C_{i,j}$ (resp. right envelope of $C_{i,j}$) is the sequence of edges of the convex hull of the inside (resp. outside) pixels of $C_{i,j}$, such that the first vertex is C_i , the last vertex is C_j and the edges turn clockwise (resp. counterclockwise) around the hull.

We may now define a linear analog to a digital contour.

Definition 5. The arithmetic MLP of a digital contour C is the polygon $\text{AMLP}(C)$ defined by zones $C_{i,j}$ in C , according to its type:

zone type of $C_{i,j}$	associated part of $\text{AMLP}(C)$
(\wedge, \wedge) convex	union of the left envelope of each quadrant word of $C_{i,j}$
(\vee, \vee) concave	union of the right envelope of each quadrant word of $C_{i,j}$
(\wedge, \vee) cvx-inflexion	segment joining the inside pixel of C_i to the outside pixel of C_j
(\vee, \wedge) ccv-inflexion	segment joining the outside pixel of C_i to the inside pixel of C_j

Lemma 1. $\text{AMLP}(C)$ is a closed polygonal line with vertices in \mathbb{Z}^2 .

Proof. By construction, all vertices of $\text{AMLP}(C)$ are in \mathbb{Z}^2 . Clearly, $\text{AMLP}(C)$ is a polygonal line inside each zone of C . All that remains to check is that inflexion zones links the convex and concave zones correctly, and this is done using standard properties of MDSS. \square

Let U be the closed unit square centered at $(0, 0)$ and let \oplus denotes the Minkowski sum of two sets. We recall that the set $C \oplus U$ is also the one pixel wide band $L_2(C) \setminus L_1(C)^\circ$.

Lemma 2. In a (\wedge, \wedge) -zone $C_{i,j}$ of C , the corresponding edges of $\text{AMLP}(C)$ form a simple polygonal line included in $L_2(C)^\circ \setminus L_1(C)^\circ$, and in $C_{i,j} \oplus U$.

Proof. Consider a single quadrant word $Q_{a < b}$ that is part of a (\wedge, \wedge) -zone. For this part of C , the edges of $\text{AMLP}(C)$ form the digital convex hull of the inside pixels of C so that there cannot be any intersection with $L_1(C)^\circ$. On the other hand, again by the digital convex hull property, all the integer points in the $\text{AMLP}(C)$, for this zone, must belong to the set of inner pixels of C . In particular, there are no points of $L_2(C)$. \square

The following lemmata are proven similarly to the previous one.

Lemma 3. In a (\vee, \vee) -zone $C_{i',j'}$ of C , the corresponding edges of $\text{AMLP}(C)$ form a simple polygonal line included in $L_2(C) \setminus L_1(C)$, and in $C_{i',j'} \oplus U$.

Lemma 4. In a (\wedge, \vee) or (\vee, \wedge) zone $C_{i'',j''}$ of C , the corresponding edges of $\text{AMLP}(C)$ form a single straight segment included in $L_2(C) \setminus L_1(C)^\circ$, and in $C_{i'',j''} \oplus U$.

Proof. We refer the reader to Fig. 1. The $\text{AMLP}(C)$ is in this zone the thick red segment squeezed in the band that separates inside from outside pixels. The lemma follows. \square

Theorem 2. $\text{AMLP}(C)$ is a simple polygon with boundary in $L_2(C) \setminus L_1(C)^\circ$.

Proof. Lemmata 2 and 3 guarantee that the restriction of the edges of $\text{AMLP}(C)$ to these parts is always a single polygonal line in $L_2(C) \setminus L_1(C)^\circ$. Furthermore these parts have always a pairwise empty intersection. Indeed, taking any two of these parts, they are defined by two pieces of C , say C_{l_1, l_2} and $C_{l'_1, l'_2}$. These pieces are separated by at least one inflexion zone, whose length is at least 1. Therefore $C_{l_1, l_2} \cap C_{l'_1, l'_2} = \emptyset$. Moreover $(C_{l_1, l_2} \oplus U) \cap (C_{l'_1, l'_2} \oplus U)$ is either (i)

empty, (ii) an integer point, (iii) a unit segment. If (i), the inflexion zone is large enough and the two parts of $\text{AMLP}(C)$ may obviously not touch each other. If (ii) or (iii), the inflexion zone is two surfels long or one surfel long respectively. We notice furthermore that $\text{AMLP}(C)$ ends on one side at an inside pixel and on the other side at an outside pixel. It is easy to check (there is one 1 surfel configuration and three 2 surfels configuration) that the two parts of $\text{AMLP}(C)$ have an empty intersection.

Lemma 4 indicates that the edges of $\text{AMLP}(C)$ in inflexion zones are also in $L_2(C) \setminus L_1(C)^\circ$. It is also clear that at a convex junction C_k , $\text{AMLP}(C) \cap (C_k \oplus U)$ is reduced to the point in (C_k) , therefore with no other self-intersections. A symmetric result is obtained at a concave junction. Gathering everything, the so-formed polygonal line is simple. Jordan theorem concludes it has a finite inner component. $\text{AMLP}(C)$ is thus a polygon with a boundary in $L_2(C) \setminus L_1(C)^\circ$. \square

We finish by providing an algorithm to compute $\text{AMLP}(C)$.

Algorithm 1: Computation of $\text{AMLP}(C)$

```

1 Compute  $(M_i)_{i=1..N}$  the tangential cover of  $C$  ;
2 Decompose  $(M_i)_{i=1..N}$  in  $(\alpha, \beta)$ -zones; // WHERE  $\alpha, \beta \in \{\wedge, \vee\}$  (DEFINITION 3)
3  $S \leftarrow ()$ ;
4 for each  $(\alpha, \beta)$ -zone  $z$  of  $C$  do
5    $\lfloor$  add to the list  $S$  the part of  $\text{AMLP}(C)$  associate to  $z$ ; // (DEFINITION 5)
6 return  $S$ 

```

Theorem 3. *Algorithm 1 computes $\text{AMLP}(C)$ in time linear with respect to the length of C .*

Proof. Let n be the number of points of C . Computation of the tangential cover on line 1 is performed in linear time according to [13]. The computation on line 2 is clearly proportional to N which is also $O(n)$. Finally, on line 5, there are two cases to consider. Given a convex or concave zone, the $\text{AMLP}(C)$ is some convex hull of a simple polygonal line which is computed in time proportional to the length of the zone using [14] or [15]. Given an inflexion zone, the computation is reduced to a segment, thus a $O(1)$ operation. Total computation time is $O(n)$. \square

3.3 $\text{AMLP}(C)$ is the MLP of C

In order to show the equivalence between $\text{AMLP}(C)$ and the convex hull of $L_1(C)$ relatively to $L_2(C)$, we need the following technical lemma.

Lemma 5. *Convex vertices of $\text{AMLP}(C)$ are inside pixels of C (i.e. $\in \partial L_1(C)$), concave vertices of $\text{AMLP}(C)$ are outside pixels of C (i.e. $\in \partial L_2(C)$).*

Proof. We already know that in a convex zone of C , vertices of $\text{AML}(C)$ are by Definition 5 inside pixels with strictly decreasing edge directions. In a concave zone of C , vertices of $\text{AML}(C)$ are outside pixels with strictly increasing edge directions. Around an inflexion zone of direction θ_i , one may show that $\theta_{i-1} > \theta_i$, where θ_{i-1} is the slope of the previous edge of $\text{AML}(C)$, imposes a (\wedge, \vee) -zone, so the vertex is the inside pixel of an upper leaning point. The reasoning is similar when $\theta_{i-1} < \theta_i$. \square

We can now prove that the polygon $\text{AML}(C)$ is the minimum perimeter polygon of $L_1(C)$, $L_2(C)$, or the so-called MLP of C in the terminology of [4].

Theorem 4. *If C is a simple 4-connected digital contour, then $\text{AML}(C)$ is the convex hull of $L_1(C)$ relative to $L_2(C)$ or, otherwise said, $\text{AML}(C)$ is the intersection of every $L_2(C)$ -convex set containing $L_1(C)$.*

Proof. We proceed in four steps :

1. $\text{AML}(C)$ is a $L_2(C)$ -convex set containing $L_1(C)$. Using Theorem 2 along with Lemma 5 one concludes that if there exist $x, y \in \text{AML}(C)$ such that $\overline{xy} \not\subset \text{AML}(C)$ but $\overline{xy} \subset L_2(C)$ then \overline{xy} splits $L_2(C)^c$ in two disjoint components.
2. Every convex and every concave vertex of $\text{AML}(C)$ belongs to every $L_2(C)$ -convex set containing $L_1(C)$. For convex vertices, this is direct from Lemma 5. For a concave vertex v , using Lemma 3 one may find a pair of point $x, y \in L_1(C)$ such that $v \in \overline{xy} \subset L_2(C)$.
3. Every edge of $\text{AML}(C)$ belongs to every $L_2(C)$ -convex set containing $L_1(C)$. By lemmata 2, 3 and 4, no edge from $\text{AML}(C)$ intersect $L_2(C)^c$ so that the previous step of this proof allows to conclude.
4. Points (2) and (3) implies $\partial\text{AML}(C)$ is included in every $L_2(C)$ -convex set containing $L_1(C)$, which proves that $\text{AML}(C)$ is included in the intersection of every $L_2(C)$ -convex set containing $L_1(C)$. Being itself such a relative convex set (point (1)), it is necessarily the convex hull of $L_1(C)$ relative to $L_2(C)$. \square

4 Combinatorial definition

Based on the combinatorial characterization of digital convexity obtained in [15], we propose a new algorithmic definition of the minimum length polygon. Given an arbitrary ordered alphabet $\mathcal{A} = \{a_1, a_2, \dots, a_n\}$ with the order $a_1 < a_2 < \dots < a_n$, written $\mathcal{A} = \{a_1 < a_2 < \dots < a_n\}$ for short, we extend this order to words over \mathcal{A} using the lexicographic order. We note $|w|_a$ the number of occurrences of the letter a in w and $|w| = \sum_{a \in \mathcal{A}} |w|_a$ is the length of w . Let \mathcal{A}^n be the set of all words of length n over \mathcal{A} , in particular $\mathcal{A}^0 = \{\varepsilon\}$ where ε is called the *empty word*. A word w is *non-empty* if $w \neq \varepsilon$. The i -th letter of a word w is $w[i]$ and we refer to factors of w like this : $w = w[1 : i-1] \cdot w[i : i+j] \cdot w[i+j+1 : n]$, where \cdot is the concatenation and $|w| = n$. By reference to the Freeman coding, given a word $w \in \{0, 1, 2, 3\}^n$ the translation vector associated to w is $\vec{w} = (|w|_0 - |w|_2, |w|_1 - |w|_3)$.

Definition 6. A non-empty word w over the ordered alphabet \mathcal{A} is a Lyndon word if $w < v$ for any non-empty word v such that $w = uv$. We note $\mathbf{L}_{\mathcal{A}}$ the set of all Lyndon words over \mathcal{A} .

Theorem 5 ([16] Theorem 5.1.1). Any non-empty word w over \mathcal{A} admits a unique factorization as a sequence of decreasing Lyndon words : $w = l_1^{n_1} l_2^{n_2} \dots l_k^{n_k}$ with $l_1 > l_2 > \dots > l_k$ where $n_i \geq 1$ and $l_i \in \mathbf{L}_{\mathcal{A}}$ for all $1 \leq i \leq k$.

We define the function FLF, called *first Lyndon factor*, as $\text{FLF}(w, \mathcal{A}) = (l_1, n_1)$ where $w = l_1^{n_1} l_2^{n_2} \dots l_k^{n_k}$ is the unique factorization of w in decreasing Lyndon words according to the ordered alphabet \mathcal{A} .

Introduced by Christoffel in [17], Christoffel words were reinvestigated by Borel and Laubie in [18]. Since then their impressive combinatorial structure has been studied by many, see [19] for a comprehensive self-contained survey. Here is one of the many equivalent definitions of Christoffel words.⁴

Definition 7. A Christoffel word on the alphabet $\{a < b\}$ is the Freeman code of a the path joining two consecutive upper leaning points of a DSS with positive slope according to the convention that a codes an horizontal step and b codes a vertical one.

Once again, referring to the Freeman code, one defines the *slope* of a word as $\rho(w) = |w|_b / |w|_a$ with the convention that $1/0 = \infty$. In the case of Christoffel words, unlike the general case, there is a direct link between lexicographic order and the slope : $u, v \in \mathbf{C}_{a < b} \implies (u < v \iff \rho(u) < \rho(v))$.

A convex polyomino being composed of only one convex zone the quadrant words $Q_{a < b}$ provide a natural decomposition of its border in four quadrant words. Our combinatorial view of convexity is based the following result which characterizes convex quadrant words.

Theorem 6 ([15]). A hv -convex polyomino P is convex if and only if the factorization as decreasing Lyndon words of each quadrant words $Q_{a < b} = l_1^{n_1} l_2^{n_2} \dots l_k^{n_k}$ is such that $l_i \in \mathbf{C}_{a < b}$ for all $1 \leq i \leq n_k$. Moreover, in the case where P is convex, for each quadrant, the edges of its convex hull coincide with the vectors $n_i \vec{l}_i$.

4.1 Definition of the CMLP

We define the combinatorial minimum length polygon algorithmically using Algorithm 3 which simply computes vertices given by a list a edges determined by Algorithm 2. We suppose that the word w codes the boundary of a polyomino P starting from the point (x_0, y_0) the lowers point among the leftmost points of P (i.e. $x_0 = \min\{x | (x, y) \in P\}$ and $y_0 = \min\{y | (x_0, y) \in P\}$).

In order to illustrate how Algorithm 2 works, let us discuss the geometrical interpretation of the modifications performed to the alphabet A in Algorithm 2. First, notice that initially the alphabet is set as $A = \{3 < 0 < 1 < 2\}$ as we

⁴ These words are sometimes referred as *primitive lower Christoffel* words.

Algorithm 2: nextEdge

Input: (u, n, A) such that $u \in A^n$ and $A = \{a_1 < a_2 < a_3 < a_4\}$.
Output: (x, l, A) with $x \in \mathbb{Z}^2$, $l \in \mathbb{N}$.

- 1 $(v, k, inC) \leftarrow \text{FLF}(u, A)$;
- 2 $x = k\vec{v}$; $l = k|v|$;
- 3 **if** $v = a_2$ **then**
- 4 $A \leftarrow \{a_4 < a_1 < a_2 < a_3\}$;
- 5 $x \leftarrow x - \vec{v}$;
- 6 **else if** *not inC* **then**
- 7 $u[0] \leftarrow a_3$;
- 8 $A \leftarrow \{a_4 < a_3 < a_2 < a_1\}$;
- 9 $(x, l, A) \leftarrow \text{nextEdge}(u, n, A)$
- 10 **return** (x, l, A)

Algorithm 3: Computation of vertices from Algorithm 2.

Input: $w \in \{0, 1, 2, 3\}^n$ the boundary word of P .
Output: (x_0, x_1, x_2, \dots) a list of vertices that form the CMLP of P .

- 1 $x \leftarrow (0, 0)$; $i \leftarrow 0$; $A \leftarrow \{3 < 0 < 1 < 2\}$;
- 2 $w \leftarrow w \cdot 10$; $n \leftarrow n + 2$;
- 3 **while** $w \neq \varepsilon$ **do**
- 4 $(v, l, A) = \text{nextEdge}(w, n, A)$;
- 5 $x_{i+1} \leftarrow x_i + v$; $i \leftarrow i + 1$;
- 6 $w \leftarrow w[l + 1 : n]$; $n \leftarrow n - l$;
- 7 **return** $(x_0, x_1, x_2, \dots, x_i)$

consider a convex zone with quadrant word $Q_{0 < 1}$. All through the algorithm, it shall always be the case that when analysing a convex quadrant word $Q_{a_2 < a_3}$ the alphabet A is set as $\{a_1 < a_2 < a_3 < a_4\}$ so that the word a_2a_1 codes a quadrant change while a_3a_4 codes a change of convexity type.

A bijective map $\mu : \mathcal{A} \rightarrow \mathcal{A}$ over the letters \mathcal{A} extends naturally to any word $w \in \mathcal{A}^n$ as $\mu(w) = \mu(w[1])\mu(w[2]) \cdots \mu(w[n])$. Using the notation $\mu(\mathcal{A}) = \{\mu(a_1) < \mu(a_2) < \dots\}$, clearly $\mu(w) \in \mathbf{L}_{\mathcal{A}} \iff w \in \mathbf{L}_{\mu(\mathcal{A})}$ and $\mu(w) \in \mathbf{C}_{a < b} \iff w \in \mathbf{C}_{\mu(a) < \mu(b)}$. Algorithm 2 uses this fact so that instead of applying transformations to the whole word w , only the order relation over the four letter alphabet is changed. Let w be the contour word of C over $\mathcal{A} = \{3 < 0 < 1 < 2\}$ and define $r : \mathcal{A} \rightarrow \mathcal{A}$ as $r(3) = 2$, $r(0) = 3$, $r(1) = 0$ and $r(2) = 1$. One verifies that the contour coded by $r(w)$ corresponds to a rotation by $\pi/2$. This explains line 4 of Algorithm 2 which is called when a quadrant change occurs.

Similarly, let $A = \{a_1 < a_2 < a_3 < a_4\}$ and define the *bar* operator $\overline{(\)}$ as $\overline{a_1} = a_4$, $\overline{a_2} = a_3$, $\overline{a_3} = a_2$ and $\overline{a_4} = a_1$. Consider a quadrant word $Q_{a < b}$, one verifies that $\overline{Q_{a < b}}$ corresponds to a reflexion by the line $y = x$ if $\{a, b\} \in \{\{0, 1\}, \{2, 3\}\}$ or by the line $y = -x$ if $\{a, b\} \in \{\{0, 3\}, \{1, 2\}\}$. Roughly speaking, this transformation turns this part of the contour *inside out*, so that

computing the left envelope of $Q_{a<b}$ is equivalent to the computation of the right envelope of $\overline{Q_{a<b}}$. This explains line 8 of Algorithm 2.

The modification of the first letter of w at line 7 is due to the fact that w code the inter-pixel path. Since the condition at line 6 detects a change in the convexity type, the inner pixel adjacent to the step coded by $w[1]$ must now be consider as an outside pixel. This is done by switching the value of the first letter of w from a_2 to a_3 .

Definition 8. *The combinatorial MLP of a digital contour C , noted $CMLP(C)$, is obtained by joining consecutives vertices given as output of Algorithm 3.*

We suppose that the lower pixel among the leftmost pixels of the polyomino P is centered at $(0, 0)$. This ensures that point $v_0 = (0, 0)$ is a vertex of $CMLP(C)$. Moreover, in order to close the polygonal path computed, the word 10 is added at the end of w so that an extra vertex located at $(0, 0)$ is added at the end of the list closing the polygonal line.

Proposition 1. *The $AMLP(C)$ and $CMLP(C)$ are the same.*

Proof. Quadrant changes are detected by the condition at line 3 of Algorithm 2. Note that in such case, the first letter is ignored since the inter-pixel path is always one step longer than the corresponding part of the border of $L_1(C)$.

On a concave part of C , the reversal of the alphabet at line 8 of Algorithm 2 reverses the perspective and allow to use exactly the same algorithms to compute the convex hull of the outside pixels of C . Finally, it remains to check that after having performed the operations on lines 7 and 8 of Algorithm 2, the edge computed by $FLF(w, A)$ is the same as the segment of the $AMLP(C)$ corresponding to this inflexion zone. \square

4.2 An adapted implementation of FLF

In order to compute function FLF one may use Duval's algorithm [20] which computes the pair $(u, k) = FLF(w, \mathcal{A})$ for any word w with a time complexity of $O(k|u|)$. This optimal algorithm is attributed to [21] in [22]. In Algorithm 2 we compute $(u, k) = FLF(w, \mathcal{A})$ but, in the case where u is not a Christoffel word, we do not care about this specific output (u, k) . Based on this idea, we propose a modified version of Duval's algorithm. Algorithm 4 has the ability to determine dynamically if the word being read might lead to a Christoffel word and otherwise the computation immediately stops. Note that by removing lines identified by numbers from Algorithm 4, one obtains exactly Duval's algorithm. The extra variables p and q have the following interpretation: p is the biggest index smaller than j such that $w[1 : p]$ is a Christoffel word, while q is the smallest index bigger or equal to j such that $w[1 : q]$ might be a Christoffel word. The way p and q are updated is based on the following result.

Proposition 2. *Given a Christoffel word w , define $\chi_w = \{wz \in \mathbf{C}_{a<b} \mid z \neq \varepsilon\}$. For any $x \in \chi_w$, there exist $k \geq 1$ and v such that $x = w^k v z$ for some z where*

Algorithm 4: Duval++

Input: $(w, n, \{a_1 < a_2 < a_3 < a_4\})$ where $w \in \{a_2, a_3\} \cdot \{a_1, a_2, a_3, a_4\}^{n-1}$.
Output: (u, k, b) where b is a boolean such that $b \iff (u, k) = \text{FLF}(w)$.
 $i \leftarrow 1; j \leftarrow 2;$
1 $p \leftarrow 1; q \leftarrow 2;$
 while $j \leq n$ **and** $w[i] \leq w[j]$ **do**
 if $w[i] = w[j]$ **then**
 2 **if** $j = q$ **then**
 3 $q \leftarrow q + p;$
 $i \leftarrow i + 1;$
 else
 4 **if** $j \neq q$ **or** $w[j] \neq a_3$ **then**
 5 $\text{return } (\varepsilon, 0, \text{false});$
 6 $\text{tmp} \leftarrow p; p \leftarrow q; q \leftarrow 2q - \text{tmp};$
 $i \leftarrow 1;$
 $j \leftarrow j + 1;$
 return $(w[1 : j - i], \lfloor \frac{j-1}{j-i} \rfloor, \text{true});$

$w = uv$ is the unique factorization of w as two Christoffel words⁵ if $|w| \geq 2$ and $v = b$ if $|w| = 1$. Moreover, $w^l v \in \mathbf{C}_{a < b}$ for any $l \geq 1$.

Proof. This is mainly due to a result of [18, 23] stating that any $w = uv \in \mathbf{C}_{a < b}$ is obtained in a unique way as $(u, v) = H_1 \circ H_2 \circ \dots \circ H_n(a, b)$ where each H_i is either the function $(x, y) \rightarrow (xy, y)$ or $(x, y) \rightarrow (x, xy)$. \square

Proposition 3. *Algorithm 3 computes $\text{CMLP}(C)$ in time proportional to the length of C .*

Proof. It suffices to show that the time complexity of Algorithm 2 is bounded by $O(l)$ where (v, l) is its output. First, suppose there is no recursive call. In this case, time complexity is simply given by Duval's algorithm.

On the other hand, suppose that there is a recursive call. This means that the call to FLF has output $(\varepsilon, 0, \text{false})$. Let m be the number of letters read in this first call to Algorithm 4. One may show that in such case, in the recursive call the computation of FLF will provide an output of the form (v, k, true) where $k|v| \geq m/2$. \square

5 Concluding remarks

We have presented two different definitions for the MPL of a digital contour, one based on an arithmetic approach, the other based on a combinatorial one. Both are shown to be the unique MLP of the contour and linear time algorithm

⁵ The factorization of a Christoffel word as a product of two Christoffel word is equivalent to the *splitting formula* of DSS.

to compute them are provided. Even though we did not show it, our notion of MLP has no problem dealing with one pixel wide areas, i.e. holes or bars of one pixel wide, as illustrated in Fig. 2.

Finally, note that one could modify the given algorithms in order to remove aligned points. This is done in a time proportional to the number of vertices of the MLP so that the overall linearity of the algorithms is not affected.

Acknowledgment. The authors would like to thank François de Vieilleville for helpful discussions about MLP.

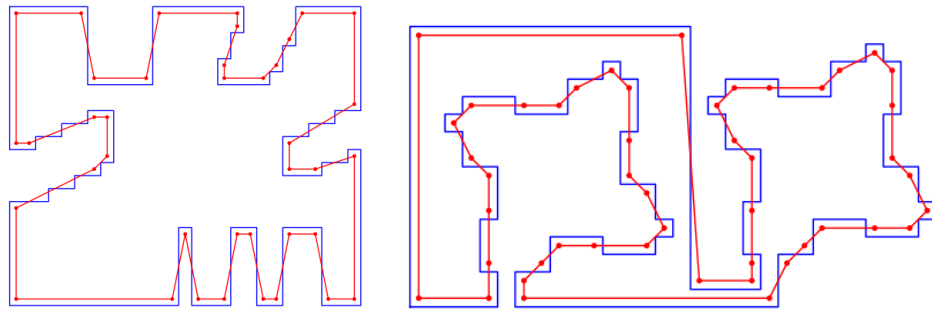


Fig. 2. Examples of CMLP

References

1. Montanari, U.: A note on minimal length polygonal approximation to a digitized contour. *Communications of the ACM* **13**(1) (1970) 41–47
2. Sklansky, J., Chazin, R.L., Hansen, B.J.: Minimum perimeter polygons of digitized silhouettes. *IEEE Trans. Computers* **21**(3) (1972) 260–268
3. Hobby, J.D.: Polygonal approximations that minimize the number of inflections. In: *SODA '93: Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, Philadelphia, PA, USA, Society for Industrial and Applied Mathematics (1993) 93–102
4. Klette, R., Yip, B.: The length of digital curves. *Machine Graphics Vision* **9**(3) (2000) 673–703 (Also research report CITR-TR-54, University of Auckland, NZ. 1999).
5. Coeurjolly, D., Klette, R.: A comparative evaluation of length estimators of digital curves. *IEEE Trans. Pattern Analysis and Machine Intelligence* **26**(2) (2004) 252–258
6. Klette, R., Rosenfeld, A.: *Digital Geometry - Geometric Methods for Digital Picture Analysis*. Morgan Kaufmann, San Francisco (2004)
7. Sloboda, F., Stoer, J.: On piecewise linear approximation of planar Jordan curves. *J. Comput. Appl. Math.* **55**(3) (1994) 369–383

8. Sloboda, F., Zaťko, B.: On one-dimensional grid continua in \mathbb{R}^2 . Technical report, Institute of Control Theory and Robotics, Slovak Academy of Sciences, Bratislava, Slovakia (1996)
9. Sloboda, F., Zaťko, B.: On approximation of Jordan surfaces in 3D. In Bertrand et al., G., ed.: Digital and Image Geometry. Volume 2243 of LNCS. Springer (2001) 365–386
10. Guibas, L.J., Hershberger, J.: Optimal shortest path queries in a simple polygon. In: SCG '87: Proceedings of the third annual symposium on Computational geometry, New York, NY, USA, ACM (1987) 50–63
11. Klette, R., Kovalevsky, V., Yip, B.: On length estimation of digital curves. In: Vision geometry VIII. Volume 3811. (1999) 117–128
12. Doerksen-Reiter, H., Debled-Rennesson, I.: Convex and concave parts of digital curves. In Klette, R., Kozera, R., Noakes, L., Weickert, J., eds.: Geometric Properties for Incomplete Data. Volume 31 of Computational Imaging and Vision., Springer (2006) 145–160
13. Lachaud, J.O., Vialard, A., de Vieilleville, F.: Fast, accurate and convergent tangent estimation on digital contours. Image and Vision Computing **25**(10) (2007) 1572 – 1587 Discrete Geometry for Computer Imagery 2005.
14. Melkman, A.A.: On-line construction of the convex hull of a simple polyline. Inf. Process. Lett. **25**(1) (1987) 11–12
15. Brlek, S., Lachaud, J.O., Provençal, X., Reutenauer, C.: Lyndon + christoffel = digitally convex. Pattern Recognition **42**(10) (2009) 2239 – 2246 Selected papers from the 14th IAPR International Conference on Discrete Geometry for Computer Imagery 2008.
16. Lothaire, M.: Combinatorics on words. Cambridge Mathematical Library. Cambridge University Press, Cambridge (1997)
17. Christoffel, E.B.: Observatio arithmetica. Annali di Matematica **6** (1875) 145–152
18. Borel, J.P., Laubie, F.: Quelques mots sur la droite projective réelle. J. Théor. Nombres Bordeaux **5**(1) (1993) 23–51
19. Berstel, J., Lauve, A., Reutenauer, C., Saliola, F.V.: Combinatorics on words. Volume 27 of CRM Monograph Series. American Mathematical Society, Providence, RI (2009) Christoffel words and repetitions in words.
20. Duval, J.P.: Factorizing words over an ordered alphabet. J. Algorithms **4**(4) (1983) 363–381
21. Fredricksen, H., Maiorana, J.: Necklaces of beads in k colors and k -ary de Bruijn sequences. Discrete Math. **23**(3) (1978) 207–210
22. Lothaire, M.: Applied combinatorics on words. Volume 105 of Encyclopedia of Mathematics and its Applications. Cambridge University Press, Cambridge (2005)
23. Berstel, J., de Luca, A.: Sturmian words, Lyndon words and trees. Theoret. Comput. Sci. **178**(1-2) (1997) 171–203