Dynamic Minimum Length Polygon

J.-O. Lachaud* and X. Provençal

Laboratoire de Mathématiques, UMR 5127 CNRS, Université de Savoie,
73376 Le Bourget du Lac, France.
[jacques-olivier.lachaud, xavier.provencal]@univ-savoie.fr

Abstract. This paper presents a formal framework for representing all reversible polygonalizations of a digital contour (i.e. the boundary of a digital object). Within these polygonal approximations, a set of local operations is defined with given properties, e.g., decreasing the total length of the polygon or diminishing the number of quadrant changes. We show that, whatever the starting reversible polygonal approximation, iterating these operations leads to a specific polygon: the Minimum Length Polygon. This object is thus the natural representative for the whole class of reversible polygonal approximations of a digital contour. Since all presented operations are local, we obtain the first dynamic algorithm for computing the MLP. This gives us a sublinear time algorithm for computing the MLP of a contour, when the MLP of a slightly different contour is known.

1 Introduction

It is often interesting to construct a polygonal approximation of digital contours (i.e. the boundary of a digital object) in order to study their geometry. We are interested in reversible polygonalizations, which have the property that they are digitized exactly as the input digital contour. Classically, a reversible polygon can be obtained by greedy decomposition of the input contour into longest digital straight segments [22, 11]. This decomposition depends on the starting point, but has at most one more edge than the reversible polygon with the minimal number of edges. A reversible polygonal approximation which minimizes the integral summed squared error can also be sought [6]. Another classical reversible polygon is the Minimum Length Polygon (MLP) [13, 17, 7]. It is a good digital tangent and length estimator [10, 3] and is proven to be multigrid convergent in $O(h)$ for digitization of convex shapes, where h is the grid step (reported in [9, 18, 19]). Several linear-time algorithms exist to compute it [14, 15].

The contributions of this paper are twofold. First, we introduce a kind of algebra within all reversible polygonal descriptions of a given digital contour. Each digital contour thus induces a class of reversible polygons which have all the same digitization. A set of valid operations is then defined to pass from one polygon to another within the same class. We show that the subset of operations we provide is necessary and sufficient to go from any element of a class to the

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

MLP of the same class. In a sense, the MLP acts as a natural representative of all the reversible polygonal approximations. Secondly, this approach leads us to a *dynamic* algorithm for computing the MLP. Since all operations are local with controlled complexity, given the MLP of a given contour C of size n, after a local perturbation on this contour, that is to add or remove one pixel to the region bounded by C, the MLP of this new region can be computed in $O(\log n)$.

Furthermore, a dynamic algorithm for computing the MLP is interesting in several applications. It has been shown that the MLP is a very good regularizer for digital deformable models [5]. Unfortunately, existing algorithms are not dynamic, and the MLP is thus recomputed at each iteration for each possible elementary deformations. A dynamic MLP thus induces a dramatic improvement in the computation speed of digital deformable models. Another application is the computation of multiscale digital representations of a digital object [16]. The MLP of each multiscale contour could thus be computed directly from the decomposition into digital straight segments computed by their analytic approach.

2 Preliminaries

We call *digital contour* C a simple 1-curve in $\mathbb{Z}^2$ (in the terminology of [9], §7.3.2, p. 243), such that its cell representation has two boundaries that are Jordan curves. The inner curve (resp. outer curve) is called the *inner polygonal curve* of C (resp. the *outer polygonal curve* of C). The *inner polygon* $\mathrm{IP}(C)$ is the inner polygonal curve with its inside in $\mathbb{R}^2$. Similarly the *outer polygon* $\mathrm{OP}(C)$ is the outer polygonal curve with its inside in $\mathbb{R}^2$. See Fig. 1 for an illustration. Remark that C, $\mathrm{IP}(C)$ and $\mathrm{OP}(C)$ are all polyominoes such that $\mathrm{IP}(C)$ and $\mathrm{OP}(C)$ have vertices in $\mathbb{Z}^2$ while the interpixel path of C has vertices in $\mathbb{Z}^2 + (1/2, 1/2))$.

One can use for instance a Freeman chain to code a digital contour as a word over the alphabet $(0, 1, 2, 3)$, the associated displacements written as: $\overrightarrow{0} = (1, 0)$, $\overrightarrow{1} = (0, 1)$, $\overrightarrow{2} = (-1, 0)$ and $\overrightarrow{3} = (0, -1)$. These words are usually called *contour words* and the contour word of a digital contour C is denoted by $F(C)$. In order to simplify the presentation, we will assume that digital contour are always encoded in a clockwise manner.

Clearly, any curve among the digital contour, the inner polygonal curve and the outer polygonal curve, completely defines the others. For instance starting with the digital contour C, the Freeman chain code of the inner polygonal curve is obtained by removing one step in each turn ab^kc of $F(C)$ with $k \geq 1$ and $(a, b, c) \in \{(0, 3, 2), (1, 0, 3), (2, 1, 0), (3, 2, 1)\}$ by $ab^{k-1}c$ and adding one step in each turn ab^kc with $k \geq 1$ and $(a, b, c) \in \{(0, 1, 2), (1, 2, 3), (2, 3, 0), (3, 0, 1)\}$ by $ab^{k+1}c$. Similarly, the Freeman chain code of the outer polygonal curve is obtained from $F(C)$ in the same way by adding one step in former case and removing one in the latter one.

2.1 Minimum length polygon

Following the works of Sloboda, Zat́ko and Stoer [18, 21, 20] (or see [8, 9]), we define the minimum length polygon (MLP) of C as the shortest Jordan curve

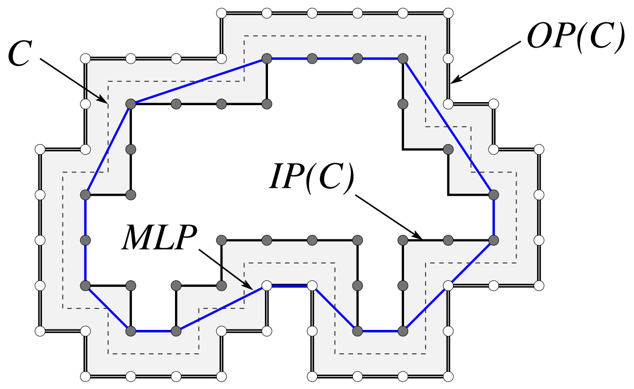


Fig. 1. A digital contour C with its inner polygon $\mathrm{IP}(C)$, its outer polygon $\mathrm{OP}(C)$ and its MLP which is, in a clockwise manner, up to circular permutation, the grid-curve $[(1,0)^3$, σ^+, $(3,2)$, $(1,0)$, σ^+, $\widetilde{(1,1)}$, $\widetilde{(1,1)}$, $(1,0)$, σ^+, $\widetilde{(1,1)}$, $(1,0)$, σ^+, $\widetilde{(2,1)}$, $(1,0)$, σ^+, $(1,1)$, $(1,0)^2$, σ^+, $(1,2)$, $(3,1)]$.

which stays inside the 1-pixel width band drawn by the cell representation of C. More precisely, letting $\mathcal{A}$ be the family of simply connected compact sets of $\mathbb{R}^2$, we define:

Definition 1. *The* minimum perimeter polygon *of two polygons V, U with $V \subset U^\circ \subset \mathbb{R}^2$ is a subset P of $\mathbb{R}^2$ such that*

$$P = \operatorname{argmin}_{A \in \mathcal{A},\ V \subseteq A,\ \partial A \subset U \setminus V^\circ} \mathrm{Per}(A), \tag{1}$$

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

Definition 2. *The* minimum length polygon (MLP) *of a digital contour C is the minimum perimeter polygon of* $\mathrm{IP}(C), \mathrm{OP}(C)$.

Other equivalent definitions for MLP may be found in [14, 15]: they are based either on arithmetic or word combinatorics. In [18], it is shown that Equation (1) has a unique solution.

3 Algebra on reversible polygonal representations

We wish to classify polygons with integer vertices according to the digital contour that they represent. Indeed, many different polygons may represent the same contour. More precisely, a *reversible polygonal representation (RPR) of C* is a polygon whose edges stays in $\mathrm{OP}(C) \setminus \mathrm{IP}(C)^\circ$, whose vertices are integer vertices of either $\mathrm{OP}(C)$ or $\mathrm{IP}(C)$ and with an extra bit of information per vertex specifying if it touches the inside or the outside of the band. It is clear that the digital contour can be reconstructed from such object by computing the set of intersected pixels. If an edge intersects the interior of a pixel, then it is part of C while if the edges intersect only the border of a pixel, to the abovementioned extra bit of information allows to determine if this pixel is part of C or not.

We wish now to find a natural representative for all the RPR of a given contour C. Our approach is to look for the shortest one. Since the MLP of C is one RPR of C, our unique representative will be the MLP. We thus provide a notation for RPR and a set of operations within RPR. These operations are designed so that they shorten the RPR and they act locally. Furthermore, we show that they preserve the digital contour.

3.1 Grid-vector, grid-curve

Definition 3. *A* grid-vector *is a triplet $x = ((p,q),k,\delta_x) \in \mathbb{N}^2 \times \mathbb{N} \times \mathbb{B}$ where* $\gcd(p,q) = 1$, *q/p is the* slope *of x (with the convention that $\infty = 1/0$), $k \geq 1$ is its number of repetitions and the boolean δ_x is true when one endpoint of x lies on the inner polygonal curve and the other endpoint lies on the outer polygonal curve.*

Such grid-vector $x = ((p,q),k,\delta_x)$ is noted $(p,q)^k$ if δ_x is F and $\widetilde{(p,q)}^k$ if δ_x is T, the $\sim$ meaning that a side change has occurred. A grid-vector has no specific orientation since its slope is in $[0,\infty[$. Any grid-vector with slope 0 or ∞ is called *trivial.* The translation associated to a grid-vector is always given relatively to a pair of letter (a,b) from the alphabet $\{0,1,2,3\}$ and is denote using $\xrightarrow{(a,b)}$. These letters represents the current orientation context, i.e. the oriented quadrant. The symbol $\sim$ specifies a side change and thus inverts the current orientation context. Therefore, after this edge, the context is changed (see Fig. 2). Translations for arbitrary grid vectors are given by the formulae:

$$\overset{(a,b)}{\overrightarrow{(p,q)^k}} = k(p\,\overrightarrow{a} + q\,\overrightarrow{b}), \quad \text{and} \quad \overset{(a,b)}{\overrightarrow{\widetilde{(p,q)}^k}} = k(p\,\overrightarrow{b} + q\,\overrightarrow{a}).$$

The reversed point of view induced by the symbol $\sim$ explains the different formulae for translation.

Of course, such translation looses any geometrical interpretation when the pair of letters (a,b) represent elementary steps in the opposite direction as in the case of $(0,2)$ or $(1,3)$. This situation should never happen.

In order to completely describe a circular band, grid-vectors alone are not enough. We need extra information regarding quadrant changes. That is why we introduce the σ operators that act solely on the current orientation context (i.e., on the pairs of letters):

$\sigma^-(a,b) = (b,\overline{a})$: this operator is a turn toward the exterior,
$\sigma^+(a,b) = (\overline{b},a)$: this operator is a turn toward the interior,
with the convention $\overline{0} = 2, \overline{1} = 3, \overline{2} = 0, \overline{3} = 1$.

In order to unify the notations, we associate the nil vector to any operator so that $\overrightarrow{\sigma^+} = \overrightarrow{\sigma^-} = \overrightarrow{(0,0)}$. Similarly, it is convenient to see each grid-vector as an operator on a pair of letters acting in the following way, let $x = ((p,q),k,\delta_x)$,

$$x(a,b) = \begin{cases} (b,a) \text{ if } \delta_x \text{ is } \mathtt{T}, \\ (a,b) \text{ otherwise.} \end{cases}$$

An ordered list of grid-vectors and operators defines a 1-pixel width band in the following way.

Definition 4. *A* grid-curve Γ *is a list* $\Gamma = [l_0, l_1, \ldots, l_{n-1}]$ *where each* l_i *is either a grid-vector or one of the operators* σ^+, σ^-.

The geometrical interpretation of a grid-curve $\Gamma = [l_0, l_1, \ldots, l_{n-1}]$ starting with the pair of letters (a,b) is the polygonal curve $P_\Gamma = [P_0, P_1, \ldots, P_n]$ where each point P_i is given by:

- $P_0 = (0,0)$ and $(a_0, b_0) = (a,b)$.
- $P_{i+1} = P_i + \overset{(a_i,b_i)}{\overrightarrow{l_i}}$ and $(a_{i+1}, b_{i+1}) = l_i(a_i, b_i)$ for each $i \in \{0, 1, \ldots, n-1\}$.

Figure 2 illustrates this construction. This figure also highlights the fact that each grid-curve defines a one pixel-wide band bounded on each side by a digital curve and that the segment associated with each grid-vector $x = ((p,q),k,\delta_x)$ intersects one of these digital curves exactly $k+1$ times. More precisely, each segment starts on one of the two digital curves that form the band, and it intersects one curve exactly every $p+q$ steps. In order to simplify the notations when concatenating two grid-curves, we write $\Gamma(a,b) = (a_n, b_n)$.

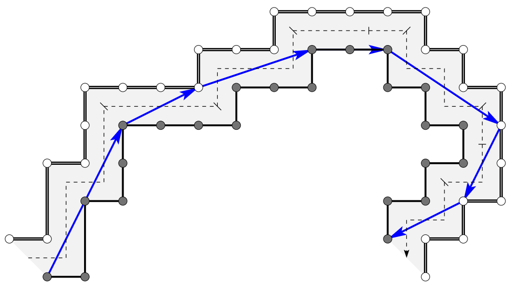
3.2 Christoffel words, interpixel path, RPR

In order to digitize a grid-vector as a piece of digital contour, we make use of a well-known discrete analog to straight segments, the Christoffel words [2, 12]. They have a lot of equivalent definitions, for instance they are the 4-connected Freeman chaincode between two consecutive upper leaning points of a digital straight line, or they are the Lyndon factors of sturmian words (e.g., see [1, 14]).

The Christoffel word of slope q/p over the alphabet (a,b) is denoted by $\mathcal{C}_{q/p}^{(a,b)}$. We then associate to each element l_i of a grid-curve $\Gamma = [l_0, l_1, \ldots, l_{n-1}]$ a word $F_{(a,b)}(l_i)$ defined as

$$\begin{aligned} F_{(a,b)}\left((p,q)^k\right) &= \left(\mathcal{C}_{q/p}^{(a,b)}\right)^n, & F_{(a,b)}(\sigma^-) &= \overline{b}, \\ F_{(a,b)}\left(\widetilde{(p,q)^k}\right) &= a\overline{b}F_{(b,a)}\left((p,q)^k\right), & F_{(a,b)}(\sigma^+) &= a. \end{aligned}$$

The word $F_{(a,b)}(\Gamma)$ is defined by gluing all $F(l_i)$. It may contain factors of the form $a\overline{a}$ which have no real geometrical interpretation. We thus project this word in the free group in order to remove these back and forth moves. This reduced



$$
\begin{aligned}
F_{(0,1)}\left(\left((1,2),2,\mathtt{F}\right)\right) &= (011)^2,\\
F_{(0,1)}\left(\left((1,2),1,\mathtt{T}\right)\right) &= 03\cdot 100,\\
F_{(1,0)}\left(\left((3,1),1,\mathtt{T}\right)\right) &= 12\cdot 0001,\\
F_{(0,1)}\left(\left((1,0),2,\mathtt{F}\right)\right) &= 0^2,\\
F_{(0,1)}\left(\sigma^+\right) &= 0,\\
F_{(3,0)}\left(\left((3,2),1,\mathtt{T}\right)\right) &= 32\cdot 00303,\\
F_{(0,3)}\left(\sigma^-\right) &= 1,\\
F_{(3,2)}\left(\left((2,1),1,\mathtt{F}\right)\right) &= 332,\\
F_{(3,2)}\left(\left((2,1),1,\mathtt{T}\right)\right) &= 30\cdot 223,
\end{aligned}
$$

$$
\begin{aligned}
w &= 011011\cdot 0\,\cancel{31}\,00\cdot 1\,\cancel{20}\,001\cdot 00\cdot 0\cdot 3\,\cancel{20}\,030\,\cancel{31}\,332\cdot 3\,\cancel{02}\,23,\\
&= 011011\cdot 000\cdot 1001\cdot 00\cdot 0\cdot 3030\cdot 3\cdot 32\cdot 323.
\end{aligned}
$$

Fig. 2. Illustration of the grid-curve $\varGamma = [(1,2)^2, \widetilde{(1,2)}, \widetilde{(3,1)}, (1,0)^2, \sigma^+, \widetilde{(3,2)}, \sigma^-, (2,1), \widetilde{(2,1)}]$ starting with the letters $(0,1)$. Each vector of the polygonal curve $P_\varGamma$ is represented by an arrow. The interpixel path w obtained by the concatenation of the Freeman code associated to each element of the grid-curve $\varGamma$ is shown with a dashed line.

word (each $a\bar{a} = \varepsilon$, the empty word) is called the *interpixel path* $F^{\varepsilon}_{(a,b)}(\varGamma)$. See Fig. 2.

Now, let (Q_i) be the clockwise vertices of some RPR of C and let λ_i be the boolean that is true when Q_i lies on $\mathrm{IP}(C)$. The following sequence defines the *grid-curve* of Q, for all i:

1. if $\overrightarrow{Q_{i-1}Q_i}$ and $\overrightarrow{Q_iQ_{i+1}}$ are not in the same quadrant:
   - if $\lambda_i = \mathtt{T}$
     - if $\overrightarrow{Q_iQ_{i+1}}$ is to the right of $\overrightarrow{Q_{i-1}Q_i}$, append as many σ^+ as the number of clockwise $\pi/2$ rotations to bring these two vectors in the same quadrant.
     - else append as many σ^- as the number of counterclockwise $\pi/2$ rotations to bring these two vectors in the same quadrant.
   - else append the opposite operators of above paragraph.
2. letting (a,b) be the current orientation context, then $\overrightarrow{Q_iQ_{i+1}}$ is some $u\vec{a} + v\vec{b}$; let also $p = \gcd(u,v)$.
   - if $\lambda_i = \lambda_{i+1}$ append $(u/p, v/p)^p$,
   - else append $\widetilde{(v/p, u/p)}^p$.

The following proposition formalizes the fact that the previous construction does build the correct digital contour.

Proposition 1. *For any RPR Q of a digital contour C, the interpixel path of the grid-curve of Q has the same chaincode as C up to conjugacy.*

For space reasons, we do not detail the proof here. Using this property we will be able to validate simplification operations defined on grid-curves by showing that they preserve the digital contour.

In order to adopt a local approach to process RPR and the associated grid-curve, we will consider sublists of such grid-curves. Note that they are also grid-curves (but they do have extremities unlike grid-curves defined by RPR).

We introduce an equivalence relation between these objects, because they geometrically define the same one pixel-wide band and preserve the orientation context.

Definition 5. *Two grid-curves $\Gamma = [l_0, l_1, \dots, l_{n-1}]$ and $\Gamma' = [l'_0, l'_1, \dots, l_{m-1}]$ are* equivalent*, noted $\Gamma \equiv \Gamma'$, if $F^{\varepsilon}_{(0,1)}(\Gamma) = F^{\varepsilon}_{(0,1)}(\Gamma')$ and $\Gamma(0,1) = \Gamma'(0,1)$.*

For instance, $[(1,1),(1,1)] \equiv [(1,1)^2]$ since $01 \cdot 01 = (01)^2$ and both curves leave the alphabet unchanged. On the other hand, Fig. 5 shows a less trivial example.

4 Simplification rules

In order to compute the canonical representation of a grid path, we define simplification rules, which are detailed in the following subsections:

– Merging rules: given by equations (2) and (3).
– Splitting rules: given by equation (4).
– Operator simplification rules: given in Section 4.4.

When applied to a grid curve Γ, each of these rules generates another grid curve Γ' such that $F^{\varepsilon}_{(a,b)}(\Gamma) = F^{\varepsilon}_{(a,b)}(\Gamma')$ but in each case, we can ensure that $\|\Gamma\| < \|\Gamma'\|$ (the euclidean length is smaller), or $\|\Gamma\| = \|\Gamma'\|$ and $|F_{(a,b)}(\Gamma)| < |F_{(a,b)}(\Gamma')|$ (same euclidean length but shorter word).

4.1 Grid-vectors fusion rules

Given two vectors $\overrightarrow{u} = (p,q)$ and $\overrightarrow{v} = (r,s)$ it is well know that the area of the oriented parallelogram defined by $\overrightarrow{u}$ and $\overrightarrow{v}$ is simply $ps - qr$. According to this, we define the product of $x = ((p,q), k, \delta_x)$ and $y = ((r,s), l, \delta_y)$ as

$$x \otimes y = \begin{cases} ps - qr & \text{if } \delta_y = \mathtt{F}, \\ pr - qs & \text{if } \delta_y = \mathtt{T}. \end{cases}$$

The sign, positive or negative, of $x \otimes y$ has the following geometrical interpretation:

– $x \otimes y < 0$. In such case, the grid-curve $[x, y]$ defines a convex vertex of the *RPR* that is optimal in the sense that it may not be replaced by a shorter polygonal curve that stays within $\mathrm{OP}(C) \setminus \mathrm{IP}(C)^{\circ}$.
– $x \otimes y = 0$. In such case, x and y are co-linear. When $\delta_y = \mathtt{F}$ then y may simply be added to the number of repetitions of x so that $[x, y]$ may be replaced by $[((p,q), k+l, \delta_x)]$:

$$[((p,q), k, \delta_x), ((p,q), l, \mathtt{F})] \equiv [((p,q), k+l, \delta_x)]. \qquad (2)$$

On the other hand, if $\delta_y = \mathtt{T}$ then no simplification is possible and $[x, y]$ is optimal.

- $x \otimes y > 0$. In such case, $[x, y]$ is not optimal and there exist some grid-curve Γ such that $\|\Gamma\| < \|[x, y]\|$.

Therefore we focus our attention on pairs of grid-vectors x and y such that $x \otimes y > 0$. There are two cases to consider, whether $x \otimes y = 1$ or $x \otimes y > 1$. In the first case, the two grid-vectors are compatible: there is no integer points in the triangle formed by (p, q) and (r, s) and these two segments may be replaced by the segment $(p + r, q + s)$. This is detailed in Section 4.2. In the second case however, the latter segment would go outside of $\mathrm{OP}(C) \setminus \mathrm{IP}(C)^{\circ}$. Hence, $[x, y]$ has to be replaced by some grid-curve $\Gamma = [l_1, l_2, \ldots, l_n]$ in which for each i from 1 to $n - 1$, $l_i \otimes l_{i+1} \leq 0$. In section 4.3 we show how to compute Γ in time proportional to the depth of the continued fraction development of the slopes of x and y.

4.2 Merging grid-vectors

The following merging operation is based on the well known *splitting formula* of digital straight segments (see [23]). This is equivalently known in the field of word combinatorics as the *standard factorization* of Christoffel words (see [1, 2]).

Let $x = ((p, q), 1, \mathtt{F})$ and $y = ((r, s), 1, \mathtt{F})$ be two grid-vectors such that $x \otimes y = 1$. In such case, x and y may be merged in order to form the grid-vector $z = ((p + r, q + s), 1, \mathtt{F})$ so that

$$[x, y] \equiv [z] \text{ and } \|[x, y]\| > \|[z]\|.$$

In particular, a grid-curve of the form $[x, x, \ldots, x, y]$ with k copies of x may be simplified to the shorter $[((kp + r, kq + s), 1, \delta_x)]$. Similarly, $[x, y, y, \ldots, y]$, with l copies of y, is merged to $[((p + lr, q + ls), 1, \mathtt{F})]$. On the other hand, if both x and y are repeated more then one time, a more complex simplification operation (explained in Section 4.3) is needed. By taking into account the possible changes from the inner to the outer polygon and vice versa, we obtain the following merging rule:

- Let $x = ((p, q), k, \delta_x)$ and $y = ((r, s), l, \delta_y)$ with either $\delta_y = \mathtt{F}$ and $\min(k, l) = 1$ or $\delta_y = \mathtt{T}$ and $l = 1$, then

$$[x, y] \equiv [z] \text{ where } z = \begin{cases} ((kp + lr, kq + ls), 1, \delta_x) & \text{if } \delta_y = \mathtt{F}. \\ ((kp + ls, kq + lr), 1, \neg\delta_x) & \text{otherwise.} \end{cases} \tag{3}$$

4.3 Split and merge formulae

In order to simplify a grid-curve into a shorter one, we introduce splitting operations. Given two grid-vectors x and y such that $x \otimes y > 1$, both x and y are split in a specific manner. The resulting grid-curve is such that merging operations are sufficient in order to obtain the shortest equivalent grid-curve.

Let $x = ((p, q), 1, \mathtt{F})$ be a non-trivial grid-vector and let $[u_0; u_1, \ldots, u_n]$ be the continued fraction development of q/p. We note q_i/p_i is the i-th convergent of q/p

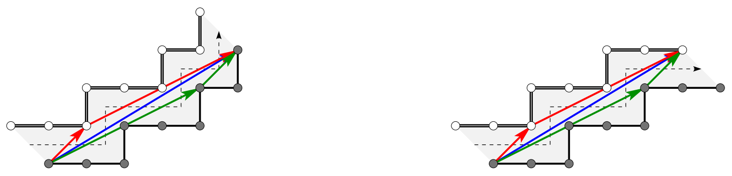


Fig. 3. Illustration of the upper and lower splitting operations. On the left, $(3/5)$ is split to $S_\downarrow((3/5)) = [\,(1/2)^2, (1/1)\,]$ and $S^\uparrow((3/5)) = [\,\widetilde{(1/1)}, (2/1), \widetilde{(1/2)}\,]$. On the right, the segment $\widetilde{(5/3)}$ is split to $S_\downarrow(\widetilde{(5/3)}) = [(1/2)^2, \widetilde{(1/1)}]$ and $S^\uparrow(\widetilde{(5/3)}) = [\widetilde{(1/1)}, (2/1)^2]$. On both examples, the upper splitting $S^\uparrow$ is shown in red while the lower splitting $S_\downarrow$ is shown in green.

that is the fraction $q_i/p_i = [u_0; u_1, \ldots, u_i]$. In order to lighten the presentation, we introduce the following notations: $x_i = (p_i, q_i)$, $x_{-1} = (0, 1)$, $x_{-2} = (1, 0)$. Also, more generally, if $y = ((r, s), l, \delta_y)$ then $y^{-1} = ((s, r), l, \delta_y)$ and $\widetilde{y} = ((r, s), l, \neg\delta_y)$. The following operation is exactly the opposite operation of (3).

Definition 6. *The* basic splitting *of the grid-vector x_n is the grid-curve:*

$$S(x_n) = \begin{cases} [x_{2m-2}, x_{2m-1}^{u_{2m}}] \text{ if } n = 2m, \\ \\ [x_{2m}^{u_{2m+1}}, x_{2m-1}] \text{ if } n = 2m+1, \end{cases}$$

Computing the inverse of any number from its continued fraction development is an easy task. This observation leads to following formula:

$$\text{if } S(x) = [u^k, v^l] \text{ then } S(x^{-1}) = [(v^{-1})^l, (u^{-1})^k].$$

The basic splitting operation is only defined on grid-vector of the form $x = ((p, q), 1, \delta_x)$ with $\delta_x = \mathtt{F}$. We extend it to all segments of multiplicity one.

Definition 7. *The upper splitting $S^\uparrow$ and the lower splitting $S_\downarrow$ are defined as follow, let $[u^k, v^l] = S\,((p, q), 1, F))$ then*

	$x = ((p, q), 1, F)$	$x = ((q, p), 1, T)$
$S^\uparrow(x)$	$[\,\widetilde{(v^{-1})^l}, (u^{-1})^{k-1}, \widetilde{u}\,]$	$[\,\widetilde{(v^{-1})^l}, (u^{-1})^k\,]$
$S_\downarrow(x)$	$[\,u^k, v^l\,]$	$[\,u^k, v^{l-1}, \widetilde{v^{-1}}\,]$

By repetition of the above splittings, any grid-vector may be decomposed in order to isolate a trivial grid-vector on the left or the right. See Fig. 4 for an illustration of the following definition.

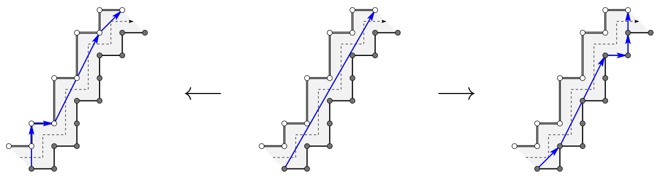


Fig. 4. In the center, an illustration of the grid-vector $x = \widetilde{(7,4)}$. On the left, the left splitting $S_{\leftarrow}(x) = [\widetilde{(1,0)}^2, (0,1), \widetilde{(2,1)}^2, (1,1)]$. On the right, the right splitting $S_{\rightarrow}(x) = [(1,1), (1,2)^2, (1,0), (0,1), \widetilde{(1,0)}]$.

Definition 8. *Let $x = ((p,q), k, \delta_x)$ be a grid-vector, the* left splitting $S_{\leftarrow}(x)$ *and the* right splitting *are defined as follows:*

- *if $k = 1$ and x is trivial, $S_{\leftarrow}(x) = S_{\rightarrow}(x) = [x]$;*
- *if $k \geq 2$,* $S_{\leftarrow}(x) = S_{\leftarrow}(((p,q),1,\delta_x)) + [((p,q),k-1,\mathtt{F})]$,
  $S_{\rightarrow}(x) = [((p,q),k-1,\delta_x))] + S_{\rightarrow}(((p,q),1,\mathtt{F}))$;
- *otherwise, let $S^{\uparrow}(x) = [l_1, l_2, \ldots, l_n]$ and $S_{\downarrow}(x) = [l'_1, \ldots, l'_m]$* (4)

$$S_{\leftarrow}(x) = S_{\leftarrow}(l_1) + [l_2, \ldots, l_n],$$
$$S_{\rightarrow}(x) = \begin{cases} [l_1, \ldots, l_{n-1}] + S_{\rightarrow}(l_n) \text{ if } \delta_x = \mathtt{F}, \\ [l'_1, \ldots, l'_{m-1}] + S_{\rightarrow}(l'_m) \text{ if } \delta_x = \mathtt{T}. \end{cases}$$

Proposition 2. *Let $x = ((p,q), k, \delta_x)$ be a grid-vector, both left and right splittings of x are such that: $[x] \equiv [S_{\leftarrow}(x)] \equiv [S_{\rightarrow}(x)]$.*

Sketch of the proof. These equivalences come from successive applications of splitting formula on digital straight segments.

Proposition 3. *The number of grid-vector in both left and right splittings of $x = ((p,q), k, \delta_x)$ is $\Theta(n)$ where n is the depth of the continued fraction development of q/p.*

Proof. It suffices to see that each time a basic splitting operation is performed (Definition 6), the depth of the continued fraction development of the slopes decreases by one or two. □

Since the depth n of a continued fraction $q/p = [u_0; u_1, \ldots, u_n]$ is smaller than $\log_2(p+q)$, a weaker form of the previous proposition states that the number of grid-vector is some $O(\log N)$ where N is the length of the contour word.

Algorithm 1 illustrates how to simplify a grid-cruve by the use of our *split and merge* formulae. The function `Merge` called on line 15 of Algorithm 1 performs the following task: the two grid-curves given as input being a right splitting,

Algorithm 1: `Simplification`

```
Input: $\mathit{\Gamma} = [l_0, l_1, \ldots, l_{n-1}]$ where each $l_i$ is a grid-vector
1  $\mathit{\Delta} = [\,]$;
2  while $\mathit{\Gamma}$ is not empty do
3      $y \leftarrow \mathit{\Gamma}.pop\_front()$ ;
4      if $\mathit{\Delta}$ is empty then
5          $\mathit{\Delta}.push\_back(y)$;
6      else
7          $x \leftarrow \mathit{\Delta}.pop\_back()$;
8          if $x \otimes y < 0$ or $(x \otimes y = 0$ and $\delta_y = T)$ then
9              $\mathit{\Delta}.push\_back(x)$;
10             $\mathit{\Delta}.push\_back(y)$;
11         else
12             if there exist $z$ such that $[z] \equiv [x, y]$ then
13                 $\mathit{\Gamma}.push\_front(z)$;
14             else
15                 $\mathit{\Gamma} \leftarrow$ Merge $(S_{\rightarrow}(x), S_{\leftarrow}(y)) + \mathit{\Gamma}$;
16 return $\mathit{\Delta}$;
```

$S_{\rightarrow}(x) = [r_1, r_2, \ldots, r_n]$ and a left splitting, $S_{\leftarrow}(y) = [l_1, l_2, \ldots, l_m]$, both r_n and l_1 are trivial and may be replaced by the grid-vector $(1, 1)$. Boths lists are then concatenated into $C = [r_1, \ldots, r_{n-1}, (1, 1), l_2, \ldots, l_m]$. Finally, if there is a pair of consecutive grid-vectors u, v in C such that, according to the fusion rules described by equations (2) and (3), there exist z satisfying $[u, v] \equiv [z]$, then the pair u, v is replaced by z. This last step is performed iteratively until there are no such pairs left.

The following proposition states that whenever line 15 of Algorithm 1 is executed, the grid-curve is simplified in the sense that output curve is strictly shorter.

Proposition 4. *Given two grid-vectors x and y such that $x \otimes y > 1$, the grid-curve $\mathit{\Gamma} =$ `Merge` $(S_{\rightarrow}(x), S_{\leftarrow}(y))$ is such that $\|\mathit{\Gamma}\| < \|[x, y]\|$.*

Sketch of the proof. Consider the grid-curve $\mathit{\Gamma} = S_{\rightarrow}(x) + S_{\leftarrow}(y)$ and the associated polygon $P_{\mathit{\Gamma}} = [P_0, P_1, \ldots, P_n]$, as defined in Section 3.1. Let P_i be the point between $S_{\rightarrow}(x)$ and $S_{\leftarrow}(y)$. One checks that all points P_1, P_2, $\ldots$, P_{i-1}, P_{i+1}, $\ldots$, P_{n-1} lies on the same polygonal contour (inner or outer) while P_i lies on the other one. The fusion of two grid-vectors removes a points from the associated polygon. In particular, the first operation performed by `Fusion` is to remove P_i from $P_{\mathit{\Gamma}}$. Each pair of consecutive grid-vectors u, v from C is such that either $u \otimes v \leq 0$ or the pair u, v is replaced by a single grid-vector. When the process stops, the resulting grid-curve $\mathit{\Delta}$ defines a convex region.

Finally, let X be some point far enough from P_0 in the direction $\overrightarrow{y} - \overrightarrow{x}$. Consider $\mathit{\Pi}_{\mathit{\Delta}}$ the polygon defined by $\mathit{\Delta}$ followed by the line segments $\overline{P_n X}$ and

$\overline{XP_0}$ and, similarly, Π_{xy} the polygon $P_0P_iP_nA$. The polygon Π_Δ is a convex polygon strictly included in Π_{xy} and thus its perimeter is strictly smaller.

4.4 Simplification rules for operators

As mentioned previously, operator σ^+ codes a quadrant change toward the inside which means that a part of a RPR of the form $[x, \sigma^+, y]$ is locally optimal. On the other hand, operator σ^- may not appear in a MLP since it codes a quadrant change toward the outside (See Fig. 2). We define local rules to remove the σ^- operators in a grid-curve.

First of all, using the left and right splitting operations defined in the previous section, we can easily update the grid-curve so that around a σ^- operator there are only trivial grid-vectors. Also, by using the relation $[(0,1)] \equiv [\sigma^-, (1,0), \sigma^+]$ we may only consider trivial grid-vectors with slope 0.

Simplification rules for operator σ^- are all local and thus treated in constant time. These rules are of three types:

Push to the right. The following rules create a shorter grid-curve by replacing a pattern of the form $_\rightarrow\!\uparrow$ by $\nearrow$.

$$[(1,0), \sigma^-, (1,0)] \equiv [(1,1), \sigma^-], \quad \text{and} \quad [\widetilde{(1,0)}, \sigma^-, (1,0)] \equiv [\widetilde{(1,1)}, \sigma^-]. \tag{5}$$

Cancellation rules. Given an occurrence of σ^- in a grid-curve, the trivial grid-vectors right before and after may go back and forth within a single pixel. Such situation appears in a *locally-closed pattern*. It is a grid-curve such that: (i) it includes exactly one or two trivial grid-vectors before σ^- and one or two more after σ^-, (ii) it is closed in the sense that the first and last points of P_Γ are the same. Given such locally-closed pattern Γ, if there exists $\Delta \in \{[\,], [\sigma^+], [\sigma^+, \sigma^+]\}$ such that $\Gamma \equiv \Delta$ then replace Γ by Δ. When implemented these rules may be tabulated so as to apply them in constant time.

Correction rules. When performing a left or right splitting operation, we make the assumption that all points below the grid-vector belongs to the same polygon (inner or outer) and all those above lies in the other polygon. Although this is true in general, it may not be the case for extremities. For example, consider the grid-vector $(1,4)$ from Fig. 5. In order to obtain a trivial grid-vector right before σ^-, $(1,4)$ is replaced by $S_\rightarrow(x) = [\widetilde{(1,0)}^4, \widetilde{(1,0)}]$ while the correct substitution would be to replace $(1,4)$ by $[\widetilde{(1,0)}^3, \widetilde{(0,1)}, (1,0)]$. One could modify the splitting operations in order to take these situation into account but we prefer to use above splitting operations as is and eventually correct the curve afterward. These errors are locally-closed patterns in which the curve changes from one polygon (inner or outer) to the other one an odd number of times. Geometrically, this would imply that a point belongs to both polygons at the same time, which is impossible. Our splitting operations may cause only two types of these faulty locally-closed patterns. Let $\Delta = [l_1, \ldots, l_k]$ be a locally-closed pattern of the grid-curve Γ:

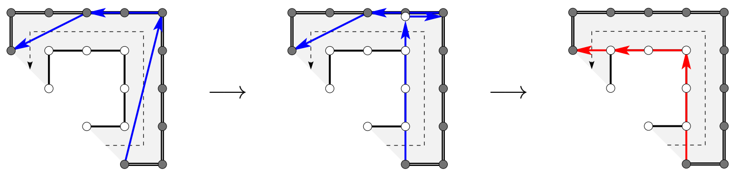


Fig. 5. Illustration of a grid-curve simplification. On the left $\Gamma = [(1,4), \sigma^-, (0,1)^2, \sigma^-, (2,1)]$ and on the right $\Delta = [\widetilde{(1,0)^3}, \sigma^+, \sigma^+, (0,1)^2, \widetilde{(1,0)}]$ and $\Gamma \equiv \Delta$ since $01111 \cdot 3 \cdot 0 \cdot 22223 = 03 \cdot (1)^3 \cdot 1 \cdot 2 \cdot (2)^2 \cdot 30 \cdot 2$ and both curves transform the alphabet $(0,1)$ into $(2,3)$. On the other hand the euclidean length of Δ is smaller than the length of Γ.

– if $F^{\varepsilon}_{(0,1)}(\Delta) = 0$ and $\Delta((0,1)) = (2,1)$. In this case, let x be the trivial grid-vector right after Δ in Γ, the pattern $\Delta + [x] = [l_1, \ldots, l_k, x]$ is replaced by $[\sigma^-, \widetilde{x}]$.
– if $F^{\varepsilon}_{(0,1)}(\Delta) = 0$ and $\Delta((0,1)) = (3,2)$. In this case, let y be the trivial grid-vector with multiplicity one right before Δ in Γ, the pattern $[y] + \Delta$ is replaced by $[\sigma^+, \widetilde{y}, \sigma^-]$.

5 Concluding remarks

We have presented three types of simplification rules (merging rules, splitting rules and operators simplification rules) that allows to compute the MLP of any given RPR with local operations. Starting from the interpixel path of some discrete region, one may use these rules in order to compute its MLP. On the other hand, the overall computation time would be significantly lower by using the algorithms presented in [14, 15] since their approach are more straightforward than the iterative one obtained from our local operations. Nevertheless, given some discrete contour C and its MLP, if a local perturbation is performed on C, for instance change a factor 01 by 10 in the contour word, a RPR which is not the MLP can be deduced directly from the previous one. Using the techniques presented in this paper, a dynamic computation of the new MLP from this RPR is possible in time sublinear with respect to the length of the section over which both MLPs differ. We plan to use this algorithm in digital deformable partition models where the length of region boundaries have to be computed after each modification [5, 4].

References

1. Jean Berstel, Aaron Lauve, Christophe Reutenauer, and Franco V. Saliola. *Combinatorics on words*, volume 27 of *CRM Monograph Series*. American Mathematical Society, Providence, RI, 2009. Christoffel words and repetitions in words.
2. J.-P. Borel and F. Laubie. Quelques mots sur la droite projective réelle. *J. Théor. Nombres Bordeaux*, 5(1):23–51, 1993.

3. D. Coeurjolly and R. Klette. A comparative evaluation of length estimators of digital curves. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 26(2):252–258, 2004.
4. G. Damiand, A. Dupas, and J.-O. Lachaud. Combining topological maps, multi-label simple points and minimum length polygon for efficient digital partition model. Submitted to IWCIA 2011.
5. F. de Vieilleville and J.-O. Lachaud. Digital deformable model simulating active contours. In *Proc. DGCI*, volume 5810 of *LNCS*, pages 203–216. 2009.
6. F. Feschet. Fast guaranteed polygonal approximations of closed digital curves. In *Proc. SCIA*, volume 3540 of *LNCS*, pages 910–919. 2005.
7. J. D. Hobby. Polygonal approximations that minimize the number of inflections. In *Proc. SODA*, pages 93–102, 1993.
8. R. Klette, V. Kovalevsky, and B. Yip. On length estimation of digital curves. In *Vision geometry VIII*, volume 3811, pages 117–128, 1999.
9. R. Klette and A. Rosenfeld. *Digital Geometry - Geometric Methods for Digital Picture Analysis.* Morgan Kaufmann, San Francisco, 2004.
10. R. Klette and B. Yip. The length of digital curves. *Machine Graphics Vision*, 9(3):673–703, 2000.
11. M. Lindenbaum and A. Brückstein. On recursive, o(n) partitioning of a digitized curve into digital straight segments. *IEEE Trans. On Pattern Analysis and Machine Intelligence*, 15(9):949–953, 1993.
12. M. Lothaire. *Algebraic combinatorics on words*, volume 90 of *Encyclopedia of Mathematics and its Applications.* Cambridge University Press, Cambridge, 2002.
13. U. Montanari. A note on minimal length polygonal approximation to a digitized contour. *Communications of the ACM*, 13(1):41–47, 1970.
14. X. Provençal and J.-O. Lachaud. Two linear-time algorithms for computing the minimum length polygon of a digital contour. In *Proc. DGCI*, volume 5810 of *LNCS*, pages 104–117, 2009.
15. T. Roussillon, L. Tougne, and I. Sivignon. What does digital straightness tell about digital convexity? In *Proc. IWCIA*, volume 5852 of *LNCS*, pages 43–55. 2009.
16. M. Said, J.-O. Lachaud, and F. Feschet. Multiscale discrete geometry. In *Proc. DGCI*, volume 5810 of *LNCS*, pages 118–131. 2009.
17. J. Sklansky, R. L. Chazin, and B. J. Hansen. Minimum perimeter polygons of digitized silhouettes. *IEEE Trans. Computers*, 21(3):260–268, 1972.
18. F. Sloboda and J. Stoer. On piecewise linear approximation of planar Jordan curves. *J. Comput. Appl. Math.*, 55(3):369–383, 1994.
19. F. Sloboda and B. Zaťko. On one-dimensional grid continua in Rˆ2. Technical report, Institute of Control Theory and Robotics, Slovak Academy of Sciences, Bratislava, Slovakia, 1996.
20. F. Sloboda and B. Zaťko. On approximation of Jordan surfaces in 3D. In G. Bertrand et al., editor, *Digital and Image Geometry*, volume 2243 of *LNCS*, pages 365–386. Springer, 2001.
21. F. Sloboda, B. Zaťko, and J. Stoer. On approximation of planar one-dimensional continua. In R. Klette, A. Rosenfeld, and F. Sloboda, editors, *Advances in Digital and Computational Geometry*, pages 113–160, 1998.
22. A. W. M. Smeulders and L. Dorst. Decomposition of discrete curves into piecewise straight segments in linear time. In R. A. Melter, A. Rosenfeld, and P. Bhattacharya, editors, *Vision Geometry*, volume 119 of *Contempory Mathematics*, pages 169–195. Am. Math. Soc., 1991.
23. K. Voss. *Discrete images, objects, and functions in* $\mathbf{Z}^n$, volume 11 of *Algorithms and Combinatorics.* Springer-Verlag, Berlin, 1993.