Multiscale Discrete Geometry*

Mouhammad Said[1,2], Jacques-Olivier Lachaud[1], and Fabien Feschet[2]

[1] Laboratoire de Mathématiques, UMR 5127 CNRS, Université de Savoie,
73376 Le Bourget du Lac, France
mouhammadsaiid@hotmail.com, jacques-olivier.lachaud@univ-savoie.fr
[2] LAIC, Univ. Clermont-Ferrand,
IUT, Campus des Cézeaux, 63172 Aubière Cedex, France
feschet@laic.u-clermont1.fr

Abstract. This paper presents a first step in analyzing how digital shapes behave with respect to multiresolution. We first present an analysis of the covering of a standard digital straight line by a multi-resolution grid. We then study the multi-resolution of Digital Straight Segments (DSS): we provide a sublinear algorithm computing the exact characteristics of a DSS whenever it is a subset of a known standard line. We finally deduce an algorithm for computing a multiscale representation of a digital shape, based only on a DSS decomposition of its boundary.

Keywords: multiscale geometry, digital contours, standard lines, digital straight segment recognition, Stern-Brocot tree, multi-resolution.

1 Introduction

Multiscale analysis is a classical tool in image processing [15,11]. It has been introduced for addressing the fact that derivatives or geometric features are sensitive to the notion of scale. They do not intrinsically exist but they are often present only at a finite number of scales. To compute this representation, the original signal is embedded in a parameterized family of derived signals, obtained by convolutions of a parameterized family of Gaussian kernels, whose variance is the scale. At any scale, smooth versions of the derivatives are computed. Combined together, they provide a multiscale representation of the feature geometry, thus providing information about its significance for later interpretation or post-processing. Originally defined in the continuous space, these approaches have been extended to the discrete case [13] to better handle both the finite support and the finite resolution of digital images.

The use of Gaussian convolutions does not process well binary data such as digital curves. So, another approach must be followed. We can consider the approach of Vacavant *et al.* [14] as a multiscale approach in the sense that geometric objects are represented by a multi-resolution set of rectilinear tiles. However, the set of tiles is not intrinsic and is computed from the digital objects as the result of an optimization process. Indeed, they determine the minimal

* Part of this work was funded by ANR project GeoDIB (ANR-06-BLAN-0225-03).

S. Brlek, C. Reutenauer, and X. Provençal (Eds.): DGCI 2009, LNCS 5810, pp. 118–131, 2009.

number of rectangles whose union covers the considered object. In this way, they define geometric primitives such as lines and use them to analyze thick digital objects. Its potential is yet unclear for multiscale analysis of digital object features, and the constructed objects are not analytically defined.

We believe that, in the context of digital geometry, geometric primitives such as lines, circles or polynomials are of a great importance. Pieces of digital lines are excellent tangent estimators [7,12], circular arcs estimate curvature [1]. It is thus fundamental to keep them in the multiscale analysis of digital boundaries. Our point of view is therefore similar to the one of Figueiredo [8] who studied the behavior of 8-connected lines when changing the resolution of the grid. He was the first to characterize when the image of such a digital line in various resolution grids is another digital line.

In the present paper, we pursue on this idea and we provide in Section 2 an analytic description of how 4-connected digital straight lines, called *digital standard lines* (DSL), behave when the resolution of the grid is changed by an arbitrary factor. We prove first that their subsampling is also a standard line, whose parameters can all be analytically defined (Theorems 1 and 2).

The multiresolution of a digital straight segment (DSS) is addressed in Section 3. However, a full analytic approach seems out of reach at the moment. This is why we take an algorithmic approach. We present a new algorithm which computes the exact characteristics of a DSS that is a subset of a known DSL, given the endpoints of the DSS. Not surprisingly, it is based on continued fractions. Its worst-case computational complexity is $\Theta(\sum_{i=0}^{k} u_i)$, where $[u_0; u_1, \ldots, u_k]$ is the continued fraction of the slope of the input DSL. The expectation of this sum for fractions $\frac{a}{b}$ with $a + b = n$ is experimentally lower than $\log^2 n$, and this sum is upper bounded by n. This is to compare with classical DSS recognition algorithms (e.g., see [4]), whose complexities are at best $\Theta(n)$.

Section 4 applies the preceding results to the multiscale computation of a digital shape. We show there that if the digital shape was previously polygonalized as a sequence of M DSS, its exact multiresolution is computed in time linear with M times the average of the sums of the partial cofficients of its output DSS slopes. In most cases, this is clearly sublinear, and at worst, linear in the size of the contour. This paper is therefore a first step towards the multiscale computation of the tangential cover [6,12], a fundamental tool for representing and analyzing digital curves.

2 Covering of a Standard Line by a Lower Resolution Grid

A first step for defining multiscale discrete geometry is to find the covering of a digital straight line by a lower resolution grid. We shall prove here that this covering is indeed a digital straight line with computable characteristics (Theorem 1) and that this line is standard (Theorem 2). The proof is essentially technical and is based on the same principle as the proof of Figueiredo [8].

Let $D(a, b, \mu)$ be a standard digital line such that $0 < a < b$ and $\gcd(a, b)$=1. Let us consider the subgroup $S(h, v)$= (Xh, Yv) of $\mathbb{Z}^2$ (where X, Y, h, v all are integers). Obviously the fondamental domain $[0, h) \times [0, v)$ of $S(h, v)$ and its translations by the vectors $X(h, 0) + Y(0, v)$ induce a tiling of $\mathbb{Z}^2$ where each tile contains exactly one point of $S(h, v)$ (for which reason we will refer indifferently to the tile itself or to the point of $S(h, v)$ it contains). We are interested in the set of tiles of $S(h, v)$ that intersect $D(a, b, \mu)$. The set of tiles is a discrete line in the subgroup $S(h, v)$. We denote this line by Δ and call it the *covering line* of $D(a, b, \mu)$ in $S(h, v)$.

The tiling generated by $S(h, v)$ on $\mathbb{Z}^2$ induces a new coordinate system where coordinates (X, Y) are related to the canonical coordinates of $\mathbb{Z}^2$ by the following obvious relations where $\left[\frac{x}{h}\right]$ is the quotient of the euclidean division of x by h and $\left\{\frac{x}{h}\right\}$ is the remainder of this division:

$$X = \left[\frac{x}{h}\right] \qquad Y = \left[\frac{y}{v}\right] \qquad x = hX + \left\{\frac{x}{h}\right\} \qquad y = vY + \left\{\frac{y}{v}\right\}.$$

By definition, $D(a, b, \mu)$ can be written as

$$\mu \leq ax + by < \mu + a + b \tag{1}$$

Hence the equation of Δ in the coordinate system related to S writes:

$$\mu \leq a(hX + \left\{\frac{x}{h}\right\}) + b(vY + \left\{\frac{y}{v}\right\}) < \mu + a + b \tag{2}$$

In order to simplify (2) let us introduce

$$m_x = \left\{\frac{x}{h}\right\} \qquad m_y = \left\{\frac{y}{v}\right\} \tag{3}$$

Since m_x and m_y vary when x steps through $\mathbb{Z}$, equation (2) becomes:

$$\mu - \max_{x \in \mathbb{Z}}(am_x + bm_y) \leq ahX + bvY < \mu + a + b - \min_{x \in \mathbb{Z}}(am_x + bm_y) \tag{4}$$

Now to fully determine this equation that defines the covering of the digital line by the tiling, the exact range of $am_x + bm_y$, i.e, the values of $\min_{x\in\mathbb{Z}}(am_x + bm_y)$ and $\max_{x\in\mathbb{Z}}(am_x + bm_y)$, need to be calculated.

By definition of m_x and m_y, we have $0 \leq m_x \leq h - 1$ and $0 \leq m_y \leq v - 1$. Using these bounds in (4) suggests for Δ an equation of the form:

$$\mu - a(h - 1) - b(v - 1) \leq ahX + bvY < \mu + a + b. \tag{5}$$

But since $ahX + bvY$ can only assume values that are multiples of $g = \gcd(ah, bv)$, the bounds of equation (4) can be refined. We further denote $\frac{ah+bv}{g}$ by p.

However since m_x and m_y are linked through (2) and (3), the precise bounds of $am_x + bm_y$ when x steps through $\mathbb{Z}$, need to be determined. In fact, we show in what follows, that even though $am_x + bm_y$ does not reach the absolute bounds 0 and $a(h - 1) + b(v - 1)$ in some cases, equation (20) (given on page 123) always holds.

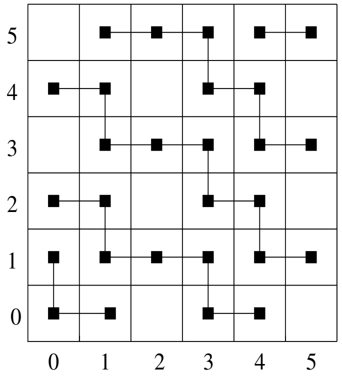

Fig. 1. Determination of the range of $\mathbf{2}m_x\mathbf{+3}m_y$. The black squares represent the family of standard digital lines defined by $5 - 6t \leq 2m_x + 3m_y < 10 - 6t$ restricted to $[0, 6) \times [0, 6)$ and hence the possible values of (m_x, m_y).

Inverting (3) yields $x = kh + m_x, k \in \mathbb{Z}$ and $y = lv + m_y, l \in \mathbb{Z}$. Hence, after insertion into (1):

$$\mu - akh - blv \leq am_x + bm_y < \mu + a + b - akh - blv \tag{6}$$

The term $akh + blv$ has values which are multiples of g. Equation (6) can thus be rewritten as

$$\mu - tg \leq am_x + bm_y < \mu + a + b - tg, t \in \mathbb{Z} \tag{7}$$

Equation (7) describes a family of digital lines of direction (a, b) parameterized by $t \in \mathbb{Z}$, which we denote by D_t. The intersection of D_t with the domain of (m_x, m_y), $[0, h) \times [0, v)$ defines the possible pairs (m_x, m_y) and therefore the range of possible values for the sum $am_x + bm_y$ (see Figure 1).

Let us denote with (n_x, n_y) the pair of $D_t \cap [0, h) \times [0, v)$ that yields the maximum value for the sum $am_x + bm_y$. Let us also denote by d the difference between $am_x + bm_y$ evaluated at $(h - 1, v - 1)$ and at (n_x, n_y):

$$d = a(h - 1) + b(v - 1) - (an_x + bn_y) \tag{8}$$

Determining the maximum value of t verifying (7) for the point (n_x, n_y) provides a more precise expression of the lower bound of d. (8) becomes

$$d = (p + t)g - (\mu + 2a + 2b - 1).$$

To find the maximum value, we take $d = 0$ which implies that $(n_x, n_y) = (h - 1, v - 1)$, and

$$t = -p + Q_2 + \frac{R_2}{g}, \tag{9}$$

where $Q_2 = \left[\frac{\mu+2a+2b-1}{g}\right]$ and $R_2 = \left\{\frac{\mu+2a+2b-1}{g}\right\}$. Now, we must distinguish two possible cases for R_2.

1. If g divides $\mu + 2a + 2b - 1$. In this case, (9) becomes $t = -p + Q_2$ and we insert this equality in (4).

$$ahX + bvY \geq \mu - \max_{x \in \mathbb{Z}}(am_x + bm_y) \geq (-p + Q_2)\, g - a - b + 1$$

Let $\alpha = \frac{ah}{g}$ and $\beta = \frac{bv}{g}$. Then the previous equation can be rewritten as:

$$\alpha X + \beta Y \geq -p + Q_2 - Q_1 \tag{10}$$

where $Q_1 = \left[\frac{a+b-1}{g}\right]$ and $R_1 = \left\{\frac{a+b-1}{g}\right\}$.

2. If g does not divide $\mu + 2a + 2b - 1$. In this case, we insert (9) directly in (4).

$$\alpha X + \beta Y \geq -p + Q_2 - Q_1 + \frac{R_2}{g} - \frac{R_1}{g}$$

We calculate the difference between the division of R_2 by g and the division of R_1 by g, by comparing R_1 and R_2:

$$\text{If } R_2 \leq R_1, \text{ then } \alpha X + \beta Y \geq -p + Q_2 - Q_1 \tag{11}$$

$$\text{If } R_2 > R_1, \text{ then } \alpha X + \beta Y \geq -p + Q_2 - Q_1 + 1 \tag{12}$$

A similar reasoning can be applied with the minimum value of $am_x + bm_y$ throughout $D_t \cap [0, h) \times [0, v)$ obtained at point (n'_x, n'_y). Let us also denote by d' the difference between $am_x + bm_y$ evaluated at 0 and at (n'_x, n'_y):

$$d' = 0 - (an'_x + bn'_y). \tag{13}$$

Determining the maximum value of t verifying (7) for the point (n'_x, n'_y) provides a more precise expression of the upper bound of d'. (13) becomes

$$d' = 0 - (\mu - tg) = (\mu + 2a + 2b - 1) + tg - (2\mu + 2a + 2b - 1)$$

We take $d' = 0$ to find the minimum value, implying $(n'_x, n'_y) = (0, 0)$, then:

$$t = -Q_2 - \frac{R_2}{g} + \left(\frac{2\mu + 2a + 2b - 1}{g}\right) \tag{14}$$

At this point, we must distinguish two possible cases according to R_2.

1. If g divides $\mu + 2a + 2b - 1$. In this case, (14) becomes $t = -Q_2 + \left(\frac{2\mu+2a+2b-1}{g}\right)$ and we insert this equality in (4):

$$ahX + bvY < \mu + a + b - (\mu - tg) < -Q_2 g + 2\mu + 3a + 3b - 1$$

and,

$$\alpha X + \beta Y < -Q_2 + Q_3 + \frac{R_3}{g}, \tag{15}$$

where $Q_3 = \left[\frac{2\mu+3a+3b-1}{g}\right]$ and $R_3 = \left\{\frac{2\mu+3a+3b-1}{g}\right\}$.
We must further distinguish two possible cases for R_3.

(a) If g divides $2\mu + 3a + 3b - 1$. In this case Equation (15) becomes

$$\alpha X + \beta Y < -Q_2 + Q_3. \tag{16}$$

(b) If g does not divide $2\mu + 3a + 3b - 1$. In this case Equation (15) becomes

$$\alpha X + \beta Y < -Q_2 + Q_3 + 1. \tag{17}$$

2. If g does not divide $\mu + 2a + 2b - 1$. In this case we insert (14) directly in (4):

$$ahX + bvY < \mu + a + b - (\mu - tg) < -Q_2 g - Q_1 + 2\mu + 3a + 3b - 1.$$

Then the previous equation can be rewritten as:

$$\alpha X + \beta Y < Q_3 - Q_2 + \frac{R_3}{g} - \frac{R_2}{g}.$$

We calculate the difference between the division of R_3 by g and the division of R_2 by g, by comparing R_2 and R_3:

$$\text{If } R_3 \leq R_2, \text{ then } \alpha X + \beta Y < -Q_2 + Q_3 \tag{18}$$

$$\text{If } R_3 > R_2, \text{ then } \alpha X + \beta Y < -Q_2 + Q_3 + 1 \tag{19}$$

More precisely, eleven equations involving $\alpha X + \beta Y$ can be obtained by combining the lower bound (equations (10), (11), (12)) and the upper bound (equations (16), (17), (18), (19)) of $\alpha X + \beta Y$. Among the eleven possible cases, four cases are impossible. For the remaining cases we have also conditions which yield the same result. Therefore those equations can be formulated as a single expression given below.

Theorem 1. *The digital straight line Δ of $S(h, v)$ covering the standard digital line $D(a, b, \mu)$ of $\mathbb{Z}^2$ is defined by:*

$$-p + Q_2 - Q_1 + SI \leq \alpha X + \beta Y < Q_3 - Q_2 + SS \tag{20}$$

Where $\alpha = \frac{ah}{g}$, $\beta = \frac{bv}{g}$, $g = \gcd(ah, bv)$, $p = \alpha + \beta$ *and*

$$SI = \begin{cases} 0 \text{ if } & R_2 \leq R_1 \\ 1 & \textit{otherwise} \end{cases} \qquad SS = \begin{cases} 0 \text{ if } & R_3 \leq R_2 \\ 1 & \textit{otherwise} \end{cases}$$

Figure 2 illustrates the covering of $D(7, 9, 6)$ in the subsampling $(6, 6)$.

Theorem 2. *The covering line Δ is standard.*

In order to prove that the difference between the upper and lower bounds of (20) is equal to $\alpha + \beta$, we first need the following proposition and lemma.

Proposition 1.

$$\frac{R_2 - R_1 - R}{g} = \begin{cases} 0 \text{ if } & R_1 \leq R_2 \\ -1 & \textit{otherwise.} \end{cases} \qquad \textit{and} \qquad \frac{R_3 - R_2 - R}{g} = \begin{cases} 0 \text{ if } & R_2 \leq R_3 \\ -1 & \textit{otherwise.} \end{cases}$$

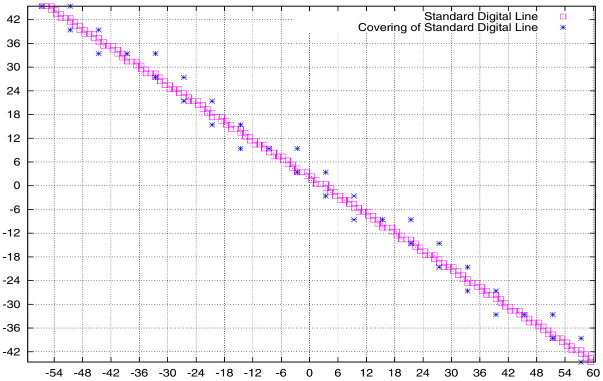

Fig. 2. The covering line of $D(7, 9, 6)$ by $S(6, 6) : -12 \leq 7X + 9Y < 4$

Proof. Since $0 \leq R_1 < g$, $0 \leq R_2 < g$ and $0 \leq R < g$, then $0 \leq \frac{R_1}{g} < 1$, $0 \leq \frac{R_2}{g} < 1$ and $0 \leq \frac{R}{g} < 1$. Thus we can write $-2 < \frac{R_2 - R_1 - R}{g} < 1$. To specify the exact integer value of $\frac{R_2 - R_1 - R}{g}$, we must compare R_1 and R_2.

- If $R_1 \leq R_2$, then $R_2 = (\mu + 2a + 2b - 1) \bmod g = (\mu + a + b) \bmod g + (a + b - 1) \bmod g = R + R_1$. Therefore $R_2 - R_1 - R = 0$ and $\frac{R_2 - R_1 - R}{g} = 0$.
- If $R_1 > R_2$, then $R_2 = (\mu + 2a + 2b - 1) \bmod g = (\mu + a + b) \bmod g + (a + b - 1) \bmod g - g = R + R_1 - g$. Therefore $R_2 - R_1 - R = -g$ and $\frac{R_2 - R_1 - R}{g} = -1$.

The proof of the second condition is analogous to the first one. □

Lemma 1.

$$Q_2 - Q_1 = Q + \begin{cases} 0 \text{ if } & R_1 \leq R_2 \\ 1 & \textit{otherwise.} \end{cases} \quad \textit{and} \quad Q_3 - Q_2 = Q + \begin{cases} 0 \text{ if } & R_2 \leq R_3 \\ 1 & \textit{otherwise.} \end{cases}$$

Proof. We only provide the proof of the first result and the second result has a similar proof, $\mu + a + b = (\mu + 2a + 2b - 1) - (a + b - 1)$, then by definition of the Euclidean division, we can write $\mu + a + b = gQ + R$, $a + b - 1 = gQ_1 + R_1$, and $\mu + 2a + 2b - 1 = gQ_2 + R_2$. So, the first equality can be rewritten as $gQ + R = (gQ_2 + R_2) - (gQ_1 + R_1)$. Then,

$$g(Q - Q_2 + Q_1) = R_2 - R_1 - R$$

To reduce this equality, we must compare R_1 and R_2

1. If $R_1 \leq R_2$, then by Proposition 1, $\frac{R_2 - R_1 - R}{g} = 0$ which implies that $Q = Q_2 - Q_1$,
2. If $R_1 > R_2$, then by Proposition 1, $\frac{R_2 - R_1 - R}{g} = -1$ which implies that $Q = Q_2 - Q_1 - 1$. □

Proof (Theorem 2). We want to prove that the difference between the upper and lower bounds of (20) is equal to $\alpha + \beta$. There are 5 cases to consider. From Proposition 1 and Lemma 1, we get:

- If $R_3 < R_2 < R_1$, then $SI = 0$, $SS = 0$, $Q_2 - Q_1 = Q + 1$ and $Q_3 - Q_2 = Q + 1$.
- If $R_2 < R_1$ and $R_2 < R_3$, then $SI = 0$, $SS = 1$, $Q_2 - Q_1 = Q + 1$ and $Q_3 - Q_2 = Q$.
- If $R_1 < R_2 < R_3$, then $SI = 1$, $SS = 1$, $Q_2 - Q_1 = Q$ and $Q_3 - Q_2 = Q$.
- If $R_1 < R_2$ and $R_3 < R_2$, then $SI = 1$, $SS = 0$, $Q_2 - Q_1 = Q$ and $Q_3 - Q_2 = Q + 1$.
- If $R_1 = R_2 = R_3$, then $SI = 0$, $SS = 0$, $Q_2 - Q_1 = Q$ and $Q_3 - Q_2 = Q$.

It is easy to see for all cases that, $(Q_3 - Q_2 + SS) - (-p + Q_2 - Q_1 + SI) = p$.
□

3 A Fast DSS Recognition Algorithm

We focus now our attention on the multiresolution of digital straight segments (DSS), which are finite connected parts of digital lines. In the first subsection, we recall first some links between DSS, patterns, continued fraction of the slope, and Stern-Brocot tree representation of fractions. Let S be some DSS included in a line D. We then obtain easily Proposition 3, which states that the slope of the (h, v)-multiresolution S' of S is some reduced fraction of the slope of the (h, v)-multiresolution D' of D. However, the full analytic description of S' seems out of reach at the moment. This is why we propose an algorithmic approach in the second subsection. All the characteristics of S' can be determined in time sublinear in the number of points of S' (Algorithm 2 and Proposition 4).

3.1 DSS, Patterns, Irreducible Fractions and Continued Fractions

Given a standard line (a, b, μ), we call *pattern* of characteristics (a, b) the succession of Freeman moves between any two consecutive upper leaning points. The Freeman moves defined between any two consecutive lower leaning points is the previous word read from back to front and is called the *reversed pattern.* As noted by several authors (e.g. see [10], or the work of Berstel reported in [2]), the pattern of any slope can be constructed from the continued fraction of the slope. We recall that a *simple continued fraction* is an expression:

$$z = \frac{a}{b} = [u_0, u_1, u_2, ..., u_i, ..., u_n] = u_0 + \frac{1}{u_1 + \frac{1}{... + \frac{1}{u_{n-1} + \frac{1}{u_n}}}},$$

where n is the *depth* of the fraction, and u_0, u_1, etc, are all integers and called the *partial quotients.* We call *k-th convergent* the simple continued fraction formed of the k first partial quotients: $z_k = \frac{p_k}{q_k} = [u_0, u_1, u_2, ..., u_k]$.

The role of partial quotients can be visualized with a structure called *Stern-Brocot tree* (see [9] for a complete definition) which is a hierarchy containing all the positive irreductible rational fractions. The idea under its construction is to begin with the two fractions $\frac{0}{1}$ and $\frac{1}{0}$ and to repeat the insertion of the median of these two fractions as follows: insert the median $\frac{m+m'}{n+n'}$ between $\frac{m}{n}$ and $\frac{m'}{n'}$. The sequence of partial quotients defines the sequence of right and left moves down the tree. Many works deal with the relations between irreducible rational fractions and digital lines (see [5] for characterization with Farey series, and [16] for a link with decomposition into continuous fractions). In [3], Debled and Réveillès first introduced the link between this tree and the recognition of digital line. Recognizing a piece of digital line is like going down the Stern-Brocot tree up to the directional vector of the line. To sum up, the classical online DSS recognition algorithm **DR95** [3] (reported in [10]) updates the DSS slope when adding a point that is just exterior to the current line (weak exterior points). The slope evolution is analytically given by next property.

Proposition 2. *[2] The slope evolution in **DR95** depends on the parity of the depth of its slope, the type of weakly exterior point added to the right. This is summed up in the table below, where the slope is $[0, u_1, ..., u_k]$, $k = 2i$ even or $k = 2i+1$ odd, δ pattern(s) and δ' reversed pattern(s):*

	Even k	*Odd k*
Upper weakly exterior	$[0, u_1, ..., u_{2i}, \delta]$	$[0, u_1, ..., u_{2i+1} - 1, 1, \delta]$
Lower weakly exterior	$[0, u_1, ..., u_{2i} - 1, 1, \delta']$	$[0, u_1, ..., u_{2i+1}, \delta']$

A slope evolution during recognition is then clearly equivalent to descending in the Stern-Brocot tree, to the left or to the right. Since S' is a piece of a DSL of slope $\frac{a'}{b'}$, we therefore obtain:

Proposition 3. *If $D'(a', b', \mu')$ is the covering of some digital line D by a tiling $S(h, v)$, then any segment $S \subset D$ induces a segment $S' \subset D'$ by the same tiling, such that the slope of S' is $\frac{a'}{b'}$ or one of its ancestors in the Stern-Brocot tree.*

3.2 Fast DSS Recognition When DSL Container Is Known

We provide a general algorithmic solution to the multiresolution of a DSS that is sublinear in the number of its points (Algorithm 2). We exploit the fact that in our case, we *know* that it is a piece of a standard digital line of known characteristics (with Theorems 1 and 2). Therefore most of the points tested in **DR95** are useless since they do not lead to any slope evolution. Proposition 3 tells us that we must find the slope of S' in the ancestors of the slope of D'. We must also determine the shift to origin of S'.

Starting from the initial correct quadrant, Algorithm 2 determines progressively the positions of the weakly exterior points. They are related to the Bézout coefficient of the current DSS slope $\frac{p_k}{q_k}$ (line 1). Since we update at each step the continued fraction of the slope, these coefficients are given in $O(1)$ time by the formula

$$\text{Bezout}(p, q, k) = \begin{cases} (q_k - q_{k-1}, p_k - p_{k-1}), \text{when } k \text{ is even} \\ (q_{k-1}, p_{k-1}), \text{when } k \text{ is odd} \end{cases} .$$

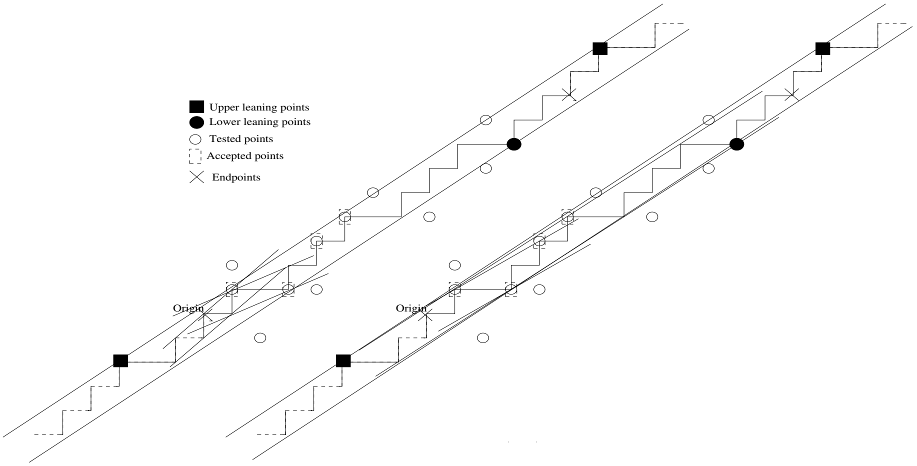

Fig. 3. A digital straight line $D(13, 17, -5)$ with an odd slope. Computes the characteristics (a, b, μ) of a DSS S that is the subset of D between the origin and the point $(12, 9)$. The intermediate slopes are drawn with solid lines on the left and on the right, tested points are circled.

```
Action UpdateSlope( In $uw$, $\delta$, InOut $k$, $u$, $p$, $q$ ) ;
$uw$ : boolean /* True iff upper weak leaning point */ ;
$\delta$ : integer /* number of (reversed) patterns */ ;
$k$ : integer /* depth of slope continued fraction */ ;
$u, p, q$ : array of integers /* slope cont. fraction */ ;
begin
    if ($uw = true$ and $k$ is odd) or ($uw = false$ and $k$ is even) then
        $u_k \leftarrow u_k - 1$, $p_k \leftarrow p_k - p_{k-1}$, $q_k \leftarrow q_k - q_{k-1}$ ;
        $u_{k+1} \leftarrow 1$, $p_{k+1} \leftarrow p_k + p_{k-1}$, $q_{k+1} \leftarrow q_k + q_{k-1}$ ;
        $k \leftarrow k + 1$;
    if $\delta = 1$ then
        $u_k \leftarrow u_k + 1$, $p_k \leftarrow p_k + p_{k-1}$, $q_k \leftarrow q_k + q_{k-1}$ ;
    else
        $u_{k+1} \leftarrow \delta$, $p_{k+1} \leftarrow \delta p_k + p_{k-1}$, $q_{k+1} \leftarrow \delta q_k + q_{k-1}$ ;
        $k \leftarrow k + 1$;
end
```

Algorithm 1. Updates in $O(1)$ the slope of a DSS according to the addition of an upper leaning point (uw is $true$) or lower leaning point (uw is $false$), to the number of patterns or reversed patterns δ, and to the current continued fraction of the slope

The algorithm then checks in sequence upper and lower weakly exterior points so as to find the first in the DSL (repeat block at line 2). The number of iteration gives the number of pattern repetitions. Once such a point is found, Proposition 2 indicates how to update the slope, depending if it is a slope increase (line 3) or

```
Action SmartDSS( In $D$, In $P, Q$, Out $S$ ) ;
$D$ : DSL $(\alpha, \beta, \mu')$, $P, Q$ : Point of $\mathbb{Z}^2 S$ : DSS $(a, b, \mu)$ ;
Var $u, p, q$ : array of integers /* Cont. fraction $\frac{a}{b} = [u_0, \ldots, u_k] = \frac{p_k}{q_k}$ */ ;
Var $U, L, U', L'$ : Point of $\mathbb{Z}^2$ ;
Var $ulu, lul, inside$ : boolean ;
Var $k, loop$ : integer ;
begin
    $k \leftarrow 0$, $u_0 \leftarrow 0$, $p_0 \leftarrow 0$, $q_0 \leftarrow 1$, $p_{-1} \leftarrow 1$, $q_{-1} \leftarrow 0$ ;
    $U \leftarrow P$, $L \leftarrow P$ ;
    $inside \leftarrow true$, $ulu \leftarrow true$, $lul \leftarrow true$ ;
    while $inside$ and $p_k \neq \alpha$ do
        $(a, b) \leftarrow (p_k, q_k)$ ;
1       $(b', a') \leftarrow$ Bézout$(p, q, k)$ /* $ab' - ba' = 1$ */ ;
        $U' \leftarrow U + (b - b', a - a')$;
        $L' \leftarrow L + (b', a')$;
        $\delta \leftarrow 1$, $loop \leftarrow 0$;
2       repeat
            $U' \leftarrow U' + (b, a)$, $L' \leftarrow L' + (b, a)$;
            if $U'_y \leq Q_y$ and $U' \in D$ then $loop \leftarrow 1$ ;
            else if $L'_x \leq Q_x$ and $L' \in D$ then $loop \leftarrow 2$ ;
            else $\delta \leftarrow \delta + 1$ ;
        until $loop \neq 0$ or $U'_y \geq Q_y$ or $L'_x \geq Q_x$ ;
3       if $loop = 1$ then
            /* Increase slope with weak upper leaning point $U'$ */ ;
            UpdateSlope($true, \delta, k, u, p, q$ ) ;
4           $L \leftarrow L' - (b', a')$ ;
            if not $lul$ then $L \leftarrow L - (b, a)$;
            $ulu \leftarrow true$, $lul \leftarrow false$ ;
5       if $loop = 2$ then
            /* Decrease slope with weak lower leaning point $L'$ */ ;
            UpdateSlope($false, \delta, k, u, p, q$) ;
6           $U \leftarrow U' - (b - b', a - a')$ ;
            if not $ulu$ then $U \leftarrow U - (b, a)$;
            $ulu \leftarrow false$, $lul \leftarrow true$ ;
        else
            $inside \leftarrow false$;
    $a \leftarrow p_k$, $b \leftarrow q_k$, $\mu \leftarrow aU_x - bU_y$ ;
end
```

Algorithm 2. Computes the characteristics (a, b, μ) of a DSS that is some subset of a DSL D, given a starting point P and an ending point Q $(P, Q \in D)$. The computation time is linear with the sum $\sum_{i=0}^{k} u_k$ with $\frac{a}{b} = [u_0, \ldots, u_k]$.

decrease (line 5). This is effectively implemented in $O(1)$ time with Algorithm 1. It remains to update consistently the new first lower leaning point (line 4) or upper leaning point (line 6). We have to be a little careful whether the current DSS is a pattern (ULU) or a reversed pattern (LUL): it is just to adapt δ to be the number of patterns or the number of reversed patterns. The algorithm stops either when point Q has been reached or when DSL slope has been reached.

An execution of this algorithm is illustrated on Figure 3. 11 points have been tested compared with a DSS length of 23. We show below that the worst-case time complexity of Algorithm 2 is related to the partial quotients of the DSS slope and not to the slope itself.

Proposition 4. *Let S be a DSS of slope $\frac{a}{b} = [u_0, u_1, ..., u_k]$, $T(n)$ the number of points on S tested by Algorithm 2 to recognize $\frac{a}{b}$ with $\sum_{i=0}^{k} u_i = n$, then $T(n) \leq 2n$ (it only depends on the sum of u_i).*

Proof. We prove by induction on n that $T(n) \leq 2n$. The initial state $T(0) = 0$ is obvious since P and the quadrant is known. Assume that $T(n) \leq 2n$ and we shall prove that $T(n+1) \leq 2n+2$.

As $T(n) = 2n$, then $\frac{a}{b}$ is some $[u_0, u_1, ..., u_k]$. According to Stern-Brocot tree and Proposition 2, there are only two possible evolutions for the slope, either $[u_0, u_1, ..., u_k, 1]$ or $[u_0, u_1, ..., u_k - 1, 1, 1]$. The computation of the next slope on $n+1$ depends of the parity of the slope depth and the type of weakly exterior point added.

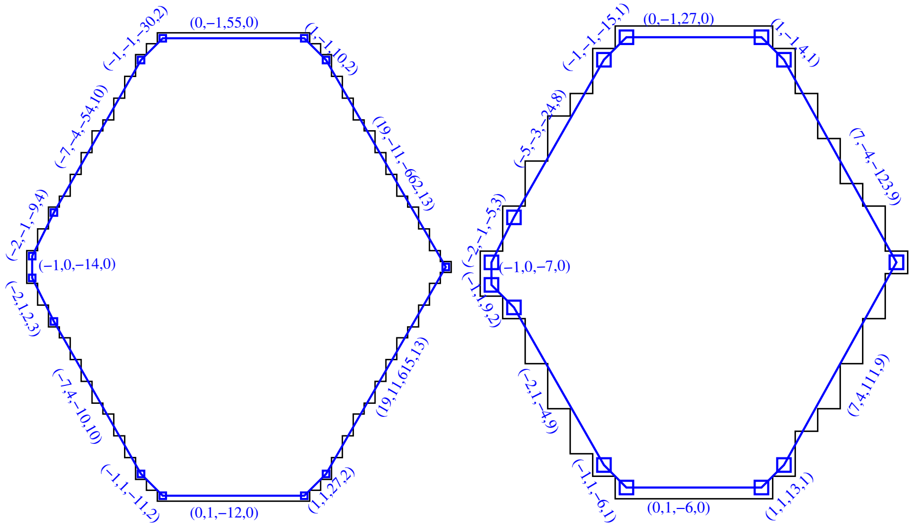

Fig. 4. (a) Multiscale computation of the boundary of a digital shape according to several tiling (h, v). (b) $(h, v) = (1, 1)$ and original DSL is $(4, 7, -6)$. (c) $(h, v) = (2, 2)$, DSL is $(4, 7, -8)$ and extracted DSS is $(4, 7, -8)$. (d) $(h, v) = (3, 4)$, DSL is $(16, 21, -32)$ and extracted DSS is $(3, 4, -6)$.

Assume first that the slope has even depth $k = 2i$. If this point is upper weakly exterior, then the new slope increases to $[u_0, u_1, ..., u_{2i}, 1]$ and $T(n+1) = T(n)+1$. In the other case, we test the lower weakly exterior point, then the slope decreases to $[u_0, u_1, ..., u_{2i} - 1, 1, 1]$ and $T(n+1) = T(n) + 2$. Therefore we conclude that $T(n+1) \leq T(n) + 2 \leq 2n + 2$.

The reasonning is similar in the other case. We note that the loop at line 2 agglomerates a sequence of δ patterns, instead of doing it one by one as in this proof. □

4 Multiscale Covering of a Digital Contour

The preceding results allow us to compute an exact multiscale representation of a digital contour C with $\#C$ points, if this contour has been beforehand decomposed into a sequence of M digital straight segments. The process is illustrated on Figure 4. The 4-connected inner pixel contour of a digital shape is first extracted and decomposed greedily into digital straight segments. Taking for instance DSL $(19, -11, -662)$ associated to a DSS of length 32, we compute two (h, v) coverings of this segment. The $(1, 1)$-covering requires 13 tests and the $(2, 2)$ covering requires 9 tests. A multiresolution algorithm tracing the original DSS would have made about 32 tests, whichever the chosen covering.

Our technique allows us to compute the exact multiscale representation of a digital contour in a time proportional to $M \times \overline{U}$, where $\overline{U}$ is the average of the partial quotient sum of the *output* subsampled DSS. By definition, $\overline{U}$ cannot exceed $O(\#C/(hv))$, and is in most case some $O(\log \#C)$. In worst case, our algorithm is linear with $\#C$.

5 Conclusion

In this paper, we have presented new results about the covering of discrete objects by regular tilings. Our approach is based on the algebraic equation of a standard digital line and provides analytic formulas. We have further presented a novel fast DSS recognition algorithm, in the case when the digital line containing it is known. Sublinear recognition is possible for most slopes. Finally, these properties have been used to compute the exact multiscale covering of a digital contour. Our algorithm is sensitive to the size of the output subsampled contour, and generally sublinear.

References

1. Coeurjolly, D., Grard, Y., Reveills, J.-P., Tougne, L.: An elementary algorithm for digital arc segmentation. Discrete Applied Mathematics 139, 31–50 (2004)
2. de Vieilleville, F., Lachaud, J.-O.: Revisiting digital straight segment recognition. In: Kuba, A., Nyúl, L.G., Palágyi, K. (eds.) DGCI 2006. LNCS, vol. 4245, pp. 355–366. Springer, Heidelberg (2006)
3. Debled-Rennesson, I.: Etude et reconnaissance des droites et plans discrets. PhD thesis, Université Louis Pasteur, Strasbourg (1995)
4. Debled-Rennesson, I., Reveillès, J.-P.: A linear algorithm for segmentation of discrete curves. International Journal of Pattern Recognition and Artificial Intelligence 9, 635–662 (1995)
5. Dorst, L., Smeulders, A.W.M.: Discrete representation of straight lines. IEEE transactions Pattern Analysis Machine Intelligence 6, 450–463 (1984)

6. Feschet, F.: Canonical representations of discrete curves. Pattern Anal. Appl. 8(1-2), 84–94 (2005)
7. Feschet, F., Tougne, L.: Optimal time computation of the tangent of a discrete curve: Application to the curvature. In: Bertrand, G., Couprie, M., Perroton, L. (eds.) DGCI 1999. LNCS, vol. 1568, pp. 31–40. Springer, Heidelberg (1999)
8. Figueiredo, O.: Advances in discrete geometry applied to the extraction of planes and surfaces from 3D volumes. PhD thesis, EPFL, Lausanne (1994)
9. Hardy, G.H., Wright, E.M.: An introduction to the Theory of Numbers. Oxford Society (1989)
10. Klette, R., Rosenfeld, A.: Digital Geometry - Geometric Methods for Digital Picture Analysis. Morgan Kaufmann, San Francisco (2004)
11. Koenderink, J.J.: The structure of images. Biol. Cyb. 50, 363–370 (1984)
12. Lachaud, J.-O., Vialard, A., de Vieilleville, F.: Fast, accurate and convergent tangent estimation on digital contours. Image Vision Comput. 25(10), 1572–1587 (2007)
13. Lindeberg, T.: Discrete derivative approximations with scale-space properties: A basis for low-level features extraction. Journal of Mathematical Imaging and Vision 3(4), 349–376 (1993)
14. Vacavant, A., Coeurjolly, D., Tougne, L.: Dynamic reconstruction of complex planar objects on irregular isothetic grids. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Remagnino, P., Nefian, A., Meenakshisundaram, G., Pascucci, V., Zara, J., Molineros, J., Theisel, H., Malzbender, T. (eds.) ISVC 2006. LNCS, vol. 4292, pp. 205–214. Springer, Heidelberg (2006)
15. Witkin, A.P.: Scale-space filtering. In: Proc. 8th Int. Joint Conf. Art. Intell., Larlsruhe, Germany, pp. 1019–1022 (1983)
16. Yaacoub, J.: Enveloppes convexes de réseaux et applications au traitement d'images. PhD thesis, Université Louis Pasteur, Strasbourg (1997)