Université de Savoie
U.F.R Sciences Fondamentales et Appliquées
Laboratoire de Mathématiques LAMA
École Doctorale MSTII

THESE

pour obtenir le grade de

Docteur de l'Université de Savoie

Discipline:Mathématiques - Informatiques

présentée et soutenue publiquement

par

MOUHAMMAD SAID

le 8 décembre 2010

Géométrie multirésolution des objets discrets bruités

COMPOSITION DU JURY

Acknowledgments

Last thing to do :-)

Humayoun *Humayoun* `Humayoun` test.

Abstract

fsdf sdf sdf **sd** All this research work has been implemented in a commercial software (Imago from DOSIsoft), allowing us to validate our results in clinical conditions.

Keywords: Atlas-based Segmentation, non rigid registration, radiotherapy, atlas creation

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Motivation and Goals

1.2 Overview of the Thesis

1.3 Road map

Chapter 2
Background

Contents

2.1 Continued Fractions

At first glance nothing seems simpler or less significant than writing a number, for example $\frac{3}{7}$, in the formal

$$\frac{3}{7} = 0 + \frac{1}{\frac{7}{3}} = 0 + \frac{1}{2 + \frac{1}{3}} = 0 + \cfrac{1}{2 + \cfrac{1}{2 + \cfrac{1}{1}}}.$$

It turns out, however, that fractions of this form, called *continued fractions*, provide much insight into many mathematical problems, particularly into the nature of numbers.

Continued fractions were studied by the great mathematicians of the seventeenth and eighteenth centuries and are a subject of active investigation today.

This section shows how continued fractions might be discovered accidentally, and then, by means of examples, how rational fractions can be expanded into continued fractions. Gradually more general notation is introduced and preliminary theorems are stated and proved.

2.1.1 Notations and Definitions

The theory of continued fractions deals with a special algorithm that is one of the most important tools in analysis, probability theory, mechanics, and especially, number theory. The history of the use of continued fractions is as long as the history of the use of Euclid's algorithm, because the process of finding the greatest commun divisor for two positive integers a and b is the same process as calculating the continued fractions expansion of $\frac{a}{b}$. For more historical information see Brezinski (1991) [?], Flajolet et al. (2000) [?], and Vardi (1998) [?].

The purpose of the present elementary text is to acquaint the reader only with the so-called regular or simple continued fractions, that is, those of the form

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + ...}}$$

The letters $a_0, a_1, a_3, ...,$ in most general treatment of the subject, denote independent variables. In particular cases, these variables may be allowed to take values only in certain specified domains. Thus, $a_0, a_1, a_3, ...$ may be assumed to be real or complex numbers, functions of one or several variable, and so on. For the purposes of the PhD, we shall always assume $a_1, a_2, ...$ to be positive integers; a_0 equal to zero. We shall call these numbers the *elements* of the given continued fraction. The number of elements may be either finite or infinite. In this chapter, we shall write the given continued fraction in the form

$$a_0 + \cfrac{1}{a_1 + \cfrac{1}{... + \cfrac{1}{a_{n-1} + \cfrac{1}{a_n}}}} = [a_0; a_1, a_2, ..., a_n]$$

and call it a finite continued fraction, where n is the depth of the fraction, and a_0, a_1, etc, are all intergers and called *partial quotients.*

2.1.2 Convergents

Every finite continued fractions,

$$[a_0; a_1, a_2, ..., a_n]$$

being the result of a finite number of rational operations on its elements, is a rational funtion of these elements and, consequently, can be repesented as the ratio of two polynomials

$$\frac{P(a_0; a_1, a_2, ..., a_n)}{Q(a_0; a_1, a_2, ..., a_n)}$$

in $a_0, a_1, a_2, ..., a_n$, with integral coefficients. If the elements have numerical values, the given continued fractions is then represented in the form of an ordinary fraction $\frac{p}{q}$. However, such a representation is, of course, not unique. For what follows, it will be important for us to have a *definite* representation of a finite continued fraction in the form of a simple fraction - a representation which we shall call *canonical*. We shall define such a representation by induction.

For a zeroth-order continued fraction,

$$[a_0] = a_0,$$

we take as our canonical representation the fraction $a_0/1$. Suppose now that canonical representations are defined for continued fractions of order less than n.

$$[a_0; a_1, a_2, ..., a_n] = [a_0; r_1] = a_0 + \frac{1}{r_1}.$$

Here,

$$r_1 = [a_1; a_2, a_3, ..., a_n]$$

is an (n-1)st-order continued fraction, for which, consequently, the canonical representation is already defined. Let us represent it as

$$r_1 = \frac{p'}{q'}$$

then,

$$[a_0; a_1, a_2, ..., a_n] = a_0 + \frac{q'}{p'} = \frac{a_0 p' + q'}{p'}$$

We shall take this last fraction as our canonical representation of the continued fraction $[a_0; a_1, a_2, ..., a_n]$. Thus, by setting,

$$[a_0; a_1, a_2, ..., a_n] = \frac{p}{q},$$
$$r_1 = [a_1; a_2, ..., a_n] = \frac{p'}{q'},$$

we have the following expressions for the numerators and denomirators of these canonical representations:

$$p = a_0 p' + q', \qquad q = p'. \tag{2.1}$$

Thus, we have uniquely defined canonical representations of continued fractions of all orders.

In the theory of continued fractions, an especially important role is played by the canonical representations of the segments of a given finite continued fraction. We shall denote by $\frac{p_k}{q_k}$ the canonical representation of the segment

$$s_k = [a_0; a_1, a_2, ..., a_k]$$

of the continued fraction, and we shall call it the kth-order convergent of the continued fraction α. For an nth-order continued fraction α, obviously

$$\frac{p_n}{q_n} = \alpha$$

such a continued fraction has n+1 convergents (of orders, 0, 1, 2, ..., n).

Theorem 1 *(the rule for the formation of the convergents). For arbitrary $k \geq 2$,*

$$\begin{aligned} p_k &= a_k p_{k-1} + p_{k-2}, \\ q_k &= a_k q_{k-1} + q_{k-2}. \end{aligned} \tag{2.2}$$

Proof. In the case of $k = 2$, the formulas in 3.3 are easily verified directly. Let us suppose that they are true for all $k < n$. Let us then consider the continued fractions

$$[a_0; a_1, a_2, ..., a_n]$$

and let us denote by p'_r/q'_r its rth-order convergent. On the basis of the formulas in 3.6,

$$\begin{aligned} p_n = a_0 p'_{n-1} + q'_{n-1}, \\ q_n = p'_{n-1}. \end{aligned} \tag{2.3}$$

And since, by hypothetis,

$$\begin{aligned} p'_{n-1} = a_n p'_{n-2} + p'_{n-3}, \\ q'_{n-1} = a_n q'_{n-2} + q'_{n-3}. \end{aligned} \tag{2.4}$$

(here, we have a_n rather than a_{n-1} bacause the fraction $[a_0; a_1, a_2, ..., a_n]$ begins with a_1 and not with a_0), it follows on the basis of 3.6 that

$$\begin{aligned} p_n &= a_0(a_n p'_{n-2} + p'_{n-3}) + (a_n q'_{n-2} + q_{n-3}), \\ &= a_n(a_0 p'_{n-2} + q'_{n-2}) + (a_0 p'_{n-3} + q_{n-3}), \\ &= a_n p_{n-1} + p_{n-2}, \\ q_n &= a_n p'_{n-2} + p'_{n-3}, \\ &= a_n q_{n-1} + q_{n-2}. \end{aligned} \tag{2.5}$$

which completes the proof.

These recursion formulas 3.3, which express the numerator and denomirator of an nth-order convergent in terms of the element a_n and the numerators and denomerators of the two preceding convergents, serve as the formal basis of the entire theory of continued fractions.

Remark 1 *It is sometimes convenient to consider a convergent of order -1; in this case, we set* $p_{-1} = 1$ *and* $q_{-1} = 0$*. Obviously, with this convention (and only then), the formulas of* 3.3 *retain their validity for* $k = 1$*.*

Theorem 2 *For all* $k \geq 0$*,*

$$q_k p_{k-1} - p_k q_{k-1} = (-1)^k. \tag{2.6}$$

Theorem 3 *For all* $k \geq 1$*,*

$$q_k p_{k-2} - p_k q_{k-2} = (-1)^{k-1} a_k. \tag{2.7}$$

2.1.3 The Stern-Brocot Tree

The role of partial quotients can be visualized with a structure called *Stern-Brocot tree* (see [?] and [?] 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. An illustration of this tree is proposed in figure 2.1. The node labeled by $\frac{m}{n}$ is a descendant of the node $\frac{k}{l}$ (and $\frac{k}{l}$ is then called an ancestor of $\frac{m}{n}$) if there is a path leading upwards from $\frac{m}{n}$ to $\frac{k}{l}$. Each fraction in the Stern-Brocot tree, except the first two, is of the form $\frac{m+m'}{n+n'}$, where $\frac{m}{n}$ is the nearest ancestor above and to the left, and $\frac{m'}{n'}$ is is the nearest ancestor above and to the right.

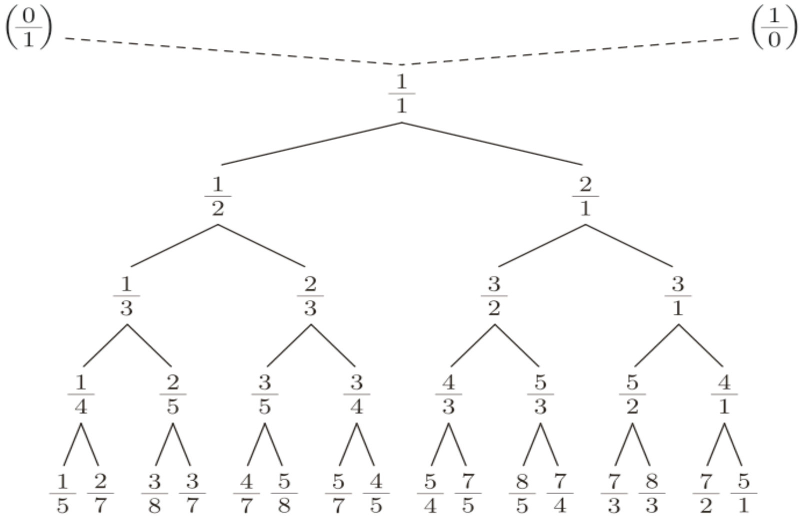
The left-hand side of the Stern-Brocot tree is also called the *Farey tree* [?]. The Stern-Brocot tree has a number of interesting properties, which can be found in [?]. For example, all the fractions appearing in it are in lowest terms. Each positive fraction $\frac{m}{n}$ where m and n are relatively prime, appears in the tree exactly once and non of them is omitted. The construction of the tree preserves the order $<$ in $\mathbb{Q}$.

Any node with a value between 0 and 1 is obtained by finite successive moves from the $\dfrac{1}{1}$ node. The moves can be of two types: L is a mowe toward the left child, R is a move toward the right child. Those moves determine the type of node when they end paths: even (resp. odd) nodes end with a R (resp. L) move. It is known that those nodes have a development in continued fraction, and it such that:

- $[0, u_1, u_2, ..., u_{2k}, u_{2k+1}, ..., u_{2i-1}, u_{2i}] = R^0 L^{u_1} ... R^{u_{2k}} L^{u_{2k+1}} ... L^{u_{2i-1}} R^{u_{2i}-1}$
- $[0, u_1, u_2, ..., u_{2k}, u_{2k+1}, ..., u_{2i}, u_{2i+1}] = R^0 L^{u_1} ... R^{u_{2k}} L^{u_{2k+1}} ... R^{u_{2i}} L^{u_{2i+1}-1}$

.

Of course odd nodes have an odd depth in their development in continued fractions, similarly even nodes have an even depth. When descending the tree the depth of a child



$\binom{0}{1}$
$\binom{1}{0}$
$\frac{1}{1}$
$\frac{1}{2}$ $\frac{2}{1}$
$\frac{1}{3}$ $\frac{2}{3}$ $\frac{3}{2}$ $\frac{3}{1}$
$\frac{1}{4}$ $\frac{2}{5}$ $\frac{3}{5}$ $\frac{3}{4}$ $\frac{4}{3}$ $\frac{5}{3}$ $\frac{5}{2}$ $\frac{4}{1}$
$\frac{1}{5}$ $\frac{2}{7}$ $\frac{3}{8}$ $\frac{3}{7}$ $\frac{4}{7}$ $\frac{5}{8}$ $\frac{5}{7}$ $\frac{4}{5}$ $\frac{5}{4}$ $\frac{7}{5}$ $\frac{8}{5}$ $\frac{7}{4}$ $\frac{7}{3}$ $\frac{8}{3}$ $\frac{7}{2}$ $\frac{5}{1}$

Figure 2.1: Stern-Brocot tree: positive irreductible rational fractions

changes if the move used to reach it differs from the last move used to reach its father. Consider the node $\frac{1}{2}$ whose depth equals one, the last move used to reach it is a L move, its left child $\frac{1}{3}$ has the same depth. The right child of $\frac{1}{2}$ is obtained by the successive moves $R^0L^1R^1$ and has a depth that equals two. We can classify the nodes of the Stern-Brocot tree according to the depth of their developmant in contnued fraction (Figure 2.1).

2.2 Discrete Geometry

Digital geometry deals with geometric properties of subsets of digital pictures and with the approximation of geometric properties of objects by making use of the properties of the digital picture subsets the represent the objects.

Digital geometry can be viewed as a special branch of discrete geometry that deals with graph-theoretical or combinatorial concepts. It can also viewed as approximate Euclidean geometry on the basis of the fact that picture analysis generally makes use of ideas about Euclidean space. However, digital geometry differs from approximation theory in its use of digitized input data (grid points that are not necessaily on the original curve) rather than sampled input data (sample points that are on the curve but that are not necessarily grid points) and in its focus on understanding the data in digital terms rather than approximating the data with the use of polynomials. Digital geometry also differs from computational geometry, which deals with finite sets of geometric objects in Euclidean space.

In this section, we present some basic notions of discrete geometry that are used in this work. We first introduce the Discrete Line and then present the notion of Freeman's codes we use to recognize discrete lines.

2.2.1 Discrete Lines

The arithmetic definition of a discrete line was introduced by J.O. Reveillès [1, ?, ?].

Definition 1 *Let D a discrete line of parameters (a, b, μ) with slope $\frac{a}{b}$ with $a \leq b$, $b \neq 0$ and $pgcd(a, b) = 1$, is the set of integer points (x, y) verifying: (see Figure ?? for more details)*

$$\mu \leq ax + by < \mu + w$$

where,

- *if $w < max(|a|, |b|)$, D is not connected;*
- *if $w = max(|a|, |b|)$, D is a 8-connected discrete line;*
- *if $max(|a|, |b|) < w < |a| + |b|$, D is a 4- or 8-connected discrete line;*
- *if $w = |a| + |b|$, D is a 4-connected and is called standard discrete line;*
- *if $w > |a| + |b|$, D is a thick.*

In this thesis, we are interested in Standard digital lines which verify $w = a + b$, we shall note them $D(a, b, \mu, w)$.

2.2.2 Freeman's codes

The first characterisation of a digital straight segment was given by Freeman [?]. This characterisation is descriptive and makes use of codes which are defined for all possible moves in the 4-neighbourhood. The particular structure of a sequence of such codes (i.e. the chain code) is then used to characterise discrete straightness.

Definition 2 (Freeman'codes and chain code) *All possible moves in the 4-neighbourhood are numbered successively conterclockwise from 0 to 3, as shown in Figure 2.2.*

The encoding $c_{i i=1,\dots,n}(c_i \in 0, 1, 2, 3)$ of a given sequence of 4moves defined by the discrete points $p_{i i=0,\dots,n}$ is called the chain-code of this sequence.

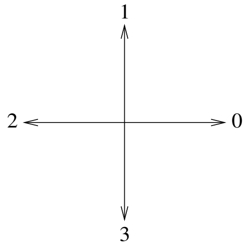


Figure 2.2: Freeman's codes in the 4-neighbourhood

2.2.3 CF-based descriptions of Digital Lines

Many works deal with the relations between irreducible rational fractions and digital lines (see [?] for characterization with Farey series, and [?] for a link with decomposition into continuous fractions). In [?], 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** [?] (also reported in [?]) updates the DSS slope when adding a point that is just exterior to the current line (weak exterior points). Sivignon et al. used the previous results to study the geometrical and arithmetical properties of the intersection of two digital lines or planes. More precisely, some results about the connectivety, periodicity and minimal parameters of this intersection have been reported [?, ?].

In [?], Brons used a linguistic methods for the description of a Straight Line on a grid. He provided an algorithm that constructs a string representing a straight line on a grid in Freeman's coding scheme. Some number-theoretical aspects of this algorithm are treated Euclid's algorithm, Farey series and continued fractions.

In 1982, Wu formulated a theorem describing digital straightness by a set of conditions (called the *DSS property*) which the corresponding chain code must fulfill; see Klette and Rosenfeld [5] [55, pp. 208-209, Th. 14]. Proofs of this theorem based on continued fractions were published by Bruckstein [?] described digital straightness by a number of transformations preserving it. Some of these transformations were defined by means of continued fractions.

Dorst and Duin (1984) [?] presented an algorithm for drawing digital straight line, which is valid for irrational slopes.

In [?], De Vieilleville and Lachaud have used the results proposed by Debled-Rennesson and J.-P. Reveillès in 1995 [1]. They have revisited a classical arithmetically-based DSS recognition algorithm with new parameters related to a combinatorial representation of DSS. New analytic relations have been established and the relation with the Stern-Brocot tree has been made explicit.

In [?], Hanna Uscka-Wehlou presented the run-hierarchical structure of digital lines with irrational slopes, based on the continued fractions expansions of the slopes. The theoretical part of the paper was based on [?] and the examples were based on the literature concerning continued fractions [?]. The examples show how to use the theory in finding digitization patterns. This description can also be useful in theoretical research on digital lines with irrational slopes. For example, in [?] Hanna U-K has examined some classes of digital lines, defined by the continued fractions expansions of their slopes.

In [?], Voss presented a recursive transformations for computing the pattern of a standard line from the *simple continued fractions* of its slope.

2.3 Multiscale Analysis of Digital Geometry

2.4 Maximal Digital Straight Segments

Maximal segments defined on digital curves and edges of convex digital polygons [?, ?, ?, ?]. Maximal segments of a digital curve are DSS not strictly included in many other DSS of the curve. Efficient algorithms have been proposed to extract them and compute their characteristics [1] as well as optimal algorithms to recover the whole set of maximal segments. They are useful when estimating the local geometry of digital curves like tangent direction or curvature [?, ?]. Through them digital curves can be polygonized into the minimal number of straight segments [3]. Maximal segments can be used to decide whether or not a polyomino is convex [?]. The main result of this section is Theorem [1] which bounds the slope difference between two consecutive maximal segments.

Definition 3 (*Réveillès* [?]) *The set of points (x, y) of the digital plane verifying $\mu \leq ax - by < \mu + |a| + |b|$, with a,b and μ integer numbers, is called the standard line with slope a/b and shift μ.*

The standard lines are the 4-connected discrete lines. The quantity $ax - by$ is called the remainder of the line. The points whose remainder is μ (resp. $\mu + |a| + |b| + 1$) are called upper (resp. lower) leaning points. Finite connected portions of digital lines define *digital straight line.*

Definition 4 (*Réveillès* [?]) *A set of successive points $[C_iC_j]$ of C is a digital straight segment DSS iff there exists a standard line $D(a, b, \mu)$ containing them. The predicate $[C_iC_j]$ is a DSS and is denoted by S(i,j).*

The principal upper and lower leaning points are defined as those with extremal x values. The first index j, $i \leq j$, such that $S(i, j)$ and $\neg S(i, j + 1)$ is called the front of i. The map associating any i to its front is denoted by F. Symmetrically, the first index i such that $S(i, j)$ and $\neg S(i - 1, j)$ is called the back of j and the corresponding mapping is denoted by B. Maximal Segments are defined as those DSS not strictly included in another DSS.

These relations give the four equivalent characterisations of maximal segments:

Definition 5 *Any set of points $[C_iC_j]$ is called a maximal segment iff any of the following equivalent characterisations holds:* $(1)S(i,j)$ *and* $\neg S(i,j{+}1)$ *and* $\neg S(i{-}1,j)$*,* $(2)B(j)=i$ *and* $F(i)=j$*,* $(3)\exists k, i=B(k)$ *and* $j=F(B(k))$*,* $(4)\exists k', i=B(F(k'))$ *and* $j=F(k')$*.*

Consecutive maximal segments overlap and often on more than two points. The digital path that belongs to two consecutive maximal segments is called a *common part*, its associated maximal segments are $C_{B(j),j}$ and $C_{i,F(i)}$ if $C_{i,j}$ is the common part. Note that $C_{i-1,j+1}$ is not a DSS. A common part is never empty. In fact they knowed the type of the limiting points for all common parts, as shown below:

Lemma 1 *[?] If $C_{i,j}$ is the common part of two consecutive maximal segments, then the points i-1 and j+1 are both upper or both lower weakly exterior.*

See Theorem 1 [?] for more details about an analytic bounds on slopes of two consecutive maximal segments.

Chapter 3

Multi-Scale of Straight Digital Line

Contents

3.1 Introduction

Multiscale analysis is a classical tool in image processing [?, ?]. 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 [?] 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.* [?] 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 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 [?, 6], circular arcs estimate curvature [?]. 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 [?] 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 this chapter, 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 4, 5 and 6).

3.2 Covering of a standard digital 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 4) and that this line is standard (Theorem 5). The proof is essentially technical and is based on the same principle as the proof of Figueiredo [?]. The preceding theorems are valid for digital straight lines in the second and fourth quadrants. For completeness, we also provide the equivalent theorem for digital straight lines in the first and third quadrants (Theorem 6).

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 $\mathbb{S}(h, v)$= $(Xh,\ Yv)$ of $\mathbb{Z}^2$ (where X, Y, h, v all are integers). Obviously the fondamental domain $[0, h) \times [0, v)$ of $\mathbb{S}(h, v)$ 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 $\mathbb{S}(h, v)$ (for which reason we will refer indifferently to the tile itself or to the point of $\mathbb{S}(h, v)$ it contains). We are interested in the set of tiles of $\mathbb{S}(h, v)$ that intersect $D(a, b, \mu)$. We denote this set by Δ. Although we have not proven yet that it is indeed a digital straight line, we already call it the *covering line* of $D(a, b, \mu)$ in $\mathbb{S}(h, v)$. More generally, given any subset O of $\mathbb{Z}^2$, the set of tiles of $\mathbb{S}(h, v)$ that intersects O is called its (h, v)-*covering.*

The tiling generated by $\mathbb{S}(h, v)$ 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:

$$\begin{aligned} X &= \left[\frac{x}{h}\right] \\ Y &= \left[\frac{y}{v}\right] \end{aligned} \tag{3.1}$$

which can be inverted as follows:

$$\begin{aligned} x &= hX + \left\{\frac{x}{h}\right\} \\ y &= vY + \left\{\frac{y}{v}\right\} \end{aligned} \tag{3.2}$$

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

$$\mu \leq ax + by < \mu + a + b \tag{3.3}$$

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

$$\begin{gathered} \mu \leq a(hX + \left\{\frac{x}{h}\right\}) + b(vY + \left\{\frac{y}{v}\right\}) < \mu + a + b \\ \mu - a\left\{\frac{x}{h}\right\} - b\left\{\frac{y}{v}\right\} \leq ahX + bvY < \mu + a + b - a\left\{\frac{x}{h}\right\} - b\left\{\frac{y}{v}\right\} \end{gathered} \tag{3.4}$$

In order to simplify (3.4) let us introduce

$$\begin{aligned} m_x &= \left\{\frac{x}{h}\right\} \\ m_y &= \left\{\frac{y}{v}\right\} \end{aligned} \tag{3.5}$$

Since m_x and m_y vary when x steps through $\mathbb{Z}$, equation (3.4) becomes:

$$\mu - max_{x\in\mathbb{Z}}(am_x + bm_y) \leq ahX + bvY < \mu + a + b - min_{x\in\mathbb{Z}}(am_x + bm_y) \tag{3.6}$$

Now to fully determine this equation that defines the covering of the digital line by the tiling, the exact rang 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.

Determination of the range of $am_x + bm_y$:

By definition in equation(3.5), it is clear that:

$$\begin{gathered} 0 \leq m_x \leq h - 1 \\ 0 \leq m_y \leq v - 1 \end{gathered} \tag{3.7}$$

Using these bounds in (3.6) suggests for Δ an equation of the form as

$$0 \leq am_x + bm_y \leq a(h-1) + b(v-1)$$

Then

$$\mu - a(h-1) - b(v-1) \leq ahX + bvY < \mu + a + b \tag{3.8}$$

But since $ahX + bvY$ can only assume values that are multiples of $g = \gcd(ah, bv)$, the bounds of equation (3.6) can be refined. we further denote $\dfrac{ah+bv}{g}$ by p.
However since m_x and m_y are linked through (3.5), 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 to absolute bounds 0 and $a(h-1) + b(v-1)$ in somes cases,
Inverting equation (3.5) yields

$$\begin{aligned} x &= kh + m_x \quad ,k \in \mathbb{Z} \\ y &= lv + m_y \quad ,l \in \mathbb{Z} \end{aligned} \tag{3.9}$$

Hence, after insertion into (3.3):

$$\begin{aligned} \mu \le ax + by \;&\Rightarrow\; \mu \le a(kh + m_x) + b(lv + m_y) \\ &\Rightarrow\; \mu - akh - blv \le am_x + bm_y \end{aligned}$$

$$\begin{aligned} ax + by < \mu + a + b \;&\Rightarrow\; a(kh + m_x) + b(lv + m_y) \le \mu + a + b \\ &\Rightarrow\; am_x + bm_y < \mu + a + b - akh - blv \end{aligned}$$

Then

$$\mu - akh - blv \le am_x + bm_y < \mu + a + b - akh - blv \tag{3.10}$$

The term $kah + lbv$ has values which are multiples of $g = \gcd(ah, bv)$:

$$\begin{aligned} ah = p_1 g, p_1 \in \mathbb{Z} \;\; and \;\; bv = p_2 g, p_2 \in \mathbb{Z} \;&\Rightarrow\; kah = kp_1 g \;\; and \;\; lbv = lp_2 g \\ &\Rightarrow\; kah + lbv = kp_1 + lp_2 g = (kp_1 + lp_2)g = tg, \end{aligned}$$

Then

$$kah + lbv = tg \;\; ,t \in \mathbb{Z} \tag{3.11}$$

Equation (3.10) can thus be rewritten as

$$\mu - tg \le am_x + bm_y < \mu + a + b - tg, t \in \mathbb{Z} \tag{3.12}$$

Equation (3.12) 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 3.1).

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 $am_x + bm_y$, i.e.

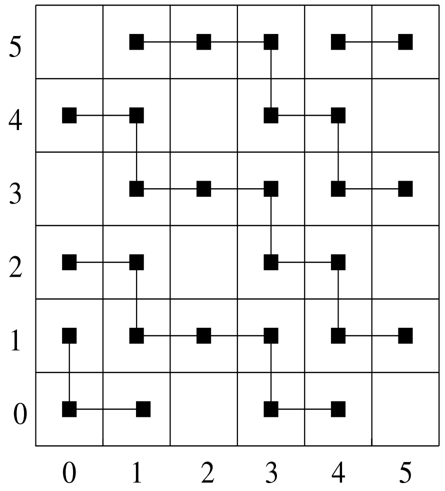


Figure 3.1: **Determination of the range of $2m_x$+$3m_y$.** The black squares represent the family of standard digital lines defined by $5-6t \leq 2m_x+3m_y < 10-6t$ restricted to $[0,6)\times[0,6)$ and hence the possible values of (m_x, m_y)

$an_x + bn_y = max_{(m_x,m_y)\in D_t\cap[0,h)\times[0,v)}(am_x + bm_y)$
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):

$$\mu - tg \leq am_x + bm_y \leq \mu + a + b - tg - 1 \tag{3.13}$$

$am_x + bm_y$ reaches the minimal value $\mu - tg$ and the maximal value $\mu + a + b - tg - 1$, Then:

$$d \;=\; a(h-1) + b(v-1) - (an_x + bn_y) \tag{3.14}$$

Determining the maximum value of t verifying (3.14) for the point (n_x, n_y) provides a more precise expression of the lower bound of d. (3.14) becomes

$$\begin{aligned} d &= a(h-1) + b(v-1) - (\mu + a + b - tg - 1) \\ &= ah - a + bv - b - \mu - a - b + tg + 1 \\ &= pg + tg - 2a - 2b + 1 - \mu \\ &= (p+t)g - (\mu + 2a + 2b - 1). \end{aligned}$$

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

$$\begin{aligned}
(p+t)g-(\mu+2a+2b-1) &= 0 \\
p+t-\left(\frac{\mu+2a+2b-1}{g}\right) &= 0 \\
t &= -p+\left(\frac{\mu+2a+2b-1}{g})\right) \\
t &= -p+\frac{1}{g}\left(\left[\frac{\mu+2a+2b-1}{g}\right]g+\left\{\frac{\mu+2a+2b-1}{g}\right\}\right) \\
t &= -p+\left[\frac{\mu+2a+2b-1}{g}\right]+\frac{\left\{\frac{\mu+2a+2b-1}{g}\right\}}{g} \\
t &= -p+Q_2+\frac{R_2}{g},
\end{aligned} \tag{3.15}$$

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 divided $\mu+2a+2b-1$. In this case, (3.15) becomes

$$t = -p + Q_2$$

And we insert this equality in (3.6).

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

Divided by g, then

$$\frac{ahX}{g}+\frac{bvY}{g} \geq -p+Q_2-\frac{a+b-1}{g} \tag{3.16}$$

Where a,b and g being three integers, we denote with $\left[\frac{a+b-1}{g}\right]$ the quotient of the euclidean division of $a+b-1$ by g while $\left\{\frac{a+b-1}{g}\right\}$ denotes the remainder of this division.
The fundamental relation between these two values writes:

$$a+b-1 = g\left[\frac{a+b-1}{g}\right]+\left\{\frac{a+b-1}{g}\right\}$$

Let $\alpha = \dfrac{ah}{g}$ and $\beta = \dfrac{bv}{g}$. Then Equation(3.16) can be rewritten as:

$$\alpha X + \beta Y \geq -p + Q_2 - Q_1 - \frac{R_1}{g} \tag{3.17}$$

Where $Q_1 = \left[\frac{a+b-1}{g}\right]$ and $R_1 = \left\{\frac{a+b-1}{g}\right\}$. We conclude that the division of remainders of $a+b-1$ by g by g is between 0 and 1 strict, and all terms in Equation(3.17) are integers, so we get:

$$\alpha X + \beta Y \quad \geq \quad -p + Q_2 - Q_1 \tag{3.18}$$

2. If g does not divide $\mu + 2a + 2b - 1$. In this case, (3.15) becomes

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

Hence Equation (3.6) can be rewritten as

$$\begin{aligned} ahX + bvY &\geq tg - a - b + 1 \\ &\geq \left(-p + Q_2 + \frac{R_2}{g}\right) g - a - b + 1 \end{aligned}$$

Divided by g, then,

$$\begin{aligned} \alpha X + \beta Y &\geq -p + Q_2 + \frac{R_2}{g} - \frac{a+b-1}{g} \\ &\geq -p + Q_2 - Q_1 + \frac{R_2}{g} - \frac{R_1}{g} \end{aligned}$$

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:

- If $R_2 \leq R_1$,then,

$$\alpha X + \beta Y \geq -p + Q_2 - Q_1 \tag{3.19}$$

- If $R_2 > R_1$,then,

$$\alpha X + \beta Y \geq -p + Q_2 - Q_1 + 1 \tag{3.20}$$

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).
$an^{'}_x + bn^{'}_y = min_{(m_x,m_y) \in D_t \cap [0,h) \times [0,v)}(am_x + bm_y)$

Let us also denote by d' the difference between $am_x + bm_y$ evaluated at $(0,0)$ and at $(n^{'}_x, n^{'}_y)$, Then:

$$d' = 0 - (an^{'}_x + bn^{'}_y) \tag{3.21}$$

Determining the maximum value of t verifying (3.21) for the point (n'_x, n'_y) provides a more precise expression of the upper bound of d'. (3.21) becomes

$$\begin{aligned} d' &= 0 - (\mu - tg) \\ &= -\mu + tg = \mu + 2a + 2b - 1 - 2\mu - 2a - 2b + 1 + tg \\ &= (\mu + 2a + 2b - 1) + tg - (2\mu + 2a + 2b - 1) \end{aligned}$$

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

$$\begin{aligned} t &= -\left(\frac{\mu + 2a + 2b - 1}{g}\right) + \left(\frac{2\mu + 2a + 2b - 1}{g}\right) \\ &= -Q_2 - \frac{R_2}{g} + \left(\frac{2\mu + 2a + 2b - 1}{g}\right) \end{aligned} \tag{3.22}$$

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

1. If g divides $\mu + 2a + 2b - 1$. In this case (3.22) becomes

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

   We insert this equality in (3.6) and can be rewritten as

   $$\begin{aligned} ahX + bvY &< \mu + a + b - (\mu - tg) \\ &< tg + a + b \\ &< \left(-Q_2 + \left(\frac{2\mu + 2a + 2b - 1}{g}\right)\right) g + a + b \\ &< -Q_2 g + 2\mu + 3a + 3b - 1 \end{aligned}$$

   Divided by g, then,

   $$\alpha X + \beta Y < -Q_2 + Q_3 + \frac{R_3}{g} \tag{3.23}$$

   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 (3.23) becomes

$$\alpha X + \beta Y < -Q_2 + Q_3 \tag{3.24}$$

(b) If g does not divide $2\mu + 3a + 3b - 1$. In this case Equation (3.23) becomes

$$\alpha X + \beta Y < -Q_2 + Q_3 + 1 \tag{3.25}$$

2. If g does not divide $\mu + 2a + 2b - 1$. In this case Equation (3.22) becomes

$$t = -\left[\frac{\mu + 2a + 2b - 1}{g}\right] - \frac{\left\{\dfrac{\mu + 2a + 2b - 1}{g}\right\}}{g} + \left(\frac{2\mu + 2a + 2b - 1}{g}\right)$$

Hence Equation (3.14) can be rewritten as

$$\begin{aligned} ahX + bvY &< \mu + a + b - (\mu - tg) \\ &< tg + a + b \\ &< -Q_2 g - R_2 + 2\mu + 3a + 3b - 1 \end{aligned} \tag{3.26}$$

Where μ,a,b and g being four integers, we denote with $\left[\dfrac{2\mu + 3a + 3b - 1}{g}\right]$ the quotient of the euclidean division of $\mu + 3a + 3b - 1$ by g while $\left\{\dfrac{2\mu + 3a + 3b - 1}{g}\right\}$ denotes the remainder of this division.
The fundamental relation between these two values can be rewriten as:

$$2\mu + 3a + 3b - 1 = g\left[\frac{2\mu + 3a + 3b - 1}{g}\right] + \left\{\frac{2\mu + 3a + 3b - 1}{g}\right\}$$

Let $\alpha = \dfrac{ah}{g}$ and $\beta = \dfrac{bv}{g}$. Then Equation (3.26) 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:

- If $R_3 \leq R_2$, then,

$$\alpha X + \beta Y < -Q_2 + Q_3 \tag{3.27}$$

- If $R_3 > R_2$, then,

$$\alpha X + \beta Y < -Q_2 + Q_3 + 1 \tag{3.28}$$

More precisely, eleven equations involving $\alpha X + \beta Y$ can be obtained by combining the lower bound (equations 3.18, 3.19, 3.20) and the upper bound (equations 3.24, 3.25, 3.27, 3.28) of $\alpha X + \beta Y$. Then we have the following cases:

1. If $(R_2 = 0)$ and $(R_3 \neq 0)$
$$-p + Q_2 - Q_1 \leq \alpha X + \beta Y < Q_3 - Q_2 + 1$$

2. If $(R_2 = 0)$ and $(R_3 = 0)$
$$-p + Q_2 - Q_1 \leq \alpha X + \beta Y < Q_3 - Q_2$$

3. If $(R_2 \neq 0)$ and $(R_1 = R_2)$ and $(R_2 = R_3)$
$$-p + Q_2 - Q_1 \leq \alpha X + \beta Y < Q_3 - Q_2$$

4. If $(R_2 \neq 0)$ and $(R_1 = R_2)$ and $(R_3 < R_2)$
$$-p + Q_2 - Q_1 \leq \alpha X + \beta Y < Q_3 - Q_2$$

5. If $(R_2 \neq 0)$ and $(R_1 = R_2)$ and $(R_3 > R_2)$
$$-p + Q_2 - Q_1 \leq \alpha X + \beta Y < Q_3 - Q_2 + 1$$

6. If $(R_2 \neq 0)$ and $(R_2 < R_1)$ and $(R_3 = R_2)$
$$-p + Q_2 - Q_1 \leq \alpha X + \beta Y < Q_3 - Q_2$$

7. If $(R_2 \neq 0)$ and $(R_2 < R_1)$ and $(R_3 < R_2)$
$$-p + Q_2 - Q_1 \leq \alpha X + \beta Y < Q_3 - Q_2$$

8. If $(R_2 \neq 0)$ and $(R_2 < R_1)$ and $(R_3 > R_2)$
$$-p + Q_2 - Q_1 \leq \alpha X + \beta Y < Q_3 - Q_2 + 1$$

9. If $(R_2 \neq 0)$ and $(R_2 > R_1)$ and $(R_3 = R_2)$
$$-p + Q_2 - Q_1 + 1 \leq \alpha X + \beta Y < Q_3 - Q_2$$

10. If $(R_2 \neq 0)$ and $(R_2 > R_1)$ and $(R_3 < R_2)$
$$-p + Q_2 - Q_1 + 1 \leq \alpha X + \beta Y < Q_3 - Q_2$$

11. If $(R_2 \neq 0)$ and $(R_2 > R_1)$ and $(R_3 > R_2)$

$$-p + Q_2 - Q_1 + 1 \leq \alpha X + \beta Y < Q_3 - Q_2 + 1$$

Remark 2 *We want to prove that the cases 4,5,6 and 9 are impossible*
Proof of case 4.

$$\begin{aligned}
R_1 &= R_2 \\
\left\{\frac{a+b-1}{g}\right\} &= \left\{\frac{\mu+2a+2b-1}{g}\right\} \\
a+b-1 &\equiv \mu+2a+2b-1(mod\ g) \\
\mu+a+b &\equiv 0(mod\ g)
\end{aligned}$$

$$\begin{aligned}
R_3 &< R_2 \\
\left\{\frac{2\mu+3a+3b-1}{g}\right\} - \left\{\frac{\mu+a+b-1}{g}\right\} &< 0 \\
\left\{\frac{\mu+a+b}{g} + \frac{\mu+a+b-1}{g}\right\} - \left\{\frac{\mu+a+b-1}{g}\right\} &< 0 \\
\left\{0 + \frac{\mu+a+b-1}{g}\right\} - \left\{\frac{\mu+a+b-1}{g}\right\} &< 0 \\
\left\{\frac{\mu+a+b-1}{g}\right\} - \left\{\frac{\mu+a+b-1}{g}\right\} &< 0 \\
0 &< 0 \qquad \textbf{\textit{absurd}}
\end{aligned}$$

Similarly, we prove that the cases 5,6 and 9 are impossible

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

$$-p + Q_2 - Q_1 + SI \leq \alpha X + \beta Y < Q_3 - Q_2 + SS \tag{3.29}$$

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

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

As an example, let us study the covering of the standard digital line

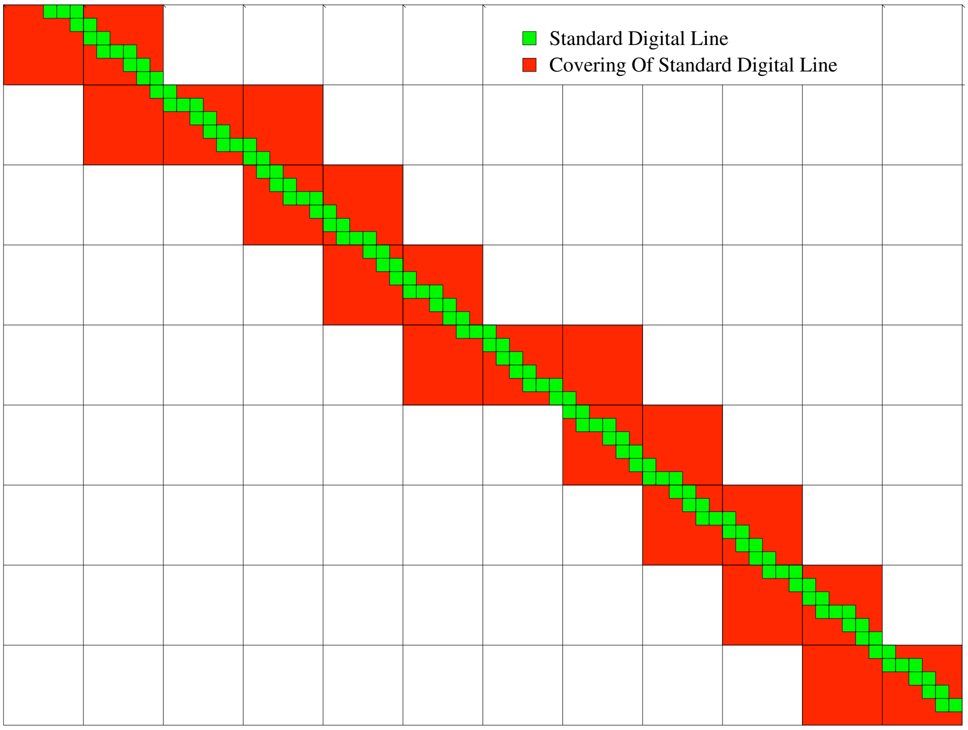


Figure 3.2: The covering line of $D(7,9,6)$ by $\mathbb{S}(6,6): -12 \leq 7X + 9Y < 4$

$$D(7,9,6)\colon\ 6 \leq 7x + 9y < 22$$

Figure 3.2 and Figure 3.3 illustrates respectively the $(6,6)$-covering and $(3,4)$-covering of $D(7,9,6)$.

Definition 6 (The Euclidean Division) *Let $a, b \in \mathbb{Z}$, $b > 0$, then there exists unique $q, r \in \mathbb{Z}$ such that $a = bq + r$, $0 \leq r < b$. Here q is called quotient of the integer division of a by b, and r is called remainder.*

Theorem 5 *The covering line Δ is standard.*

In order to prove that the difference between the upper and lower bounds of (3.29) is equal to $\alpha + \beta$, we first need the following proposition and lemma.

Proposition 1

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

Where $R = \left\{ \dfrac{\mu + a + b}{g} \right\}$.

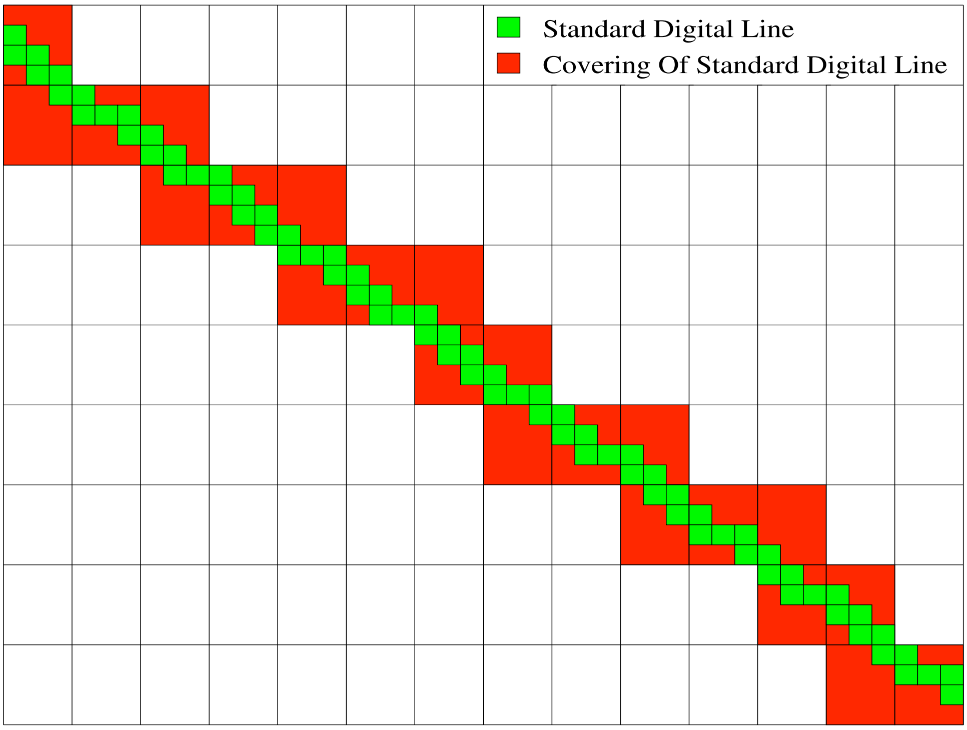


Figure 3.3: The covering line of $D(7,9,6)$ by $\mathbb{S}(3,4) : -11 \leq 7X + 12Y < 8$

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 2

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

Where $\left[\dfrac{\mu + a + b}{g} \right]$.

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 $Q = Q_2 - Q_1 - 1$.

Theorem 5. We want to prove that the difference between the upper and lower bounds of (3.29) is equal to $\alpha + \beta$. There are 5 cases to consider. From Proposition 1 and Lemma 2, 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, $(Q_3 - Q_2 + SS) - (-p + Q_2 - Q_1 + SI) = p$. □

Theorem 6 *The digital straight line Δ of $\mathbb{S}(h, v)$ covering the standard digital line $D(a, b, \mu)$ of $\mathbb{Z}^2$ on the first and the third quadrant is defined by:*

$$-p_1 + Q'_2 - Q'_1 + SI \leq \alpha X - \beta Y < p_2 + Q'_3 - Q'_2 + SS \tag{3.30}$$

Where $\alpha = p_1 = \frac{ah}{g}$, $\beta = p_2 = \frac{bv}{g}$, $g = \gcd(ah, bv)$,
$Q'_k = \left\{ \left[\frac{(k-1)\mu + ka + b - 1}{g} \right], k = 1, 2, 3 \right\}$, $R'_k = \left\{ \left\{ \frac{(k-1)\mu + ka + b - 1}{g} \right\}, k = 1, 2, 3 \right\}$ *and*

$$SI = \begin{cases} 0 & \text{if} \quad R'_2 \leq R'_1 \\ 1 & \quad \textit{otherwise} \end{cases} \qquad SS = \begin{cases} 0 & \text{if} \quad R'_3 \leq R'_2 \\ 1 & \quad \textit{otherwise} \end{cases}$$

proof. see Appendix A.

3.3 Conclusion

In this chapter, we have presented new results about the covering of discrete lines by regular and irregural tilings in all quadrants. Our approach is based on the arithmetic equation of a standard digital line and provides analytic formulas. These latter results will be used, in chapter 4 to determine the multiresolution of DSS by examining the multiresolution of the intersection of two DSL, and in chapter 5 to compute the exact multiscale covering of a digital contour C can be achevied directly from an arbitrary decomposition of C into digital straight segments.

Chapter 4

Multiscale Analysis of Digital Segments by Intersection of 2D Digital Lines

Contents

4.1 Motivation

Digital Straight Lines (DSL) and Digital Straight Segments (DSS) are useful to describe the geometry of a digital shape (coding, geometric estimators, feature detection) and this explains why they have been so deeply studied (see the survey [5] or [?]). When a straight line is digitized on a grid, we obtain a sequence of grid points defining a digital straight line segment. Methods of recognizing digital straight segments DSS are known since long. In one of the first methods, Freeman [?] suggested to analyze the regularity in the pattern of the directions in the chain code [?] of a digital curve. Anderson and Kim [?] have presented a deep analysis of the properties of the DSS's and suggested a different algorithm based on calculating the convex hull of the points of digital curves to be analyzed. In [?], Kovalevsky presented a *new classification of digital curves* into boundary curves and visual curves. Boundary curves and lines are a useful means for fast drawing of regions defined by their boundaries.

It is well known that shapes should be studied at different scales, and this has led to the development of regular and irregular pyramids for shape analysis and scene understanding (e.g. [?]). However there exists no analytical description of the multiresolution of a digital shape, contrary to the famous scale-space analysis in the continuous world [?, ?]. One of the contribution of this chapter is to give new analytical results on the multiresolution of DSL and DSS. A byproduct is new results about digital line intersection.

Discrete geometry is different from Euclidean geometry in many ways, and the differences between the intersection of two Euclidean lines and two Digital lines is often used to illustrate this difference. Indeed, while the intersection of two Euclidean lines is a Euclidean point, the intersection of two digital lines can be a discrete point, a set of discrete points or even empty on regular grids. Examples of digital lines intersection are depicted in Figure 4.1

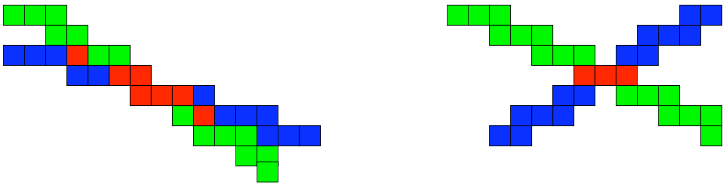


Figure 4.1: Intersection of two Standard Digital Lines D_1 (light green boxes) and D_2 (blue boxes), their intersection are drawn as red boxes. On the left (resp. right), the two lines are in the same quadrant (in two differents quadrants).

In [?], Debled et al. presented a definition of the set of intersection pixels of two digital lines using a unimodular matrix. This definition enables the design of an efficient algorithm to determine all the pixels of an intersection, given the parameters of two lines. Sivignon et al. studied the geometrical and arithmetical properties of the intersection of two digital lines or planes. More precisely, some results about the connectivety, periodicity and minimal parameters of this intersection have been reported [?, ?]. Figueiredo [?] first provided an analytical description of the multiresolution of 8-connected DSL.

Analytical formulae for DSS appear to be a much harder problem: since DSS are finite parts of DSL, they are more sensitive to arithmetic peculiarites. We therefore follow an indirect path to DSS multiresolution. In Section 2, given a DSS, we build two DSL whose intersection contains it and whose main connected part has the same arithmetic characteristics as well as the same number of patterns. We note here that we propose new results about the combinatorics of such digital line intersections, that are complementary to the results of Sivignon *et al.* [?]. Section 3 determines the multiresolution of DSS by examining the multiresolution of the intersection of these two DSL. We give a new analytical description of this set with arithmetic inequalities.

This chapter is a first step toward the multiresolution of a DSS in constant time by analytical formulae.

4.2 Standard Digital Lines Intersection

We recall here some definitions and properties about DSL, their relations with rational fractions, and a combinatoric definition of a DSS. We restrict our study of DSS to the main connected part (say S) of the intersection of two well-chosen standard DSL. These two lines are related to the downward moves in the Stern-Brocot tree during a DSS recognition. We finally show that S can be built so that it has the characteristics of any given DSS. Note that intersection of digital lines can be complex and may not be connected.

4.2.1 DSS, patterns, irreducible fractions and continued fractions

We here recall a few properties about *patterns* composing DSS and their close relations with continued fractions. They constitute a powerful tool to describe discrete lines with rational slopes [?]. All definitions and propositions stated below hold for DSS with slopes in the fourth quadrant.

Given a standard line (a, b, μ), we call *pattern* of characteristics (a, b) the succession of Freeman moves between any two consecutive lower leaning points. The Freeman moves defined between any two consecutive upper 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 [?], or the work of Berstel reported in [?]), 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:

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

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: $z_k = \frac{p_k}{q_k} = [u_0, u_1, u_2, ..., u_k]$.

We recall a few more relations regarding the way convergents are related:

$$\forall k \geq 1 \quad p_k q_{k-1} - p_{k-1} q_k = (-1)^{k+1} \tag{4.1}$$

$$p_0 = 0 \;\; p_{-1} = 1 \quad \forall k \geq 1 \quad p_k = u_k p_{k-1} + p_{k-2} \tag{4.2}$$

$$q_0 = 1 \;\; q_{-1} = 0 \quad \forall k \geq 1 \quad q_k = u_k q_{k-1} + q_{k-2} \tag{4.3}$$

Continued fractions can be finite or infinite, we focus on the case of rational slopes of lines in the fourth quadrant, that is finite continued fractions between 0 and 1. Then for each i, u_i is a strictly positive integer. In order to have a unique writing we consider that the last *partial coefficient* is greater or equal to two except for slope $1 = [0, 1]$.

Let us now explain how to compute the *pattern* associated with a rational slope z in the fourth quadrant.

Consider E a mapping from the set of positive rational number smaller than one onto

Freeman code words defined as follows. First terms are stated as $E(z_0) = 0$ and $E(z_1) = 0^{u_1}3$ and others are expressed recursively:

$$E(z_{2i+1}) = E(z_{2i})^{u_{2i+1}} E(z_{2i-1}) \tag{4.4}$$

$$E(z_{2i}) = E(z_{2i-2}) E(z_{2i-1})^{u_{2i}} \tag{4.5}$$

The role of partial quotients can be visualized with a structure called *Stern-Brocot tree* Figure 2.1 (see [?] 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 [?] for characterization with Farey series, and [?] for a link with decomposition into continuous fractions). In [?], 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** [?] (reported in [?]) 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.

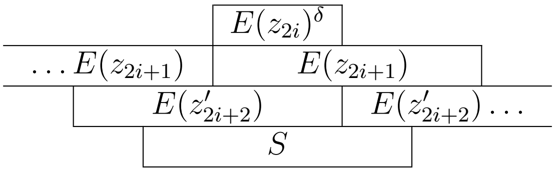
Proposition 2 *[?] The slope evolution in **DR95** depends on the parity of the complexity of its slope, the type of weakly exterior point added to the right (UWE and LWE stands respectively for upper and lower weakly exterior). This is summed up in the table below, where the slope is $[0, u_1, ..., u_n]$, $n = 2i$ even or $n = 2i+1$ odd, δ pattern(s) and δ' reversed pattern(s):*

	Even n	*Odd n*
UWE	$[0, u_1, ..., u_{2i}, \delta]$	$[0, u_1, ..., u_{2i+1} - 1, 1, \delta]$
LWE	$[0, u_1, ..., u_{2i} - 1, 1, \delta']$	$[0, u_1, ..., u_{2i+1}, \delta']$

4.2.2 Segments by digital line intersection.

In order to study a DSS composed of δ patterns of slope z_n, we build a very similar DSS which includes it as the intersection of two DSL with carefully chosen slopes. Their patterns are placed so that one starts at the first lower leaning point and the other ends at the last lower leaning point (see Fig. 4.2).

Proposition 3 *The main connected part S of the intersection between $E(z_{2i+1})$ with $z_{2i+1} = [0, u_1, ..., u_{2i}, \delta]$ and $E(z'_{2i+2})$ with $z'_{2i+2} = [0, u_1, ..., u_{2i} - 1, 1, \delta]$ is defined as their common part as placed below:*



The word S is exactly $w_1E(z_{2i})^{\delta}w_2$, with $w_1 = E(z_1)^{u_2} \cdots E(z_{2i-2k-1})^{u_{2i-2k}} \cdots E(z_{2i-3})^{u_{2i-2}}\ E(z_{2i-1})^{u_{2i}-1}$ and $w_2 = E(z_{2i-2})^{u_{2i-1}} \cdots E(z_{2i-2k})^{u_{2i-2k+1}} \cdots E(z_2)^{u_3}\ E(z_0)^{u_1}$. We remark that it contains the pattern $E(z_{2i}) = E(z'_{2i})$ repeated δ times (Figure 4.2 exemplifies the construction of this intersection).

Proof. From (4.4) and (4.5), we have $E(z_{2i+1}) = E(z_{2i})^{\delta}E(z_{2i-1})$ and $E(z^{'}_{2i+2}) = E(z^{'}_{2i})E(z^{'}_{2i+1})^{\delta} = E(z_{2i})E(z_{2i+1})^{\delta} = E(z_{2i-2})E(z_{2i-1})^{u_{2i}-1}E(z_{2i})^{\delta}$. It is clear that $E(z_{2i})^{\delta}$ is the first common part between $E(z_{2i+1})$ and $E(z^{'}_{2i+2})$. In the following, we calculate the largest prefix intersection between $E(z_{2i-1})$ and $E(z_{2i-2})E(z_{2i-1})^{u_{2i}-1}$ in the right of $E(z_{2i})^{\delta}$ (resp. the largest suffix intersection between $E(z_{2i})E(z_{2i-1})$ and $E(z_{2i-2})E(z_{2i-1})^{u_{2i}-1}$ in the left of $E(z_{2i})^{\delta}$).
In the first case, we have:

$$\begin{aligned}
E(z_{2i-1}) &= E(z_{2i-2})^{u_{2i-1}}E(z_{2i-3})\\
&= E(z_{2i-2})^{u_{2i-1}}E(z_{2i-4})^{u_{2i-3}}E(z_{2i-5})\\
&= \cdots\\
&= E(z_{2i-2})^{u_{2i-1}}E(z_{2i-4})^{u_{2i-3}}\cdots E(z_{2i-2k})^{u_{2i-2k+1}}\cdots E(z_2)^{u_3}E(z_1)
\end{aligned}$$

$$\begin{aligned}
E(z_{2i-2})E(z_{2i-1})^{u_{2i}-1} &= E(z_{2i-2})E(z_{2i-1})E(z_{2i-1})^{u_{2i}-2}\\
&= E(z_{2i-2})E(z_{2i-2})^{u_{2i-1}}E(z_{2i-3})E(z_{2i-1})^{u_{2i}-2}\\
&= E(z_{2i-2})^{u_{2i-1}}E(z_{2i-2})E(z_{2i-3})E(z_{2i-1})^{u_{2i}-2}\\
&= \cdots\\
&= E(z_{2i-2})^{u_{2i-1}}E(z_{2i-4})^{u_{2i-3}}\cdots E(z_{2i-2k})^{u_{2i-2k+1}}...E(z_2)^{u_3}E(z_0)\\
&\quad E(z_1)E(z_1)^{u_2}\cdots E(z_{2i-2k-1})^{u_{2i-2k}}E(z_{2i-5})^{u_{2i-4}}E(z_{2i-3})^{u_{2i-2}}\cdots\\
&\quad E(z_{2i-1})^{u_{2i}-2}
\end{aligned}$$

Then their prefix intersection is $E(z_{2i-2})^{u_{2i-1}}E(z_{2i-4})^{u_{2i-3}}...E(z_{2i-2k})^{u_{2i-2k+1}}...$
$E(z_2)^{u_3}E(z_0)^{u_1}$.

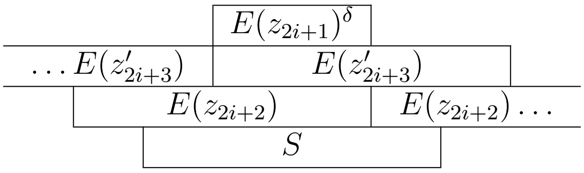
In the second case, we have:

$$\begin{aligned}
E(z_{2i})E(z_{2i-1}) &= E(z_{2i-2})E(z_{2i-1})^{u_{2i}}E(z_{2i-1}) \\
&= E(z_{2i-2})E(z_{2i-1})^2E(z_{2i-1})^{u_{2i}-2}E(z_{2i-1}) \\
&= E(z_{2i-2})E(z_{2i-1})E(z_{2i-2})^{u_{2i-1}}E(z_{2i-3})E(z_{2i-1})^{u_{2i}-1} \\
&= \cdots \\
&= E(z_{2i-2})E(z_{2i-1})E(z_{2i-2})^{u_{2i-1}-1}E(z_{2i-4})^{u_{2i-3}}E(z_{2i-6})^{u_{2i-5}} \\
&\quad \ldots E(z_{2i-2k})^{u_{2i-2k+1}} \cdots E(z_2)^{u_3}E(z_0)E(z_1)E(z_1)^{u_2}E(z_3)^{u_4} \cdots \\
&\quad E(z_{2i-2k-1})^{u_{2i-2k}} \cdots E(z_{2i-3})^{u_{2i-2}}E(z_{2i-1})^{u_{2i}-1}
\end{aligned}$$

$$\begin{aligned}
E(z_{2i-2})E(z_{2i-1})^{u_{2i}-1} &= E(z_{2i-4})E(z_{2i-3})^{u_{2i-2}}E(z_{2i-1})^{u_{2i}-1} \\
&= E(z_{2i-6})E(z_{2i-5})^{u_{2i-4}}E(z_{2i-3})^{u_{2i-2}}E(z_{2i-1})^{u_{2i}-1} \\
&= \cdots \\
&= E(z_0)E(z_1)^{u_2}E(z_3)^{u_4} \cdots E(z_{2i-2k-1})^{u_{2i-2k}}E(z_{2i-3})^{u_{2i-2}} \\
&\quad \cdots E(z_{2i-1})^{u_{2i}-1}
\end{aligned}$$

Then their suffix intersection is $E(z_1)^{u_2}E(z_3)^{u_4}...E(z_{2i-2k-1})^{u_{2i-2k}}...E(z_{2i-3})^{u_{2i-2}}$ $E(z_{2i-1})^{u_{2i}-1}$. □

Proposition 4 *The main connected part S of the intersection between $E(z_{2i+2})$ with $z_{2i+2} = [0, u_1, ..., u_{2i+1}, \delta]$ and $E(z'_{2i+3})$ with $z'_{2i+3} = [0, u_1, ..., u_{2i+1} - 1, 1, \delta]$ is defined as their common part as placed below:*



The word S is exactly $w_1E(z_{2i+1})^{\delta}w_2$, with $w_1 = E(z_1)^{u_2}$ $E(z_3)^{u_4}$ $\cdots$ $E(z_{2i-2k-1})^{u_{2i-2k}}$ $\cdots$ $E(z_{2i-3})^{u_{2i-2}}$ $E(z_{2i-1})^{u_{2i}}$. and $w_2 = E(z_{2i})^{u_{2i+1}-1}E(z_{2i-2})^{u_{2i-1}}E(z_{2i-4})^{u_{2i-3}}$ $\cdots$ $E(z_{2i-2k})^{u_{2i-2k+1}}$ $\cdots$ $E(z_2)^{u_3}E(z_0)^{u_1}$.
We remark that it contains the pattern $E(z_{2i+1}) = E(z'_{2i+1})$ repeated δ times (Figure 4.3 exemplifies the construction of this intersection).

Proof. From (4.4) and (4.5), we have $E(z_{2i+2}) = E(z_{2i})E(z_{2i+1})^{\delta}$ and $E(z'_{2i+3}) = E(z'_{2i+2})^{\delta}E(z'_{2i+1}) = E(z_{2i+1})^{\delta}E(z_{2i})^{u_{2i+1}-1}E(z_{2i-1})$. It is clear that $E(z_{2i+1})^{\delta}$ is the first common part between $E(z_{2i+2})$ and $E(z'_{2i+3})$. In the following, we calculate the largest prefix intersection between $E(z_{2i})E(z_{2i+1})$ and $E(z_{2i})^{u_{2i+1}-1}E(z_{2i-1})$ in the right of $E(z_{2i+1})^{\delta}$ (resp. the largest suffix intersection between $E(z_{2i})$ and $E(z_{2i})^{u_{2i+1}-1}E(z_{2i-1})$ in the left of $E(z_{2i+1})^{\delta}$). In the same way as Proposition 3, we pursue the proof of this proposition. □

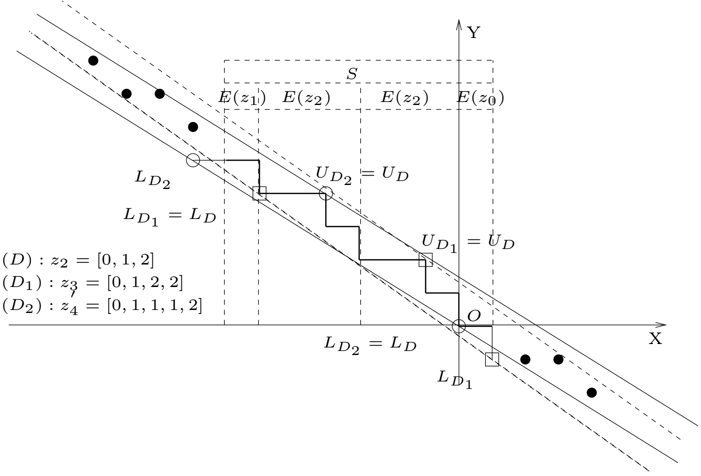


Figure 4.2: Intersection of two patterns $E(z_3)$ and $E(z_4')$, where S is the main connected part of their intersection. The leaning points of D_1 (resp. of D_2) are drawn as boxes (resp. as circles).

Theorem 7 *[?] If the DSL D of even slope $\frac{p_{2i}}{q_{2i}} = [0, u_1, ..., u_{2i}]$ (or of odd slope $\frac{p_{2i+1}}{q_{2i+1}} = [0, u_1, ..., u_{2i+1}]$) is the common part of two standard digital lines D_1 and D_2, then their slopes are:*

	D_1	D_2
D has an even slope	$\frac{\delta p_{2i} + p_{2i-1}}{\delta q_{2i} + q_{2i-1}}$	$\frac{(\delta+1)p_{2i} - p_{2i-1}}{(\delta+1)q_{2i} - q_{2i-1}}$
D has an odd slope	$\frac{\delta p_{2i+1} + p_{2i}}{\delta q_{2i+1} + q_{2i}}$	$\frac{(\delta+1)p_{2i+1} - p_{2i}}{(\delta+1)q_{2i+1} - q_{2i}}$

Definition 7 *Let a and b are two prime numbers, the coefficients of Bézout of the couple (a, b) are the smallest couple of intergers (x, y) such as $ax + by = 1$.*

The writing of $\frac{a}{b}$ in continuous gives the Bezout coefficient of (a, b):

Lemma 3 *The coefficients of Bézout relating to the couple of integers (a, b) are written according to the parity of the complexity of $\frac{a}{b}$, and denoted by $\frac{p_k}{q_k}$ the k-th convergent of $\frac{a}{b}$:*

$$\begin{array}{ll} (q_{2n}, -p_{2n}) & \text{if } \frac{a}{b} \text{ is of complexity } 2n+1 \\ (b - q_{2n-1}, p_{2n-1} - a) & \text{if } \frac{a}{b} \text{ is of complexity } 2n \end{array}$$

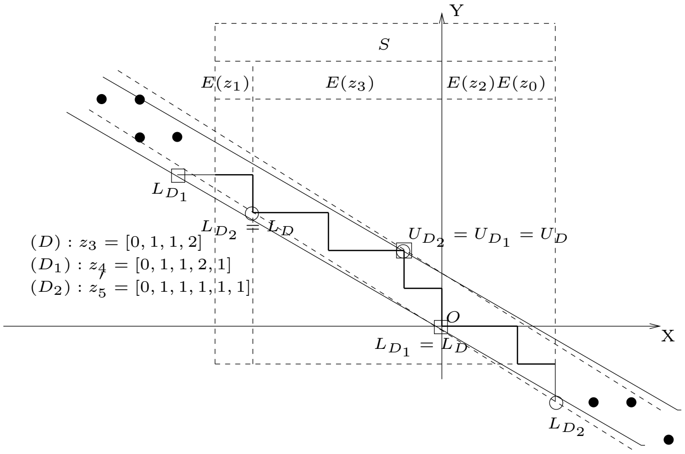

Y
S
$E(z_1)$
$E(z_3)$
$E(z_2)E(z_0)$
L_{D_1}
$L_{D_2} = L_D$
$U_{D_2} = U_{D_1} = U_D$
$(D) : z_3 = [0, 1, 1, 2]$
$(D_1) : z_4 = [0, 1, 1, 2, 1]$
$(D_2) : z_5' = [0, 1, 1, 1, 1, 1]$
O
X
$L_{D_1} = L_D$
L_{D_2}

Figure 4.3: Intersection of two patterns $E(z_4)$ and $E(z_5')$, where S is the main connected part of their intersection. The leaning points of D_1 (resp. of D_2) are drawn as boxes (resp. as circles).

Lemma 4 *On the arithmetic straight lines D_1 and D_2. If $D(a, b, \mu)$ has an even complexity with the remainder $\mu = a(x - x_0) + b(y - y_0)$ where (x, y) is the first point of D and (x_0, y_0) defines the origin of the pixels in $\mathbb{Z}^2$. Then the points of D_1 and D_2 of remainders $\delta\mu + \mu - \delta$ and $\delta\mu$ respectively (If D has an odd complexity, then the points of D_1 and D_2 of remainders $\delta\mu$ and $\delta\mu + \mu - \delta$ respectively) have their coordinates given by (An illustration of this lemma is given in Figure 4.2 and Figure 4.3).*

	D_1	D_2
D has an even slope	$(\mu(b - q_{2i-1}) - \delta q_{2i}, \mu(p_{2i-1} - a) + \delta p_{2i}) + k(-\delta q_{2i} - q_{2i-1}, \delta p_{2i} + p_{2i-1})$	$(\mu(b - q_{2i-1}), \mu(p_{2i-1} - a)) + k(-(\delta + 1)q_{2i} + q_{2i-1}, (\delta + 1)p_{2i} - p_{2i-1})$
D has an odd slope	$(\mu q_{2i}, -\mu p_{2i}) + k(-\delta q_{2i+1} - q_{2i}, \delta p_{2i+1} + p_{2i})$	$(\mu q_{2i} - \delta q_{2i+1}, -\mu p_{2i} + \delta p_{2i+1}) + k(-(\delta + 1)q_{2i+1} + q_{2i}, (\delta + 1)p_{2i+1} - p_{2i})$

Proof.

* We calculate the remainders of those points in D_1:

- Case where D has the complexity $2n$:
$r_{D_1}((\mu(b - q_{2i-1}) - \delta q_{2i}, \mu(p_{2i-1} - a) + \delta p_{2i}) + k(-\delta q_{2i} - q_{2i-1}, \delta p_{2i} + p_{2i-1})) =$
$r_{D_1}((\mu(b - q_{2i-1}) - \delta q_{2i}, \mu(p_{2i-1} - a) + \delta p_{2i})) + 0 = \delta\mu + \mu - \delta$.
- Case where D has the complexity $2n + 1$:
$r_{D_1}((\mu q_{2i}, -\mu p_{2i}) + k(-\delta q_{2i+1} - q_{2i}, \delta p_{2i+1} + p_{2i}) = r_{D_1}((\mu q_{2i},$
$- \mu p_{2i})) + 0 = \delta\mu$.

* We calculate the remainders of those points in D_2:

  - Case where D has the complexity $2n$:
    $r_{D_2}((\mu(b-q_{2i-1}),\mu(p_{2i-1}-a))+k(-(\delta+1)q_{2i}+q_{2i-1},(\delta+1)p_{2i}-p_{2i-1}))=$
    $r_{D_2}((\mu(b-q_{2i-1}),\mu(p_{2i-1}-a)))+0=\delta\mu.$
  - Case where D has the complexity $2n+1$:
    $r_{D_2}((\mu q_{2i}-\delta q_{2i+1},-\mu p_{2i}+\delta p_{2i+1})+k(-(\delta+1)q_{2i+1}+q_{2i},(\delta+1)p_{2i+1}-p_{2i})=$
    $r_{D_2}((\mu q_{2i}-\delta q_{2i+1},-\mu p_{2i}+\delta p_{2i+1}))+0=\delta\mu+\mu-\delta.$

Proposition 5 *Let $D_1(a_1,b_1,\mu_1)$ and $D_2(a_2,b_2,\mu_2)$ be two standard DSL of slopes $\frac{a_1}{b_1}=[0,u_1,...,u_{2i},\delta]$ and $\frac{a_2}{b_2}=[0,u_1,\cdots,u_{2i}-1,1,\delta]$ with $\mu_1=\delta\mu+\mu-\delta$ and $\mu_2=\delta\mu$. Then their main connected part is a DSS of slope z_{2i} with δ patterns and shift μ.*

Proof. Let us denote $L=E(z_1)^{u_2}E(z_3)^{u_4}\cdots E(z_{2i-2k-1})^{u_{2i-2k}}\cdots E(z_{2i-3})^{u_{2i-2}}E(z_{2i-1})^{u_{2i}-1}$ and $R=E(z_{2i-2})^{u_{2i-1}}E(z_{2i-4})^{u_{2i-3}}\cdots E(z_{2i-2k})^{u_{2i-2k+1}}\cdots E(z_2)^{u_3}E(z_0)^{u_1}$ two factors (Left and Right) of the main connected part S of an common intersection of D_1 and D_2. From proposition 3, we further get that R is a left factor of $E(z_{2i-1})$ (resp. L is a right factor of $E(z_{2i})E(z_{2i-1})$). Moreover $E(z_{2i-1})$ is a right factor of $E(z_{2i})$, so we can see that L and R are two factors of $E(z_{2i})$, and the slope of pattern $E(z_{2i})$ does not change when we add L and R to this. Therefore, S is a DSS of slope z_{2i}.

Proposition 6 *Let $D_1(a_1,b_1,\mu_1)$ and $D_2(a_2,b_2,\mu_2)$ be two standard DSL of slopes $\frac{a_1}{b_1}=[0,u_1,...,u_{2i+1},\delta]$ and $\frac{a_2}{b_2}=[0,u_1,...,u_{2i+1}-1,1,\delta]$ with $\mu_1=\delta\mu$ and $\mu_2=\delta\mu+\mu-\delta$. Then their main connected part is a DSS of slope z_{2i+1} with δ patterns and shift μ.*

Proof. The proof of this proposition is similar to the proof of Proposition 5.

4.3 Multiscale of digital lines intersection

We are now in position to study the multiresolution of a DSS defined by the intersection of two DSL, as specified in Proposition 5. We again denote these two DSL by $D_1(a_1,b_1,\mu_1)$ and $D_2(a_2,b_2,\mu_2)$ with $\mu_1=\delta\mu+\mu-\delta$ and $\mu_2=\delta\mu$.

The tiling generated by $\mathbb{S}(h,v)$ 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 obvious relations $X=\left[\frac{x}{h}\right]$ and $Y=\left[\frac{y}{v}\right]$, where $\left[\frac{x}{h}\right]$ is the quotient of the euclidean division of x by h. Furthermore we denote by $\left\{\frac{x}{h}\right\}$ the remainder of this division. An (h,v)-covering of a set of points of $\mathbb{Z}^2$ is the set of tiles of $\mathbb{S}(h,v)$ which intersect it. Let $\Delta_1(a'_1,b'_1,\mu'_1)$ and $\Delta_2(a'_2,b'_2,\mu'_2)$ be the two digital straight lines that are the (h,v)-covering of D_1 and D_2

respectively. Theorem 1 of [?] states that these two lines are standard and gives their arithmetic inequalities:

$$\begin{array}{ll} -p^1 + Q_2^1 - Q_1^1 + SI^1 \leq a_1'X + b_1'Y < Q_3^1 - Q_2^1 + SS^1 & (\Delta_1) \\ -p^2 + Q_2^2 - Q_1^2 + SI^2 \leq a_2'X + b_2'Y < Q_3^2 - Q_2^2 + SS^2 & (\Delta_2) \end{array}$$

where, for $i \in \{1,2\}, p^i = a_i' + b_i'$, $g^i = gcd(a_ih, b_iv)$, $a_i' = \dfrac{a_ih}{g^i}$, $b_i' = \dfrac{b_iv}{g^i}$, $\mu_i = \delta\mu + (2 - i)(\mu - \delta)$,
for $k = 1,2,3$, $Q_k^i = \left[\frac{(k-1)\mu_i + k(a_i+b_i)-1}{g^i}\right]$, $R_k^i = \left\{\frac{(k-1)\mu_i + k(a_i+b_i)-1}{g^i}\right\}$ and

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

To simplify equations, we set $A^i = -p^i + Q_2^i - Q_1^i + SI^i$ and $B^i = Q_3^i - Q_2^i + SS^i$ for $i = 1, 2$. To find their intersection we have thus to solve the following system of equations:

$$\begin{array}{l} A^1 \leq a_1'X + b_1'Y < B^1 \\ A^2 \leq a_2'X + b_2'Y < B^2 \end{array} \tag{4.6}$$

Since a_1' and b_1' are relatively prime, there exist u_1 and v_1 such that $a_1'u_1 + b_1'v_1 = 1$. We introduce $U = \begin{pmatrix} u_1 & -b_1' \\ v_1 & a_1' \end{pmatrix}$ and the change of coordinates $\begin{pmatrix} X' \\ Y' \end{pmatrix} = U^{-1}\begin{pmatrix} X \\ Y \end{pmatrix}$. Thus Equation (4.6) can be rewritten as:

$$\begin{pmatrix} A^1 \\ A^2 \end{pmatrix} \leq \begin{pmatrix} 1 & 0 \\ u_1a_2' + v_1b_2' & a_1'b_2' - a_2'b_1' \end{pmatrix} \begin{pmatrix} X' \\ Y' \end{pmatrix} < \begin{pmatrix} B^1 \\ B^2 \end{pmatrix} \tag{4.7}$$

Let $\lambda_1 = u_1a_2' + v_1b_2'$ and $\lambda_2 = a_1'b_2' - a_2'b_1'$. The solution of the intersection can be computed with a double loop in X' and Y' and formulated as expressions given below.

Theorem 8 *The digital intersection of two digital straight lines Δ_1 and Δ_2 of $S(h,v)$ covering respectively the two digital straight lines D_1 and D_2 of $\mathbb{Z}^2$ is defined by:*

$$A^1 \leq X' < B^1 \tag{4.8}$$

The expression of the boundaries of Y' depends on the sign of λ_2:

$$\lambda_2 > 0, -\left[\frac{-A^2 + \lambda_1X'}{\lambda_2}\right] \leq Y' < -\left[\frac{-B^2 + \lambda_1X'}{\lambda_2}\right] \tag{4.9}$$

$$\lambda_2 < 0, \left[\frac{B^2 - \lambda_1X'}{\lambda_2}\right] + 1 \leq Y' < \left[\frac{A^2 - \lambda_1X'}{\lambda_2}\right] + 1 \tag{4.10}$$

X’	-2		-1			0			1		2			3			4	
Y’	-1	0	-1	0	1	-1	0	1	0	1	0	1	2	0	1	2	1	2
X	-2	-6	1	-3	-7	4	0	-4	3	-1	6	2	-2	9	5	1	8	4
Y	1	4	-1	2	5	-3	0	3	-2	1	-4	-1	2	-6	-3	0	-5	-2

Table 4.1: Points of intersection of $\Delta_1(3, 4, -2)$ and $\Delta_2(3, 5, -3)$

Example. Let $D(2, 3, 2)$ be a standard digital line of slope $\frac{2}{3} = [0, 1, 2]$. For instance, suppose $\delta = 1$ and $(h, v) = (2, 2)$, Proposition **??** gives $D_1(3, 4, 3)$ and $D_2(3, 5, 2)$. Their $(2, 2)$-covering is the two lines $\Delta_1(3, 4, -2)$ and $\Delta_2(3, 5, -3)$. We apply Theorem 8 to determine the set of points of their intersection.
Hence $-2 \leq X' < 5$, since $\lambda_2 = 3 > 0$, then the value of Y' is given by the equation below:

$$-\left[\frac{3 - X'}{3}\right] \;\leq\; Y' \;<\; -\left[\frac{-5 - X'}{3}\right] \tag{4.11}$$

Finally, we have applied the unimodular matrix $U = \begin{pmatrix} 3 & -4 \\ -2 & 3 \end{pmatrix}$ on (X', Y') to get the final result given in the table 4.1 and illustrated on Fig. 4.5.

According to Lemma 4, the coordinates of the first and the last points of the pattern P (i.e. a subset of the main connected part S of the intersection of D_1 and D_2) of D are $(x_f, y_f) = (1, 0)$ and $(x_l, y_l) = (4, -2)$ (see Figure 4.4). Let P' be a covering of P by the tiling $(h, v) = (2, 2)$, i.e. a subset of the main connected part S' of the intersection of Δ_1 and Δ_2 (S' is a covering of S by the same tiling). Therefore, the first and last points of P' are $(X_f, Y_f) = (0, 0)$ and $(X_l, Y_l) = (2, -1)$, and every point of P' is shadowed in the table.

Let Δ be a covering of D by the same tiling. We have calculated the characteristics (a, b, μ) of Δ that is equal to (3,4,-5) by using Theorem 1 of [?]. Finally, using Algorithm *NewSmartDSS* that is explained in chapter 5 to compute the characteristics of P' that is some subset of a DSL Δ, given a starting point (X_f, Y_f) and an ending point (X_l, Y_l) $((X_f, Y_f), (X_l, Y_l) \in \Delta)$, and are equal to $(1, 1, 0)$.

Property 1 *Let* $(a_1, a_2) \in \mathbb{Z} \times \mathbb{Z}$ *and* $b \in \mathbb{Z}^*$ *then:*

$$\left[\frac{a_1}{b}\right] - \left[\frac{a_2}{b}\right] = \begin{cases} \left[\dfrac{a_1 - a_2}{b}\right] & \text{if} \quad R_1 \geq R_2 \\ \left[\dfrac{a_1 - a_2}{b}\right] + 1 & \textit{otherwise} \end{cases} \tag{4.12}$$

Where $R_1 = \left\{\frac{a_1}{b}\right\}$ *and* $R_2 = \left\{\frac{a_2}{b}\right\}$.

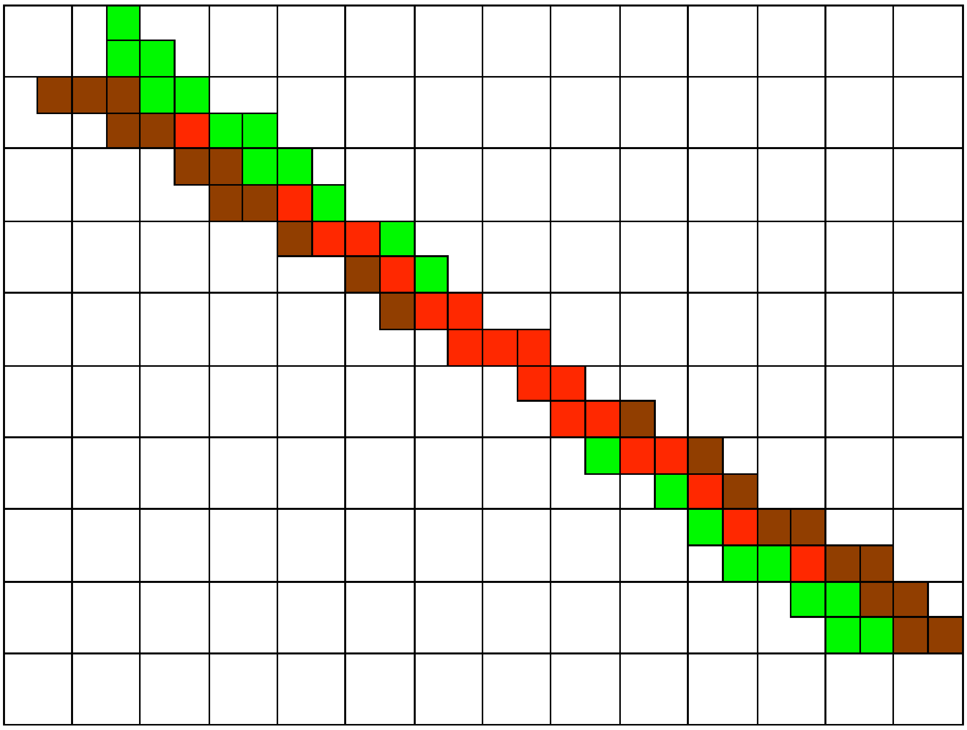


Figure 4.4: Intersection of $D_1(3,4,3)$ drawn as light green boxes and $D_2(3,5,2)$ drawn as brown boxes, their intersection is drawn by red boxes.

Proof. The proof of this proposition is similar to the proof of Proposition 1 and Lemma 1 of [?].

Proposition 7 *Let $n_{IP}(\Delta_1, \Delta_2)$ be a number of intersection points of Δ_1 and Δ_2 and $n_P(X^{'})$ be the number of points who have the same abscissa $X^{'}$. Then,*

$$n_{IP}(\Delta_1, \Delta_2) = \sum_{X^{'}=A^1}^{B^1-1} n_P(X^{'})$$

where,

- *If $\lambda_2 > 0$, then,*

$$n_P(X^{'}) = \begin{cases} \left\lfloor \dfrac{a_2^{'} + b_2^{'}}{\lambda_2} \right\rfloor & \text{if} \quad R_1 \geq R_2 \\ \left\lfloor \dfrac{a_2^{'} + b_2^{'}}{\lambda_2} \right\rfloor + 1 & otherwise \end{cases}$$

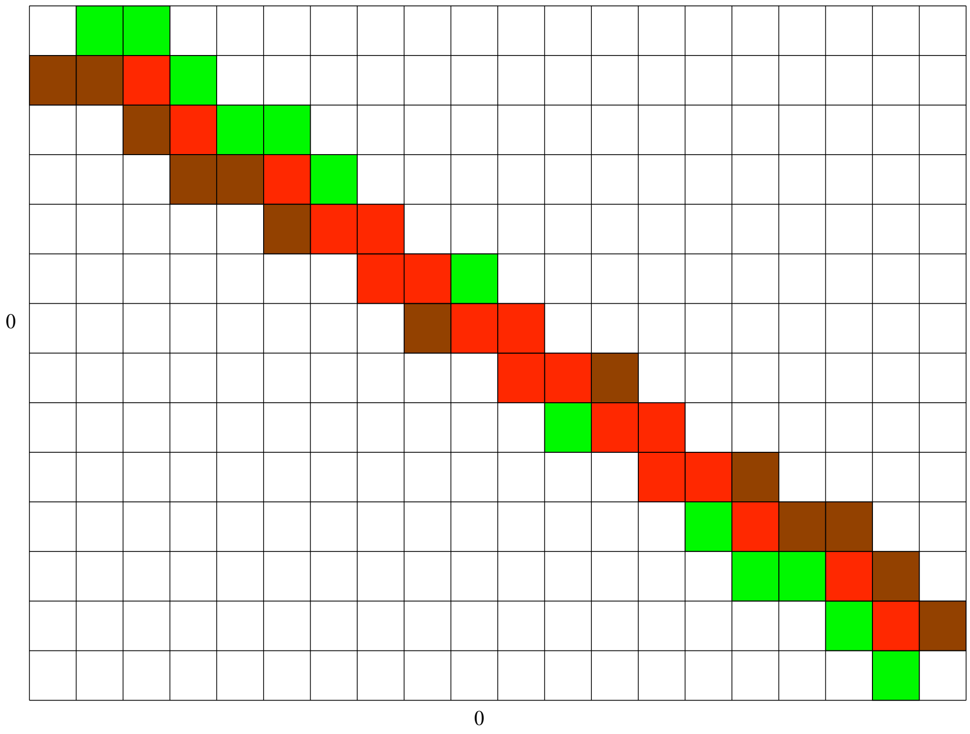


Figure 4.5: Intersection of $\Delta_1(3,4,-2)$ drawn as light green boxes and $\Delta_2(3,5,-3)$ drawn as brown boxes, and their intersection is drawn by red boxes.

- *If* $\lambda_2 < 0$, *then,*

$$n_P(X^{'}) = \begin{cases} \left[\dfrac{-a_2^{'} - b_2^{'}}{\lambda_2}\right] & \text{if} \quad R_1 \geq R_2 \\ \left[\dfrac{-a_2^{'} - b_2^{'}}{\lambda_2}\right] + 1 & \textit{otherwise} \end{cases}$$

with $R_1 = \left\{\dfrac{-A^2 + \lambda_1 X^{'}}{\lambda_2}\right\}$ *and* $R_2 = \left\{\dfrac{-B^2 + \lambda_1 X^{'}}{\lambda_2}\right\}$.

Proof. If $\lambda_2 > 0$, then from Equation (4.9) we get:

$$-\left[\frac{-A^2 + \lambda_1 X'}{\lambda_2}\right] \leq Y' < -\left[\frac{-B^2 + \lambda_1 X'}{\lambda_2}\right]$$

Thus, $n_P(X^{'}) = -\left[\dfrac{-B^2 + \lambda_1 X'}{\lambda_2}\right] + \left[\dfrac{-A^2 + \lambda_1 X'}{\lambda_2}\right]$.

By using property 1, we clearly get:

$$n_P(X^{'}) = \begin{cases} \left[\dfrac{-A^2 + B^2}{\lambda_2}\right] & \text{if} \quad R_1 \geq R_2 \\ \left[\dfrac{-A^2 + B^2}{\lambda_2}\right] + 1 & \text{otherwise} \end{cases}$$

with $R_1 = \left\{\dfrac{-A^2 + \lambda_1 X'}{\lambda_2}\right\}$ and $R_2 = \left\{\dfrac{-B^2 + \lambda_1 X'}{\lambda_2}\right\}$.
Since D_2 is a standard digital line and their covering Δ_2 by the tiling (h, v) is also standard digital line (see Theorem 1 of [?]), then we have: $B^2 - A^2 = a_2^{'} + b_2^{'}$.
It is now easy to see that

$$n_P(X^{'}) = \begin{cases} \left[\dfrac{a_2^{'} + b_2^{'}}{\lambda_2}\right] & \text{if } \ R_1 \geq R_2 \\ \left[\dfrac{a_2^{'} + b_2^{'}}{\lambda_2}\right] + 1 & \text{otherwise} \end{cases}$$

On the other hand, Equation (4.8) indicates that the values of $X^{'}$ varies between A^1 and $B^1 - 1$.
We are now able to calculate the number of intersection points of Δ_1 and Δ_2 wrt the number of points $(X^{'}, Y^{'})$. Thus we get:

$$n_{IP}(\Delta_1, \Delta_2) = \sum_{X'=A^1}^{B^1-1} n_P(X^{'})$$

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

4.4 Conclusion

In this chapter, we have presented new results about the intersection of two standard digital lines. Moreover, we have calculated the coordinates of the first and the last points, all the characteristics (a, b, μ) of these lines and their intersection. From these results we have computed all the characteristics of the (h,v)-covering of these lines by using Theorem 1 of [?]. Finally, the obtained properties can be used to determine all the intersection pixels of P' that is a covering of the initial pattern P by a multiresolution grid.

Chapter 5

Two Sublinear ~~Fast~~ DSS recognition Algorithm when DSL container is known

Contents

5.1 Introduction

In this chapter, We focus our attention on the multiresolution of digital straight segments (DSS), which are finite connected parts of digital lines. Theorems 4, 5 and 6 give~~s~~ us an immediate property on the multiresolution of a DSS.

We present here new algorithms *SmartDSS* and *RSmartDSS* which compute the exact characteristics of a DSS that is a subset of a known DSL, given the endpoints of the DSS. The worst-case computational complexity of the *SmartDSS* algorithm is $\Theta(\sum_{i=0}^{k}$ [illegible] 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. On the other hand, the best case is $\Theta(1)$. Whereas the worst-case computational complexity of the *RSmartDSS* algorithm is $\Theta(k-k')$, where $[u_0; u_1, \ldots, u_{k'}]$ is the continued fraction of the slope of the output DSS. On the other hand, the best case is $\Theta(1)$. This is to compare with classical DSS recognition algorithms (e.g., see [1]), whose complexities are at best $\Theta(n)$. The methods were tested and various experimental results show that *RSmartDSS* algorithm performs better from than the *SmartDSS* algorithm. Furthermore, we have shown Proposition 9 in order to speed up in somes cases Algorithm 3.

Section 5.4 applies the preceding results to the multiscale computation of a digital shape inscribed in some domain $D \times D$. We assume that the digital shape was previously polygonalized as a sequence of M DSS, for instance by a simple greedy DSS segmentation algorithm or with the Minimum Perimeter Polygon [7, 12]. We show that its exact multiresolution can be computed in time linear with MT, where T is the time to process one DSS. The time T in the algorithm *NewSmartDSS* is at best $\Theta(\log D)$ and at worst $\Theta(\log D + \sum_{i=0}^{k} u_i)$, where $[u_0; u_1, \ldots, u_k]$ is the continued fraction of the slope of the output DSS. In most cases, this is clearly sublinear, and at worst, linear in the size of the contour. Whereas The time T in the algorithm *ReversedSmartDSS* is at best $\Theta(\log D)$ and at worst $\Theta(\log D + k - k')$. In most cases, this is clearly sublinear, and at worst, linear in the size of the partial quotients for each slope of the contour. Finally, we have proved Theorem 9, which ensures that the multiresolution of any digital contour is exactly computed. This work is therefore a first step towards the multiscale computation of the tangential cover [2, 6], a fundamental tool for representing and analyzing digital curves.

5.2 A refinement algorithm for computing the multiresolution of a DSS

Corollary 1 *The (h, v)-covering of a standard DSS S of supporting line $D(a, b, \mu)$ and endpoints P, Q is a standard DSS S', which is the 4-connected part of the covering line of D in the tiling $\mathbb{S}(h, v)$ going from $P' = \mathbb{S}(h, v)(P)$ to $Q' = \mathbb{S}(h, v)(Q)$.*

The DSS S' is called the *covering segment in* $\mathbb{S}(h, v)$ of the DSS S. However, we cannot conclude immediately on its minimal characteristics. To do so, we need other tools that are described in the next section.

5.2.1 Number of patterns in the image by the multiresolution

Let S be some DSS included in a line D. We then obtain Proposition 8, which states that the slope of the (h, v)-covering S' of S is some reduced fraction of the slope of the (h, v)-covering D' of D, or the slope of D' itself. Furthermore, we have determined the minimum number of patterns of S starting from which the slope of S' is exactly the slope of D'. If the number of patterns is below this value, we have proposed an output-sensitive algorithmic approach `SmartDSS` which computes all the characteristics of S' in a time sublinear in the number of points of S' (Algorithm 3). All these results are gathered in function `NewSmartDSS` (Algorithm 4).

The following proposition will help us in determining how many patterns are contained in the image S' of S by the multiresolution.

Proposition 8 *v divides $\Delta_k a$ and h divides $\Delta_k b$, for the following values of Δ_k by cases:*

- *$\gcd(h,v) \neq 1$ and $\Delta_k = k\frac{hvg}{g_1g_2g_3g_4}$, for all $k = 1,2,...$ where $g = \gcd(h,v)$, $g_1 = \gcd(a,v)$, $g_2 = \gcd(b,h)$, $g_3 = \gcd(h, \frac{v}{g_1})$ and $g_4 = \gcd(v, \frac{h}{g_2})$.*
- *$\gcd(h,v) = 1$ and $\Delta_k = k\frac{hv}{g_1g_2}$, for all $k = 1,2,....$*
- *$h = v$ and $\Delta_k = kv$, for all $k = 1,2,....$*

proof.

- We assume that $\gcd(h,v) \neq 1$. Therefore we have four cases:

  1. If $\gcd(a,v) = 1$ and $\gcd(b,h) = 1$, then Δ_k is a common multiple of h and v. Since $\gcd(h,v) \neq 1$, then Δ_k is a multiple of the smallest common multiple of h and v. So $\exists k = 1,2,...$ such that $\Delta_k = k\frac{hv}{g}$ where $g = \gcd(h,v)$.
  2. If $\gcd(a,v) \neq 1$ and $\gcd(b,h) = 1$, then Δ_k is a common multiple of h and $\frac{v}{g_1}$, where $g_1 = \gcd(a,v)$. Since $\gcd(h,v) \neq 1$, then $\exists k = 1,2,...$ such that $\Delta_k = k\frac{hv}{g_1}$.
  3. If $\gcd(a,v) = 1$ and $\gcd(b,h) \neq 1$, then Δ_k is a common multiple of v and $\frac{h}{g_2}$, where $g_2 = \gcd(b,h)$. Since $\gcd(h,v) \neq 1$, then $\exists k = 1,2,...$ such that $\Delta_k = k\frac{hv}{g_2}$.
  4. If $\gcd(a,v) \neq 1$ and $\gcd(b,h) \neq 1$, then Δ_k is a common multiple of $\frac{v}{g_1}$ and $\frac{h}{g_2}$. Since $\gcd(h,v) \neq 1$, then $\exists k = 1,2,...$ such that $\Delta_k = k\frac{hv}{g_1g_2}$.

  Therefore, those values of Δ_k can be formulated as a single expression $\Delta_k = k\frac{hvg}{g_1g_2g_3g_4}$, $k = 1,2,....$ □

- We assume now that $\gcd(h,v) = 1$. Therefore we have four cases:

  1. If $\gcd(a,v) = 1$ and $\gcd(b,h) = 1$, then Δ_k is a common multiple of h and v. Since $\gcd(h,v) = 1$, then Δ_k is a multiple of hv. So $\exists k = 1,2,\ldots$ such that $\Delta_k = k\frac{hv}{g_1g_2}$ where $g1 = \gcd(a,v)$ and $g2 = \gcd(b,h)$.
  2. If $\gcd(a,v) \neq 1$ and $\gcd(b,h) = 1$, then Δ_k is a multiple of $\frac{hv}{g_1}$. So $\exists k = 1,2,\ldots.$ such that $\Delta_k = k\frac{hv}{g_1g_2}$.
  3. If $\gcd(a,v) = 1$ and $\gcd(b,h) \neq 1$, then Δ_k is a multiple of $\frac{hv}{g_2}$. So $\exists k = 1,2,\ldots.$ such that $\Delta_k = k\frac{hv}{g_1g_2}$.
  4. If $\gcd(a,v) \neq 1$ and $\gcd(b,h) \neq 1$, then Δ_k is a multiple of $\frac{hv}{g_1g_2}$. So $\exists k = 1,2,\ldots.$ such that $\Delta_k = k\frac{hv}{g_1g_2}$.

Therefore, $\Delta_k = k\frac{hv}{g_1 g_2}$, $k = 1, 2, \ldots$ for all cases. □

The proof of the third statement is analogous to the first one.

We are now in position to determine whenever the slope of S' is the same as the slope of D'. Let δ_{min} be the first positive integer for which v divides $\delta_{min}a$ and h divides $\delta_{min}b$ (i.e. $\delta_{min} = \Delta_1$). We also denote by δ_c the smallest number of patterns for which the slope of S' is equal to the slope of D'.

Proposition 9 *If $\delta \geq 2\delta_{min}$, then the slope of S' is equal to the slope of D'. Otherwise (i.e. $\delta < 2\delta_{min}$), then the slope of S' either is one the ancestors of the slope of D' in the Stern-Brocot tree or is equal to the slope of D'.*

proof. Let S a digital straight segment of slope $\frac{a}{b}$ and S' its covering by tiling (h, v). Suppose $h = v$, and $\delta = 2\delta_{min}$. According to Theorem 4 and Proposition 8, the slope of D' is equal to the slope of S, and $\delta = 2h$, then the number of the vertical (resp. horizontal) step of S is equal to $2ha$ (resp. $2hb$). So the number of the vertical (resp. horizontal) step of S' is equal to $2a$ (resp. $2b$). We have two possibilities:

- If the first point of S' is an upper or lower leaning point, then the slope of S' is equal to the slope of D' and the number of patterns of S' is equal to 2.

- Otherwise, the slope of the subset before the first upper or lower leaning point of S' is one of the ancestors of the slope of S' in the Stern-Brocot tree. Therefore, the slope of S' is equal to the slope of D' and the number of patterns of S' is equal to 1.

The proofs of the other cases are analogous to the first one and exploits the other cases of Proposition 8.

This is an immediate corollary.

Corollaire 2 *Assume S contains $\delta_c + n\delta_{min}$ patterns in D, then the pattern of S' is repeated $n + 1$ times.*

Conjecture 1 *Let μ_c a critical value of lower bound. Suppose $\delta = \delta_c$ in the case $\mu = \mu_c$. Similarly $\delta = \delta_c$ for all the following cases:*

- *$(h = v)$ and $(\mu = \mu_c + nv)$, where $n \in \mathbb{Z}$.*

- *$\gcd(h, v) = 1$ and $(\mu = \mu_c + nhv)$.*

- *$\gcd(h, v) \neq 1$ and $(\mu = \mu_c + n\frac{hv}{g})$, where $g = \gcd(h, v)$.*

5.2.2 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 3). 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 4 and 5). Therefore most of the points tested in **DR95** are useless since they do not lead to any slope evolution. Proposition 9 tells us that we must find the slope of S' in the ancestors of the slope of D' when the number of patterns of S' is less than the $2\delta_{min}$. We must also determine the shift to origin of S'.

Starting from the initial correct quadrant, Algorithm 3 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 function `Bezout` (Algorithm 1), assuming a computation model where standard arithmetic operations are in $O(1)$.[1]

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 decrease (line 5). This is effectively implemented in $O(1)$ time with Algorithm 2. 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 5.1. 11 points have been tested compared with a DSS length of 23. We show below that the worst-case time complexity of Algorithm 3 is related to the partial quotients of the DSS slope and not to the slope itself. We say that a DSS is *primitive* whenever it contains one pattern of its slope or one reversed pattern of its slope.

Proposition 10 *Let S be a primitive DSS of slope $\frac{a}{b} = [u_0, u_1, ..., u_k]$. We denote by $T(S)$ the number of points on S tested by Algorithm 3 to recognize its slope $\frac{a}{b}$. If $\sum_{i=0}^{k} u_i = n$, then $T(S) \leq 2n$ (it only depends on the sum of the partial quotients).*

proof. We prove by induction on n that $T(S) \leq 2n$. The initial state $T(S) = 0$ is obvious since S is just a one step horizontal segment and has slope 0. Algorithm 3 exits immediately of the loop at line 2 since tests $U'_y \leq Q_y$ and $L'_x \leq Q_x$ are false. Since P and the quadrant is known, the output is correct while no test "$\in S$" has been made. Assume that $T(S) \leq 2n$ for all primitive DSS S of slope $[u_0, u_1, ..., u_k]$ with $\sum_{i=0}^{k} u_i = n$. We shall prove that for any primitive DSS S' with a sum $n+1$, we have $T(S') \leq 2n+2$.

[1] A reasonnable assumption since all integers are bounded by D, the size of the image domain.

Let S' be such a DSS, with slope $a'/b' = [u'_0, u'_1, ..., u'_{k'}]$ and sum $n+1$. The recognition process (Algorithm 3) is incremental, and corresponds exactly to a progressive descent in the Stern-Brocot tree. Therefore, the recognition process visits the father a/b of the node a'/b' at some point. According to Stern-Brocot tree and Proposition 2, there are only two possible evolutions for a slope of a DSS, either $[u_0, u_1, ..., u_k, \delta]$ or $[u_0, u_1, ..., u_k - 1, 1, \delta]$. Therefore, we have two cases for the partial quotients of $a/b = [u_0, u_1, ..., u_k]$:

- $k = k' - 1$, $\forall i = 0 \ldots k, u_i = u'_i$, $u'_{k'} = \delta$.

- $k = k' - 2$, $\forall i = 0 \ldots k, u_i = u'_i$, $u'_{k'-1} = 1$, $u'_{k'} = \delta$.

In both cases, $\sum_{i=0}^{k} u_i = n + 1 - \delta$, $\delta \geq 1$. Therefore, this first subpart of S' (called S hereafter) is itself a DSS with slope a/b and sum $\leq n$. This subpart of S' is also primitive, even if S itself contains δ repetitions of this pattern. We can apply the inductive hypothesis and we obtain $T(S) \leq 2(n+1-\delta)$. At each iteration of the loop starting line 2, zero, one or two points are tested for their inclusion in D. The number of iterations is exactly δ. It follows that at most 2δ points are tested between the recognition of S and the recognition of the slope of S'. Furthermore, once this slope is recognized, no other point is tested since S' was assumed primitive.

Then, $T(S') \leq T(S) + 2\delta \leq 2(n + 1 - \delta) + 2\delta = 2n + 2$.

The following proposition naturally extends to arbitrary finite DSS the previous proposition and its proof is similar.

Proposition 11 *Let S be a DSS of slope $\frac{a}{b} = [u_0, u_1, ..., u_k]$, containing δ patterns $\frac{a}{b}$ (or reversed patterns). If $\sum_{i=0}^{k} u_i = n$, then $T(S) \leq 2n + 2\delta$.*

We sum up the previous results in function `NewSmartDSS` (Algorithm 4), whose input is a DSS $S(a, b, \mu)$, the endpoints (P, Q) of S and the tiling (h, v). First, the line D has the same characterisitics as S, and we compute the characteristics of D' with Theorem 4. We then calculate the positions of the first and the last points of S' as the images of the endpoints P and Q by the tiling (h, v) (line 1,2). According to Proposition 9, we have computed the first value of Δ (δ_{min}) where v divides Δa and h divides Δb (line 3). It remains to compute the exact characteristics of S'. If the number of patterns or the number of reversed patterns in S is greater than $2\delta_{min}$, then the slope of S' is equal to the slope of D' (line 4). Otherwise, we call function `SmartDSS` (line 5).

An execution of this algorithm is illustrated on Figure 5.2. 3 points have been tested compared with a contour length of 30. For example, on S'_0, since the number of patterns (equal to 5) is greater than $2\delta_{min}$ (δ_{min} is equal to 2), then the slope of S'_0 is equal to the slope of D'_0 and the number of tested points of this segment is equal to 0.

Algorithm 1: Computes in $O(1)$ the Bézout coefficients (b', a') of (p_k, q_k), i.e. $p_k b' - q_k a' = 1$, using the partial quotients $\frac{p_i}{q_i}$ of $\frac{p_k}{q_k}$.

```
Function Bezout( In $p$, $q$, $k$ ) : $(b', a')$ ;
$p, q$ : array of integer /* partial quotients */ ;
$k$ : integer /* depth of continued fraction */ ;
begin
    if $k$ is even then
        $b' \leftarrow q_k - q_{k-1}$ ;
        $a' \leftarrow p_k - p_{k-1}$ ;
    else
        $b' \leftarrow q_{k-1}$ ;
        $a' \leftarrow p_{k-1}$ ;
end
```

Algorithm 2: 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.

```
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 3: 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 $\frac{a}{b} = [u_0, \ldots, u_k]$.

```
Action SmartDSS( In $D$ : DSL $(\alpha, \beta, \mu')$, In $P, Q$ : Points of $\mathbb{Z}^2$,
                 Out $S$: DSS $(a, b, \mu)$ ) ;
Var $u, p, q$ : array of integers /* Cont. frac. $\frac{a}{b} = [u_0, \ldots, u_k] = \frac{p_k}{q_k}$ */ ;
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       $(b', a') \leftarrow$ Bezout$(p, q, k)$ /* $ab' - ba' = 1$ */ ;
        $U' \leftarrow U + (b - b', a - a')$;
        $L' \leftarrow L + (b', a')$;
        L $\delta \leftarrow 1$, $loop \leftarrow 0$;
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
```

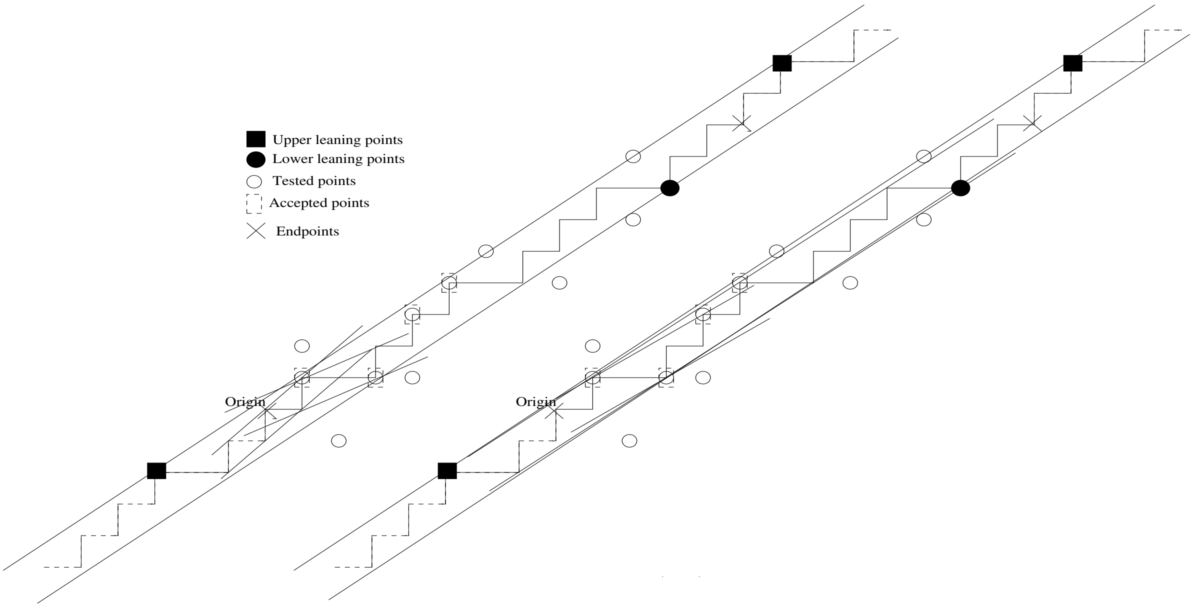


Figure 5.1: A digital straight line $D(13, 17, -5)$ with an odd slope. Computes the characteristics (a, b, μ) of a DSS S that is the subset of D between the origin and the point $(12, 9)$. The intermediate slopes are drawn with solid lines on the left and on the right, tested points are circled.

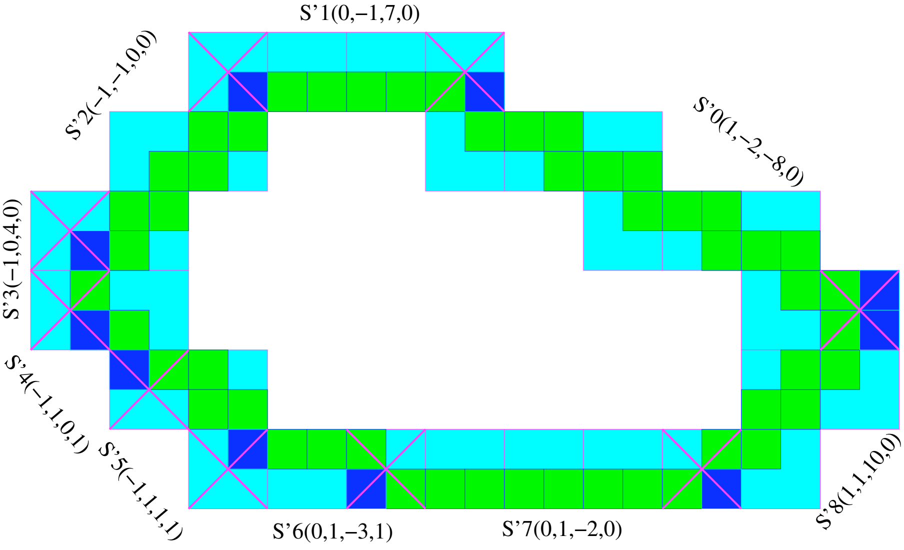


Figure 5.2: Multiscale computation of the boundary of a digital shape is given in green color, having 60 points, according to the tiling $(h, v) = (2, 2)$. The dark boxes and crosses respectively represent the endpoints of the DSS of the initial contour and of its covering.

Algorithm 4: Computes the characteristics (a', b', μ') of a DSS S' that is some subset of a DSL D', given a starting point P' and an ending point Q' $(P', Q' \in D')$.

```
    Action NewSmartDSS( In $S$, In $P, Q, h, v$, Out $S'$ ) ;
    $S$ : DSS $(a, b, \mu)$, $P, Q$ : Point of $\mathbb{Z}^2$ , $S'$ : DSS $(a', b', \mu')$ ;
    Var $\delta$ : integer /* Number of (reversed) patterns in S */ ;
    Var $\delta_{min}$ : integer /* The first delta where $v$ divides $\Delta a$ and $h$ divides $\Delta b$ */ ;
    Var $\alpha, \beta, \mu''$ : integer /* $D'(\alpha, \beta, \mu'')$ is the covering of $D$ by tiling $(h, v)$ */ ;
    Var $P', Q'$ : integers /* The endpoints of $S'$ */ ;
    begin
        $D \leftarrow DSL(S)$ /* $D(a, b, \mu)$ */ ;
        MultiScale$(D, h, v, D')$ /* Theorem 4 */ ;
1       $P'_x \leftarrow \frac{P_x}{h}$, $P'_y \leftarrow \frac{P_y}{v}$ ;
2       $Q'_x \leftarrow \frac{Q_x}{h}$, $Q'_y \leftarrow \frac{Q_y}{v}$ ;
3       $\delta_{min} \leftarrow$ deltamin $(a, b, h, v)$ /* Proposition 8 */ ;
        if $\delta \geq 2\delta_{min}$ then
4           $S' \leftarrow D'$; /* The slope of $S'$ is equal to the slope of $D'$ */
        else
5           SmartDSS$(D', P', Q', S')$;
    end
```

5.3 A coarsening algorithm for computing the multiresolution of a DSS

In this section, we ~~have~~ propose~~d~~ algorithm~~s~~ to find the characteristics (a', b', μ') of the DSS S' in ~~the~~ second quadrant, that is the covering of S by the tiling $\mathbb{S}(h, v)$. We have used for ~~the~~ first step the same framework ~~that~~ presented in the previous section (for example line 1,2 of algorithm 9), and ~~according to~~ Proposition 9 which states that the slope of S' is either one of the ancestors of S or equal to the slope it self. So we pursue the method bottom-up way of Stern-Brocot Tree and called *Reversed Smart DSS*.

Algorithm 9 is the general algorithm to compute the characterisitcs (a', b', μ') of a DSS S', that is some subset of a DSL $D'(\alpha', \beta')$ (D' is the covering of D by the tiling (h, v) and S is some subset of D). After the computation of the characterisitcs of D', the endpoints A' and B' of S', by using Theorem 4 (line 1,2), we compute the lower leaning points L_1 before A' and L_2 after B', these values are given in O(1) time by the function *FirstLowerBound* (line 3) and *SecondLowerBound* (line 4) (see Figure 5.3 for more details). It remains to [illegible] the function **RSmartDSS** (Algorithm 8) that returns the characterisitcs (a', b', μ') of S' corresponding to the bottom-up way of the Stern-Brocot Tree.

Algorithm 8 then checks ~~firstly~~ the horizontal distance between L_1 and L_2. However, if this distance is (1) three times greathter than β', (2) egal to two times β', A' and L_1 are confused, or B' and L_2 are confused, or (3) A' and L_1 are confused, and B' and L_2 are confused. Then the algorithm stops and returns ~~(α', β')~~ (line 1,2). Otherwise, if the horizontal distance is egal to two times β', then we apply the function **TwiceSlope** (line 3). This is effectively implemented in $O(n - n')$ time with Algorithm 7 where n is the size of the partial quotients of the input slope $\frac{\alpha'}{\beta'} = [u_0, u_1, \ldots, u_n]$, and n' is the size of the partial quotients of the ouput slope $\frac{a'}{b'} = [u_0, u_1, \ldots, u_{n'}]$. Otherwise, we use the method of reduced slope, it means that L_1 is ~~moved~~ towards A' and L_2 towards B', corresponding to the parity and the previous slope of slope (line 4,5). If L_1 and L_2 do not change, we thus execute the final condition (line 6) that returns (a', b', μ') of S'. At each step, when we change the value of L_1 ~~and L_2~~, we compute the new values of (α', β') and so on until this algorithm is stopped.

Algorithm 6 ~~computes in~~ $O(1)$ the characteristics of the lower bound in the left (or right according to the action $Left$) of L_2. In this algorithm, we fix L_2, move L_1 right towards A' (or L_3 ~~left towards~~ B' in the other case) (line 1,2) ~~and~~ calculate the value of the new slope between new L_1 and L_2 (or L_2 ~~and new~~ L_3).

Algorithm 5 determines in $O(1)$ the positions of the new upper leaning points LU and RU, according to the parity of slope (line 1,3). ~~Either~~ we have moved LU and RU towards the left in the odd case~~s~~ or towards the right in the even case~~s~~. But before moving, we must calculate the number of subpatterns, that is covering the point A' in the even case (line 4) or B' in the odd case (line 2).

Algorithm 7 returns the characteristics (a', b', μ') of a DSS $A'B'$ that is some subset of a DSS of pattern (α', β') repeated twice. From the two lower leaning points L_1 and L_3, we find its middle lower leaning point L_2 (line 1). We have calculated the first and the second upper leaning points (line 1,2) (~~two~~ patterns have three lower and two upper leaning points). We can start an execution at the same time for three algorithms, Algorithm 6 computes the left slope and stops when the lower leaning point L_1 overtakes or reaches A', Algorithm 6 computes the right slope and stops when the lower leaning point L_3 overtakes or reaches B', and Algorithm 5 computes the upper slope and stops when the upper point RU overtakes towards the left or reaches B' and the upper point LU overtakes towards the right or reaches A'. In this case when one of the three algorithms is stopped, we thus calculate the deepest slope and compare its value and the slope of the algorithm stopped. If these values are equals, then ~~Algorithm~~ 7 stopped and returns the characteristics (a', b', μ'). Otherwise, we ~~have continued~~ with the other algorithms until all these algorithms are stopped.

Example. Let us look at a run of Algorithm 9 for a digital straight segment S of slope $\frac{13}{18} = [0, 1, 2, 1, 1, 2]$ illustrated in the left of Figure 5.3, such that A and B are the endpoints of S. For instance, suppose $\delta = 1$ and $(h, v) = (2, 2)$, then its

(2,2)-covering is illustrated as S' in the right, where A' and B' are the endpoints of S' and also the (2,2)-covering respectively of A and B. First, as the slope of S' either is one the ancestors of the slope of D' in the Stern-Brocot tree or is equal to the slope of D' (D' is (2,2)-covering of D and has the same slope as D), ~~therefore~~ we must calculate the lower leaning points L_1 before A', L_3 after B' and L_2 between A' and B' (In this example, the slope $\frac{13}{18}$ is repeated twice, it means that S' is included in two patterns $\frac{13}{18}$). Since the condition on line 3 of Algorithm 8 is fulfilled, ~~therefore~~ we have to compute the characteristics (a', b', μ') of S' by using Algorithm 7. For the first step, we apply left lower slope (call Algorithm 6 (line 3) and returns the slope $\frac{3}{4}$) (see Figure 5.5), right lower slope (call Algorithm 6 (line 4) and returns the slope $\frac{5}{7}$) (see Figure 5.6), and upper slope (call Algorithm 5 (line 5) and returns the slope $\frac{5}{7}$) (see Figure 5.4). As Algorithm 5 is stopped, then we compute the deepest slope of $\frac{3}{4}$, $\frac{5}{7}$ and $\frac{5}{7}$ that is equal to $\frac{5}{7}$. And as the value of the deepest slope is equal to the value of the upper slope, then Algorithm 7 stops and returns the final result $(5, 7, 6)$.

Let us now give some explanations of Figure 5.5. In the first step, we fix L_{11} at L_1 and L_{22} at L_2, and we compute the previous slope $PS = \frac{5}{7}$ of $S = \frac{13}{18}$. Since the parity of S is odd, ~~then~~ we calculate the number k of subpatterns $\frac{5}{7}$ that is covering A' from L_{11} to the right, such that this covering reach or overtakes A'. It is impossible to take $k = 3$, because in this case this displacement from L_{11} overtakes L_{22}. So we take $k = 2$, and L_{11} is moved two times subpatterns $\frac{5}{7}$ toward the right. We have now a new DSS $S(3, 4)$ from new L_{11} to L_{22}. In the second step, we calculate the previous slope $PS = \frac{1}{1}$ of $S = \frac{3}{4}$. Since the parity of S is even, then we calculate the number k of subpatterns $\frac{1}{1}$ that is covering A' from L_{22} to the left, such that this covering reach or overtakes A'. It is impossible to take $k = 4$, because in this case this displacement from L_{22} overtakes L_{11}. So we take $k = 3$, and L_{11} is moved three times subpatterns $\frac{1}{1}$ toward the left and we obtain a new DSS from the new L_{11} to L_{22}. As L_{11} reach A', therefore this algorithms stops and returns the left slope $\frac{1}{1}$.

It is easy to see ~~that~~ all the operations of ~~these~~ four algorithms can be executed in linear time in the [illegible] of ~~the~~ partial quotient of continued fraction. The worst-case computational complexity is $O(n - n')$, where $[u_0, u_1, ..., u_n]$ is the continued ~~fraction~~ of the input DSL and $[u_0, u_1, ..., u_{n'}]$ is the continued fraction of the output DSL.

Algorithm 5: Computes in O(1) the Upper characteristics (a'_1, b'_1) of a DSS that is some subset of a DSL D' (U_1 and U_2 are two upper leaning points of D'), given a starting point A' and an ending point B' ($A', B' \in D'$).

```
Function UpperSlope( In $U_1, U_2$ : Upper bound of $D'$ : DSL $(\alpha', \beta')$, In $A', B'$ :
Points of $\mathbb{Z}^2$) : $(a'_1, b'_1)$ ;
Var $LU, RU$ : Point of $\mathbb{Z}^2$ /* Left and Right upper leaning points */ ;
Var $S, PS$ : Point of $\mathbb{Z}^2$ /*  PS the previous slope of S  */ ;
Var $k$ : integer /* Number of subpatterns covering A' or B' */ ;
Var $parity$ : boolean /* parity of slope continued fraction */ ;
begin
    $LU \leftarrow U_1$, $RU \leftarrow U_2$ ;
    if ($LU > A'$ and $RU < B'$) then
        return $(\alpha', \beta')$ ;
    $S \leftarrow (\alpha', \beta')$ ;
    $parity \leftarrow$Parity($S$) ;
    $PS \leftarrow$PreviousSlope($S$) ;
    if ($parity$ is odd) then
        if ($RU > B'$) then
            $k \leftarrow$NumberOfCoveringSubPatterns($RU, B', PS, true, false$);
            $RU \leftarrow RU - k(PS_y, -PS_x)$ ;
        if ($LU < A'$) then
            $LU \leftarrow RU - (PS_y, -PS_x)$ ;
    else
        if ($LU < A'$) then
            $k \leftarrow$NumberOfCoveringSubPatterns($LU, A', PS, true, false$);
            $LU \leftarrow LU - k(-PS_y, PS_x)$ ;
        if ($RU > B'$) then
            $RU \leftarrow LU - (-PS_y, PS_x)$ ;
    $(\alpha', \beta') \leftarrow (LU_y - RU_y, RU_x - LU_x)$ ;
    return $(\alpha', \beta')$ ;
end
```

Algorithm 6: Computes in O(1) the Lower (Left or Right) characteristics (a'_2, b'_2) of a DSS that is some subset of a DSL D', given a starting point A' and an ending point L_2 (Left part) or given a starting point L_2 and an ending point B' (Right part) $(A', B' \in D')$ (Left pattern is L_1L_2 and Right pattern is L_2L_3).

```
Function LowerSlope( In $L_1, L_2, L_3$ : Lower bound of $D'$ : DSL $(\alpha', \beta')$, In $X(A'$ or $B')$ : Point of $\mathbb{Z}^2$, Left : Boolean): $(a'_2, b'_2)$ ;
Var $P, L_{11}, L_{22}, S, PS$ : Point of $\mathbb{Z}^2$ /* $L_{11}$ and $L_{22}$ two lower leaning points */ ;
Var $k$ : integer ;
Var $parity$ : boolean ;
begin
    $S \leftarrow (\alpha', \beta')$ ;
    $P \leftarrow$Left ? $L_1$ : $L_3$ ;
    $L_{11} \leftarrow$Left ? $L_1$ : $L_2$ ;
    $L_{22} \leftarrow$Left ? $L_2$ : $L_3$ ;
    $parity \leftarrow$Parity$(S)$ ;
    $PS \leftarrow$PreviousSlope$(S)$ ;
    if ($parity$ is odd) then
        $k \leftarrow$NumberOfCoveringSubPatterns$(L_{11}, X, PS, true, false)$;
        $P \leftarrow L_{11} - k(-PS_y, PS_x)$ ;
    else
        $k \leftarrow$NumberOfCoveringSubPatterns$(L_{22}, X, PS, true, false)$;
        $P \leftarrow L_{22} - k(PS_y, -PS_x)$;
    if Left then $L_{11} = P$ ;
    else $L_{22} = P$ ;
    $(\alpha', \beta') \leftarrow (|P_y - L_{2_y}|, |L_{2_x} - P_x|)$ ;
    $L_1 \leftarrow$Left ? $L_{11}$ : $L_1$ ;
    $L_3 \leftarrow$Left ? $L_3$ : $L_{22}$ ;
    return $(\alpha', \beta')$ ;
end
```

Algorithm 7: Computes in O(n-n') the characteristics (a', b', μ') of a DSS S' with $\frac{a'}{b'} = [u_0, ..., u_{n'}]$, that is some subset of a DSL D' with $\frac{\alpha'}{\beta'} = [u_0, ..., u_n]$ repeated twice, given a starting point A' and an ending point B' ($A', B' \in D'$).

```
Function TwiceSlope( In $L_1, L_3$ : Lower bound of $D'$ : DSL $(\alpha', \beta')$, In $A', B'$ :
Point of $\mathbb{Z}^2$) : $S'$ DSS $(a', b', \mu')$;
Var $U_1, U_2, L_2, S, S_1, S_2, S_3, DS$ : Point of $\mathbb{Z}^2$; L, R, U : boolean;
Var $D'_1, D'_2, D'_3$ : DSL $(\alpha', \beta')$, CF : Continued Fraction of $S$;
begin
    $S \leftarrow (\alpha', \beta')$, $L \leftarrow$true, $R \leftarrow$true, $U \leftarrow$true, i $\leftarrow$0 ;
    $L_2 \leftarrow$SecondLowerBound$(L_1, S)$, $U_1 \leftarrow$FirstUpperBound$(D', L_1, L_2)$;
    $U_2 \leftarrow$SecondUpperBound$(U_1, S)$, $CF \leftarrow$ContinuedFraction$(S)$ ;
    while $i < /CF/$ do
        if $(L$ and $R$ and $U)$ then
            $S_1 \leftarrow$LowerSlope$(D'_1, L_1, L_2, L_3, A', true)$ /* Left Lower Slope */;
            $S_2 \leftarrow$LowerSlope$(D'_2, L_1, L_2, L_3, B', false)$ /* Right Lower Slope */;
            $S_3 \leftarrow$UpperSlope$(D'_3, U_1, U_2, A', B')$ /* Upper Slope */;
            if $(L_1 >= A'$ or $L_3 <= B'$ or $(U_1 >= A$ and $U_2 <= B))$ then
                $DS \leftarrow$DeepestSlope( $S_1$ , $S_2$ , $S_3$ )/* DS The deepest slope */;
            if $L_1 >= A'$ then $L \leftarrow$false; if $L_3 <= B'$ then $R \leftarrow$false;
            if $(U_1 >= A$ and $U_2 <= B)$ then $U \leftarrow$false;
            if $(L_1 >= A'$ and $DS == S_1)$ or $(L_3 <= B'$ and $DS == S_2)$ or
            $(U_1 >= A$ and $U_2 <= B$ and $DS == S_3)$ then break;
        else if $((L$ and $R)$ or $(L$ and $U)$ or $(R$ and $U))$ then
            $S_1 \leftarrow(R$ and $U)$ ? LowerSlope$(D'_2, L_1, L_2, L_3, B', false)$ :
            LowerSlope$(D'_1, L_1, L_2, L_3, A', true)$ ;
            $S_2 \leftarrow(L$ and $R)$ ? LowerSlope$(D'_2, L_1, L_2, L_3, B', false)$ :
            UpperSlope$(D'_3, U_1, U_2, A', B')$;
            if $((((L$ and $U)$ or $(L$ and $R))$ and $(L_1 >= A'))$ or $(((R$ and $U)$ or $(L$
            and $R))$ and $(L_3 <= B'))$ or $(((R$ and $U)$ or $(L$ and $U))$ and $(U_1 >= A'$
            and $U_2 <= B')))$ then $DS \leftarrow$DeepestSlope( $S_1$ , $S_2$ );
            if $(((R$ and $U)$ and $((L_3 <= B'$ and $DS == S_1)$ or $(U_1 >= A'$ and
            $U_2 <= B'$ and $DS == S_2)))$ or $((L$ and $U)$ and $((L_1 >= A'$ and
            $DS == S_1)$ or $(U1 >= A'$ and $U_2 <= B'$ and $DS == S_2)))$ or $((L$ and
            $R)$ and $((L_1 >= A'$ and $DS == S_1)$ or $(L_3 <= B'$ and $DS == S_2))))$
            then break;
            if $(((L$ and $U)$ or $(L$ and $R))$ and $(L_1 >= A'))$ then $L \leftarrow$false;
            if $(((R$ and $U)$ or $(L$ and $R))$ and $(L_3 <= B'))$ then $R \leftarrow$false;
            if $(((R$ and $U)$ or $(L$ and $U))$ and $(U_1 >= A'$ and $U_2 <= B'))$ then
                $U \leftarrow$false;
        else
            if $L$ then $DS \leftarrow$LowerSlope$(D'_1, L_1, L_2, L_3, A', true)$;
            else if $R$ then $DS \leftarrow$LowerSlope$(D'_2, L_1, L_2, L_3, B', false)$;
            else $DS \leftarrow$UpperSlope$(D'_3, U_1, U_2, A', B')$;
            if $((L$ and $L_1 >= A')$ or$(R$ and $L_3 <= B')$ or $(U$ and $U_1 >= A'$ and
            $U_2 <= B'))$ then break;
    $a' \leftarrow DS_x$, $b' \leftarrow DS_y$, $\mu' \leftarrow a' L_{2_x} + b' L_{2_y}$;
end
```

Algorithm 8: Computes in O(n-n') the characteristics (a', b', μ') of a DSS S' that is some subset of a DSL D', given a starting point A' and an ending point B' $(A', B' \in D')$.

```
Function RSmartDSS( In $L_1, L_2$ : Lower bound of $D'$ : DSL $(\alpha, \beta, \mu'')$, In $A', B'$ :
Points of $\mathbb{Z}^2$) : $S'$: DSS $(a', b', \mu')$ ;
Var $Lp_1, Lp_2, L, V_1, V_2, S, PS, PPS$ : Point of $\mathbb{Z}^2$ ;
Var $k, k_1, k_2, gcd, dL, X, Y, \alpha', \beta'$ : integer ;
Var $parity, covering_A, covering_B$ : boolean ;
begin
      if $(A'_x == B'_x)$ then return $(1, 0, A'_x)$;
      if $(A'_y == B'_y)$ then return $(0, 1, A'_y)$;
      $\alpha' \leftarrow \alpha$, $\alpha' \leftarrow \beta$ /* $D'' \leftarrow (\alpha', \beta')$ */ ;
      $dL \leftarrow L_{2_x} - L_{1_x}$;
1     if $(dL \geq 3\beta')$ or $(dL == 2\beta'$ and $(A' == L_1$ or $B' == L_2))$ or $(A' == L_1$ and
2     $B' == L_2)$ then return $(\alpha', \beta', \alpha' L_{1_x} + \beta' L_{1_y})$ ;
3     if $(dL == 2\beta')$ then return TwiceSlope$(D', A', B', L_1, L_2)$;
      $parity \leftarrow$Parity$(S \leftarrow (\alpha', \beta'))$, $PS \leftarrow$PreviousSlope$(S)$ ;
      $L \leftarrow parity$ ? $L_1$ : $L_2$, $covering_{A'} \leftarrow parity$ ? $true$ : $false$;
      $covering_{B'} \leftarrow parity$ ? $false$ : $true$;
      $k_1 \leftarrow$NumberOfCoveringSubPatterns$(L, A', PS, covering_{A'}, true)$;
      $k_2 \leftarrow$NumberOfCoveringSubPatterns$(L, B', PS, covering_{B'}, true)$;
4     if $(parity$ is odd$)$ then
            $Lp_1 \leftarrow L_1 - k_1(-PS_y, PS_x)$, $V_2 \leftarrow L_1 - k_2(-PS_y, PS_x)$;
            $Lp_2 \leftarrow (V_2 \leq L_2)$ ? $V_2$ : $L_2$ ;
5     else
            $Lp_2 \leftarrow L_2 - k_2(PS_y, -PS_x)$, $V_1 \leftarrow L_2 - k_1(PS_y, -PS_x)$;
            $Lp_1 \leftarrow (V_1 \geq L_1)$ ? $V_1$ : $L_1$ ;
      $k \leftarrow$( parity is odd and $Lp_2 != L_2$) or ( parity is even and $Lp_1 != L_1$) ?
      $(Lp_{2_x} - Lp_{1_x})/PS_y$ : 1 ;
      $S \leftarrow ((Lp_{1_y} - Lp_{2_y})/k, (Lp_{2_x} - Lp_{1_x})/k)$;
6     if $(Lp_1 == L_1)$ and $(Lp_2 == L_2)$ then
            PreviousPreviousSlope( $S, PS, PPS$ ), parity = Parity(S);
            if $(Lp_2 == B')$ then
                  $Lp_1 \leftarrow Lp_1$ - ( $(parity)$ ? $(-PS_y, PS_x)$ : $(-PPS_y, PPS_x)$ );
                  $(a', b') \leftarrow (Lp_{1_y} - Lp_{2_y}, Lp_{2_x} - Lp_{1_x})$, $\mu' \leftarrow a' L_{2_x} + b' L_{2_y}$ ;
            else
                  if $(Lp_1 == A')$ then
                        $Lp_2 \leftarrow Lp_2$ - ( $(parity)$ ? $(PPS_y, -PPS_x)$ : $(PS_y, -PS_x)$ );
                        $(a', b') \leftarrow (Lp_{1_y} - Lp_{2_y}, Lp_{2_x} - Lp_{1_x})$, $\mu' \leftarrow a' L_{1_x} + b' L_{1_y}$ ;
            else
                  $(a', b') \leftarrow PS$;
                  $\mu' \leftarrow a'(parity$ ? $L_{1_x}$ : $L_{2_x}) +$ $b'(parity$ ? $L_{1_y}$ : $L_{2_y})$ ;
            return $(a', b', \mu')$;
7     RSmartDSS($Lp_1$, $Lp_2$, $(S_x, S_y)$, $A'$,$B'$) ;
end
```

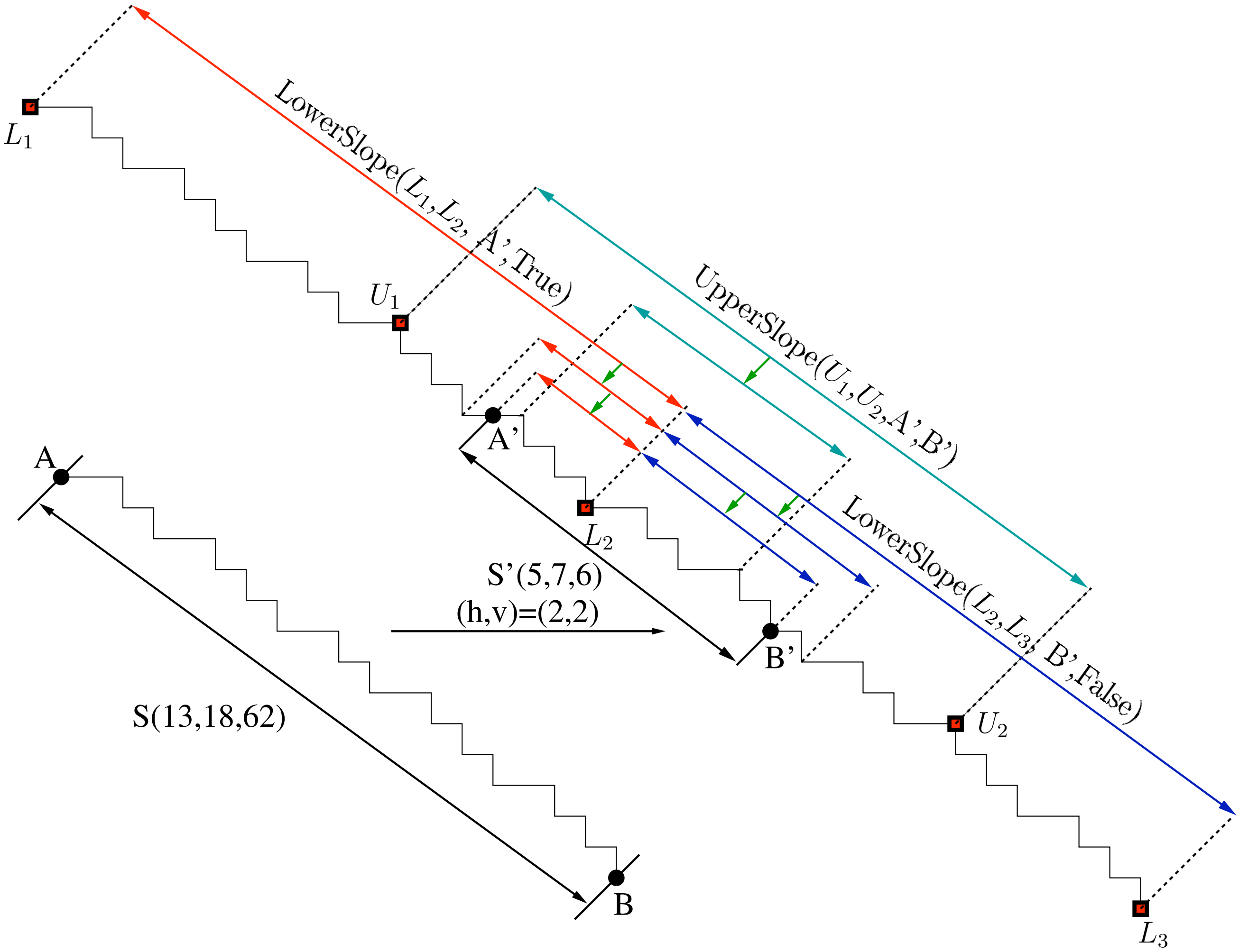

LowerSlope(L_1,L_2,A',True)
UpperSlope(U_1,U_2,A',B')
LowerSlope(L_2,L_3,B',False)
L_1
U_1
L_2
U_2
L_3
A
A'
B
B'
S'(5,7,6)
(h,v)=(2,2)
S(13,18,62)

Figure 5.3: A digital straight segment $S(13, 18, 62)$ with an odd depth slope, taken between A and B. Computes the characteristics (a', b', μ') of a DSS S' between A' and B', that is both the covering of S by the tiling $(h, v) = (2, 2)$, and some subset of $D'(13, 18, 16)$ (D' is the covering of D by the same tiling). The Lower and Upper leaning points in the right DSS are drawn as red boxes. The (red, blue or cyan) arrows ~~are represented in terms of~~ bottom-up move on the Stern-Brocot Tree.

Algorithm 9: Computes the characteristics (a', b', μ') of a DSS S' that is some subset of a DSL D', given a starting point A' and an ending point B' $(A', B' \in D')$.

```
Function ReversedSmartDSS( In $S$ : DSS $(a, b, \delta)$, In $O$: Point of $\mathbb{Z}^2$, In $A, B$ :
the endpoints of $S$, In $h, v$ : Integer ) : $S'$ : DSS $(a', b', \mu')$ ;
Var $\alpha, \beta, \mu''$ : integer  /* $D'(\alpha, \beta, \mu'')$ is the covering of $D$ by tiling $(h, v)$ */ ;
Var $A', B'$ : Point of $\mathbb{Z}^2$  /* The endpoints of $S'$ */ ;
Var $L_1, L_2$ : Point of $\mathbb{Z}^2$  /* Lower bounds of $S'$ */ ;
Var $g$ : integer ;
begin
    $\mu \leftarrow a(A_x - O_x) + b(A_y - O_y)$ ;
    $D \leftarrow DSL(S)$  /* $D(a, b, \mu)$ */ ;
    MultiScale($D, h, v, D'$) ;
    $A' \leftarrow (\frac{A_x}{h}, \frac{A_y}{v})$, $B' \leftarrow (\frac{B_x}{h}, \frac{B_y}{v})$ ;
    $L_1 \leftarrow$FirstLowerBound($D', A'$) ;
    $L_2 \leftarrow$SecondLowerBound($D', L_1, B'$) ;
    $S' \leftarrow$RSmartDSS($D', A', B', L_1, L_2$) ;
    return $S'(a', b', \mu')$;
end
```

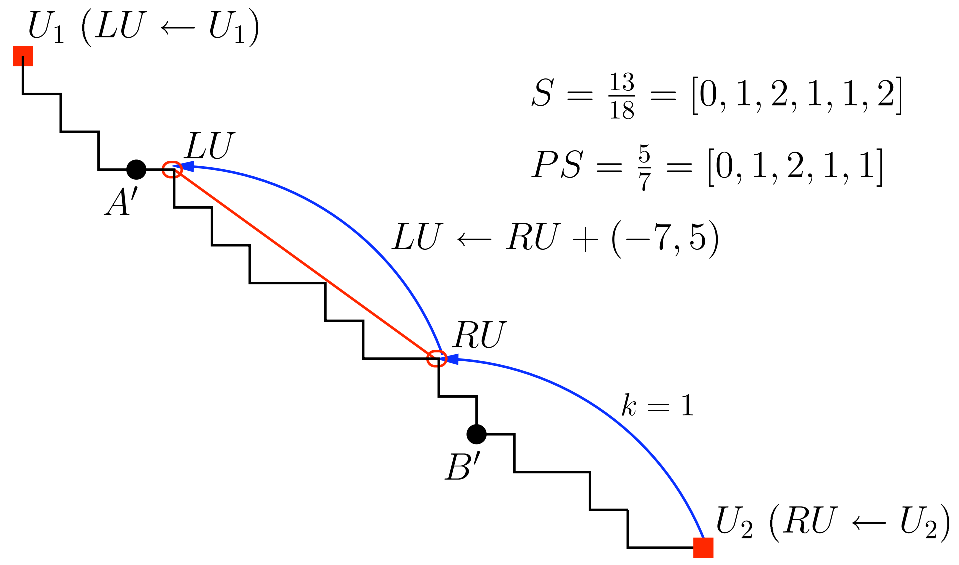

U_1 $(LU \leftarrow U_1)$
$S = \frac{13}{18} = [0, 1, 2, 1, 1, 2]$
$PS = \frac{5}{7} = [0, 1, 2, 1, 1]$
LU
A'
$LU \leftarrow RU + (-7, 5)$
RU
$k = 1$
B'
U_2 $(RU \leftarrow U_2)$

Figure 5.4: A digital straight segment $U_1U_2(13, 18, 16)$, some subset of L_1L_3 of Figure 5.3. Computes the upper characteristics (a'_1, b'_1) of a DSS $A'B'$, where each blue arrow represents the move toward the new upper leaning points, and k the number of subpatterns covering B'.

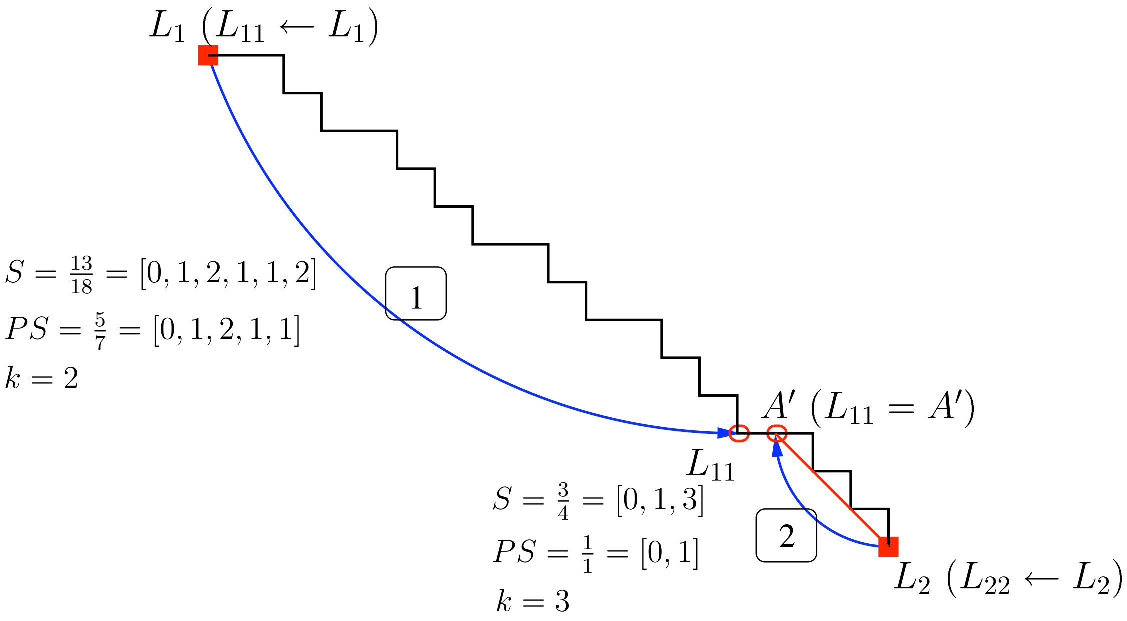

L_1 $(L_{11} \leftarrow L_1)$
$S = \frac{13}{18} = [0,1,2,1,1,2]$
$PS = \frac{5}{7} = [0,1,2,1,1]$
$k = 2$
1
A' $(L_{11} = A')$
L_{11}
$S = \frac{3}{4} = [0,1,3]$
$PS = \frac{1}{1} = [0,1]$
$k = 3$
2
L_2 $(L_{22} \leftarrow L_2)$

Figure 5.5: A digital straight segment $L_1L_2(13,18,16)$, some subset of L_1L_3 of Figure 5.3. Computes the left lower characteristics (a'_2, b'_2) of a DSS $A'L_2$, where each blue arrow represents the move toward the new lower leaning points, and k the number of subpatterns covering A'.

5.4 Multiscale covering of a digital contour

The preceding results allow us to compute an exact multiscale representation of a digital contour C with N points. More precisely, an exact covering of a digital contour C can be achieved directly from an arbitrary decomposition of C into digital straight segments (S_j).

$$\begin{tikzcd}
C \arrow[rr, "\text{A. }(h,v)\text{-covering}"] \arrow[d, "\text{B1. DSS decomp.}"'] & & C' \\
(S_j)_{j=1..M} \arrow[rr, "\text{B2. covering segment in }\mathbb{S}(h,v)"] & & (S'_j)_{j=1..M} \arrow[u, "\text{B3. set of pixels}"']
\end{tikzcd}$$

Indeed, given a digital contour $C = ((x_i, y_i))_{i=1...N}$, we can prove that the sets of digital points obtained directly with method A and indirectly with method B are the same:

A. The set of points C' is exactly the set $\{(x_i \div h, y_i \div v), i = 1 \dots N\}$, by definition of the covering by~~hi~~ the tiling $\mathbb{S}(h,v)$.

B. 1. The digital contour is first decomposed into M DSS S_j. A simple greedy

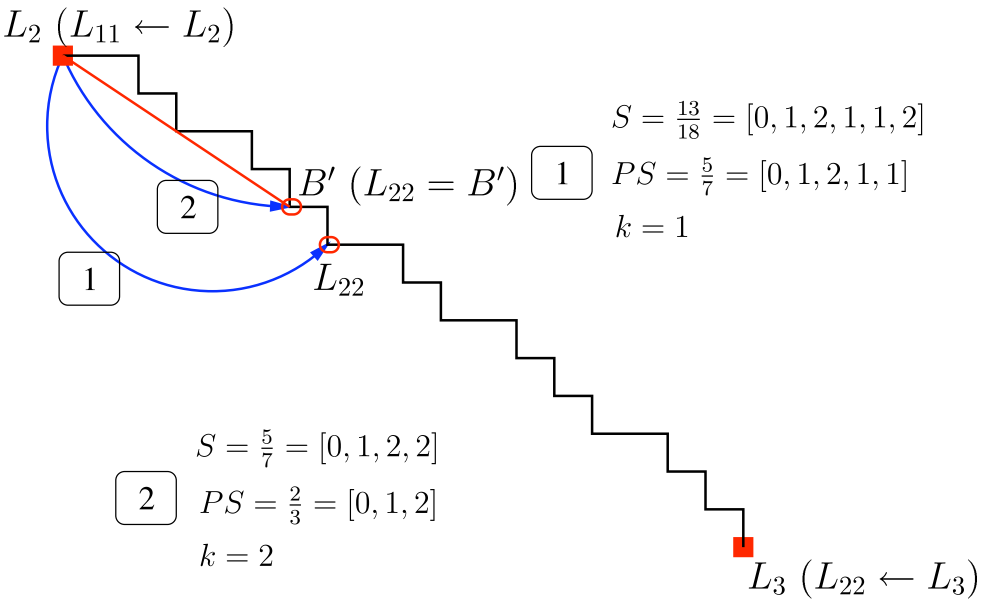

L_2 $(L_{11} \leftarrow L_2)$
B' $(L_{22} = B')$
1
$S = \frac{13}{18} = [0,1,2,1,1,2]$
$PS = \frac{5}{7} = [0,1,2,1,1]$
$k = 1$
2
1
L_{22}
$S = \frac{5}{7} = [0,1,2,2]$
2
$PS = \frac{2}{3} = [0,1,2]$
$k = 2$
L_3 $(L_{22} \leftarrow L_3)$

Figure 5.6: A digital straight segment $L_2L_3(13, 18, 16)$, some subset of L_1L_3 of Figure 5.3. Computes the right lower characteristics (a'_3, b'_3) of a DSS between L_2B', where each blue arrow represents the move toward the new lower leaning points, and k the number of subpatterns covering B'.

decomposition into DSS could be used (picking a first point, and extending the DSS at most in one direction, then repeating the process).

2. For each DSS S_j, we compute its covering segment S'_j in $\mathbb{S}(h, v)$ with `NewSmartDSS`(Algorithm 4).
3. Each DSS S'_j is decomposed into its set of pixels in $\mathbb{S}(h, v)$.

Theorem 9 *Given a tiling $\mathbb{S}(h, v)$, method B outputs the (h, v)-covering of contour C (method A).*

proof. By definition of the decomposition into DSS, $C = \cup_{j=1...M} S_j$ (DSS are viewed as sets of pixels). Step 2 to 3 comes from the fact that the covering by a tiling $\mathbb{S}(h, v)$ commutes with the set union. Step 3 to 4 comes from Corollary 1.

$$\begin{aligned} C &= \cup_{j=1...M} S_j \\ \Rightarrow \mathbb{S}(h,v)(C) &= \mathbb{S}(h,v)(\cup_{j=1...M} S_j) \\ \Rightarrow C' &= \cup_{j=1...M} \mathbb{S}(h,v)(S_j) \\ \Rightarrow C' &= \cup_{j=1...M} S'_j. \end{aligned}$$

5.4.1 Experiments

The preceding result is useful for computing the (h, v)-multiresolution of a digital shape given by its inner 4-connected contour. Method B is significantly faster than method A when the input contour has already been decomposed into a sequence of DSS, which is often the case when analysing digital shapes. A standard method to decompose a digital contour into a set of DSS is a greedy approach: given a starting point, one builds the longest DSS in one direction, then the algorithm starts again at the last point of this DSS till looping around the contour. This technique has the advantage of producing either the smallest possible number of DSS covering the input contour or just one more [9, 5]. For optimality, the minimum number of DSS is achievable in linear time [3].

The input digital contour is sometimes defined as the interpixel contour separating pixels of the shape from pixels of the background. In this case, a good DSS decomposition is given by the minimum length polygon (MLP) or minimum perimeter polygon (MPP) of the contour [7, 11], which is also computable in linear time ([8] and independently in [10]).

Note that the covering in $\mathbb{S}(h, v)$ induces a surjective map from C to C'. This is more detailed in [4], where this multiscale representation of a digital shape is used to detect noise.

Method B is illustrated on Figure 5.7. The first image (a) is a 4-connected inner pixel contour of a digital polygon which is composed of 13 digital straight segments. In (b,c) we have computed two (h, v)-covering of this contour. Taking for intance the DSL $(19, -11, -673)$ associated to a DSS of length 31. According to the *NewSmartDSS* algorithm, in $(1, 1)$-covering, 13 points are tested. In $(2, 2)$-covering, its characteristics will be $(7, -4, -134)$ and require 9 tests. In $(4, 4)$-covering, its characteristics will be $(5, -3, -35)$ and require 6 tests. The image (d) is a circle made of 19 DSS. Taking the DSL $(2, -1, -78)$, since the number of patterns of this segment is equal to 4, then for the $(2, 2)$-covering (image (e)), $\delta \geq 2\delta_{min}$ ($\delta_{min} = 2$) and its characteristics are $(2, -1, -41)$ with 0 test. But in the $(4, 4)$-covering (image (f)), its characteristics are $(2, -1, -22)$ with 2 tests.

Our technique for the first algorithm *NewSmartDSS* allows us to compute the exact multiscale representation of a digital contour in a time between $\Omega(M)$ and $O(M \times \overline{U})$, where $\overline{U}$ is the average of the partial quotient sum of the *output* subsampled DSS (Proposition 11). By definition, $\overline{U}$ cannot exceed $O(N/(hv))$, and is in most case some $O(\log^2(N/(hv)))$. The latter bound is observed when taking all possible slopes in a box $N/(hv) \times N/(hv)$. In worst case, our algorithm is linear with N.

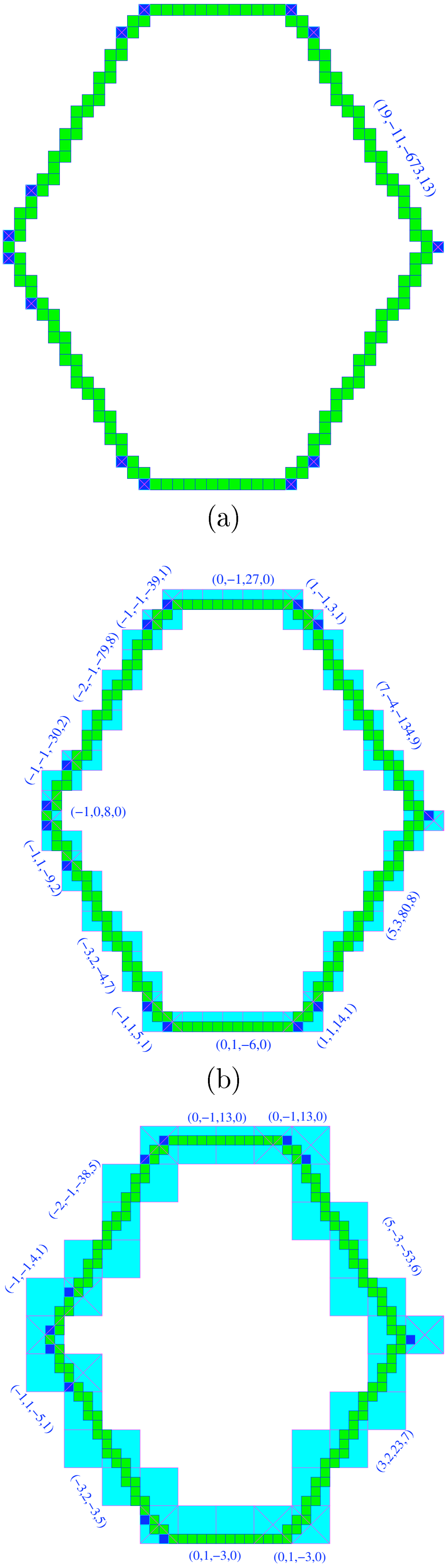
Timing measures. Execution times were measured for some contours (Table 5.1). These times were obtained on a 2.10 GHz Intel Core 2 Duo. The listed numbers include the computation time for subsampling contours and the associated number of tested points in the *NewSmartDSS* algorithm and the associated s[illegible]f the partial quotients in the *ReversedSmartDSS* algorithm.

Shape	Flower			Circle			Polygon		
# points	67494			16004			15356		
# segments	1991			574			44		
h, v	2	4	10	2	4	10	2	4	10
# points (h, v)	33744	16870	6750	8000	4000	1600	7676	3840	1532
Smart DSS									
# points tested	19352	11254	4367	5413	2977	1019	782	667	527
timings (ms)	5.2262	4.9694	4.6250	1.6282	1.5502	1.4294	0.2802	0.2734	0.2586
New Smart DSS									
# points tested	16507	10383	4199	4977	2873	1011	526	461	527
timings (ms)	4.9510	4.8946	4.5990	1.5698	1.5209	1.4130	0.2654	0.2514	0.2578
Reversed Smart DSS									
timings (ms)	4.9330	5.3098	4.8602	1.4702	1.6190	1.3234	0.1538	0.1538	0.1702

Table 5.1: Computation times of the (h, v)-covering of various digital shapes with our proposed approach. The digital shapes are: a circle of radius 2000; a flower with 5 extremeties, mean radius 5000 and variability of radius 7000; a polygon with 8 sides and radius 2000. The symbol # stands for "number of".

5.5 Conclusion

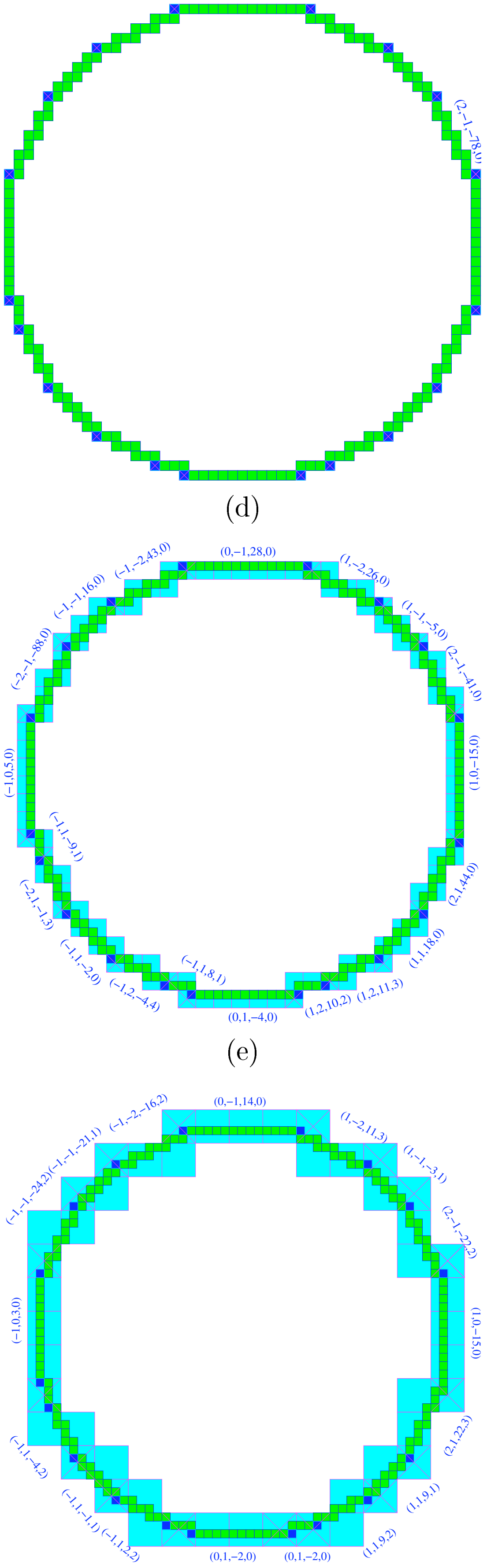
As the complexity of the proposed classical DSS recognition algorithms by Debled et al. for computing the slope of a DSS at best $\theta(n)$, where n is the number of points of the *DSS*, and as most of the points tested in DR95 are unless since they do not lead to any slope evolution. We have further presented two novel fast DSS recognition algorithms *NewSmartDSS* and *ReversedSmartDSS* according respectively to the ~~up-bottom~~ and bottom-up way in the Stern Brocot Tree, that are applicable when the digital line containing it has known characteristics. Sublinear time recognition is possible for most slopes. Finally, these properties have been used to compute the exact multiscale covering of a digital contour. Our algorithms are sensitive respectively to the size and the size of the partial quotients of the output subsampled contour, and are generally sublinear.


(19,−11,−673,13)
(0,−1,27,0)
(−1,−1,−39,1)
(1,−1,3,1)
(−2,−1,−79,8)
(7,−4,−134,9)
(−1,−1,−30,2)
(−1,0,8,0)
(−1,1,−9,2)
(5,3,80,8)
(−3,2,−4,7)
(−1,1,5,1)
(0,1,−6,0)
(1,1,14,1)
(0,−1,13,0)
(0,−1,13,0)
(−2,−1,−38,5)
(5,−3,−53,6)
(−1,−1,4,1)
(−1,1,−5,1)
(3,2,23,7)
(−3,2,−3,5)
(0,1,−3,0)
(0,1,−3,0)

(a)

(b)

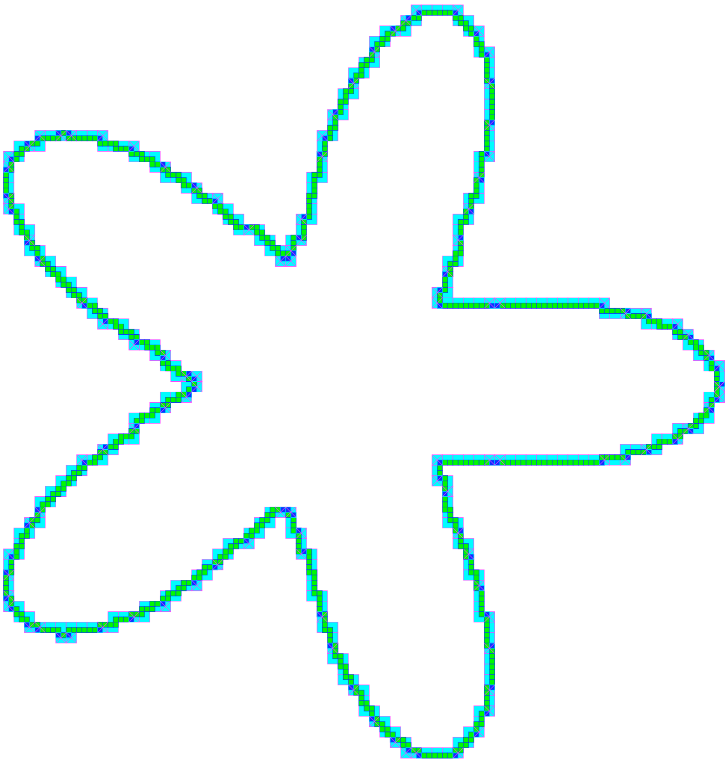
(c)


(2,−1,−78,0)
(0,−1,28,0)
(1,−2,26,0)
(1,−1,−5,0)
(2,−1,−41,0)
(1,0,−15,0)
(2,1,44,0)
(1,1,18,0)
(1,2,11,3)
(1,2,10,2)
(0,1,−4,0)
(−1,1,8,1)
(−1,2,−4,4)
(−1,1,−2,0)
(−2,1,−1,3)
(−1,1,−9,1)
(−1,0,5,0)
(−2,−1,−88,0)
(−1,−1,16,0)
(−1,−2,43,0)
(0,−1,14,0)
(1,−2,11,3)
(1,−1,−3,1)
(2,−1,−22,2)
(1,0,−15,0)
(2,1,22,3)
(1,1,9,1)
(1,1,9,2)
(0,1,−2,0)
(0,1,−2,0)
(−1,1,2,2)
(−1,1,−1,1)
(−1,1,−4,2)
(−1,0,3,0)
(−1,−1,−24,2)(−1,−1,−21,1)
(−1,−2,−16,2)

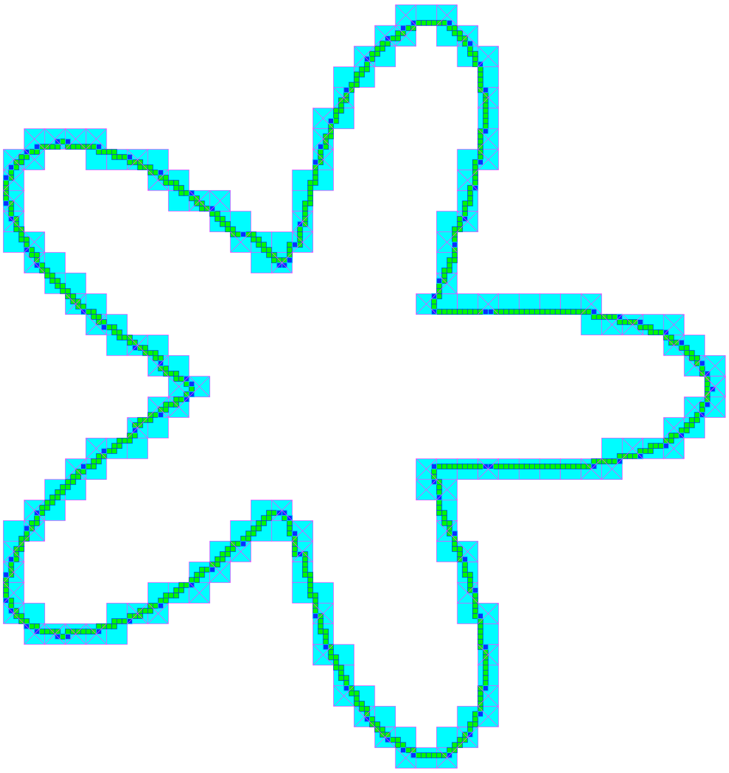
(d)

(e)

(f)



(g)



(h)

Figure 5.7: Covering of polygon, circle and a flower for $(h, v) \in \{(2, 2), (4, 4)\}$. For each shape, the endpoints of each covering segment are drawn by crosses. The associated pixels in the initial shape are drawn as filled boxes.

Chapter 6

Application to multiscale computation of digital contours with Blurred Segments

Chapter 7

Conclusions and Future Research

Contents

7.1 Summary of results

7.2 Proposed future work

Appendix A

Appendix Proofs of Theorems

A.1 proof of the Multiscale of Straight Digital Line by a lower resolution grid in the first and third quadrants

We here provide the proof of the equivalent theorem for digital straight lines in the first and third quadrants (Theorem 4).

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 $\mathbb{S}(h, v)$= $(Xh,\ Yv)$ of $\mathbb{Z}^2$ (where X, Y, h, v all are integers). Obviously the fondamental domain $[0, h) \times [0, v)$ of $\mathbb{S}(h, v)$ 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 $\mathbb{S}(h, v)$ (for which reason we will refer indifferently to the tile itself or to the point of $\mathbb{S}(h, v)$ it contains). We are interested in the set of tiles of $\mathbb{S}(h, v)$ that intersect $D(a, b, \mu)$. We denote this set by Δ. Although we have not proven yet that it is indeed a digital straight line, we already call it the *covering line* of $D(a, b, \mu)$ in $\mathbb{S}(h, v)$. More generally, given any subset O of $\mathbb{Z}^2$, the set of tiles of $\mathbb{S}(h, v)$ that intersects O is called its (h, v)*-covering*.

The tiling generated by $\mathbb{S}(h, v)$ 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:

$$\begin{aligned} X &= \left[\frac{x}{h}\right] \\ Y &= \left[\frac{y}{v}\right] \end{aligned} \tag{A.1}$$

which can be inverted as follows:

$$\begin{aligned} x &= hX + \left\{\frac{x}{h}\right\} \\ y &= vY + \left\{\frac{y}{v}\right\} \end{aligned} \tag{A.2}$$

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

$$\mu \leq ax - by < \mu + a + b \tag{A.3}$$

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

$$\mu \leq a(hX + \left\{\frac{x}{h}\right\}) - b(vY + \left\{\frac{y}{v}\right\}) < \mu + a + b$$
$$\mu - a\left\{\frac{x}{h}\right\} + b\left\{\frac{y}{v}\right\} \leq ahX - bvY < \mu + a + b - a\left\{\frac{x}{h}\right\} + b\left\{\frac{y}{v}\right\} \tag{A.4}$$

In order to simplify ($A.4$) let us introduce

$$\begin{aligned} m_x &= \left\{\frac{x}{h}\right\} \\ m_y &= \left\{\frac{y}{v}\right\} \end{aligned} \tag{A.5}$$

Since m_x and m_y vary when x steps through $\mathbb{Z}$, equation ($A.4$) becomes:

$$\mu - max_{x\in\mathbb{Z}}(am_x - bm_y) \leq ahX - bvY < \mu + a + b - min_{x\in\mathbb{Z}}(am_x - bm_y) \tag{A.6}$$

Now to fully determine this equation that defines the covering of the digital line by the tiling, the exact rang 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.

Determination of the range of $am_x - bm_y$:

By definition in equation($A.5$), it is clear that:

$$\begin{aligned} 0 \leq m_x \leq h - 1 \\ 0 \leq m_y \leq v - 1 \end{aligned} \tag{A.7}$$

Using these bounds in ($A.6$) suggests for Δ an equation of the form as

$$-b(v-1) \leq am_x - bm_y \leq a(h-1)$$

Then

$$\mu - a(h-1) \leq ahX - bvY < \mu + a + b + b(v-1) \tag{A.8}$$

But since $ahX - bvY$ can only assume values that are multiples of $g = \gcd(ah, bv)$, the bounds of equation ($A.6$) can be refined. we further denote $\frac{ah}{g}$ by p_1 and $\frac{bv}{g}$ by p_2.
However since m_x and m_y are linked through ($A.5$), 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 to absolute bounds $-b(v-1)$ and $a(h-1)$ in somes cases,
Inverting equation ($A.5$) yields

$$\begin{aligned} x &= kh + m_x \quad, k \in \mathbb{Z} \\ y &= lv + m_y \quad, l \in \mathbb{Z} \end{aligned} \tag{A.9}$$

Hence, after insertion into ($A.3$):

$$\begin{aligned}\mu \leq ax - by \ &\Rightarrow\ \mu \leq a(kh + m_x) - b(lv + m_y)\\ &\Rightarrow\ \mu - akh + blv \leq am_x - bm_y\end{aligned}$$

$$\begin{aligned}ax - by < \mu + a + b \ &\Rightarrow\ a(kh + m_x) - b(lv + m_y) \leq \mu + a + b\\ &\Rightarrow\ am_x - bm_y < \mu + a + b - akh + blv\end{aligned}$$

Then

$$\mu - akh + blv \leq am_x - bm_y < \mu + a + b - akh + blv \tag{A.10}$$

The term $kah - lbv$ has values which are multiples of $g = \gcd(ah, bv)$:
$ah = p_{11}g$, $p_{11} \in \mathbb{Z}$ and $bv = p_{22}g$, $p_{22} \in \mathbb{Z}$

$$\begin{aligned}&\Rightarrow kah = kp_{11} \quad \text{and} \quad lbv = lp_{22}g\\ &\Rightarrow kah - lbv = kp_{11}g - lp_{22}g = (kp_{11} - lp_{22})g = tg,\end{aligned}$$

Then

$$kah - lbv = tg \ \ , t \in \mathbb{Z} \tag{A.11}$$

Equation ($A.10$) can thus be rewritten as

$$\mu - tg \leq am_x - bm_y < \mu + a + b - tg, t \in \mathbb{Z} \tag{A.12}$$

Equation ($A.12$) 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 $am_x - bm_y$ (see Figure A.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 $am_x - bm_y$, i.e.

$an_x - bn_y = max_{(m_x,m_y)\in D_t\cap[0,h)\times[0,v)}(am_x - bm_y)$

Let us also denote by d the difference between $am_x - bm_y$ evaluated at $(h - 1, 0)$ and at (n_x, n_y):

$$\mu - tg \leq am_x - bm_y \leq \mu + a + b - tg - 1 \tag{A.13}$$

$am_x - bm_y$ reaches the minimal value $\mu - tg$ and the maximal value $\mu + a + b - tg - 1$, Then:

$$d \ = \ a(h - 1) - (an_x - bn_y) \tag{A.14}$$

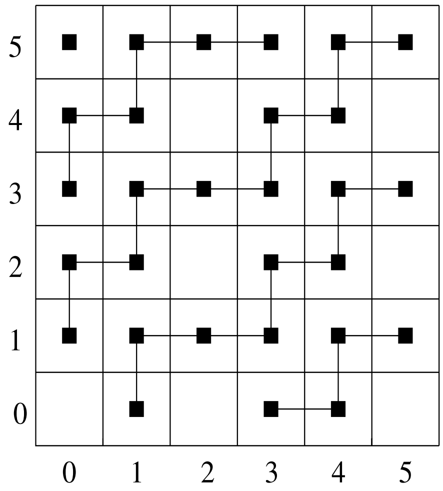


Figure A.1: **Determination of the range of 2m_x-3m_y.** The black squares represent the family of standard digital lines defined by $5-6t \leq 2m_x - 3m_y < 10-6t$ restricted to $[0,6) \times [0,6)$ and hence the possible values of (m_x, m_y)

Determining the maximum value of t verifying ($A.14$) for the point (n_x, n_y) provides a more precise expression of the lower bound of d. ($A.14$) becomes

$$\begin{aligned} d &= a(h-1) - (\mu + a + b - tg - 1) \\ &= ah - a - \mu - a - b + tg + 1 \\ &= (p_1 + t)g - (\mu + 2a + b - 1). \end{aligned}$$

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

$$\begin{aligned} (p_1 + t)g - (\mu + 2a + b - 1) &= 0 \qquad \text{(A.15)} \\ p_1 + t - \left(\frac{\mu + 2a + b - 1}{g}\right) &= 0 \\ t &= -p_1 + \left(\frac{\mu + 2a + b - 1}{g}\right) \\ t &= -p_1 + \frac{1}{g}\left(\left[\frac{\mu + 2a + b - 1}{g}\right] g + \left\{\frac{\mu + 2a + b - 1}{g}\right\}\right) \\ t &= -p_1 + \left[\frac{\mu + 2a + b - 1}{g}\right] + \frac{\left\{\frac{\mu + 2a + b - 1}{g}\right\}}{g} \\ t &= -p_1 + Q_2 + \frac{R_2}{g}, \end{aligned}$$

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

1. If g divided $\mu + 2a + b - 1$. In this case, ($A.15$) becomes

$$t = -p_1 + Q_2$$

And we insert this equality in ($A.6$).

$$\begin{aligned} ahX - bvY &\geq \mu - max_{x\in\mathbb{Z}}(am_x - bm_y) \\ &\geq \mu - (\mu + a + b - tg - 1) \\ &\geq \mu - \mu - a - b + tg + 1 \\ &\geq tg - a - b + 1 \\ &\geq (-p_1 + Q_2)g - a - b + 1 \end{aligned}$$

Divided by g, then

$$\frac{ahX}{g} - \frac{bvY}{g} \geq -p_1 + Q_2 - \frac{a+b-1}{g} \tag{A.16}$$

Where a,b and g being three integers, we denote with $\left[\frac{a+b-1}{g}\right]$ the quotient of the euclidean division of $a+b-1$ by g while $\left\{\frac{a+b-1}{g}\right\}$ denotes the remainder of this division.
The fundamental relation between these two values writes:

$$a + b - 1 = g\left[\frac{a+b-1}{g}\right] + \left\{\frac{a+b-1}{g}\right\}$$

Let $\alpha = \frac{ah}{g}$ and $\beta = \frac{bv}{g}$. Then Equation($A.16$) can be rewritten as:

$$\alpha X - \beta Y \geq -p_1 + Q_2 - Q_1 - \frac{R_1}{g} \tag{A.17}$$

Where $Q_1 = \left[\frac{a+b-1}{g}\right]$ and $R_1 = \left\{\frac{a+b-1}{g}\right\}$. We conclude that the division of remainders of $a+b-1$ by g by g is between 0 and 1 strict, and all terms in Equation($A.17$) are integers, so we get:

$$\alpha X - \beta Y \quad \geq \quad -p_1 + Q_2 - Q_1 \tag{A.18}$$

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

$$t = -p_1 + Q_2 + \frac{R_2}{g}$$

Hence Equation ($A.6$) can be rewritten as

$$\begin{aligned} ahX - bvY &\geq tg - a - b + 1 \\ &\geq \left(-p_1 + Q_2 + \frac{R_2}{g}\right) g - a - b + 1 \end{aligned}$$

Divided by g, then,

$$\begin{aligned} \alpha X - \beta Y &\geq -p_1 + Q_2 + \frac{R_2}{g} - \frac{a + b - 1}{g} \\ &\geq -p_1 + Q_2 - Q_1 + \frac{R_2}{g} - \frac{R_1}{g} \end{aligned}$$

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:

- If $R_2 \leq R_1$,then,

$$\alpha X + \beta Y \geq -p_1 + Q_2 - Q_1 \tag{A.19}$$

- If $R_2 > R_1$,then,

$$\alpha X + \beta Y \geq -p_1 + Q_2 - Q_1 + 1 \tag{A.20}$$

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).
$an^{'}_x - bn^{'}_y = min_{(m_x,m_y) \in D_t \cap [0,h) \times [0,v)}(am_x - bm_y)$
Let us also denote by d' the difference between $am_x - bm_y$ evaluated at $(0, v - 1)$ and at $(n^{'}_x, n^{'}_y)$, Then:

$$d' = -b(v - 1) - (an^{'}_x - bn^{'}_y) \tag{A.21}$$

Determining the maximum value of t verifying ($A.21$) for the point (n'_x, n'_y) provides a more precise expression of the upper bound of d'. ($A.21$) becomes

$$\begin{aligned} d' &= -b(v - 1) - (\mu - tg) \\ &= (\mu + 2a + b - 1) + tg - bv - (2\mu + 2a - 1) \end{aligned}$$

We take $d' = 0$ to find the minimum value, implying $(n'_x, n'_y) = (0, v - 1)$, then:

$$\begin{aligned} t &= -\left(\frac{\mu + 2a + b - 1}{g}\right) + \left(\frac{2\mu + 2a - 1}{g}\right) + \frac{bv}{g} \\ &= -Q_2 - \frac{R_2}{g} + \left(\frac{2\mu + 2a - 1}{g}\right) + \frac{bv}{g} \end{aligned} \tag{A.22}$$

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

1. If g divides $\mu + 2a + b - 1$. In this case ($A.22$) becomes

$$t = p_2 - Q_2 + \left(\frac{2\mu + 2a - 1}{g}\right)$$

We insert this equality in ($A.6$) and can be rewritten as

$$\begin{array}{rcl} ahX - bvY & < & \mu + a + b - (\mu - tg) \\ & < & tg + a + b \\ & < & p_2 g - Q_2 g + 2\mu + 2a - 1 + a + b \\ & < & p_2 g - Q_2 g + 2\mu + 3a + b - 1 \end{array}$$

Divided by g, then,

$$\alpha X - \beta Y < p_2 - Q_2 + Q_3 + \frac{R_3}{g} \tag{A.23}$$

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

(a) If g divides $2\mu + 3a + b - 1$. In this case Equation ($A.23$) becomes

$$\alpha X - \beta Y < p_2 + Q_3 - Q_2 \tag{A.24}$$

(b) If g does not divide $2\mu + 3a + b - 1$. In this case Equation ($A.23$) becomes

$$\alpha X - \beta Y < p_2 + Q_3 - Q_2 + 1 \tag{A.25}$$

2. If g does not divide $\mu + 2a + b - 1$. In this case Equation ($A.22$) becomes

$$t = p_2 - \left[\frac{\mu + 2a + b - 1}{g}\right] - \frac{\left\{\frac{\mu + 2a + b - 1}{g}\right\}}{g} + \left(\frac{2\mu + 2a - 1}{g}\right)$$

Hence Equation ($A.14$) can be rewritten as

$$\begin{array}{rcl} ahX - bvY & < & \mu + a + b - (\mu - tg) \\ & < & tg + a + b \\ & < & p_2 g - Q_2 g - R_2 + 2\mu + 3a + b - 1 \end{array} \tag{A.26}$$

Where μ,a,b and g being four integers, we denote with $\left[\frac{2\mu + 3a + b - 1}{g}\right]$ the quotient of the euclidean division of $\mu + 3a + b - 1$ by g while $\left\{\frac{2\mu + 3a + b - 1}{g}\right\}$

denotes the remainder of this division.
The fundamental relation between these two values can be rewriten as:

$$2\mu + 3a + b - 1 = g\left[\frac{2\mu + 3a + b - 1}{g}\right] + \left\{\frac{2\mu + 3a + b - 1}{g}\right\}$$

Let $\alpha = \dfrac{ah}{g}$ and $\beta = \dfrac{bv}{g}$. Then Equation ($A.26$) can be rewritten as:

$$\alpha X - \beta Y \quad < \quad p_2 + Q_3 - Q_2 + \frac{R_3}{g} - \frac{R_2}{g}$$

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:

- If $R_3 \leq R_2$, then,

$$\alpha X - \beta Y < p_2 + Q_3 - Q_2 \tag{A.27}$$

- If $R_3 > R_2$, then,

$$\alpha X - \beta Y < p_2 + Q_3 - Q_2 + 1 \tag{A.28}$$

$Q_k = \left\{\left[\dfrac{(k-1)\mu + k(a+b) - 1}{g}\right], k = 1,2,3\right\}$
of R_1, R_2 and R_3. We will now present all the possible solutions of $\alpha X + \beta Y$:
More precisely, eleven equations involving $\alpha X - \beta Y$ can be obtained by combining the lower bound (equations $A.18$, $A.19$, $A.20$) and the upper bound (equations $A.24$, $A.25$, $A.27$, $A.28$) of $\alpha X - \beta Y$. Therefore those equations can be formulated as single expression given below:

Theorem 10 *The digital straight line Δ of $\mathbb{S}(h,v)$ covering the standard digital line $D(a,b,\mu)$ of $\mathbb{Z}^2$ on the first and the third quadrant is defined by:*

$$-p_1 + Q'_2 - Q'_1 + SI \leq \alpha X - \beta Y < p_2 + Q'_3 - Q'_2 + SS \tag{A.29}$$

Where $\alpha = p_1 = \frac{ah}{g}$, $\beta = p_2 = \frac{bv}{g}$, $g = \gcd(ah, bv)$,
$Q'_k = \left\{\left[\frac{(k-1)\mu+ka+b-1}{g}\right], k = 1,2,3\right\}$, $R'_k = \left\{\left\{\frac{(k-1)\mu+ka+b-1}{g}\right\}, k = 1,2,3\right\}$ *and*

$$SI = \begin{cases} 0 & \text{if} \quad R'_2 \leq R'_1 \\ 1 & \quad otherwise \end{cases} \qquad SS = \begin{cases} 0 & \text{if} \quad R'_3 \leq R'_2 \\ 1 & \quad otherwise \end{cases}$$

As an example, let us study the covering of the standard digital line

$$D(3,5,-2)\colon\ -2 \leq 3x - 5y < 6$$

Figure A.2 and Figure A.3 illustrates the $(4,4)$-covering and $(3,5)$-covering respectively of $D(3,5,-2)$.

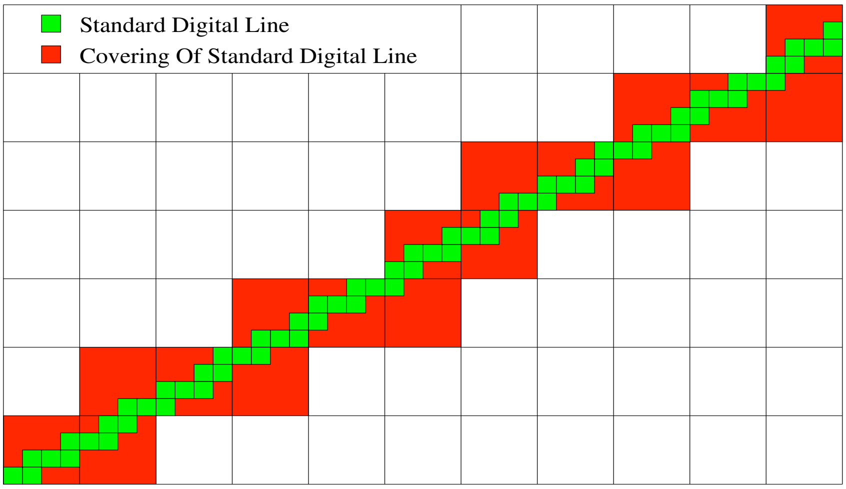


Figure A.2: The covering line of $D(3,5,-2)$ by $S(4,4)$: $-2 \leq 3X - 5Y < 6$

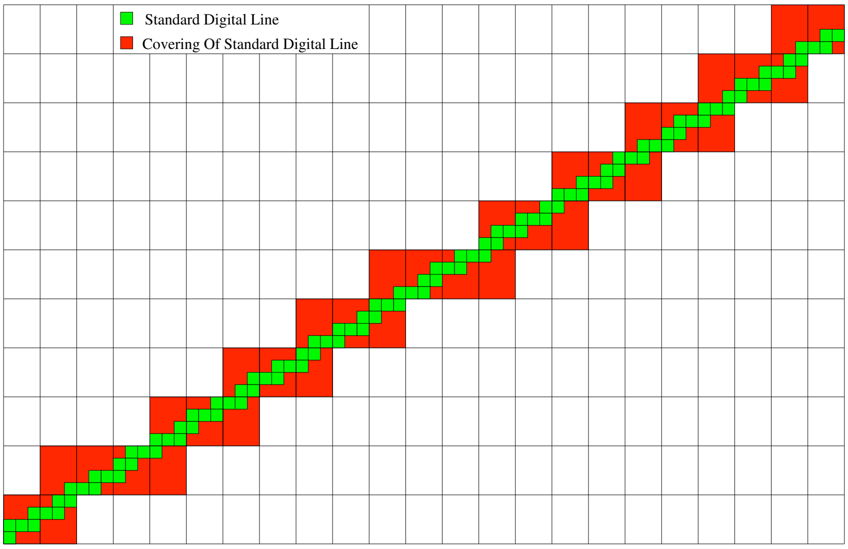


Figure A.3: The covering line of $D(3,5,-2)$ by $S(3,4)$: $-8 \leq 9X - 20Y < 21$

Bibliography

[1] T.A. Anderson and C.E. Kim. Representation of digital line segments and their preimages. 30(3):279–288, June 1985. 29

[2] Jean Berstel and Aldo de Luca. Sturmian words, lyndon words and trees. *Theor. Comput. Sci.*, 178(1-2):171–203, 1997. 31

[3] Jean Berstel and Aldo de Luca. Sturmian words, lyndon words and trees. *Theor. Comput. Sci.*, 178(1-2):171–203, 1997. 32

[4] C. Brezinski. *History of Continued Fractions and Padé Approximations.* Printed in USA, Springer, Heidelberg, 1991. 11

[5] C. Brezinski. *History of continued Fractions and Palé Approximants.* Springer-Verlag Berlin Heidelberg, 1991. 4

[6] R. Brons. Linguistic methods for the description of a straight line on a grid. 3(1):48–62, March 1974. 10

[7] Alfred M. Bruckstein. Self-similarity properties of digitized straight lines. 10

[8] L. Brun and W. Kropatsch. Contraction kernels and combinatorial maps. *Pattern Recognition Letters*, 24(8):1051 – 1057, 2003. 29

[9] D. Coeurjolly, Y. Gérard, J.-P. Reveillès, and L. Tougne. An elementary algorithm for digital arc segmentation. *Discrete Applied Mathematics*, 139:31–50, 2004. 14

[10] F. de Vieilleville and J.-O. Lachaud. Revisiting digital straight segment recognition. In A. Kuba, K. Palágyi, and L.G. Nyúl, editors, *Proc. Int. Conf. Discrete Geometry for Computer Imagery (DGCI'2006), Szeged, Hungary*, volume 4245 of *LNCS*, pages 355–366. Springer, October 2006. 10, 12, 31, 32, 35

[11] François de Vieilleville, Jacques-Olivier Lachaud, and Fabien Feschet. Maximal digital straight segments and convergence of discrete geometric estimators. In *SCIA*, pages 988–997, 2005. 11

[12] François de Vieilleville, Jacques-Olivier Lachaud, and Fabien Feschet. Convex digital polygons, maximal digital straight segments and convergence of discrete geometric estimators. *Journal of Mathematical Imaging and Vision*, 27(2):139–156, 2007. 11

[13] François de Vieilleville, Jacques-Olivier Lachaud, and Fabien Feschet. Maximal digital straight segments and convergence of discrete geometric estimators. *CoRR*, abs/0906.2716, 2009. 11

[14] I. Debled and J.P. Reveilles. A new approach to digital planes. In *Proc. Spie's Internat. Symposium on Photonics and Industrial Applications-Technical conference vision geometry 3*, Boston, 1994. 30

[15] I. Debled-Rennesson. *Etude et reconnaissance des droites et plans discrets.* PhD thesis, Université Louis Pasteur, Strasbourg, 1995. 10, 32

[16] I. Debled-Rennesson and J.-P. Reveillès. A linear algorithm for segmentation of discrete curves. *International Journal of Pattern Recognition and Artificial Intelligence*, 9:635–662, 1995. 9, 10, 11, 44

[17] L. Dorst and R.P.W. Duin. Spirograph theory: A framework for calculations on digitized straight lines. 6(5):632–639, September 1984. 10

[18] L. Dorst and A. W. M. Smeulders. Discrete representation of straight lines. *IEEE transactions Pattern Analysis Machine Intelligence*, 6:450–463, 1984. 10, 32

[19] F. Feschet. Canonical representations of discrete curves. *Pattern Anal. Appl.*, 8(1-2):84–94, 2005. 44

[20] F. Feschet and L. Tougne. Optimal time computation of the tangent of a discrete curve: Application to the curvature. In *Proc 8th Int. Conf. Discrete Geometry for Computer Imagery (DGCI'99)*, number 1568 in LNCS, pages 31–40. Springer Verlag, 1999. 11, 14

[21] F. Feschet and L. Tougne. On the min dss problem of closed discrete curves. *Discrete Applied Mathematics*, 151(1-3):138 – 153, 2005. IWCIA 2003-Ninth International Workshop on Combinatorial Image Analysis. 11, 63

[22] O. Figueiredo. *Advances in discrete geometry applied to the extraction of planes and surfaces from 3D volumes.* PhD thesis, EPFL, Lausanne, 1994. 14, 30

[23] Philippe Flajolet, Brigitte VallÃ©e, and Ilan Vardi. Continued fractions from euclid to the present day, 2000. 4

[24] H. Freeman. On the encoding of arbitrary geometric configurations. 10(2):260–268, June 1961. 29

[25] H. Freeman. Boundary encoding and processing. 1970. 9

[26] Herbert Freeman. Computer processing of line-drawing images. *ACM Comput. Surv.*, 6(1):57–97, 1974. 29

[27] Ronald L. Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics: A Foundation for Computer Science.* Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1994. 7

[28] G. H. Hardy and E. M. Wright. *An introduction to the Theory of Numbers.* Oxford Society, 1989. 7

[29] B. Kerautret and J.-O. Lachaud. Multi-scale analysis of discrete contours for unsupervised noise detection. In *IWCIA*, pages 187–200, 2009. 63

[30] C.E. Kim. Digital convexity, straightness, and convex polygons. 4(6):618–626, November 1982. 11

[31] R. Klette and A. Rosenfeld. *Digital Geometry - Geometric Methods for Digital Picture Analysis.* Morgan Kaufmann, San Francisco, 2004. 10, 29, 31, 32

[32] R. Klette and A. Rosenfeld. Digital straightness–a review. *Discrete Applied Mathematics*, 139(1-3):197 – 230, 2004. The 2001 International Workshop on Combinatorial Image Analysis. 10, 29, 63

[33] J. J. Koenderink. The structure of images. *Biol. Cyb.*, 50:363–370, 1984. 13, 29

[34] V. Kovalevsky. Applications of digital straight segments to economical image encoding. In *Proc. DGCI*, LNCS, pages 51–62, London, UK, 1997. 29

[35] J.-O. Lachaud, A. Vialard, and F. de Vieilleville. Fast, accurate and convergent tangent estimation on digital contours. *Image Vision Comput.*, 25(10):1572–1587, 2007. 14, 44

[36] Jacques-Olivier Lachaud, Anne Vialard, and François de Vieilleville. Analysis and comparative evaluation of discrete tangent estimators. In *DGCI*, pages 240–251, 2005. 11

[37] Jeffrey C. Lagarias. Number theory and dynamical systems. In S. A. Burr, editor, *The Unreasonable Effectiveness of Number Theory.* Am. Math. Soc., Proceedings of Symposia in Applied Mathematics, Vol. 46., 1992. 7

[38] T. Lindeberg. 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. 13

[39] U. Montanari. A note on minimal length polygonal approximation to a digitized contour. *Communications of the ACM*, 13(1):41–47, 1970. 44, 63

[40] X. Provençal and J.-O. Lachaud. Two linear-time algorithms for computing the minimum length polygon of a digital contour. In S. Brlek, C. Reutenauer, and X. Provençal, editors, *Proc. Int. Conf. Discrete Geometry for Computer Imagery (DGCI'2009), Montréal, Québec*, volume 5810 of *Lecture Notes in Computer Science*, pages 104–117. Springer, 2009. 63

[41] H. Reiter-Doerksen and I. Debled-Rennesson. *Convex ans concave parts of digital curves.* In Dagstuhl Seminar "Geomatric Properties from Incomplete Data", March 2004. 11

[42] J.-P. Réveillès. *Géométrie discrète, calcul en nombres entiers et algorithmique.* PhD thesis, Thèse d'etat, Université Louis Pasteur, Strasbourg, 1991. 9, 11

[43] J.-P. Réveillès. Structure des droites discrètes. *In Journées mathématique et informatique, Marseille-Luminy*, 495, Octobre 1989. 9

[44] A. Rosenfeld and R. Klette. Digital straightness. *Electronic Notes in Theoretical Computer Science*, 46:1 – 32, 2001. IWCIA 2001, 8th International Workshop on Combinatorial Image Analysis. 63

[45] T. Roussillon, I. Sivignon, and L. Tougne. What does digital straightness tell about digital convexity ? In *Proc. Int. Workshop. on Combinatorial Image Analysis (IWCIA'2009), Mexico*, volume 5852 of *Lecture Notes in Computer Science*, pages 43–55. Springer, 2009. 63

[46] M. Said, J.-O. Lachaud, and F. Feschet. Multiscale Discrete Geometry. In *Proc. International Conference on Discrete Geometry for Computer Imagery (DGCI2009) International Conference on Discrete Geometry for Computer Imagery*, volume 5810 of *Lecture Notes in Computer Science*, pages 118–131, Montréal, Québec Canada, 2009. Springer. 38, 39, 40, 42

[47] I. Sivignon, F. Dupont, and J.-M. Chassery. Digital intersections: minimal carrier, connectivity, and periodicity properties. *Graphical Models*, 66(4):226–244, 2004. 10, 30

[48] Isabelle Sivignon, Florent Dupont, and Jean-Marc Chassery. New results about digital intersections. In *Proc. DGCI*, pages 102–113, 2003. 10, 30

[49] J. Sklansky, R. L. Chazin, and B. J. Hansen. Minimum perimeter polygons of digitized silhouettes. *IEEE Trans. Computers*, 21(3):260–268, 1972. 63

[50] 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. 44

[51] Hanna Uscka-Wehlou. Continued fractions and digital lines with irrational slopes. In *DGCI*, pages 93–104, 2008. 11

[52] Hanna Uscka-Wehlou. Run-hierarchical structure of digital lines with irrational slopes in terms of continued fractions and the gauss map. *Pattern Recogn.*, 42(10):2247–2254, 2009. 10

[53] Hanna Uscka-Wehlou. Two equivalence relations on digital lines with irrational slopes. a continued fraction approach to upper mechanical words. *Theor. Comput. Sci.*, 410(38-40):3655–3669, 2009. 11

[54] A. Vacavant, D. Coeurjolly, and L. Tougne. Dynamic reconstruction of complex planar objects on irregular isothetic grids. In *Proc. 2nd Int. Symp. in Visual Computing (ISVC'2006)*, volume 4292 of *LNCS*, pages 205–214, 2006. 13

[55] Ilan Vardi. Archimedes' cattle problem. *American Mathematical Monthly*, 101:629–639, 1998. 4

[56] K. Voss. *Discrete Images, Ojects, and Functions in* $\mathbb{Z}^n$. Springer-Verlag, 1993. 11

[57] A. P. Witkin. Scale-space filtering. In *Proc. 8th Int. Joint Conf. Art. Intell., Karlsruhe, Germany*, pages 1019–1022, 1983. 13, 29

[58] J. Yaacoub. *Enveloppes convexes de réseaux et applications au traitement d'images.* PhD thesis, Université Louis Pasteur, Strasbourg, 1997. 10, 32