Generation and Recognition of Digital Planes using Multi-dimensional Continued Fractions

Thomas Fernique

LIRMM, CNRS and Univ. Montpellier 2
161 rue Ada, 34392 Montpellier, France
LIF, CNRS and Univ. de Provence
39 rue Joliot-Curie, 13453 Marseille, France

Abstract

This paper extends, in a multi-dimensional framework, pattern recognition techniques for generation or recognition of digital lines. More precisely, we show how the connection between chain codes of digital lines and continued fractions can be generalized by a connection between tilings and multi-dimensional continued fractions. This leads to a new approach for generating and recognizing digital hyperplanes.

Key words: Brun algorithm, continued fractions, digital plane generation, digital plane recognition, discrete geometry, dual map, substitution.

Introduction

Discrete (or digital) geometry mainly deals with discrete sets considered to be digitized objects of the Euclidean space. A challenging problem is to decompose a discrete set into elementary ones, which could be easily stored and from which one can easily reconstruct the original discrete set. Good candidates for such elementary discrete sets are digitizations of Euclidean hyperplanes, in particular *arithmetic discrete hyperplanes* (see ,*e.g.*, [1,2,9,10,15,17]). We thus need efficient algorithms generating arbitrarily big patches of such digitizations from given parameters and, conversely, recognizing parameters of given digitizations.

In the particular case of digitizations of lines, among other techniques, so-called *linguistic techniques* provide a nice connection with word theory and

Email address: thomas.fernique@lif.univ-mrs.fr (Thomas Fernique).

Preprint submitted to Elsevier *15 October 2008*

continued fractions. Let us briefly detail this. A (connected) digital line made of horizontal or vertical unit segments can be encoded by a two-letter word, called *chain code* or *Freeman code*. For example, if a horizontal (resp. vertical) unit segment is encoded by 0 (resp. 1), then a segment of slope 1 can be encoded by a word of the form $10\dots10 = (10)^k$. Basic transformations on words correspond then to basic operations on slopes of the segments they encode. For example, replacing each 0 by 01 and each 1 by 0 in the previous word leads to the word $(001)^k$, which encodes a segment of slope $1/2$. There are algorithms using this approach for both recognition and generation of digital lines. Continued fraction expansions of slopes of segments turn out to play a central role there (see *e.g.* [6,19,21] or references in [16]).

In higher dimensions, there are also various techniques for generating or recognizing digital hyperplanes as, for example, linear programming, computational geometry or preimage techniques (see *e.g.* [11] and references therein). However, these approaches do not extend the connection between word theory and continued fractions. The aim of this paper is to introduce an approach which does.

The paper is organized as follows.
The three first sections introduce main objects and tools. More precisely, Sec. 1 defines stepped planes, which are the digital planes that we consider, and formally states what we mean by *generate* or *recognize* such a digital plane. Then, Sec. 2 recalls the *dual map* notion, a powerful tool introduced in [4] as a generalization of substitutions on words. Some useful properties proved in [7] are also given. Last, Sec. 3 recalls the *Brun algorithm*, one of the existing multi-dimensional continued fractions algorithms.
The two following sections present the main results of this paper. In Sec. 4, we show how to construct a finite subset of a stepped plane which tiles it – one speaks about *generation* of digital plane. This subset is obtained by successive applications of dual maps related to the Brun expansion of the normal vector of the stepped plane. Conversely, in Sec. 5, we show how to decide whether a given object is a stepped plane or not, and how to compute its parameters if it is a stepped plane – one speaks about *recognition* of digital plane. Roughly speaking, the idea is to compute the Brun expansion of a *potential* normal vector directly on the given objects by successive applications of dual maps; the process eventually either fails if it is not a stepped plane, or leads to a trivial stepped plane, from which parameters of the initial stepped plane (in fact, the Brun expansion of its normal vector) are deduced.
A short conclusion resumes advantages and drawbacks of both generation and recognition algorithms.

1 Stepped planes

In this section, we introduce our basic digital objects, namely *binary functions* and *stepped planes*. Formally, it is convenient to consider the set of functions from $\mathbb{Z}^d \times \{1, \ldots, d\}$ to $\mathbb{Z}$, denoted by $\mathfrak{F}_d$. Then, we define:

Definition 1 *A* binary function *is a function in $\mathfrak{F}_d$ which takes values in $\{0, 1\}$. The* size *of a binary function $\mathcal{B}$, denoted by $|\mathcal{B}|$, is the cardinality of its* support, *that is, the subset of $\mathbb{Z}^d \times \{1, \ldots, d\}$ where $\mathcal{B}$ takes value one. We denote by $\mathfrak{B}_d$ the set of binary functions. For $\vec{x} \in \mathbb{Z}^d$ and $i \in \{1, \ldots, d\}$, we call* face of type i located in $\vec{x}$ *the binary function denoted by $(\vec{x}, i^*)$ whose support is $\{(\vec{x}, i)\}$.*

Note that binary functions (resp. functions of $\mathfrak{F}_d$) can be seen as sums of faces (resp. weighted sums of faces). Let us now provide a geometric interpretation of binary functions. Let $(\vec{e}_1, \ldots, \vec{e}_d)$ denote the canonical basis of $\mathbb{R}^d$. The geometric interpretation of a face $(\vec{x}, i^*)$ is the closed subset of $\mathbb{R}^d$ defined by:

$$\{\vec{x} + \vec{e}_i + \sum_{j \neq i} \lambda_j \vec{e}_j \mid 0 \leq \lambda_j \leq 1\}.$$

This subset is a hyperface of the unit cube of $\mathbb{R}^d$ whose lowest vertex is $\vec{x}$ (see Fig. 1). Then, the geometric interpretation of a binary function, that is, of a sum of faces, is the union of the geometrical interpretations of these faces.

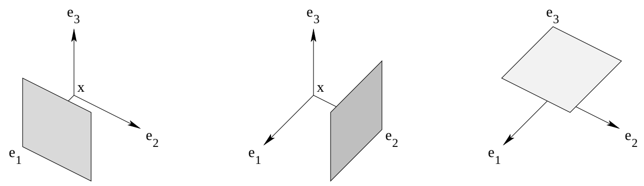

Fig. 1. Geometrical interpretations of faces $(\vec{x}, i^*)$, for $i = 1, 2, 3$ (from left to right).

Among binary functions, we are especially interested in so-called *stepped planes*, as introduced in [20]:

Definition 2 *Let $\vec{\alpha} \in \mathbb{R}^d_+ \backslash \{\vec{0}\}$ and $\rho \in \mathbb{R}$. The* stepped plane *of* normal vector $\vec{\alpha}$ *and* intercept ρ, *denoted by $\mathcal{P}_{\vec{\alpha},\rho}$, is the binary function defined by:*

$$\mathcal{P}_{\vec{\alpha},\rho}(\vec{x}, i) = 1 \ \Leftrightarrow \ \langle \vec{x} | \vec{\alpha} \rangle < \rho \leq \langle \vec{x} + \vec{e}_i | \vec{\alpha} \rangle,$$

where $\langle | \rangle$ is the dot product. We denote by $\mathfrak{P}_d$ the set of stepped planes.

Fig. 2 depicts the geometrical interpretation of a stepped plane. It is not hard to check that the vertices of a stepped plane $\mathcal{P}_{\vec{\alpha},\rho}$, that is, the integers vectors which belong to its geometrical interpretation, form a *standard arithmetic discrete plane of parameters* $(\vec{\alpha}, \rho)$ (see [1,15,17]). Moreover, one checks that the orthogonal projection along $\vec{e}_1 + \ldots + \vec{e}_d$ maps the geometrical representation of a stepped plane onto a tiling of $\mathbb{R}^{d-1}$ whose tiles are projections of geometrical representations of faces (see also Fig. 2).

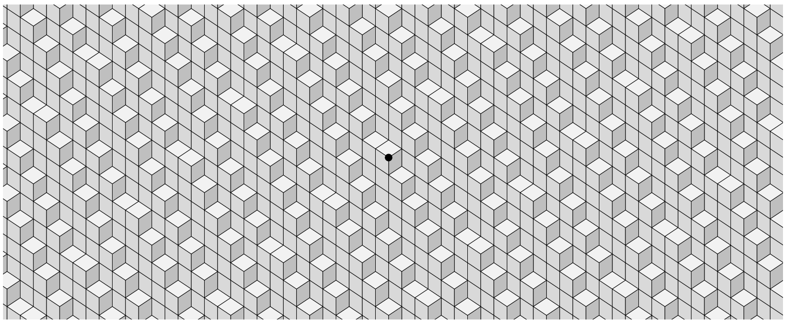

Fig. 2. Geometrical interpretation of the stepped plane $\mathcal{P}_{(24,9,10),0}$ (highlighted origin). This is a union of faces of unit three-dimensional cubes.

We can now formally state the problem of generation and recognition of digital planes that we consider throughout this paper. The *generation problem* consists, given a normal vector $\vec{\alpha} \in \mathbb{R}^d_+ \backslash \{\vec{0}\}$ and an intercept $\rho \in \mathbb{R}$, in designing an algorithm for computing any finite subsets of the stepped plane $\mathcal{P}_{\vec{\alpha},\rho}$. In particular, shapes of subsets plays an important role. In fact, since any finite subset of a stepped plane can be seen as a subset of a stepped plane with *rational* normal vector and intercept, we will consider further only the case of stepped planes with rational parameters. Conversely, the *recognition problem* consists, given a finite binary function $\mathcal{B} \in \mathfrak{B}_d$, in computing a normal vector $\vec{\alpha} \in \mathbb{R}^d_+ \backslash \{\vec{0}\}$ and an intercept $\rho \in \mathbb{R}$ such that $\mathcal{B}$ is a subset of the stepped plane $\mathcal{P}_{\vec{\alpha},\rho}$. Such a normal vector and intercept turning out to be not unique, we will also consider the problem of finding all of them.

2 Dual maps

In this section, we introduce the main tool of this paper, namely *dual maps*. Dual maps have been introduced in [4] under the name *generalized substitutions* (see also [3,5]) and extended to the framework here considered in [12].

Let us first recall some basic definitions and notations. We denote by F_d the free group generated by the alphabet $\{1, \ldots, d\}$, with the concatenation as a composition rule and the empty word as unit. An endomorphism of F_d is a *substitution* if it maps any letter to a non-empty concatenation of letters with non-negative powers. The *Parikh* or *Abelianization* map $\vec{f}$ is defined by:

$$\forall w \in F_d, \quad \vec{f}(w) = (|w|_1, \ldots, |w|_d),$$

where $|w|_i$ is the sum of the powers of the occurrences of the letter i in w. Then, the *incidence matrix* of an endomorphism σ of F_d, denoted by M_σ, is the $d \times d$ integer matrix whose i-th column is the vector $\vec{f}(\sigma(i))$. Last, an endomorphism of F_d is said to be *unimodular* if its incidence matrix has determinant ± 1.

Example 1 *Let σ be the endomorphism of F_3 defined by $\sigma(1) = 12$, $\sigma(2) = 13$ and $\sigma(3) = 1$. Note that σ is a substitution (often called* Rauzy *substitution). One computes, for example, $\sigma(1^{-1}2) = \sigma(1)^{-1}\sigma(2) = 2^{-1}1^{-1}13 = 2^{-1}3$, and $\vec{f}(2^{-1}3) = \vec{e}_3 - \vec{e}_2$. This substitution is unimodular since its incidence matrix M_σ (below) has determinant 1:*

$$M_\sigma = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

We are now in a position to recall from [12] the definition of *dual maps*:

Definition 3 *The* dual map *of a unimodular endomorphism σ of F_d, denoted by $E_1^*(\sigma)$, is the endomorphism of $\mathfrak{F}_d$ defined on a function $\mathcal{F}$ by:*

$$E_1^*(\sigma)(\mathcal{F}) : (\vec{x}, i) \mapsto \sum_{j|\sigma(i)=p\cdot j\cdot s} \mathcal{F}(M_\sigma\vec{x} + \vec{f}(p), j) - \sum_{j|\sigma(i)=p\cdot j^{-1}\cdot s} \mathcal{F}(M_\sigma\vec{x} + \vec{f}(p) - \vec{e}_j, j).$$

Note that the value of $E_1^*(\sigma)(\mathcal{F})$ in $(\vec{x}, i)$ is finite since it depends only on the values of $\mathcal{F}$ over a finite subset of $\mathbb{Z}^d \times \{1, \ldots, d\}$. This shows that $E_1^*(\sigma)$ is an endomorphism of $\mathfrak{F}_d$.

The formula defining $E_1^*(\sigma)$ can seem obfuscating: we refer to [4,5,12] for a detailed presentation. Here, let us just mention that the subscript of $E_1^*(\sigma)$ refers to the fact that this is a map acting over $(d-1)$-dim. faces of d-dim. unit cubes, while the superscript refers to a sort of duality between this map and σ. Before providing an intuitive viewpoint on dual maps, let us consider a simple example, and then give an important property of dual maps.

Example 2 *The dual map of the substitution σ introduced in Ex. 1 satisfies:*

$$E_1^*(\sigma) \;:\; \begin{cases} (\vec{0}, 1^*) \mapsto (\vec{0}, 1^*) + (\vec{0}, 2^*) + (\vec{0}, 3^*), \\ (\vec{0}, 2^*) \mapsto (-\vec{e}_3, 1^*), \\ (\vec{0}, 3^*) \mapsto (-\vec{e}_3, 2^*). \end{cases}$$

The image of any function of $\mathfrak{F}_d$, that is, of a weighted sum of faces, can then *be easily computed by linearity. Fig. 3 illustrates this.*

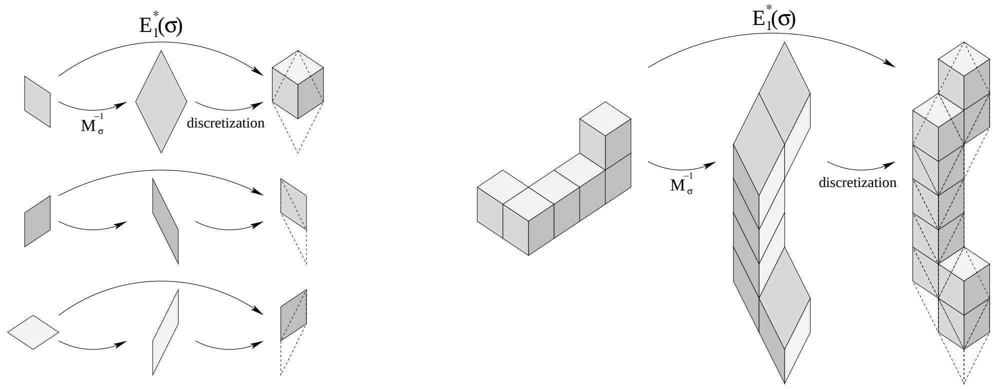

Fig. 3. Dual maps act over weighted sums of faces. Here is depicted the cases of single faces (left) and a binary function (right). The action of a dual map $E_1^*(\sigma)$ can be intuitively seen as a digitization of the linear map M_σ^{-1}.

The following theorem describes the action of dual maps on stepped planes:

Theorem 1 ([7]) *Let σ be a unimodular endomorphism of F_d. Let $\vec{\alpha} \in \mathbb{R}_+^d \setminus \{\vec{0}\}$ and $\rho \in \mathbb{R}$. If $M_\sigma^\top \vec{\alpha} \in \mathbb{R}_+^d$, then the image of the stepped plane $\mathcal{P}_{\vec{\alpha},\rho}$ by $E_1^*(\sigma)$ is the stepped plane $\mathcal{P}_{M_\sigma^\top \vec{\alpha},\rho}$. Otherwise, this image is not a binary function.*

Note that, although the image by $E_1^*(\sigma)$ of a stepped plane is a stepped plane, the image of each face of this stepped plane is a weighted sum of faces (in particular, not necessarily binary). Note also that if σ is a substitution, then $M_\sigma^\top \vec{\alpha} \in \mathbb{R}_+^d$ holds for any $\vec{\alpha} \in \mathbb{R}_+^d \setminus \{\vec{0}\}$: the image of a stepped plane by the dual map of a substitution is thus always a stepped plane.

We are now in a position to provide a more intuitive viewpoint on dual maps. On the one hand, let us recall that the *stepped* plane of normal vector $\vec{\alpha}$ and intercept ρ can be viewed as the *digitization* of the *Euclidean* plane of normal vector $\vec{\alpha}$ and intercept ρ (that is, the set $\{\vec{x} \in \mathbb{R}^d \mid \langle \vec{x} | \vec{\alpha} \rangle = \rho\}$). On the other hand, one easily checks that M_σ^{-1} maps the *real* plane of normal vector $\vec{\alpha}$ and intercept ρ onto the *real* plane of normal vector $M_\sigma^\top \vec{\alpha}$ and intercept ρ. Thus, the previous theorem leads to consider the dual map $E_1^*(\sigma)$ as a *digitization* of the linear map M_σ^{-1} (Fig. 3). This viewpoint turns out to give a good intuition for numerous properties of dual maps (by the way, note that M_σ^{-1} plays a special role in the formula defining $E_1^*(\sigma)$).

To end this section, we examine action of dual maps over subsets of stepped plane, that is, binary functions lesser than or equal to stepped planes (in the sense of the usual partial order over functions). Such binary functions are said to be *planar*. Let us first consider the case of a unimodular substitution σ (recall that substitutions are *positive* morphisms). Let $\vec{\alpha} \in \mathbb{R}_+^d \setminus \{\vec{0}\}$ and $\rho \in \mathbb{R}$, and assume that $M_\sigma^\top \vec{\alpha} \in \mathbb{R}_+^d$. Let $\mathcal{B}$ be a binary function lesser than or equal to $\mathcal{P}_{\vec{\alpha},\rho}$. We set $\mathcal{B}' = \mathcal{P}_{\vec{\alpha},\rho} - \mathcal{B}$: this is also a binary function. One has:

$$\mathcal{P}_{M_\sigma^\top \vec{\alpha},\rho} = E_1^*(\sigma)(\mathcal{P}_{\vec{\alpha},\rho}) = E_1^*(\sigma)(\mathcal{B}) + E_1^*(\sigma)(\mathcal{B}').$$

It is not hard to see that the set of binary functions is stable under the action of dual maps of substitutions (in this case, positivity is preserved by the formula defining dual maps). One thus deduces from the above equation that $E_1^*(\sigma)(\mathcal{B})$ is lesser than or equal to $\mathcal{P}_{M_\sigma^\top \vec{\alpha},\rho}$. In other words, for a substitution σ, one has:

$$0 \leq \mathcal{B} \leq \mathcal{P}_{\vec{\alpha},\rho} \implies 0 \leq E_1^*(\sigma)(\mathcal{B}) \leq E_1^*(\sigma)(\mathcal{P}_{\vec{\alpha},\rho}).$$

This generally does no more hold when σ is not a substitution (negative values appear in the formula defining dual maps). Recall that this however holds if $\mathcal{B}$ is a stepped plane (Th. 1). Let us already mention that, in Sec. 5, we will show that it also holds in a particular but useful case (dual maps of special morphisms – namely inverse of the later defined Brun substitutions – acting over specially shaped subsets of stepped planes).

3 Brun algorithm

The *Brun algorithm* is one of the numerous multi-dimensional continued fractions algorithm. For a complete exposition of these algorithms, the reader is referred to [8,18]. Here, we just briefly recall the Brun algorithm and prove a simple property (Prop. 1, below). Let us stress that the choice of the Brun algorithm is rather arbitrary: most of other matricial algorithms could be used; here, we just need the matrix linking two steps to be unimodular and finiteness of rational expansions.

Definition 4 *The* Brun map T *is the map from* $[0,1]^d \backslash \{\vec{0}\}$ *to* $[0,1]^d$ *defined on* $\vec{\alpha} = (\alpha_1, \ldots, \alpha_d)$ *by:*

$$T(\alpha_1, \ldots, \alpha_d) = \left(\frac{\alpha_1}{\alpha_i}, \ldots, \frac{\alpha_{i-1}}{\alpha_i}, \frac{1}{\alpha_i} - \left\lfloor \frac{1}{\alpha_i} \right\rfloor, \frac{\alpha_{i+1}}{\alpha_i}, \ldots, \frac{\alpha_d}{\alpha_i}\right),$$

where $i = \min\{j \mid \alpha_j = ||\vec{\alpha}||_\infty\}$. *The* Brun expansion *of a vector* $\vec{\alpha} \in [0,1]^d$ *is the sequence* $(a_n, i_n)_{n\geq 0}$ *of* $\mathbb{N}^* \times \{1, \ldots, d\}$ *defined, while* $T^n(\vec{\alpha}) \neq \vec{0}$, *by:*

$$a_n = \left\lfloor ||T^n(\vec{\alpha})||_\infty^{-1} \right\rfloor \quad \textit{and} \quad i_n = \min\{j \mid \langle T^n(\vec{\alpha}) | \vec{e}_j \rangle = ||T^n(\vec{\alpha})||_\infty\}.$$

Let us stress that, in the $d = 1$ case, the Brun map T is nothing but the classic Gauss map, and if $(a_n, i_n)_{n\geq 0}$ is the Brun expansion of $\vec{\alpha} \in [0,1]$, then $(a_n)_n$ is the continued fraction expansion of $\vec{\alpha}$, while, for all n, $i_n = 1$. We call *Brun algorithm* the process which computes Brun expansion by applying T.

Example 3 *The Brun expansion of* $(\frac{3}{8}, \frac{5}{12})$ *is* $(2,2),(1,1),(2,2),(4,1),(1,2)$.

Let us now give a matrix viewpoint. For $(a,i) \in \mathbb{N} \times \{1,\ldots,d\}$, one introduces the following $(d+1)\times(d+1)$ symmetric and unimodular matrix:

$$B_{a,i} = \begin{pmatrix} a & & 1 & \\ & I_{i-1} & & \\ 1 & & 0 & \\ & & & I_{d-i} \end{pmatrix},$$

where I_p stands for the $p\times p$ identity matrix. For $\vec{\alpha} = (\alpha_1,\ldots,\alpha_d) \in [0,1]^d \backslash \{\vec{0}\}$, $i = \min\{j \mid \alpha_j = ||\vec{\alpha}||_\infty\}$ and $a = \lfloor \alpha_i^{-1} \rfloor$, a simple computation then shows:

$$(1,\vec{\alpha}) = ||\vec{\alpha}||_\infty B_{a,i}(1, T(\vec{\alpha})), \qquad (1)$$

where, for any vector $\vec{u}$, $(1,\vec{u})$ stands for the vector obtained by adding to $\vec{u}$ a first entry equal to 1.

To end this section, we give a bound on length of Brun expansions of rational vectors (irrational ones clearly having infinite expansions):

Proposition 1 *With any rational vector* $\vec{\alpha}$ *is associated the following integer:*

$$m(\vec{\alpha}) = p_1 + \ldots + p_d + q,$$

where $p_1,\ldots,p_d$ *and* q *are co-prime integers such that* $\vec{\alpha} = (p_1/q,\ldots,p_d/q)$. *Then, the length of the Brun expansion of* $\vec{\alpha} \in [0,1]^d \cap \mathbb{Q}^d$ *is bounded by:*

$$\log_{\frac{d+2}{d+1}}(m(\vec{\alpha})),$$

where $\log_a$ *denotes the logarithm to the base* a.

Proof. We show by induction on $k \geq 1$ that if $\vec{\alpha}$ is a rational vector whose Brun expansion has length k, then one has:

$$m(\vec{\alpha}) \geq \left(\frac{d+2}{d+1}\right)^k.$$

For $k=1$, $T(\vec{\alpha}) = \vec{0}$ yields:

$$\vec{\alpha} = (0,\ldots,0,1/q,0,\ldots,0),$$

where $q \geq 1$. Hence, one has: $m(\vec{\alpha}) = 1 + q \geq 2 \geq (d+2)/(d+1)$. Let us now assume that the result holds for some $k \geq 1$. Let $\vec{\alpha} \in [0,1]^d$ be a rational vector whose Brun expansion has length $k+1$. Since $T(\vec{\alpha})$ is rational, one has:

$$T(\vec{\alpha}) = (p_1/q, \ldots, p_d/q),$$

where $p_1, \ldots, p_d$ and q are co-prime integers, that is, $\gcd(p_1, \ldots, p_d, q) = 1$. The definition of T then yields that there is $(a, i) \in \mathbb{N}^* \times \{1, \ldots, d\}$ such that:

$$\vec{\alpha} = \left(\frac{p_1}{aq+p_i}, \ldots, \frac{p_{i-1}}{aq+p_i}, \frac{q}{aq+p_i}, \frac{p_{i+1}}{aq+p_i}, \ldots, \frac{p_d}{aq+p_i}\right).$$

Since $\gcd(p_1, \ldots, p_{i-1}, q, p_{i+1}, \ldots, p_d, aq+p_i) = \gcd(p_1, \ldots, p_d, q) = 1$, one has:

$$m(\vec{\alpha}) = p_1 + \ldots + p_{i-1} + q + p_{i+1} + \ldots + p_d + aq + p_i,$$

that is, $m(\vec{\alpha}) = m(T(\vec{\alpha})) + aq$. But $a \geq 1$ and $q \geq p_1, \ldots, p_d$. Hence:

$$m(\vec{\alpha}) \geq m(T(\vec{\alpha})) + \frac{p_1 + \ldots + p_d + q}{d+1} = \frac{d+2}{d+1} m(T(\vec{\alpha})).$$

The result thus holds for $k+1$. This completes the proof. $\square$

It is not hard to find examples showing that the bound of Prop. 1 is not tight. The characterization of rational vectors of longest Brun expansions remains an open question (except in the one-dimensional case, where they are ratios of successive Fibonacci numbers).

4 Generation of stepped planes

In this section, we show how notions and results of previous sections can be used to generate stepped planes. Let us first introduce special substitutions:

Definition 5 *Let $(a, i) \in \mathbb{N}^* \times \{1, \ldots, d\}$. The* Brun substitution *associated with (a, i), denoted by $\beta_{a,i}$, is defined over F_{d+1} by:*

$$\beta_{a,i}(1) = 1^a \cdot (i+1), \quad \beta_{a,i}(i+1) = 1, \quad \forall j \notin \{1, i+1\},\ \beta_{a,i}(j) = j.$$

Note that the incidence matrix of $\beta_{a,i}$ is the matrix $B_{a,i}$ introduced in the previous section (in particular, Brun substitutions are unimodular). We thus deduce from Eq. (1) and Th. 1 (recall that $B_{a,i}$ is symmetric) that, for any $\vec{\alpha} \in [0,1]^d \backslash \{\vec{0}\}$ and $\rho \in \mathbb{R}$, one has:

$$\mathcal{P}_{(1,\vec{\alpha}),\rho} = E_1^*(\beta_{a,i})(\mathcal{P}_{||\vec{\alpha}||_\infty(1,T(\vec{\alpha})),\rho}) = E_1^*(\beta_{a,i})(\mathcal{P}_{(1,T(\vec{\alpha})),\rho/||\vec{\alpha}||_\infty}), \qquad (2)$$

where the last equality just easily follows from the definition of stepped planes.

Let us now apply this to the generation of stepped planes. The idea is that a stepped plane of normal vector $\vec{e}_1$ can be easily generated: this is nothing but a sum of faces of type 1 translated along the lattice $\mathbb{Z}\vec{e}_2+\ldots+\mathbb{Z}\vec{e}_{d+1}$. Then, it easily follows from what precedes that any stepped plane whose normal vector has a finite Brun expansion (that is, is rational) can be similarly generated. We can state this more precisely with the following theorem:

Theorem 2 *Let $\vec{\alpha} \in [0,1]^d \cap \mathbb{Q}^d$ having (finite) Brun expansion $(a_n, i_n)_{0\le n\le N}$. Let $\rho \in \mathbb{R}$ and $\rho' = \rho/||B_{a_0,i_0} \times \ldots \times B_{a_N,i_N}\vec{e}_1||_\infty$. Let $\mathcal{D}_{(1,\vec{\alpha}),\rho}$ be defined by:*

$$\mathcal{D}_{(1,\vec{\alpha}),\rho} = E_1^*(\beta_{a_0,i_0}) \circ \ldots \circ E_1^*(\beta_{a_N,i_N})(\rho'\vec{e}_1, 1^*).$$

Let $L_{(1,\vec{\alpha}),\rho} \subset \mathbb{Z}^{d+1}$ be the lattice of rank d defined by:

$$L_{(1,\vec{\alpha}),\rho} = B_{a_0,i_0}^{-1} \ldots B_{a_N,i_N}^{-1} \sum_{k=2}^{d+1} \mathbb{Z}\vec{e}_k.$$

Then, $\mathcal{P}_{(1,\vec{\alpha}),\rho}$ is the sum of $\mathcal{D}_{(1,\vec{\alpha}),\rho}$ translated along the lattice $L_{(1,\vec{\alpha}),\rho}$.

Example 4 *Fig. 4 shows the generation of the binary function $\mathcal{D}_{(1,3/8,5/12),0}$ by the dual maps of the Brun substitutions associated with the Brun expansion of the vector $(3/8, 5/12)$ (recall Ex. 3). One also computes:*

$$L_{(1,3/8,5/12),0} = \mathbb{Z}(\vec{e}_1 + 4\vec{e}_2 - 6\vec{e}_3) + \mathbb{Z}(2\vec{e}_1 - 2\vec{e}_2 - 3\vec{e}_3).$$

Th. 2 then yields that the geometrical interpretation of $\mathcal{P}_{(1,3/8,5/12),0}$ is the union of all the translations along $L_{(1,3/8,5/12),0}$ of the geometrical interpretation of $\mathcal{D}_{(1,3/8,5/12),0}$ (Fig. 5).

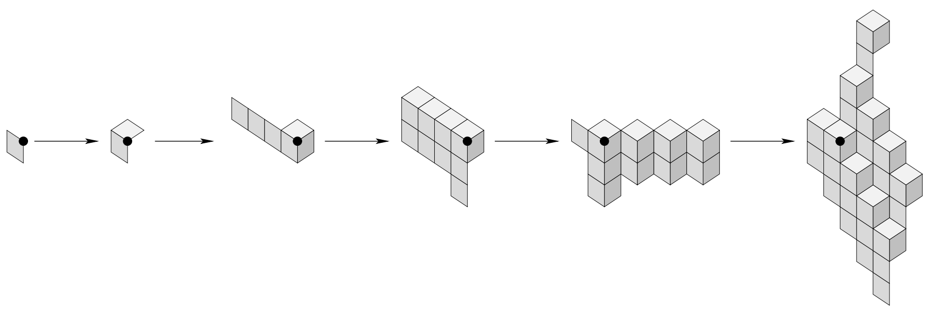

Fig. 4. Generation of $\mathcal{D}_{(1,3/8,5/12),0}$ by applications of the dual maps $E_1^*(\beta_{1,2})$, $E_1^*(\beta_{4,1})$, $E_1^*(\beta_{2,2})$, $E_1^*(\beta_{1,1})$ and $E_1^*(\beta_{2,2})$ (from left to right – highlighted origin).

Note that Th. 2 does not say anything about the *shape* of binary functions $\mathcal{D}_{(1,\vec{\alpha}),\rho}$, although it is clearly a critical point for practical generation of stepped planes. Intuitively, we would like a small and "compact" $\mathcal{D}_{(1,\vec{\alpha}),\rho}$. In fact, the size of any binary function which generate $\mathcal{P}_{(1,\vec{\alpha}),\rho}$ by translations along $L_{(1,\vec{\alpha}),\rho}$ is the size of the fundamental domain of $L_{(1,\vec{\alpha}),\rho}$ (if $\vec{\alpha} = (p_1/q, \ldots, p_d/q)$ with

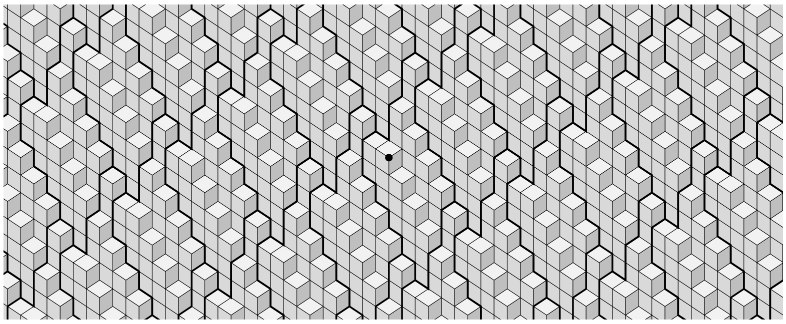

Fig. 5. According to Th. 2, the stepped plane $\mathcal{P}_{(1,3/8,5/12),0}$ can be generated by translated copies of $\mathcal{D}_{(1,3/8,5/12),0}$ (framed).

$p_1, \ldots, p_d$ and q co-primes, then this size is $q + p_1 + \ldots + p_d$). Thus, the real question deals with the shape of $\mathcal{D}_{(1,\vec{\alpha}),\rho}$. How can the compactness of $\mathcal{D}_{(1,\vec{\alpha}),\rho}$ be quantified? For example, denoting by s the size of $\mathcal{D}_{(1,\vec{\alpha}),\rho}$ and by b the number of faces of $\mathcal{P}_{(1,\vec{\alpha}),\rho}$ which are not in $\mathcal{D}_{(1,\vec{\alpha}),\rho}$ but adjacent to it (the "boundary" of $\mathcal{D}_{(1,\vec{\alpha}),\rho}$), we can consider that the smaller the ratio b^d/s^{d-1} is the better the shape is; this yields that good shapes look like balls. According to this criterion, $\mathcal{D}_{(1,\vec{\alpha}),\rho}$ seems to be generally rather compact (Fig. 6). We do not provide here bounds on such a compactness measure (it should be investigated using the notion of *boundary endomorphisms*, introduced in [5]).

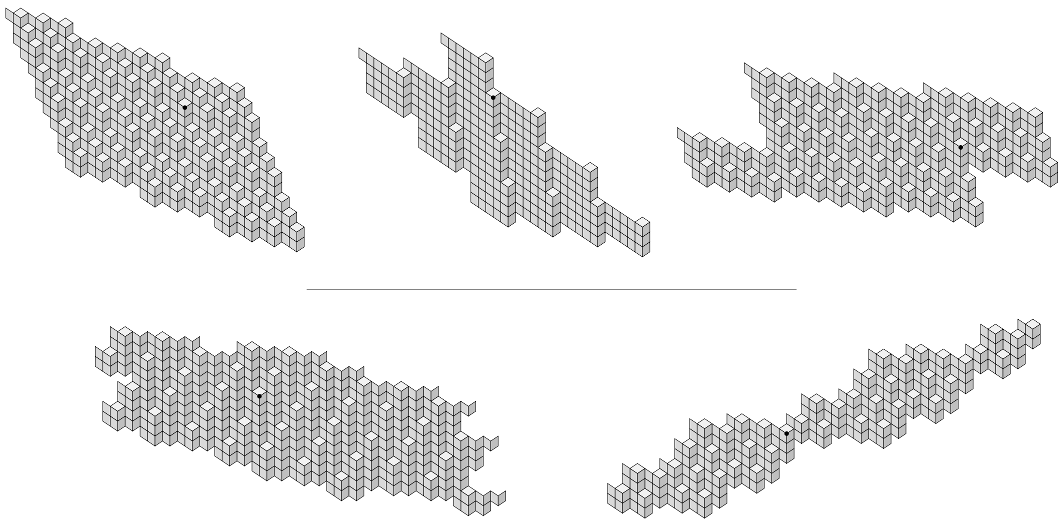

Fig. 6. Examples of binary functions $\mathcal{D}_{\vec{\alpha},\rho}$ for some rational vectors $\vec{\alpha}$ (see Th. 2). Here, ratio b^2/s takes values 25, 39, 40, 45 and 80 (from top-left to bottom-right).

To conclude this section, let us briefly mention the case where $\mathcal{P}_{(1,\vec{\alpha}),\rho}$ is an irrational stepped plane, that is, $\vec{\alpha}$ has an infinite Brun expansion $(a_n, i_n)_n$. In particular, if this expansion is periodic with period p and if $\vec{\alpha}_k$ denotes the rational vector whose Brun expansion is $(a_n, i_n)_{n \leq k \times p}$, then $(\mathcal{D}_{(1,\vec{\alpha}_k),\rho})_k$ turns out to be an increasing sequence of subsets of $\mathcal{P}_{(1,\vec{\alpha}),\rho}$ which, under some conditions, eventually contains any finite subset of $\mathcal{P}_{(1,\vec{\alpha}),\rho}$ (see [14], chap. 6).

5 Recognition of stepped planes

5.1 Principle

In the previous section, dual maps and the Brun algorithm were used to generate stepped planes from their normal vectors. Here, the idea is to process backwards. Let us first provide, in this section, an overview of the principle of the recognition algorithm which will be detailed in the next sections.

One easily checks that Brun substitutions are invertible. Eq. (2) thus can be rewritten as follows:

$$\mathcal{P}_{(1,T(\vec{\alpha})),\rho/||\vec{\alpha}||_\infty} = E_1^*(\beta_{a,i}^{-1})(\mathcal{P}_{(1,\vec{\alpha}),\rho}). \tag{3}$$

In particular, note that only the quantities a and i are required to transform a stepped plane of normal vector $(1,\vec{\alpha})$ into a stepped plane of normal vector $(1,T(\vec{\alpha}))$. Since $i = \min\{j \mid \alpha_j = ||\vec{\alpha}||_\infty\}$ and $a = \lfloor \alpha_i^{-1} \rfloor$, they depend on $\vec{\alpha}$. However, let us stress that they contain only a *partial* information on $\vec{\alpha}$. This information turns out to be easily deducible from local configurations of a stepped plane of normal vector $(1,\vec{\alpha})$ – this is the point of Sec. 5.2, below. In other words, given a stepped plane whose normal vector is unknown, we know how to obtain the first term of the Brun expansion of this vector (that is, a and i) as well as the stepped plane whose normal vector is the image of the previous one by the Brun map T. Assuming that the initial vector has a finite Brun expansion, we thus can iterate the process up to obtaining a stepped plane of normal vector $(1,\vec{0})$, which is easily recognizable, as well as the Brun expansion of the initial vector.

Note that the the previous process is rather theoretical, since stepped planes are infinite objects. Moreover, this just allows to compute the normal vector of a given stepped plane, not to decide whether a given binary function is planar or not. However, these problems can be fixed by some slight modifications of the previous process. Indeed, the way partial information on its normal vector is deduced from a stepped plane can be extended to any binary function, under some mild conditions (end of Sec. 5.2). Of course, the information obtained in this way cannot always be linked with a normal vector, since this notion makes sense only for stepped planes. In particular, this information can be "inconsistent", that is, cannot be linked with any possible normal vector: in such a case, the binary function is clearly non-planar. Otherwise, the binary function "locally looks planar" (this will be later formalized).

This yields the following process of recognition. While the current binary function "locally looks planar", we apply $E_1^*(\beta_{a,i}^{-1})$, that is, we perform what we call a *Brun step*. If we eventually get a subset of a stepped plane of normal vector $(1, \vec{0})$, then the initial binary function is planar (as an image by dual maps of substitutions of planar binary function). Else, if we eventually get a binary function which is clearly not planar, then the initial binary function is also not planar (this is not completely evident, we prove it in Sec. 5.3). This process is formalized in Sec. 5.4. Bounds on the maximal number of Brun steps that ensure termination are also proposed there.

5.2 Runs

Runs are special subsets of binary functions:

Definition 6 *An* (i, j)-run *of a binary function* $\mathcal{B}$ *is a binary function* $\mathcal{R} \leq \mathcal{B}$ *which can be written, for some vector* $\vec{x} \in \mathbb{Z}^d$ *and maximal interval* $I \subset \mathbb{Z}$*:*

$$\mathcal{R} = \sum_{k \in I} (\vec{x} + k\vec{e}_j, i^*).$$

Such a run is right-closed *if* I *has a right endpoint* b*, with* $\mathcal{B}(\vec{x} + b\vec{e}_j, j^*) = 1$*. It is* left-closed *if* I *has a left endpoint* a*, with* $\mathcal{B}(\vec{x} + (a-1)\vec{e}_j + \vec{e}_i, j^*) = 1$*. The terms* closed*,* open*,* right-open *and* left-open *are then defined as for intervals.*

In other words, the geometric interpretation of an (i, j)*-run* of a binary function $\mathcal{B}$ is a maximal sequence of contiguous faces of type i, aligned with the direction $\vec{e}_j$, and included in the geometric interpretation of $\mathcal{B}$ (see Fig. 7).

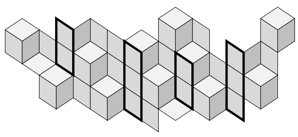

Fig. 7. This binary function has every type of $(1, 3)$-runs: left-closed, right-closed, closed and open (framed runs, from left to right).

The following proposition links runs with normal vectors of stepped planes:

Proposition 2 *A stepped plane of normal vector* $\vec{\alpha} = (\alpha_1, \ldots, \alpha_d)$ *has* (i, j)*-runs of size* $\max(\lfloor \alpha_i/\alpha_j \rfloor, 1)$ *and* $\max(\lceil \alpha_i/\alpha_j \rceil, 1)$ *if* $\alpha_j \neq 0$*, infinite otherwise.*

Proof. Let $\vec{x} \in \mathbb{Z}^d$ and $I \subset \mathbb{Z}$ such that the following binary function is an (i, j)-run of the stepped plane $\mathcal{P}_{\vec{\alpha},\rho}$:

$$\mathcal{R} = \sum_{k \in I} (\vec{x} + k\vec{e}_j, i^*).$$

Assume that I contains an interval $[a, b]$, of length $b - a + 1$. Then, one has:

$$\mathcal{P}_{\vec{\alpha},\rho}(\vec{x} + a\vec{e}_j, i) = 1 \;\Rightarrow\; \langle \vec{x} | \vec{\alpha} \rangle + a\alpha_j < \rho \leq \langle \vec{x} | \vec{\alpha} \rangle + a\alpha_j + \alpha_i,$$

$$\mathcal{P}_{\vec{\alpha},\rho}(\vec{x}+b\vec{e}_j,i)=1 \ \Rightarrow\ \langle\vec{x}|\vec{\alpha}\rangle+b\alpha_j<\rho\leq\langle\vec{x}|\vec{\alpha}\rangle+b\alpha_j+\alpha_i.$$

One deduces:

$$(b-a)\alpha_j<\rho-\langle\vec{x}|\vec{\alpha}\rangle\leq\alpha_i,$$

that is, for $\alpha_j\neq 0$:

$$b-a+1<\frac{\alpha_i}{\alpha_j}+1.$$

We thus have an upper bound on the length of I.
Let us now assume that $I=[a,b]$. Then, one has:

$$\mathcal{P}_{\vec{\alpha},\rho}(\vec{x}+a\vec{e}_j,i)=1 \ \Rightarrow\ \langle\vec{x}|\vec{\alpha}\rangle+(a-1)\alpha_j<\langle\vec{x}|\vec{\alpha}\rangle+a\alpha_j<\rho,$$

and one deduces:

$$\mathcal{P}_{\vec{\alpha},\rho}(\vec{x}+(a-1)\vec{e}_j,i)=0 \ \Rightarrow\ \rho>\langle\vec{x}|\vec{\alpha}\rangle+(a-1)\alpha_j+\alpha_i.$$

Similarly, one shows:

$$\rho\leq\langle\vec{x}|\vec{\alpha}\rangle+(b+1)\alpha_j+\alpha_i.$$

Finally, one has:

$$(a-1)\alpha_j+\alpha_i<\rho-\langle\vec{x}|\vec{\alpha}\rangle\leq(b+1)\alpha_j,$$

that is, for $\alpha_j\neq 0$:

$$b-a+1>\frac{\alpha_i}{\alpha_j}-1.$$

We thus have a lower bound on the length of I.
In conclusion, we shown that size of (i,j)-runs of $\mathcal{P}_{\vec{\alpha},\rho}$ range in the open interval $(\alpha_i/\alpha_j-1,\alpha_i/\alpha_j+1)$. The result follows (recall that, by definition, runs are of size at least 1). □

Given a stepped plane with an unknown normal vector $(1,\vec{\alpha})=(1,\alpha_1,\ldots,\alpha_d)$, this proposition allows to use runs for computing $i=\min\{j\mid\alpha_j=||\vec{\alpha}||_\infty\}$ and $a=\lfloor 1/\alpha_i\rfloor$. Indeed, i is the smallest integer such that no $(j+1,i+1)$-run has size greater than 1, for $j=1,\ldots,d$, while a is the size of the smallest $(1,i+1)$ run. Note that one should also check that there is no $(j+1,1)$-run of size greater than 1 (this proves that $\vec{\alpha}\in[0,1]^d$) and that there is at least one $(1,j+1)$-run of finite size (this proves that $\vec{\alpha}\neq\vec{0}$).

Example 5 *Consider the stepped plane of Fig. 2. Let $(1,\alpha,\beta)$ denotes its unknown*[1] *normal vector. Its $(2,1)$- and $(3,1)$-runs are of size 1, and its $(1,2)$- and $(1,3)$-runs have finite size. Thus, $(\alpha,\beta)\in[0,1]^d\backslash\{\vec{0}\}$. Its $(2,3)$-runs are*

[1] Here, in fact, we already know that $(1,\alpha,\beta)=(1,3/8,5/12)=(24,9,10)/24$. One thus can check that the first term of the Brun expansion here obtained matches with the Brun expansion of $(3/8,5/12)$ computed in Ex. 3.

all of size 1*. Thus,* $\beta \geq \alpha$*, that is,* $i = 2$*. And since its* $(1, i+1)$*-runs have size* 2 *or* 3*, one has* $a = 2$*. This finally shows that the first term of the Brun expansion of* (α, β) *is* $(a, i) = (2, 2)$*.*

Following the principle described in Sec. 5.1, we now would like to extend this to binary functions. It is not hard to deduce from Prop. 2 that planar binary functions satisfy the following proposition:

Proposition 3 *If a binary function* $\mathcal{B}$ *is a subset of a stepped plane of normal vector* $\vec{\alpha} = (\alpha_1, \ldots, \alpha_d)$*, then* $\mathcal{B}$ *has* (i, j)*-runs of size at most* $\max(\lceil \alpha_i/\alpha_j \rceil, 1)$ *and closed runs of size at least* $\max(\lfloor \alpha_i/\alpha_j \rfloor, 1)$*.*

We then introduce the notion of *local planarity*:

Definition 7 *A binary function is said to be* locally planar *if:*

(1) for $i \neq j$*, there is no* (i, j)*-run and* (j, i)*-run both of size greater than* 1*;*
(2) if a closed (i, j)*-run has size* a*, then* (i, j)*-runs have size of at most* $a+1$*.*

Local planarity means that runs do not contradict planarity. Indeed, if $\mathcal{B}$ is a subset of a stepped plane of normal vector $\vec{\alpha}$, the first condition means that one cannot have both $\alpha_i > \alpha_j$ and $\alpha_i < \alpha_j$, while the second condition means that floor and ceiling parts of α_i/α_j differ by at most 1. For example, one checks that the binary function of Fig. 7 is locally planar. In fact, local planarity corresponds to the informal "locally looks planar" of Sec. 5.1.

However, local planarity does not ensure that we can always deduce from runs enough information to perform a Brun step, as explained in Sec. 5.1. Fig. 8 (right) illustrates this. This leads to introduce 1-*recognizable* binary functions, which are those whose runs contains enough information to perform one Brun step [2] . Technically:

Definition 8 *A binary function* $\mathcal{B} \in \mathfrak{B}_{d+1}$ *is said to be* 1-recognizable *if:*

(1) there is $i \in \{1, \ldots, d\}$ *such that* $\mathcal{B}$ *has a* $(1, i+1)$*-run of size greater than* 1 *and, for any* $j \neq i$*, an* $(i+1, j+1)$*-run of size greater than* 1*;*
(2) for such an i*,* $\mathcal{B}$ *has a closed* $(1, i+1)$*-run of size* $a \in \mathbb{N}^*$ *and a* $(1, i+1)$*-run of size greater than* a*.*

The first condition ensures that, if $\mathcal{B}$ is a subset of a stepped plane of normal vector $(1, \vec{\alpha})$, then the i-th entry of $\vec{\alpha}$ is smaller than 1 and greater than any other entry. The second condition ensures that, if $\mathcal{B}$ is a subset of a stepped plane of normal vector $(1, \vec{\alpha})$, then the size of the smallest $(1, i+1)$-runs of this stepped plane is a, that is, $\lfloor 1/\alpha_i \rfloor$. For example, the binary function of

[2] The 1 of 1-*recognizable* emphasizes that this holds for *one* Brun step

Fig. 7 is 1-recognizable, with $(a, i) = (2, 2)$. We will later examine the problem of non-1-recognizable binary functions.

In all what follows, if $\mathcal{B}$ is a 1-recognizable binary function, then a and i will implicitly stand for integers a and i appearing in Def. 8.

Note that 1-recognizability and local planarity are independent notions, as illustrated by Fig. 8. The "good" case is the one of a locally planar and 1-recognizable binary function: we thus know that we must perform a Brun step using the dual map $E_1^*(\beta_{a,i}^{-1})$. We focus on this case in the next section.

Fig. 8. A binary function having a closed $(1, 3)$-run of size 1 and a $(1, 3)$-run of size 3 is not locally planar, although it can be 1-recognizable (left, with $(a, i) = (1, 2)$). Conversely, a locally planar binary function can be not 1-recognizable (right: size of runs of stepped planes containing this binary function can range in $\{2, 3, 4\}$).

5.3 Brun step and boundary problems

Let $\mathcal{B} \in \mathfrak{B}_{d+1}$ be a locally planar 1-recognizable binary function. The previous section shows that runs give integers a and i such that if $\mathcal{B}$ is a subset of a stepped plane of normal vector $(1, \vec{\alpha})$, then $E_1^*(\beta_{a,i}^{-1})$ maps this stepped plane onto a stepped plane of normal vector $(1, T(\vec{\alpha}))$. However, we would like to apply $E_1^*(\beta_{a,i}^{-1})$ on $\mathcal{B}$, not on some *hypothetical* stepped plane which would contain $\mathcal{B}$. In other words, we would like to ensure:

$$0 \leq \mathcal{B} \leq \mathcal{P}_{(1,\vec{\alpha}),\rho} \;\Rightarrow\; 0 \leq E_1^*(\beta_{a,i}^{-1})(\mathcal{B}) \leq \mathcal{P}_{(1,T(\vec{\alpha})),\rho/||\vec{\alpha}||_\infty}. \qquad (4)$$

Indeed, assuming that $E_1^*(\beta_{a,i}^{-1})(\mathcal{B})$ is 1-recognizable, we could iterate this up to recognize a planar binary function or to get a not locally planar binary function. But $\beta_{a,i}^{-1}$ is not positive (thus not a substitution), since one computes:

$$\beta_{a,i}^{-1}(1) = (i+1), \quad \beta_{a,i}^{-1}(i+1) = (i+1)^{-a} \cdot 1, \quad \forall j \notin \{1, i+1\},\ \beta_{a,i}^{-1}(j) = j.$$

Hence, we cannot rely on the end of Sec. 2 to ensure that (4) holds.

Actually, (4) fails because of boundary problems. Indeed, $E_1^*(\beta_{a,i}^{-1})$ maps each face of $\mathcal{B}$ to a sum of weighted faces (recall Def. 3). Roughly speaking, when $\mathcal{B}$ is an entire stepped plane, images of faces are sums which add up or cancel so that we finally get a stepped plane (Th. 1). But if $\mathcal{B}$ is smaller, these additions/cancellations can be incomplete, so that "artifacts" will remain near boundaries.

The idea is to avoid these boundary problems by slightly modifying $\mathcal{B}$. More precisely, let us associate with any binary function $\mathcal{B}$ the set of its *admissible parameters* $P(\mathcal{B})$:

$$P(\mathcal{B}) = \{(\vec{\alpha}, \rho) \in [0,1]^d \backslash \{\vec{0}\} \times \mathbb{R} \mid \mathcal{B} \leq \mathcal{P}_{(1,\vec{\alpha}),\rho}\}.$$

This set turns out to be a convex set of $\mathbb{R}^{d+1}$, empty if $\mathcal{B}$ is not planar. Moreover, it is open if $\mathcal{B}$ is a subset of a rational stepped plane; in particular, this is the case if $\mathcal{B}$ is planar and *finite*. Let us stress that recognizing $\mathcal{B}$ or any binary function of $P(\mathcal{B})$ is strictly equivalent. In particular, we would be especially interested in finding a binary function in $P(\mathcal{B})$ satisfying (4). The end of this section explains how this can be done.

Let us define three *transformation rules* acting over binary functions (Fig. 9):

Definition 9 *Let* $a \in \mathbb{N}^*$ *and* $i \in \{1, \ldots, d\}$*. The rule* $\phi_{a,i}$ *left-extends any right-closed and left-open* $(1, i+1)$*-run into a run of size* a*; the rule* $\psi_{a,i}$ *right-closes any right-open* $(1, i+1)$*-run of size greater than* a*; the rule* χ_i *removes any left-closed and right-open* $(1, i+1)$*-run.*

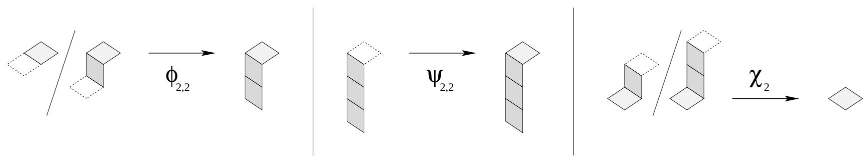

Fig. 9. Definition of transformation rules $\phi_{a,i}$, $\psi_{a,i}$ and χ_i relies on runs.

Fig. 10 shows how the locally planar 1-recognizable binary function of Fig. 7 is transformed by the three above transformation rules.

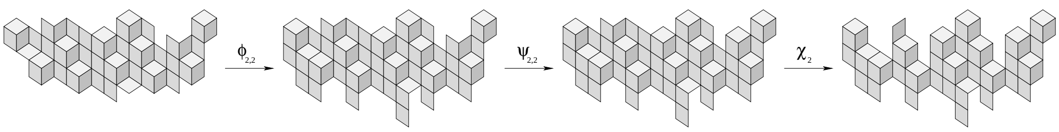

Fig. 10. Successive applications of rules $\phi_{a,i}$, $\psi_{a,i}$ and χ_i in the $(a, i) = (2, 2)$ case. Leftmost and rightmost binary functions have the same admissible parameters (Prop. 4, below), with the latter satisfying (4) (Prop. 5, below).

Proposition 4 *Let* $\mathcal{B} \in \mathfrak{B}_{d+1}$ *be a* 1*-recognizable binary function. Let* $\mathcal{B}'$ *be the binary function obtained applying* $\phi_{a,i}$*,* $\psi_{a,i}$ *and* χ_i *on* $\mathcal{B}$*. Then,* $P(\mathcal{B}) = P(\mathcal{B}')$*.*

Proof. Suppose that there is a stepped plane $\mathcal{P}$ such that $\mathcal{B} \leq \mathcal{P}$. Thus, any left-open and right-closed $(1, i+1)$-run of $\mathcal{B}$ is lesser than or equal to a closed $(1, i+1)$-run of $\mathcal{P}$. Since such a run has size at least a, this ensures that $\phi_{a,i}(\mathcal{B})$

is still lesser than or equal to $\mathcal{P}$. Conversely, if $\phi_{a,i}(\mathcal{B})$ is lesser than or equal to $\mathcal{P}$, then $\mathcal{B}$ also since $\mathcal{B} \leq \phi_{a,i}(\mathcal{B})$. This shows that $\mathcal{B} \leq \mathcal{P}$ if and only if $\phi_{a,i}(\mathcal{B}) \leq \mathcal{P}$. One similarly proceeds for $\psi_{a,i}$ and χ_i, so that, finally, $\mathcal{B} \leq \mathcal{P}$ if and only if $\mathcal{B}' \leq \mathcal{P}$. This proves $P(\mathcal{B}) = P(\mathcal{B}')$. □

Proposition 5 *Let $\mathcal{B} \in \mathfrak{B}_{d+1}$ be a 1-recognizable binary function. Let $\mathcal{B}'$ be the binary function obtained applying $\phi_{a,i}$, $\psi_{a,i}$ and χ_i on $\mathcal{B}$. If $\mathcal{B}'$ does not have open $(1, i+1)$-runs, then (4) holds, that is:*

$$0 \leq \mathcal{B}' \leq \mathcal{P}_{(1,\vec{\alpha}),\rho} \;\Rightarrow\; 0 \leq E_1^*(\beta_{a,i}^{-1})(\mathcal{B}') \leq \mathcal{P}_{(1,T(\vec{\alpha})),\rho/||\vec{\alpha}||_\infty}.$$

Proof. Assume that $\mathcal{B}'$ does not have open $(1, i+1)$-run. In this case, one checks that $\mathcal{B}'$ can be uniquely written as the image by $E_1^*(\beta_{a,i})$ of a binary function, say $\mathcal{B}''$ (rules $\phi_{a,i}$, $\psi_{a,i}$ and χ_i have been specially defined for). One also checks that $\mathcal{B}'$ is 1-recognizable, as well as $\mathcal{B}$. In particular, $E_1^*(\beta_{a,i}^{-1})(\mathcal{B}')$ is a binary function. Now, assume that there is a stepped plane $\mathcal{P}$ such that $\mathcal{B}' \leq \mathcal{P}$ and $E_1^*(\beta_{a,i}^{-1})(\mathcal{P}) \geq 0$. Let us introduce the binary function $\mathcal{C} = \mathcal{P} - \mathcal{B}'$. Since both $\mathcal{P}$ and $\mathcal{B}'$ are images by $E_1^*(\beta_{a,i})$ of binary functions, it is also the case for $\mathcal{C}$; say $\mathcal{C} = E_1^*(\beta_{a,i})(\mathcal{C}')$, for some binary function $\mathcal{C}'$. Hence, applying $E_1^*(\beta_{a,i}^{-1})$ on $\mathcal{P}$ yields:

$$E_1^*(\beta_{a,i}^{-1})(\mathcal{P}) = E_1^*(\beta_{a,i}^{-1})(\mathcal{B}' + \mathcal{C}) = E_1^*(\beta_{a,i}^{-1})(\mathcal{B}') + \mathcal{C}' \geq E_1^*(\beta_{a,i}^{-1})(\mathcal{B}') = \mathcal{B}'' \geq 0.$$

Thus, we shown that one has, for any stepped plane $\mathcal{P}$:

$$0 \leq \mathcal{B}' \leq \mathcal{P} \;\Rightarrow\; 0 \leq E_1^*(\beta_{a,i}^{-1})(\mathcal{B}') \leq E_1^*(\beta_{a,i}^{-1})(\mathcal{P}).$$

Conversely, assume that $0 \leq E_1^*(\beta_{a,i}^{-1})(\mathcal{B}') \leq E_1^*(\beta_{a,i}^{-1})(\mathcal{P})$ for some stepped plane $\mathcal{P}$. Note that the set of positive functions of $\mathfrak{F}_{d+1}$ is stable under dual maps of substitutions. Thus, since $\beta_{a,i}$ is a substitution, applying $E_1^*(\beta_{a,i})$ yields $0 \leq \mathcal{B}' \leq \mathcal{P}$. This concludes the proof. □

Fig. 11 illustrates Prop. 5.

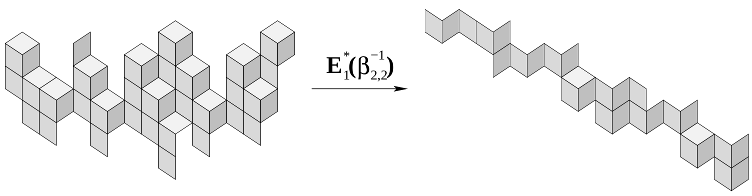

Fig. 11. Once transformation rules have been applied on a 1-recognizable binary function, the suitable dual map $E_1^*(\beta_{a,i}^{-1})$ can be used to perform a Brun step.

According to what precedes, it remains to address two problems related to iterating Brun steps:

(1) what is to be done when getting a non-1-recognizable binary function?
(2) what is to be done with open runs (recall hypothesis of Prop. 5)?

For the first problem, we rely on a generic recognition algorithm – here called `XReco` – which computes the admissible parameters of the first non-1-recognizable binary function we get (hence the term *hybrid*). For example, `XReco` can be a preimage algorithm, as described in [11]. Of course, if no Brun step is performed, then our algorithm is nothing but `XReco` itself. However, it is expected that non-1-recognizable binary functions are often small, since the bigger they are, the more information (runs) they contain. Thus, `XReco` would play a little role. But further experiments have to been carried out to make this point clearer. In particular, difficulties arise from the fact that this process heavily depends on the way initial binary functions are chosen.
For the second problem, a solution consists in removing (and storing) open runs when computing admissible parameters, and then checking how they modify these parameters (if they do).
This leads to the following algorithm, where $B'_{a,i}$ is the $(d+2)\times(d+2)$ block matrix whose first block is $B_{a,i}$ and the second the 1×1 identity matrix:

`HybridBrunReco`$(\mathcal{B})$

$n \leftarrow 0$;
$\mathcal{B}_0 \leftarrow \mathcal{B}$;
while $\mathcal{B}_n$ locally planar **and** 1-recognizable **do**
    read (a,i) on $\mathcal{B}_n$;
    $(a_n, i_n) \leftarrow (a,i)$;
    compute $\mathcal{B}'_n$;
    $L_n \leftarrow$ open runs of $\mathcal{B}'_n$;
    $\mathcal{B}_{n+1} \leftarrow E_1^*(\beta_{a_n,i_n}^{-1})(\mathcal{B}'_n - L_n)$;
    $n \leftarrow n+1$;
end while;
$P_n \leftarrow$ `XReco`$(\mathcal{B}_n)$;
for $k = n-1$ **downto** k=0 **do**
    $P_k \leftarrow B'_{a_k,i_k} P_{k+1}$;
    $P_k \leftarrow P_k \cap$ `XReco`(L_k);
end for;
return P_0;

Lines `1-10` correspond to iterated Brun steps ("Brun stage" of the algorithm), while lines `11` and `12-15` respectively handle the first and the second of the above mentioned problems ("Correction stage" of the algorithm – let us stress that the term "correction" is here informal).

We are now in a position to easily prove the correction of this algorithm:

Theorem 3 *If there are finitely many Brun steps, then* `HybridBrunReco` *returns the set of admissible parameters of the binary function in input.*

Proof. We prove it by induction on the number of Brun steps (that is, the value of n at line `11` of the pseudo-code). If $n = 0$, admissible parameters are computed by `XReco`. Assume that the result holds for some n. One has:

$$\begin{aligned}
((1,\vec{\alpha}),\rho) \in P(\mathcal{B}_0) &\Leftrightarrow 0 \leq \mathcal{B}_0 \leq \mathcal{P}_{(1,\vec{\alpha}),\rho} \\
&\Leftrightarrow 0 \leq \mathcal{B}'_0 \leq \mathcal{P}_{(1,\vec{\alpha}),\rho} \\
&\Leftrightarrow 0 \leq \mathcal{B}'_0 - L_0 \leq \mathcal{P}_{(1,\vec{\alpha}),\rho} \text{ and } 0 \leq L_0 \leq \mathcal{P}_{(1,\vec{\alpha}),\rho} \\
&\Leftrightarrow 0 \leq \mathcal{B}_1 \leq \mathcal{P}_{B^{-1}_{a_0,i_0}(1,\vec{\alpha}),\rho} \text{ and } ((1,\vec{\alpha}),\rho) \in \texttt{XReco}(L_0) \\
&\Leftrightarrow (B^{-1}_{a_0,i_0}(1,\vec{\alpha}),\rho) \in P(\mathcal{B}_1) \text{ and } ((1,\vec{\alpha}),\rho) \in \texttt{XReco}(L_0)
\end{aligned}$$

One passes from first to second line by Prop. 4, and from third to fourth line by Prop. 5, applying $E_1^*(\beta^{-1}_{a_0,i_0})$. So, finally:

$$P(\mathcal{B}_0) = B'_{a_0,i_0} P(\mathcal{B}_1) \cap \texttt{XReco}(L_0).$$

The claim follows by induction. □

Let us now prove that there are finitely many Brun steps. It is not hard to show that, at each step, the size of the considered binary function decreases. This bounds the number of Brun steps by the size of the initial binary function (that is, the one given in input). However, assuming that the initial binary function is not too sparse, we can achieve a better bound. We say that a binary function *fits into a bounding box of size* D if the distance between any two vertices of its geometric representation is lesser than D. One shows:

Proposition 6 *If the initial binary function* $\mathcal{B}_0 \in \mathfrak{B}_{d+1}$ *fits in a bounding box of side* D*, then either it is not planar, or the algorithms stops after lesser than* $\log_{\frac{d+2}{d+1}}((d+1)D^d)$ *Brun steps.*

Proof. Assume that $\mathcal{B}$ fits in a bounding box of side D and that it is a subset of a stepped plane of normal vector $(1,\vec{\alpha}) \in \mathbb{R}^{d+1}$. The key point consists in proving that entries of $\vec{\alpha}$ can be assumed to be sufficiently small integers. First, note that Def. 2 yields that any vertex $\vec{x}$ of $\mathcal{B}$ satisfies: $\rho \leq \langle \vec{x}|(1,\vec{\alpha})\rangle < \rho + ||(1,\vec{\alpha})||_1$, that is, are between hyperplanes $\langle \vec{x}|(1,\vec{\alpha})\rangle = \rho$ and $\langle \vec{x}|(1,\vec{\alpha})\rangle = \rho + ||\vec{\alpha}||_1$.

Then, let us call *leaning point* a vertex $\vec{x}$ on the lower hyperplane, that is, such that $\rho = \langle \vec{x} | (1, \vec{\alpha}) \rangle$. If, for $i \in \{2, \ldots, d+1\}$, $\mathcal{B}$ admits two leaning points $\vec{x}$ and $\vec{y}$ such that $x_i \neq y_i$, then one computes: $\alpha_{i-1} = (x_1 - y_1)/(x_i - y_i)$, that is, α_{i-1} is quotient of integer lesser than or equal to D. Otherwise, one checks that one can slightly modify $\vec{\alpha}$ and ρ in order to have two leaning points: geometrically, it suffices to "slide upwards" and "twist" the two hyperplanes containing vertices of $\mathcal{B}$, until two vertices belong to the lower hyperplane (see Fig. 12). By applying this for any i in $\{2, \ldots, d+1\}$, we obtain for $\vec{\alpha}$ a vector whose entries are quotient of integer lesser than or equal to D. The result then follows from Prop. 1. □

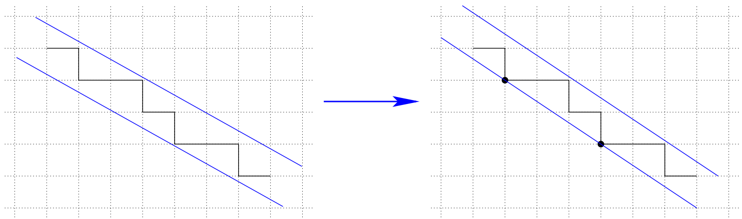

Fig. 12. Consider a subset of a stepped hyperplane of parameters $(1, \vec{\alpha})$ and ρ (left, in bold black). Its vertices lie between two euclidean hyperplanes (left, in blue). One can always slightly modify parameters (that is, slide or twist blue hyperplanes) in order to have two leaning points (right, highlighted points). If the subset fits into a bounding box of size D, then entries of the modified parameter $\vec{\alpha}$ are quotient of integer lesser than or equal to D (here, $D = 7$ and $\alpha = 3/2$).

Thus, we can slightly modify `HybridBrunReco` in order to stop if more than $\log D$ Brun steps are performed (in such a case, $\mathcal{B}_0$ is not a stepped plane, that is, the set of admissible parameters is empty). In particular, if the initial binary function $\mathcal{B}_0$ fits in a bounding box whose size is polynomial in the size of $\mathcal{B}_0$ (that is, $\mathcal{B}_0$ is "not too sparse"), then Prop. 6 shows that one can assume that the number of Brun steps is logarithmic in the size of $\mathcal{B}_0$. Moreover, since each Brun step can be performed in linear time in the size of $\mathcal{B}_n$ (which decreases), the complexity of the "Brun stage" (lines `1-10`) is quasi-linear in the size of the input.

6 Conclusion

To conclude, let us briefly recall main results as well as drawbacks of this paper. Concerning digital plane generation, we described a way to easily compute suitable finite subsets of (rational) stepped planes that allow to generate it by translated copies. However, shapes of these subsets are still not characterized. Concerning digital plane recognition, we provided an original hybrid algorithm, in the spirit of the theoretical results that we obtained in [7]. The

"Brun stage" of this algorithm (which is its original part) has quasi-linear complexity, while the complexity of the "correction stage" remains to be studied, in particular from a practical viewpoint (since this stage is expected to have much lesser importance than the Brun one). Note also that this recognition algorithm is rather complicated (because of dual maps) and not robust: stepped plane with small perturbations are not recognized as stepped planes. Thus, it is worth to ask whether a simple and robust algorithm could be found, which would rely on the main idea of the one here described (that is, local recognition and iterated encodings).

Acknowledgments. I would like to thank Valérie Berthé, Bruno Gaujal and Jacques-Olivier Lachaud for a careful reading of a preliminary version of this paper, as well as Rémy Malgouyres for a suggestion which led to Prop. 6. I would also like to thank referees of both [13] and this paper, for their useful comments.

References

[1] É. Andres, *Le plan discret*, in Proc. of Discrete Geometry for Computer Imagery DGCI'93 (1993), pp. 45–61.

[2] É. Andres, R. Acharya, C. Sibata, *Discrete analytical hyperplanes*, Graphical Models and Image Processing **59** (1997), pp. 302–309.

[3] P. Arnoux, V. Berthé, Th. Fernique, D. Jamet, *Generalized substitutions, functional stepped surfaces and flips*, Theor. Comput. Sci. **380**, pp. 251–267.

[4] P. Arnoux, S. Ito, *Pisot substitutions and Rauzy fractals*, Bull. Bel. Math. Soc. Simon Stevin **8** (2001), pp. 181–207.

[5] P. Arnoux, S. Ito, Y. Sano, *Higher dimensional extensions of substitutions and their dual maps*, J. Annal. Math. **83** (2001), pp. 183–206.

[6] J. Berstel, *Tracé de droites, fractions continues et morphismes itérés*, in *Mots*, Hermès, 1990, pp. 298-309.

[7] V. Berthé, Th. Fernique, *Brun expansions of stepped surfaces*, preprint, available at `http://www.lif.univ-mrs.fr/~fernique/info/bf.pdf.gz`

[8] A. J. Brentjes, *Multi-dimensional continued fraction algorithms*, Mathematical Centre Tracts **145**, Matematisch Centrum, Amsterdam, 1981.

[9] V. E. Brimkov, R. P. Barneva, *Plane digitization and related combinatorial problems*, Discrete Applied Mathematics **147** (2005), pp. 169–186.

[10] V. E. Brimkov, D. Cœurjolly, R. Klette *Digital planarity–a review*, Discrete Applied mathematics **155** (2007), pp. 468–495.

[11] D. Cœurjolly, V. E. Brimkov, *Computational aspects of Digital plane and hyperplane recognition*, in Proc. of International Workshop on Combinatorial Image Analysis IWCIA'06, LNCS **4040** (2006), pp. 543–562.

[12] H. Ei, *Some properties of invertible substitutions of rank d and higher dimensional substitutions*, Osaka Journal of Mathematics **40** (2003), pp. 543–562.

[13] Th. Fernique, *Generation and recognition of digital planes using multi-dimensional continued fractions*, in Proc. of Discrete Geometry for Computer Imagery DGCI'08 (2008), LNCS **4992** (2008), pp. 33–44.

[14] Th. Fernique, *Pavages, fractions continues et géométrie discrète*, Ph. D. thesis, Univ. Montpellier 2, 2007.

[15] J. Françon, *Sur la topologie d'un plan arithmétique*, Theor. Comput. Sci. **156** (1996), pp. 159–176.

[16] R. Klette, A. Rosenfeld, *Digital straightness–a review*, Elec. Notes in Theoret. Comput. Sci. **46** (2001).

[17] J.-P. Réveilles, *Calcul en nombres entiers et algorithmique*, Ph. D Thesis, Univ. Louis Pasteur, Strasbourg (1991).

[18] F. Schweiger, *Multi-dimensional continued fractions*, Oxford Science Publications, Oxford Univ. Press, Oxford (2000).

[19] A. Troesch, *Interprétation géométrique de l'algorithme d'Euclide et reconnaissance de segments*, Theor. Comput. Sci. **115** (1993), pp. 291–320.

[20] L. Vuillon, *Combinatoire des motifs d'une suite sturmienne bidimensionelle, Theoret. Comput. Sci.* **209** (1998), pp. 261–285.

[21] L.-D. Wu, *On the chain code of a line*, IEEE Transactions on Pattern Analysis and Machine Intelligence **4** (1982), pp. 359–368.