

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

Thomas Fernique

LIRMM, Univ. Montpellier 2, CNRS
161 rue Ada 34392 Montpellier - France
fernique@lirmm.fr

Abstract. This paper extends, in a multi-dimensional framework, pattern recognition technics 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.

Introduction

Discrete (or digital) geometry deals with discrete sets considered to be digitized objects of the Euclidean space. A challenging problem is to decompose a huge complicated 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 [1,7,9]). We thus need efficient algorithms which generate arbitrarily big patches of such digitizations from given parameters and, conversely, recognize parameters from given digitizations.

In the particular case of digitizations of lines, among other technics, so-called *linguistic technics* provide a nice connection with words theory and continued fractions. Let us briefly detail this. A digital line made of horizontal or vertical unit segments can be coded by a two-letter word, called *chain code* or *Freeman code*. For example, if a horizontal (resp. vertical) unit segment is coded by 0 (resp. 1), then a segment of slope 1 can be coded by a word of the form $10\dots 10 = (10)^k$. Then, basic transformations on words correspond to basic operations on slopes of the segments they code. For example, replacing each 0 by 01 and each 1 by 0 in the previous word leads to the word $(001)^k$, which codes a segment of slope $1/2$. Many algorithms use this approach for both recognition and generation of digital lines, and continued fraction expansions of slopes of segments turn out to play a central role there (see *e.g.* [11] or references in [8]).

In higher dimensions, there are also various technics for generation or recognition of digital hyperplane as, for example, linear programming, computational geometry or preimage technics (see *e.g.* [4] and references therein). However, these approaches do not extend the connection between words theory and continued fractions. The aim of this paper is to introduce an approach which does

it. Such an approach extends the case of so-called stepped surfaces (which are particular infinite digitizations), studied in [3].

The paper is organized as follows. In Sec. 1, we introduce *binary functions*, which can be seen as unions of faces of unit hypercubes. Among them, the ones called *stepped planes* ([12]) play for Euclidean hyperplanes the role played by chain codes for Euclidean lines. We also introduce *dual maps* ([2,6]), which generalize the basic transformations on chain codes mentioned above. Then, in Sec. 2, we briefly describe the *Brun algorithm*, which is one of the existing multi-dimensional continued fraction algorithms (see [10]). The Brun algorithm computes so-called *Brun expansions* of real vectors. We also introduce particular dual maps which allow the Brun algorithm to act over stepped planes. This leads, in Sec. 3, to a method for obtaining a *fundamental domain* of a stepped plane, that is, a binary function which suffices to generate by periodicity the whole stepped plane (Th. 2). In Sec. 4, we describe a method to compute so-called *Brun expansions of stepped planes*, by grabbing information from local configurations (namely *runs*). Actually, the Brun expansion of a stepped plane is nothing but the Brun expansion of its normal vector. So, the interest of this method is that it can be naturally extended to binary functions, leading to define, in Sec. 5, *Brun expansions of binary functions*. We finally use this extended notion of Brun expansion, in Sec. 6, to describe a recognition algorithm which decides whether a given binary function is a stepped plane or not (Th. 3).

1 Stepped Planes and Dual Maps

We here first introduce our basic digital objects, namely *binary functions* and *stepped planes*. Formally, it is convenient to consider the set of functions from $\mathbb{Z}^d \times \{1, \dots, d\}$ 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 $\mathbb{Z}^d \times \{1, \dots, d\}$ where \mathcal{B} takes value one. We denote by \mathfrak{B}_d the set of binary functions. For $\mathbf{x} \in \mathbb{Z}^d$ and $i \in \{1, \dots, d\}$, we call face of type i located in \mathbf{x} the binary function denoted by (\mathbf{x}, i^*) whose support is $\{(\mathbf{x}, i)\}$.*

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 $(\mathbf{e}_1, \dots, \mathbf{e}_d)$ denote the canonical basis of \mathbb{R}^d . The geometric interpretation of a face (\mathbf{x}, i^*) is the closed subset of \mathbb{R}^d defined by (see Fig. 1):

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

This subset is a hyperface of the unit cube of \mathbb{R}^d whose lowest vertex is \mathbf{x} . 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 (see Fig. 3).

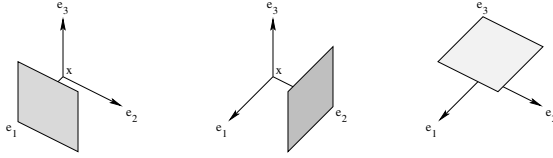


Fig. 1. Geometrical interpretations of faces (\mathbf{x}, i^*) , for $i = 1, 2, 3$ (from left to right)

Among binary functions, we are especially interested in so-called *stepped planes*:

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

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

where $\langle | \rangle$ is the canonical 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 $\mathcal{P}_{\alpha,\rho}$, that is, the integers vectors which belong to its geometrical interpretation, form a *standard arithmetic discrete plane of parameters* (α, ρ) (see [1,7,9]). Moreover, one checks that the orthogonal projection along $\mathbf{e}_1 + \dots + \mathbf{e}_d$ 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).

Let us now introduce the main tool of this paper, namely *dual maps*, which act over binary functions and stepped planes. First, let us recall some basic definitions and notations. We denote by F_d the free group generated by the alphabet $\{1, \dots, d\}$, 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 mapping* is the map \mathbf{f} from F_d to \mathbb{Z}^d defined on $w \in F_d$ by:

$$\mathbf{f}(w) = (|w|_1, \dots, |w|_d),$$

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 $\mathbf{f}(\sigma(i))$. 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 $\mathbf{f}(2^{-1}3) = \mathbf{e}_3 - \mathbf{e}_2$. This substitution is unimodular since its incidence matrix (below) has determinant 1:

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

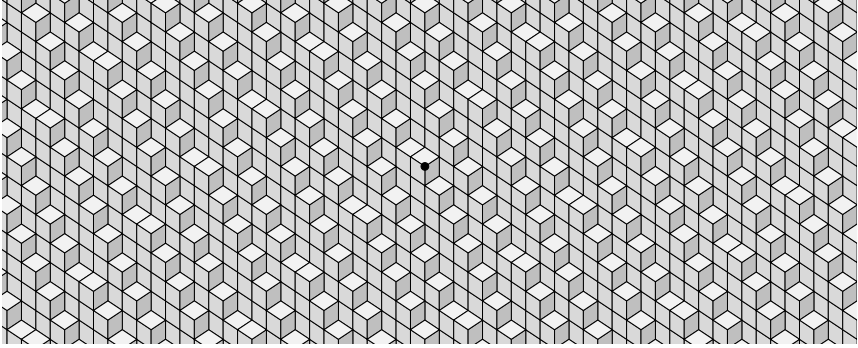


Fig. 2. Geometrical interpretation of the stepped plane $\mathcal{P}_{(24,9,10),0}$ (highlighted origin). This can be seen either as faces of unit cubes, or as a lozenge tiling of the plane.

We are now in a position to define *dual maps*:

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

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

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

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

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

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.

The following theorem, proved in [3], connects dual maps and stepped planes:

Theorem 1 ([3]). Let σ be a unimodular endomorphism of F_d . Let $\alpha \in \mathbb{R}_+^d \setminus \{\mathbf{0}\}$ and $\rho \in \mathbb{R}$. If $M_\sigma^\top \alpha \in \mathbb{R}_+^d$, then the image of the stepped plane $\mathcal{P}_{\alpha,\rho}$ by $E_1^*(\sigma)$ is the stepped plane $\mathcal{P}_{M_\sigma^\top \alpha, \rho}$. Otherwise, this image is not a binary function¹.

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 $M_\sigma^\top \alpha \in \mathbb{R}_+^d$ holds for any $\alpha \in \mathbb{R}_+^d \setminus \{\mathbf{0}\}$: the image of a stepped plane by the dual map of a substitution is thus always a stepped plane.

¹ See Def. 1.

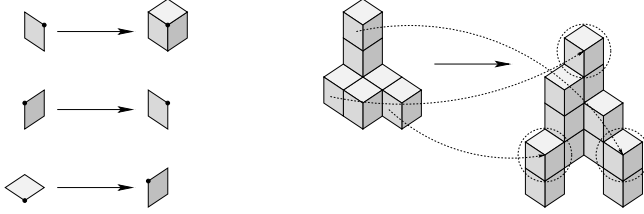


Fig. 3. Action of the dual map of Ex. 2 on faces (left) and on a binary function (right). Let us stress that the image of a binary function is not necessarily binary (unlike here).

2 Brun Expansions of Real Vectors

We here recall the Brun algorithm (see *e.g.* [10]) and use dual maps to connect it with normal vectors of stepped planes (recall Def. 2).

Definition 4. The Brun map T is the map from $[0, 1]^d \setminus \{\mathbf{0}\}$ to $[0, 1]^d$ defined on $\alpha = (\alpha_1, \dots, \alpha_d)$ by:

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

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

$$a_n = \lfloor \|T^n(\alpha)\|_\infty^{-1} \rfloor \quad \text{and} \quad i_n = \min\{j \mid \langle T^n(\alpha), e_j \rangle = \|T^n(\alpha)\|_\infty\}.$$

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 $\alpha \in [0, 1]$, then $(a_n)_n$ is the continued fraction expansion of α , while, for all n , $i_n = 1$.

Example 3. The Brun expansion of $(3/8, 5/12)$ is $(2, 2), (1, 1), (2, 2), (4, 1), (1, 2)$.

Let us mention that, as in the case of continued fractions, it turns out that a vector has a finite Brun expansion if and only if it has only rational entries. Let us now give a matrix viewpoint of the Brun map T . For $(a, i) \in \mathbb{N} \times \{1, \dots, d\}$, one introduces the following $(d + 1) \times (d + 1)$ symmetric matrix:

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

where I_p stands for the $p \times p$ identity matrix. Then, consider a vector $\alpha = (\alpha_1, \dots, \alpha_d) \in [0, 1]^d \setminus \{\mathbf{0}\}$. A simple computation shows that, with $i = \min\{j \mid \alpha_j = \|\alpha\|_\infty\}$ and $a = \lfloor \alpha_i^{-1} \rfloor$, one has:

$$(1, \alpha) = \|\alpha\|_\infty B_{a,i}(1, T(\alpha)), \quad (2)$$

where, for any vector \mathbf{u} , $(1, \mathbf{u})$ stands for the vector obtained by adding to \mathbf{u} a first entry equal to 1. Note that $B_{a,i}$ is invertible. Thus, one can rewrite the previous equation as follows:

$$(1, T(\boldsymbol{\alpha})) = \|\boldsymbol{\alpha}\|_{\infty}^{-1} B_{a,i}^{-1}(1, \boldsymbol{\alpha}). \tag{3}$$

To conclude this section, let us show that this matrix viewpoint allows to connect Brun expansions with the stepped planes and dual maps introduced in the previous section. Let us introduce *Brun substitutions*:

Definition 5. Let $a \in \mathbb{N}^*$ and $i \in \{1, \dots, d\}$. The Brun substitution $\beta_{a,i}$ is the endomorphism of F_{d+1} defined by:

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

One checks that $\beta_{a,i}$ is unimodular and has $B_{a,i}$ for incidence matrix.² Note also that $\beta_{a,i}$ is invertible, since one computes:

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

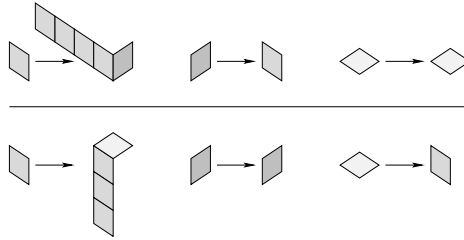


Fig. 4. Action on faces of the dual maps $E_1^*(\beta_{4,1})$ (top) and $E_1^*(\beta_{3,2})$ (bottom)

One then can consider dual maps of Brun substitutions (see Fig. 4), and one deduces from Th. 1 that Eq. (2) and (3) respectively yield:

$$E_1^*(\beta_{a,i})(\mathcal{P}_{\|\boldsymbol{\alpha}\|_{\infty}(1, T(\boldsymbol{\alpha})), \rho}) = \mathcal{P}_{(1, \boldsymbol{\alpha}), \rho}, \tag{4}$$

$$\mathcal{P}_{\|\boldsymbol{\alpha}\|_{\infty}(1, T(\boldsymbol{\alpha})), \rho} = E_1^*(\beta_{a,i}^{-1})(\mathcal{P}_{(1, \boldsymbol{\alpha}), \rho}). \tag{5}$$

3 Generation of Stepped Planes

We here show how dual maps and Brun expansions can be used to easily generate arbitrarily big patches of a stepped plane (that is, binary functions less or equal to it), provided that its normal vector has rational entries. Indeed, one proves:

² Let us recall that $B_{a,i}$ is the symmetric matrix defined Eq. (1).

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

$$\mathcal{D}_{(1, \alpha), \rho} = E_1^*(\beta_{a_0, i_0}) \circ \dots \circ E_1^*(\beta_{a_N, i_N})(\lfloor \rho' \rfloor \mathbf{e}_1, 1^*),$$

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

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

Then, the geometrical interpretation of the stepped plane $\mathcal{P}_{(1, \alpha), \rho}$ is the union of all the translations along $L_{(1, \alpha), \rho}$ of the geometrical interpretation of $\mathcal{D}_{(1, \alpha), \rho}$.

Example 4. Fig. 5 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:

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

Thus, according to Th. 2, the geometrical interpretation of the rational stepped plane $\mathcal{P}_{(1, 3/8, 5/12), 0} = \mathcal{P}_{(24, 9, 10), 0}$ (see Fig. 2) is the union of all the translations along $L_{(1, 3/8, 5/12), 0}$ of the geometrical interpretation of $\mathcal{D}_{(1, 3/8, 5/12), 0}$.

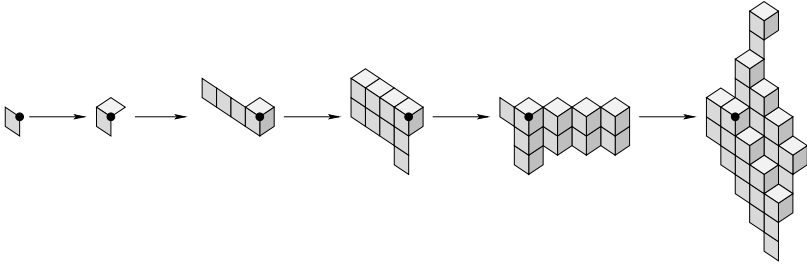


Fig. 5. Generation of $\mathcal{D}_{(1, 3/8, 5/12), 0}$ by applications of the dual maps $E_1^*(\beta_{1,2})$, $E_1^*(\beta_{4,1})$, $E_1^*(\beta_{2,2})$, $E_1^*(\beta_{1,1})$ and $E_1^*(\beta_{2,2})$ (from left to right – highlighted origin). According to Th. 2, the stepped plane $\mathcal{P}_{(1, 3/8,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

Note that, in terms of functions, one has $\mathcal{D}_{(1, \alpha), \rho} \leq \mathcal{P}_{(1, \alpha), \rho}$. This means that the geometrical interpretation of $\mathcal{D}_{(1, \alpha), \rho}$ is included in the one of $\mathcal{P}_{(1, \alpha), \rho}$. To conclude this section, let us mention that one can show that $\mathcal{D}_{(1, \alpha), \rho}$ has minimal size: such a piece of $\mathcal{P}_{(1, \alpha), \rho}$ is called a *fundamental domain* of $\mathcal{P}_{(1, \alpha), \rho}$.

4 Brun Expansions of Stepped Planes

We here show how Brun expansions of normal vectors of stepped planes can be directly computed on stepped planes relying on the notion of *run*:

Definition 6. An (i, j) -run of a binary function \mathcal{B} is a maximal sequence of contiguous faces of type i , aligned with the direction \mathbf{e}_j , whose geometric interpretation is included in the one of \mathcal{B} .

For example, the stepped plane depicted on Fig. 2 has $(1, 2)$ -runs and $(1, 3)$ -runs of size 2 or 3, and $(3, 2)$ -runs of size 1 or 2 (see also Fig. 6 in the general case of a binary function). The infimum and the supremum of the sizes (Recall Def. 1) of the (i, j) -runs of a binary function \mathcal{B} are respectively denoted by $a_{i,j}^-(\mathcal{B})$ and $a_{i,j}^+(\mathcal{B})$. The following proposition shows that runs contain information about the normal vector of a stepped plane:

Proposition 1. Let $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_d) \in \mathbb{R}_+^d \setminus \{0\}$ and $\rho \in \mathbb{R}$. Then, for $\alpha_j \neq 0$:

$$a_{i,j}^-(\mathcal{P}_{\boldsymbol{\alpha},\rho}) = \max(\lfloor \alpha_i / \alpha_j \rfloor, 1) \quad \text{and} \quad a_{i,j}^+(\mathcal{P}_{\boldsymbol{\alpha},\rho}) = \max(\lceil \alpha_i / \alpha_j \rceil, 1).$$

In particular, let us show that runs contain enough information to compute Brun expansions of normal vectors of so-called *expandable* stepped planes:

Definition 7. A stepped plane $\mathcal{P} \in \mathfrak{P}_{d+1}$ is said to be *expandable* if one has:

$$\max_{1 \leq i \leq d} a_{i+1,1}^+(\mathcal{P}) = 1 \quad \text{and} \quad \min_{1 \leq i \leq d} a_{1,i+1}^-(\mathcal{P}) < \infty.$$

In this case, we define:

$$i(\mathcal{P}) = \min_{1 \leq i \leq d} \{i \mid \max_{1 \leq j \leq d} a_{j+1,i+1}^+(\mathcal{P}) \leq 1\} \quad \text{and} \quad a(\mathcal{P}) = a_{1,i(\mathcal{P})+1}^-(\mathcal{P}).$$

Note that one easily deduces from Prop. 1 that a stepped plane is expandable if and only if its normal vector is of the form $(1, \boldsymbol{\alpha})$, with $\boldsymbol{\alpha} \in [0, 1]^d \setminus \{0\}$. Moreover, one then has:

$$i(\mathcal{P}_{(1,\boldsymbol{\alpha}),\rho}) = \min\{i \mid \alpha_i = \|\boldsymbol{\alpha}\|_\infty\} \quad \text{and} \quad a(\mathcal{P}_{(1,\boldsymbol{\alpha}),\rho}) = \lfloor \|\boldsymbol{\alpha}\|_\infty^{-1} \rfloor. \quad (6)$$

This leads to the following definition:

Definition 8. Let \tilde{T} be the map defined over expandable stepped planes by:

$$\tilde{T}(\mathcal{P}) = E_1^*(\beta_{a(\mathcal{P}),i(\mathcal{P})}^{-1})(\mathcal{P}).$$

In particular, \tilde{T} has values in \mathfrak{P}_{d+1} . More precisely, Eq. (4) yields:

$$\tilde{T}(\mathcal{P}_{(1,\boldsymbol{\alpha}),\rho}) = \mathcal{P}_{(1,T(\boldsymbol{\alpha}),\rho)}. \quad (7)$$

Thus, the Brun expansion of a vector $\boldsymbol{\alpha}$ can be computed on a stepped plane \mathcal{P} of normal vector $(1, \boldsymbol{\alpha})$, since it is nothing but the sequence $(a(\tilde{T}^n(\mathcal{P})), i(\tilde{T}^n(\mathcal{P})))_n$. By abuse, this Brun expansion is called *Brun expansion of the stepped plane \mathcal{P}* .

5 Brun Expansions of Binary Functions

Here, we show that runs allow to define Brun expansions not only of stepped planes but also of binary functions, although the latter do not have normal vectors. We first need to refine Def. 6 (see Fig. 6):

Definition 9. Let \mathcal{R} be an (i, j) -run of a binary function \mathcal{B} . Thus, there is a vector $\mathbf{x} \in \mathbb{Z}^d$ and an interval I of \mathbb{Z} (not necessarily finite) such that:

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

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

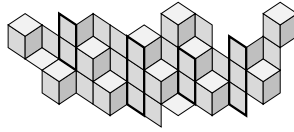


Fig. 6. This binary function has every type of $(1, 3)$ -runs: left-closed, right-closed, closed and open (framed runs, from left to right). It is moreover recognizable, with $(a, i) = (2, 2)$ (see definition below).

Then, as we previously defined Brun expansions of *expandable* stepped planes, we will here restrict to *recognizable* binary functions:

Definition 10. A binary function $\mathcal{B} \in \mathfrak{B}_{d+1}$ is recognizable if it satisfies the two following conditions. First, it shall exist $i \in \{1, \dots, d\}$ such that:

$$a_{1,i+1}^+(\mathcal{B}) \geq 2 \quad \text{and} \quad \min_{1 \leq j \leq d} a_{i+1,j+1}^+(\mathcal{B}) \geq 2.$$

Let $i(\mathcal{B})$ denotes the smallest such i . Second, \mathcal{B} shall have closed $(1, i(\mathcal{B}) + 1)$ -runs, with the smallest one having size $a_{1,i(\mathcal{B})+1}^+(\mathcal{B}) - 1$. Let $a(\mathcal{B})$ denotes this size.

Let us explain this definition. Assume that $\mathcal{B} \leq \mathcal{P}_{(1,\alpha),\rho}$, for $\alpha \in \mathbb{R}_+^d$ and $\rho \in \mathbb{R}$. Then, it is not hard to deduce from Prop. 1 that the first recognizability condition ensures that the $i(\mathcal{B})$ -th entry of α is smaller than 1 and greater than all the other entries, while the second recognizability condition ensures that $\mathcal{P}_{(1,\alpha),\rho}$ has $(1, i(\mathcal{B}))$ -runs of two different sizes, with the smallest size being equal to $a(\mathcal{B})$. In other words, recognizability ensures $\alpha \in [0, 1]^d$, $i(\mathcal{B}) = i(\mathcal{P}_{(1,\alpha),\rho})$ and $a(\mathcal{B}) = a(\mathcal{P}_{(1,\alpha),\rho})$. Thus, the formula defining \tilde{T} over stepped planes (Def. 8) can still be used to define \tilde{T} over recognizable binary functions. This leads to define the Brun expansion of a recognizable binary function \mathcal{B} as the sequence $(a(\tilde{T}^n(\mathcal{B})), i(\tilde{T}^n(\mathcal{B})))_n$, for n such that $\tilde{T}^n(\mathcal{B})$ is a recognizable binary function.

6 Recognition of Stepped Planes

We are here interested in the following recognition problem: given a binary function $\mathcal{B} \in \mathfrak{B}_{d+1}$ whose size $|\mathcal{B}|$ is finite, decide whether the following convex polytope of \mathbb{R}^{d+1} is empty or not:

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

The idea is that if the map \tilde{T} previously defined would satisfy, for any $\mathcal{B} \in \mathfrak{B}_{d+1}$:

$$0 \leq \mathcal{B} \leq \mathcal{P} \Leftrightarrow 0 \leq \tilde{T}(\mathcal{B}) \leq \tilde{T}(\mathcal{P}), \tag{8}$$

then, $P(\mathcal{B})$ would be not empty if and only if computing the sequence $(\tilde{T}^n(\mathcal{B}))_{n \geq 0}$ would lead to a binary function of the form $\sum_{\mathbf{x} \in X} (\mathbf{x}, \mathbf{1}^*)$, with the vectors of X having all the same first entries (such a binary function is easily recognizable). But Eq. (8) does not always hold. The first problem is that \tilde{T} is defined only over expandable stepped planes and recognizable binary functions. However, this problem turns out to generally appear only for binary functions whose size is small, because their runs do not contain enough information. The second problem seems more tedious: the image by \tilde{T} of a recognizable binary function less than or equal to a stepped plane \mathcal{P} is neither necessarily less than or equal to $\tilde{T}(\mathcal{P})$, nor even always a binary function. Let us first consider this problem. We introduce three rules acting over binary functions (see Fig. 7, and also Fig. 8, left):

Definition 11. *Let $a \in \mathbb{N}^*$ and $i \in \{1, \dots, d\}$. 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.*

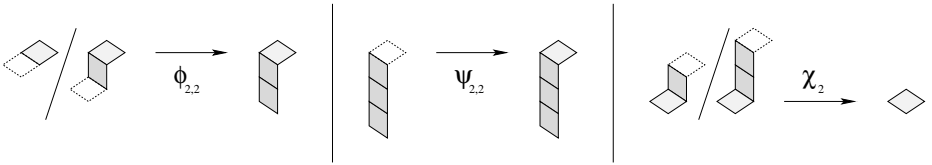


Fig. 7. The rules $\phi_{2,2}$, $\psi_{2,2}$ and χ_2 (dashed edges represent missing faces)

The following proposition then shows that one can replace any recognizable binary function \mathcal{B} by a recognizable binary function $\tilde{\mathcal{B}}$ which satisfies Eq. (8) under an additional hypothesis:

Proposition 2. *Let $\mathcal{B} \in \mathfrak{B}_{d+1}$ be a recognizable binary function and $\tilde{\mathcal{B}}$ be the binary function obtained by successively applying $\phi_{a(\mathcal{B}), i(\mathcal{B})}$, $\psi_{a(\mathcal{B}), i(\mathcal{B})}$ and $\chi_{i(\mathcal{B})}$. Then, for any stepped plane $\mathcal{P} \in \mathfrak{P}_{d+1}$, one has $\mathcal{B} \leq \mathcal{P}$ if and only if $\tilde{\mathcal{B}} \leq \mathcal{P}$. Moreover, if $\tilde{\mathcal{B}}$ does not have open $(1, i(\mathcal{B}) + 1)$ -runs, then one has:*

$$0 \leq \tilde{\mathcal{B}} \leq \mathcal{P} \Leftrightarrow 0 \leq \tilde{T}(\tilde{\mathcal{B}}) \leq \tilde{T}(\mathcal{P}).$$

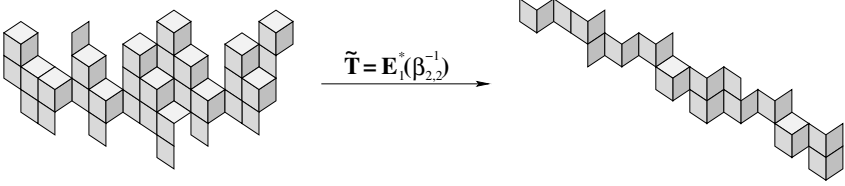


Fig. 8. The recognizable binary function \mathcal{B} of Fig. 6 is transformed by applying the rules of Fig. 7 into a binary function $\tilde{\mathcal{B}}$ (left) such that $0 \leq \mathcal{B} \leq \mathcal{P} \Leftrightarrow 0 \leq \tilde{\mathcal{B}} \leq \mathcal{P}$. Here, since $\tilde{\mathcal{B}}$ does not have open (1, 3)-run, its image by \tilde{T} (right) is such that, for any stepped plane \mathcal{P} , one has: $0 \leq \tilde{\mathcal{B}} \leq \mathcal{P} \Leftrightarrow 0 \leq \tilde{T}(\tilde{\mathcal{B}}) \leq \mathcal{P}$.

Thus, we still have the problem that a binary function \mathcal{B} is not always recognizable (Def. 10), while the second problem is now that $\tilde{\mathcal{B}}$ can have problematic open runs. However, it is expected that, in practice, both not recognizable binary functions and problematic open runs have rather small sizes. Hence, it is worth considering a hybrid algorithm. Given a recognizable binary function \mathcal{B} , we compute $\tilde{\mathcal{B}}$, remove problematic open runs and apply the map \tilde{T} . We iterate this up to obtain an unrecognizable binary function. Then, we use an already existing algorithm (*e.g.* a preimage algorithm, see [4]) to recognize this binary function and, finally, to refine the recognition by considering the previously removed open runs. More precisely, consider the following algorithm, where XReco is an algorithm which computes the set $P(\mathcal{B})$ and $B'_{a,i}$ 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})

1. $n \leftarrow 0$;
 2. $\mathcal{B}_0 \leftarrow \mathcal{B}$;
 3. **while** \mathcal{B}_n is recognizable **do**
 4. $(a_n, i_n) \leftarrow (a(\mathcal{B}_n), i(\mathcal{B}_n))$;
 5. compute $\tilde{\mathcal{B}}_n$;
 6. $L_n \leftarrow$ open runs of $\tilde{\mathcal{B}}_n$;
 7. $\mathcal{B}_{n+1} \leftarrow E_1^*(\beta_{a_n, i_n}^{-1})(\tilde{\mathcal{B}}_n - L_n)$;
 8. $n \leftarrow n + 1$;
 9. **end while**;
 10. $P_n \leftarrow \text{XReco}(\mathcal{B}_n)$;
 11. **for** $k = n - 1$ **downto** $k=0$ **do**
 12. $P_k \leftarrow B'_{a_k, i_k} P_{k+1}$;
 13. $P_k \leftarrow P_k \cap \text{XReco}(L_k)$;
 14. **end for**;
 15. **return** P_0 ;
-

One shows:

Theorem 3. *The algorithm HybridBrunReco with a binary function \mathcal{B} of finite size as input returns the set $P(\mathcal{B})$ in finite time.*

To conclude, let us discuss the computational cost of the above algorithm. Let us first focus on the “Brun” stage of the algorithm, that is, on lines 3–9. One can show that each step of this stage can be performed in time $\mathcal{O}(|\mathcal{B}_n|)$ and that the size $|\mathcal{B}_n|$ of \mathcal{B}_n strictly decreases. Thus, the whole stage can be performed in quadratic time (in the size of \mathcal{B}). However, let us stress that $(|\mathcal{B}_n|)_n$ generally decreases with an exponential rate (this is the case, for example, for any stepped plane), so that this stage is expected, in practice, to be performed in near linear time. Let us now consider the “correction” stage of the algorithm, that is, lines 10–14. Note that the sum of sizes of inputs of **XReco** is less than $|\mathcal{B}|$. Thus, assuming that **XReco** works in time no more than quadratic (this holds; for example, for a preimage algorithm, see [4]), the bound given for the first stage still holds. We also need to compute intersections of convex polytopes. The complexity of such operations is not trivial in higher dimensions, but let us stress that the intersection of k convex polytopes of \mathbb{R}^3 can be computed in time $\mathcal{O}(m \ln k)$, where m stands for the total size of these polytopes (see [5]). Moreover, let us recall that the first unrecognizable \mathcal{B}_n as well as the sum of sizes of the L_k ’s are expected to be much smaller than \mathcal{B} . In conclusion, theoretical time complexity bounds are probably much bigger than the practical efficiency of this algorithm, so that further experiments shall be performed in order to better understand the computational cost of this hybrid algorithm.

References

1. Andres, E.: Le plan discret. In: Proc. of Discrete Geometry for Computer Imagery DGCI 1993, pp. 45–61 (1993)
2. Arnoux, P., Ito, S.: Pisot substitutions and Rauzy fractals. Bull. Bel. Math. Soc. Simon Stevin 8, 181–207 (2001)
3. Berthé, V., Fernique, Th.: Brun expansions of stepped surfaces (preprint, 2008)
4. Brimkov, V., Cœ, D.: Computational aspects of Digital plane and hyperplane recognition. In: Reulke, R., Eckardt, U., Flach, B., Knauer, U., Polthier, K. (eds.) IWCMIA 2006. LNCS, vol. 4040, pp. 543–562. Springer, Heidelberg (2006)
5. Chazelle, B.: An optimal algorithm for intersecting three-dimensional convex polyhedra. SIAM J. Comput. 21, 671–696 (1992)
6. Ei, H.: Some properties of invertible substitutions of rank d and higher dimensional substitutions. Osaka Journal of Mathematics 40, 543–562 (2003)
7. Françon, J.: Discrete combinatorial surfaces. Graphical Models & Image Processing 57, 20–26 (1995)
8. Klette, R., Rosenfeld, A.: Digital straightness—a review. Elec. Notes in Theoret. Comput. Sci, vol. 46 (2001)
9. Réveillès, J.-P.: Calcul en nombres entiers et algorithmique, Ph. D Thesis, Univ. Louis Pasteur, Strasbourg (1991)
10. Schweiger, F.: Multi-dimensional continued fractions. Oxford Science Publications, Oxford Univ. Press, Oxford (2000)
11. Troesch, A.: Interprétation géométrique de l’algorithme d’Euclide et reconnaissance de segments. Theor. Comput. Sci. 115, 291–320 (1993)
12. Vuillon, L.: Combinatoire des motifs d’une suite sturmienne bidimensionnelle. Theoret. Comput. Sci. 209, 261–285 (1998)