Generation of Digital Planes Using Generalized Continued-Fractions Algorithms

Damien Jamet[1], Nadia Lafrenière[2], and Xavier Provençal[3]([✉])

[1] LORIA, Université de Lorraine, Nancy, France
[2] LaCIM, Université du Québec à Montréal, Montréal, Canada
[3] LAMA, Université Savoie Mont Blanc, Chambéry, France
`xavier.provencal@univ-smb.fr`

Abstract. We investigate a construction scheme for digital planes that is guided by generalized continued fractions algorithms. This process generalizes the recursive construction of digital lines to dimension three. Given a pair of numbers, Euclid's algorithm provides a natural definition of continued fractions. In dimension three and above, there is no such canonical definition. We propose a pair of hybrid continued fractions algorithms and show geometrical properties of the digital planes constructed from them.

Keywords: Digital planes generation · GCD algorithms · Generalized continued fractions algorithms · Recursive structure

1 Introduction

Over the last years, the study of digital straight lines has gathered much attention. Their properties have been exposed using different approaches and many applications were deduced from them. In particular, it is well known that a digital line has a recursive structure described by the continued fraction development of its slope. See [15] for a survey on digital straightness.

There has been much effort done in order to find analogous results and applications to those on 2D digital lines for 3D digital planes. See [8] for a survey on digital planarity. Works on dual substitutions [1] showed links between the structure of 3D digital planes and generalized continued fractions which lead, in particular, to generation and recognition techniques [3,12]. In this context, much effort has been directed to the study of the generation of digital planes with totally irrational normal vectors. See, for instance, [14] for a detailed work on this topic.

In [5,11], the authors investigate a process that, given a normal vector, computes the critical thickness and constructs the thinnest digital plane that is

connected. This process is completely directed by the execution of the *Fully Subtractive* algorithm on the normal vector. Even if their process computes the critical thickness for any normal vector, the construction, which we call the **FS**-construction, produces a digital plane only for a measure-zero set of vectors which excludes all integer vectors [10]. In the present paper, we propose an attempt to compute a connected digital plane for all vectors $\mathbf{v} \in \mathbb{N}^3$.

In the field of combinatorics on words, the recursive structure of digital lines has been studied, in particular, via *Christoffel words*. Recent work by Labbé and Reutenauer [17] extends Christoffel words as subgraphs of the hypercubic lattice in arbitrary dimensions. The motivation for the present paper is to provide a better understanding of the self-similarities of what is called a *Christoffel parallelogram* in [17]. This is, roughly speaking, the smallest pattern with parallel sides that tiles the digital plane by translation.

We provide a recursive construction scheme for 3D digital planes that is a generic version of the **FS**-construction. It is generic in the sense that it is parametrized by a generalized continued fraction algorithm, noted GCF-algorithm for short. A GCF-algorithm is an extension of Euclid's algorithm to higher dimensions. In [5], an inclusion relation between the result of the **FS**-construction and the construction of digital planes using dual substitutions, more precisely the E_1^* formalism, was used in order to show connectedness in specific cases. The same connection between our construction and E_1^* is expected, but left for future work. We define new *hybrid* GCF-algorithms and show that they allow to build a connected digital plane for any rational normal vector.

2   Basic Notions and Notation

Let $d \geqslant 2$ be an integer. Let $\{\mathbf{e}_i \mid k \in \{1, \ldots, d\}\}$ be the canonical basis of $\mathbb{R}^d$, and let $\langle \cdot, \cdot \rangle$ stand for the usual scalar product on $\mathbb{R}^d$. We note $\mathbf{0}$ the origin and $\mathbf{1} = \sum_{i=1}^d \mathbf{e}_i$ the vector with all coordinates equal to 1. Recall that $\mathbb{N}_+ := \mathbb{N} \backslash \{\mathbf{0}\}$. We always suppose that $\mathbf{v} \in (\mathbb{N}_+)^d$ and that the coordinates of $\mathbf{v}$ are relatively prime. The 1-norm of $\mathbf{x} \in \mathbb{R}^d$, noted $\|\mathbf{x}\|_1$, is given by the sum of the absolute values of its coordinates.

Definition 2.1. *The* digital hyperplane $\mathcal{P}(\mathbf{v}, \omega)$ *with* normal vector $\mathbf{v} \in \mathbb{R}^d$ *and* thickness $\omega \in \mathbb{R}$ *is defined by:*

$$\mathcal{P}(\mathbf{v}, \omega) = \left\{ \mathbf{x} \in \mathbb{Z}^d \mid 0 \leqslant \langle \mathbf{x}, \mathbf{v} \rangle < \omega \right\}.$$

Note that the usual definition of digital hyperplanes includes a *shift* parameter μ which we decide to omit in order to lighten the notation. Moreover, we only consider digital lines ($d = 2$) and planes ($d = 3$). For any point $\mathbf{x} \in \mathbb{Z}^d$, the quantity $\langle \mathbf{x}, \mathbf{v} \rangle$ determines if $\mathbf{x}$ belongs to $\mathcal{P}(\mathbf{v}, \omega)$ or not. We call $\langle \mathbf{x}, \mathbf{v} \rangle$ the *height* of $\mathbf{x}$ (in $\mathcal{P}(\mathbf{v}, \omega)$) and note it by $\overline{\mathbf{x}}$. See Fig. 1 for an illustration.

Two points at the same height define a *period vector*, so that $\mathcal{P}(\mathbf{v}, \omega)$ is invariant under a translation by that vector. Given $(\mathbf{b}_i)_{i \in \{1, \ldots, d-1\}}$ a basis of the
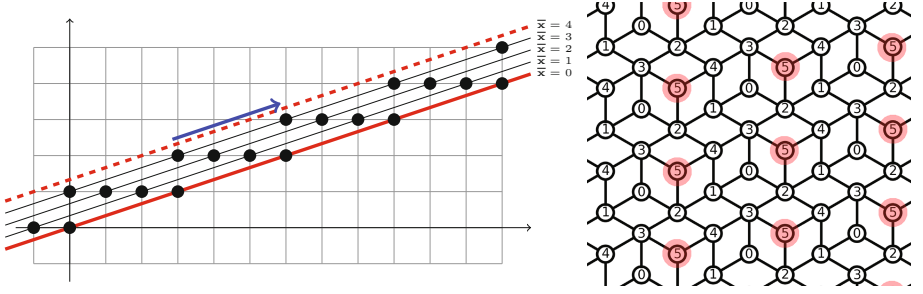
Fig. 1. The inner structure of digital lines and planes. Left: the digital line $\mathcal{P}((-1,3),4)$. The line $\overline{\mathbf{x}} = 0$ contains the lowest points, while the line $\overline{\mathbf{x}} = 3$ contains the highest points. Right: the digital plane $\mathcal{P}((1,2,3),6)$. The numbers indicate the height of each points. The highest points are highlighted showing that they form a regular lattice. In both cases, two points at same height define a period vector.

lattice $\{\mathbf{x} \in \mathbb{Z}^d \mid \overline{\mathbf{x}} = 0\}$ and a set $S \subset \mathbb{Z}^d$ such that for each integer $h \in \mathbb{Z}$, $\{\mathbf{x} \in S \mid \overline{\mathbf{x}} = h\} \neq \emptyset$ if and only if $h \in [0, \omega)$, then

$$\mathcal{P}(\mathbf{v}, \omega) = \bigcup_{\mathbf{x} \in \mathbb{Z}\mathbf{b}_1 + \cdots + \mathbb{Z}\mathbf{b}_{d-1}} S + \mathbf{x},$$

and we say that S provided with the vectors $(\mathbf{b}_i)$ *spans* $\mathcal{P}(\mathbf{v}, \omega)$.

Two points $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^d$ are said to be *adjacent* if $\|\mathbf{x} - \mathbf{y}\|_1 = 1$, which means that all their coordinates are equal except for one that differs by 1. By analogy to graph theory, we say that a subset of $\mathbb{Z}^d$ is *connected* if its adjacency graph is connected. In particular, a digital plane $\mathcal{P}(\mathbf{v}, \omega)$ is always disconnected if $\omega < \min(\mathbf{v})$ and it is always connected if $\omega > \max(\mathbf{v}) + \min\{\mathbf{v}_i \mid \mathbf{v}_i \neq 0\}$ (see [6], Lemma 5.3).

3   Construction Guided by Continued Fractions

In its additive form, Euclid's algorithm can be expressed as "*given a couple of integers (a, b), subtract the smaller to the larger one, and repeat. When both numbers are equal, their value is the* gcd". It can be expressed in a matricial expression as follows:

$$\mathbf{Euclid}\begin{pmatrix} a \\ b \end{pmatrix} = \left\{ \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix} \text{ if } b > a, \\ \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \text{ otherwise,} \right.$$

so that multiplying vector $\begin{pmatrix} a \\ b \end{pmatrix}$ by the output matrix performs one step of the algorithm. Table 1 presents the general construction scheme of digital lines and planes. It requires two inputs, namely a GCF-algorithm and a normal vector. First, we focus on dimension 2. In this case, there is a canonical GCF-algorithm: **Euclid**. The construction is simply a geometrical reinterpretation of the combinatorial construction of Christoffel words by the *Christoffel tree*,

Table 1. Recursive construction of a digital lines ($d = 2$) and plane ($d = 3$).

Input: $\mathbf{X}$ a GCF-algorithm, $\mathbf{v}$ the normal vector
Let: $\mathbf{v}_0 = \mathbf{v}$, $B_0 = \{\mathbf{0}\}$, $\mathbf{h}_0 = \mathbf{0}$, $L_0 = \{\mathbf{e}_i \mid i \in \{1, \ldots, d\}\}$ and $\mathbf{a}_0 = \mathbf{1}$.
For each step $n \geqslant 1$ **let**

Matrices:	$M_n = \mathbf{X}(\mathbf{v}_{n-1})$,
Vectors:	$\mathbf{v}_n = M_n \mathbf{v}_{n-1}$,
Indexes:	$\delta_n =$ index of the coordinate of $\mathbf{v}_{n-1}$ that is subtracted,
Translation vectors:	$\mathbf{t}_n = M_1^\top M_2^\top \cdots M_n^\top \mathbf{e}_{\delta_n}$,
Bodies:	$B_n = B_{n-1} \cup (B_{n-1} + \mathbf{t}_n)$,
Highest points:	$\mathbf{h}_n = \mathbf{t}_1 + \mathbf{t}_2 + \cdots + \mathbf{t}_n$,
Legs:	$L_n = \{\mathbf{h}_n + M_1^\top M_2^\top \cdots M_n^\top \mathbf{e}_i \mid i \in \{1, \ldots, d\}\}$.
Approximations:	$\mathbf{a}_n = M_1^{-1} M_2^{-1} \cdots M_n^{-1} \mathbf{1}$,

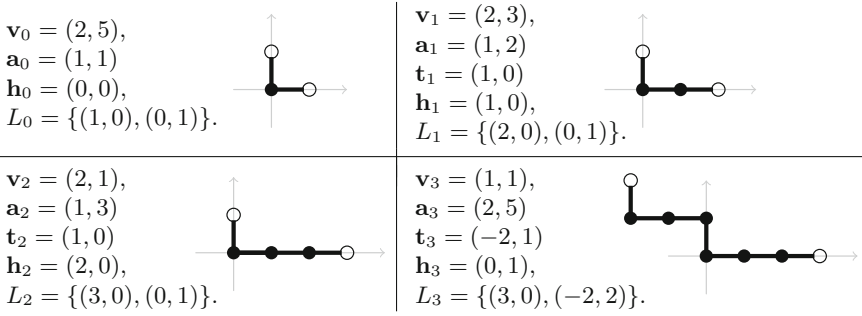|	
$\mathbf{v}_0 = (2,5)$, $\mathbf{a}_0 = (1,1)$ $\mathbf{h}_0 = (0,0)$, $L_0 = \{(1,0),(0,1)\}$.	$\mathbf{v}_1 = (2,3)$, $\mathbf{a}_1 = (1,2)$ $\mathbf{t}_1 = (1,0)$ $\mathbf{h}_1 = (1,0)$, $L_1 = \{(2,0),(0,1)\}$.
$\mathbf{v}_2 = (2,1)$, $\mathbf{a}_2 = (1,3)$ $\mathbf{t}_2 = (1,0)$ $\mathbf{h}_2 = (2,0)$, $L_2 = \{(3,0),(0,1)\}$.	$\mathbf{v}_3 = (1,1)$, $\mathbf{a}_3 = (2,5)$ $\mathbf{t}_3 = (-2,1)$ $\mathbf{h}_3 = (0,1)$, $L_3 = \{(3,0),(-2,2)\}$.

Fig. 2. Example of the construction scheme using Euclid's algorithm and the input vector $\mathbf{v} = (2,5)$. By convention, $\bullet \in B_n$ and $\circ \in L_n$. At step 3, $\mathbf{v}_3 = \mathbf{1}$ and thus $\mathbf{a}_3 = \mathbf{v}$ and $B_3 \cup L_3$ forms the main pattern of $\mathcal{P}((2,5),7)$.

see [2] (Chap. 1, Sect. 7). The result is a set of points called *pattern*. By analogy with the terminology used in [17], we say that a pattern is made of a *body*, noted B_n, and *legs*, noted

L_n. Euclid's algorithm ensures that there exists N such that $\mathbf{v}_N = (1,1)$. The *approximations* ($\mathbf{a}_n$) correspond to the values encountered while going down in the Christoffel tree (or equivalently the Stern-Brocot tree), starting from the root $\mathbf{a}_0 = (1,1)$ and ending after N steps at $\mathbf{a}_N = \mathbf{v}_0$. See Fig. 2 for an example.

The main difficulty in order to use this construction in dimension three is that there is no canonical extension of Euclid's algorithm. Instead, there exists a wide variety of GCF-algorithms (see for instance [18] or [16]). Among all GCF-algorithms, we only focus on the three following ones in the scope of this paper. Given three non-negative numbers,

– Selmer, noted **S** : subtract the smallest value to the largest one.
– Brun, noted **B** : subtract the middle value to the largest one.
– Fully Subtractive, noted **FS** : subtract the smallest value to the two others.

Obviously, the action of each of these GCF-algorithms may be expressed by a matrix product. Moreover, with these three algorithms, designating the index of the entry that is subtracted is unambiguous. Here are some basic properties of the construction given by Table 1 with $d = 3$.

Property 3.1. Let **X** be a GCF-algorithm such that each step is either **S**, **B** or **FS**. For each $n \geqslant 0$,

(1) $\mathbf{v}_n = M_n \cdots M_1 \mathbf{v}$.
(2) if $\mathbf{v}_n = \mathbf{1}$, then $\mathbf{a}_n = \mathbf{v}$.
(3) $B_n = \{\sum_{i \in I} \mathbf{t}_i \mid I \subseteq \{1, \ldots, n\}\}$.
(4) the coordinates of $\mathbf{v}_n$ are non-negative.
(5) for each $i \in \{1, 2, 3\}$, the height of $M_1^\top \cdots M_n^\top \mathbf{e}_i$ is equal to the i-th coordinate of $\mathbf{v}_n$.
(6) $\overline{\mathbf{t}}_n$, the height of $\mathbf{t}_n$, is the value of the coordinate of $\mathbf{v}_{n-1}$ that is subtracted to some other coordinate(s) by M_n, in order to compute $\mathbf{v}_n$.
(7) for each $i \in \{1, 2, 3\}$, let $\mathbf{x} = M_1^\top \cdots M_n^\top \mathbf{e}_i$. The vector $\mathbf{x}$ is a Bezout vector for $\mathbf{a}_n$, that is $\langle \mathbf{x}, \mathbf{a}_n \rangle = 1$.

Proof. Properties (1), (2) and (3) are straightforward. Property (4) is deduced from the definition of the continued fraction algorithm. For (5), $\langle M_1^\top \cdots M_n^\top \mathbf{e}_i, \mathbf{v} \rangle = \langle \mathbf{e}_i, M_n \cdots M_1 \mathbf{v} \rangle = \langle \mathbf{e}_i, \mathbf{v}_n \rangle$. For (6), the value of the coordinate of $\mathbf{v}_{n-1}$ that is subtracted is not modified by the action of M_n, in other words $\langle \mathbf{v}_{n-1}, \mathbf{e}_{\delta_n} \rangle = \langle \mathbf{v}_n, \mathbf{e}_{\delta_n} \rangle$. Finally, for (7), $\langle M_1^\top \cdots M_n^\top \mathbf{e}_i, \mathbf{a}_n \rangle = \langle \mathbf{e}_i, \mathbf{1} \rangle = 1$.   □

Obviously, geometric properties of the objects generated depend on the choice of the GCF-algorithm. Here are some properties that are expected from the generated patterns:

– They must be included in the digital plane $\mathcal{P}(\mathbf{v}, \|\mathbf{v}\|_1)$.
– They should form a connected set of points.
– Period vectors should be deduced from them.
– They should contain points at every height.
– They should be as small as possible, to avoid redundancy.

3.1   General Properties of the Construction

Even though the construction scheme from Table 1 produces infinite sequences, we only consider a finite number of steps. In dimension two, Euclid's algorithm always brings vector $\mathbf{v}$ to $\mathbf{1}$ (recall that $\mathbf{v}$ is assumed to have relatively prime coordinates). This is the halt condition for the algorithm. In dimension three, a GCF-algorithm may produce a sequence $(\mathbf{v}_n)_{n \geqslant 0}$ such that $\mathbf{1}$ does not appear in it. For instance, with the **FS** GCF-algorithm, if there exists n such that $\mathbf{v}_n = (1, 1, 2)$, then $\mathbf{v}_{n+1} = (1, 0, 2)$. Obviously, $\mathbf{v}_{n'} \neq \mathbf{1}$ for all $n' \geqslant n$. For now, we focus on the case where $\mathbf{1}$ does appear in $(\mathbf{v}_n)_{n \geqslant 0}$.

Definition 3.2. *Let* $\mathbf{X}$ *be a GCF-algorithm and* $\mathbf{v}$ *a vector with relatively prime coordinates. The* length*, noted* $\mathrm{length}_{\mathbf{X}}(\mathbf{v})$*, is, if it exists, the smallest integer* $N \geqslant 0$ *such that* $\mathbf{v}_N = \mathbf{1}$.

In Sect. 4.1, we provide two new GCF-algorithms and show (Lemma 4.1) that, if $\mathbf{X}$ is one of these, then $\mathrm{length}_{\mathbf{X}}(\mathbf{v})$ exists. Under such assumption, the points of the body are always included in a digital plane of normal vector $\mathbf{v}$.

Proposition 3.3. *Let* $\mathbf{X}$ *be a GCF-algorithm such that each step is either* $\mathbf{S}$*,* $\mathbf{B}$ *or* $\mathbf{FS}$ *and* $N = \mathrm{length}_{\mathbf{X}}(\mathbf{v})$ *exists. For each* $n \in \{0, 1, \dots, N\}$*,* $B_n \subseteq \mathcal{P}(\mathbf{v}, \|\mathbf{v}\|_1 - 2)$.

Proof. For each $n \geqslant 1$, let g_n be the value of the coordinate of $\mathbf{v}_{n-1}$ that is subtracted. From Property 3.1 (6), this value is equal to the height of $\mathbf{t}_n$. From Property 3.1 (3), given a point $\mathbf{x} \in B_n$, there exists a subset $I \subset \{1, \dots, n\}$ such that $\mathbf{x} = \sum_{i \in I} \mathbf{t}_i$ and, by linearity, $\overline{\mathbf{x}} = \sum_{i \in I} g_i$. On one hand, each g_i being non-negative, $0 \leqslant \overline{\mathbf{x}}$. On the other hand, since each g_i is subtracted to one or two coordinates of $\overline{\mathbf{v}}_{n-1}$ while keeping its coordinates non-negative, we have that $\sum_{i \in I} g_i \leqslant \|\mathbf{v}_0\|_1 - \|\mathbf{v}_N\|_1$. Thus, $\mathbf{x} \in \mathcal{P}(\mathbf{v}, \|\mathbf{v}\|_1 - 2)$. $\qquad\square$

The following proposition states that the legs define linearly independent period vectors. In Sect. 4.2, we show that these vectors may be used with the corresponding pattern in order to span a digital plane.

Proposition 3.4. *Let* $\mathbf{X}$ *be a GCF-algorithm such that each step is either* $\mathbf{S}$*,* $\mathbf{B}$ *or* $\mathbf{FS}$ *and* $N = \mathrm{length}_{\mathbf{X}}(\mathbf{v})$ *exists. For each* $n \in \{0, 1, \dots, N\}$*, let* $\{\mathbf{l}_1, \mathbf{l}_2, \mathbf{l}_3\} = L_n$ *(see Table 1). The differences* $(\mathbf{l}_2 - \mathbf{l}_1,\ \mathbf{l}_3 - \mathbf{l}_1)$ *form a basis of the lattice* $\{\mathbf{x} \in \mathbb{Z}^3 \mid \langle \mathbf{x}, \mathbf{a}_n \rangle = 0\}$.

Proof (Sketch). By Property 3.1 (7), we have $\langle \mathbf{l}_1, \mathbf{a}_n \rangle = \langle \mathbf{l}_2, \mathbf{a}_n \rangle = \langle \mathbf{l}_3, \mathbf{a}_n \rangle$, which implies that their differences are in the lattice $\{\mathbf{x} \in \mathbb{Z}^3 \mid \overline{\mathbf{x}} = 0\}$. The fact that $(\mathbf{l}_2 - \mathbf{l}_1,\ \mathbf{l}_3 - \mathbf{l}_1)$ form a basis of the lattice is deduced from the fact that the matrix $M_1^\top \cdots M_n^\top$ is unimodular, since it is a product of unimodular matrices.

4   The Choice of a GCF-Algorithm

In order to build a digital plane with normal vector $\mathbf{v}$, one should use a GCF-algorithm that reduces $\mathbf{v}$ to $\mathbf{1}$. Indeed, if $\mathbf{v}_N = \mathbf{1}$, then, by Property 3.1 (2), $\mathbf{a}_N = \mathbf{v}$ and Proposition 3.4 states that the legs L_N define linearly independent period vectors of $\mathcal{P}(\mathbf{v}, \omega)$.

The $\mathbf{FS}$ GCF-algorithm appeared naturally in the study of the topological properties of digital planes [9]. Subsequent work [4,5,10,11] showed that the geometry of a digital plane is strongly linked to the execution of $\mathbf{FS}$ on its normal vector.

Given a vector $\mathbf{v}$ for which $N = \mathrm{length}_{\mathbf{FS}}(\mathbf{v})$ is defined, a point $l \in L_N$ is such that $\overline{l} = \lfloor \frac{\|\mathbf{v}\|_1}{2} \rfloor$, the set B_N is connected and contains exactly one point at each height from 0 to $\lfloor \frac{\|\mathbf{v}\|_1}{2} \rfloor - 1$. The set $B_N \cup \{l\}$ is a minimal pattern that not only

spans $\mathcal{P}(\mathbf{v}, \frac{\|\mathbf{v}\|_1}{2})$, but *tiles* the digital plane, since the points of the translated copies of the pattern never overlap. That being said, for most of the normal vectors $\mathbf{v}$, **FS** "*fails*", which means that $\text{length}_{\mathbf{FS}}(\mathbf{v})$ is not defined [13]. For instance, in dimension 3, there are two cases where this happens. The first one is when dealing with a vector $\mathbf{v}$ such that a vector $\mathbf{u}$ of the form $(a, b, a+b+c)$, with $a, b \geqslant 1$, $c \geqslant 0$, appears in the sequence $(\mathbf{v}_n)_{n \geqslant 0}$.

In other words, when one coordinate is bigger or equal to the sum of the two others at some point of the execution of **FS**. Iterating the **FS** algorithm on $\mathbf{u}$ is equivalent to running Euclid's algorithm on (a, b) since the z-coordinate always remains the biggest one. This means that the value of c has no influence on the points computed in the sequence $(B_n)_{n \geqslant 0}$, and thus the geometry of the digital plane $\mathcal{P}(\mathbf{v}, \omega)$ cannot be described by the patterns generated. Another type of vectors which **FS** does not reduce to $\mathbf{1}$ is those such that a vector of the form (a, a, b) with $a < b$ appears in the sequence $(\mathbf{v}_n)_{n \geqslant 0}$. In such case, the action of **FS** produces a vector of the form $(a, 0, b - a)$, which is a fixed point for **FS**.

4.1   Hybrid GCF-Algorithms

The idea of mixing two different GCF-algorithms appears in [7], where the authors investigate the generation of infinite words by iteration of morphisms as approximations of digital lines. We now define two new GCF-algorithms. The main idea is to emulate **FS** as much as possible, except for the cases where it "fails" and, in such case, use **S** (Selmer) or **B** (Brun) instead.

Table 2 details the **FSS** (Fully Subtractive Selmer) and **FSB** (Fully Subtractive Brun) GCF-algorithms with the matrices provided in Table 3.

The scheme of the **FSS** (resp. **FSB**) GCF-algorithm is "*if the largest coordinate is greater or equal to the sum of the two smallest ones or if the two smallest ones are equal, apply the* **S** *(resp.* **B***) reduction; otherwise, apply the* **FS** *reduction*" (Fig. 3).

Unlike what happens with the **FS**, **S** and **B** GCF-algorithms, $\text{length}_{\mathbf{FSS}}(\mathbf{v})$ and $\text{length}_{\mathbf{FSB}}(\mathbf{v})$ always exist.

Table 2. The **FSS** and **FSB** GCF-algorithms.

Algorithm 1. Fully Subtractive Selmer	**Algorithm 2.** Fully Subtractive Brun
1: **function FSS** $(\mathbf{v})$	1: **function FSB** $(\mathbf{v})$
2: **Input:** $\mathbf{v} \in \mathbb{N}^3$	2: **Input:** $\mathbf{v} \in \mathbb{N}^3$
3: $(a, b, c) \leftarrow \text{SORTED}(\mathbf{v})$	3: $(a, b, c) \leftarrow \text{SORTED}(\mathbf{v})$
4: $\sigma_{\mathbf{v}} \leftarrow \min\{\sigma \in S_3 \mid \sigma(\mathbf{v}) = (a, b, c)\}$	4: $\sigma_{\mathbf{v}} \leftarrow \min\{\sigma \in S_3 \mid \sigma(\mathbf{v}) = (a, b, c)\}$
5: **if** $c \geqslant a + b$ or $a == b$ **then**	5: **if** $c \geqslant a + b$ or $a == b$ **then**
6: **return** $\mathbf{M}_{\sigma}^{\text{S}}$	6: **return** $\mathbf{M}_{\sigma}^{\text{B}}$
7: **else**	7: **else**
8: **return** $\mathbf{M}_{\sigma}^{\text{FS}}$	8: **return** $\mathbf{M}_{\sigma}^{\text{FS}}$
9: **end if**	9: **end if**
10: **end function**	10: **end function**

Table 3. The matrices for **FSS** and **FSB** GCF-algorithms. Each matrix is indexed by a permutation that indicates the relative order of the coordinates of the vector. We display only the matrices for the permutation 123, which corresponds to a vector (a, b, c) with $a \leqslant b \leqslant c$. Matrices for other permutations can be deduced from these.

$$\mathbf{M}_{123}^{\mathrm{FS}} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}, \mathbf{M}_{123}^{\mathrm{S}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}, \mathbf{M}_{123}^{\mathrm{B}} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}.$$

$\mathbf{v}_0 = (4, 5, 6),$
$\mathbf{a}_0 = (1, 1, 1)$
$\mathbf{h}_0 = (0, 0, 0),$
$L_0 = \{(1, 0, 0), (0, 1, 0), (0, 0, 1)\}.$

$M_1 = \mathbf{M}_{123}^{\mathrm{FS}},$
$\mathbf{v}_1 = (4, 1, 2),$
$\mathbf{a}_1 = (1, 2, 2)$
$\mathbf{t}_1 = (1, 0, 0),$
$\mathbf{h}_1 = (1, 0, 0),$
$L_1 = \{(2, 0, 0), (0, 1, 0), (0, 0, 1)\}.$

$M_2 = \mathbf{M}_{231}^{\mathrm{B}}$
$\mathbf{v}_2 = (2, 1, 2),$
$\mathbf{a}_2 = (2, 3, 3)$
$\mathbf{t}_2 = (-1, 1, 0),$
$\mathbf{h}_2 = (0, 0, 1),$
$L_2 = \{(2, 0, 0), (-1, 1, 1), (-1, 0, 2)\}.$

$M_3 = \mathbf{M}_{213}^{\mathrm{FS}},$
$\mathbf{v}_3 = (1, 1, 1),$
$\mathbf{a}_3 = (4, 5, 6)$
$\mathbf{t}_3 = (-1, 1, 0),$
$\mathbf{h}_3 = (-1, 1, 1),$
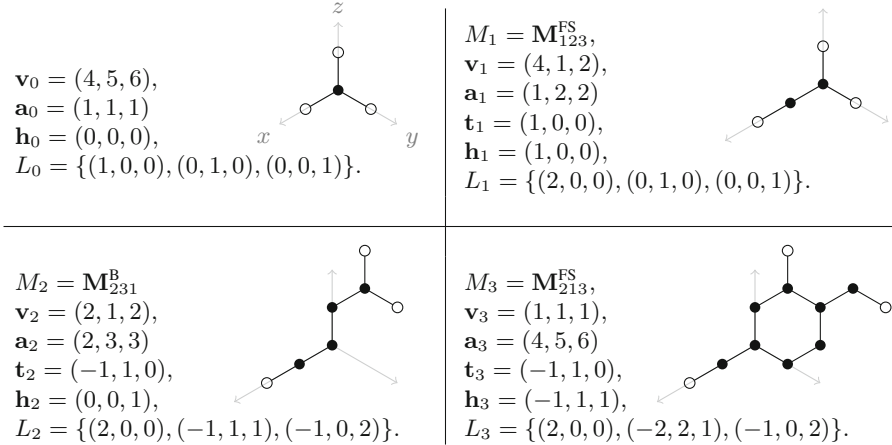$L_3 = \{(2, 0, 0), (-2, 2, 1), (-1, 0, 2)\}.$

Fig. 3. Example of the construction scheme using the **FSB** GCF-algorithm and normal vector $\mathbf{v} = (4, 5, 6)$. At step 3, $\mathbf{v}_3 = \mathbf{1}$ and thus $\mathbf{a}_3 = \mathbf{v}$. For each point $\mathbf{x} \in L_3$, $\bar{\mathbf{x}} = 8$ and the body has a point at each height from 0 to 7. The spanning $\{(B_3 \cup L_3) + k(-4, 2, 1) + l(-3, 0, 2) \mid k, l \in \mathbb{Z}\}$ is the digital plane $\mathcal{P}((4, 5, 6), 9)$.

Lemma 4.1. *Let $\mathbf{v} \in \mathbb{N}_+^3$, be a vector with relatively prime coordinates, then* length$_{\mathbf{FSS}}(\mathbf{v})$ *and* length$_{\mathbf{FSB}}(\mathbf{v})$ *both exist.*

Proof. With both algorithms, $(\|\mathbf{v}_n\|_1)_{n \geqslant 0}$ forms a decreasing integer sequence. More precisely, this sequence is strictly decreasing as long as none of the coordinates is equal to zero, which forces that, inevitably, one coordinate must reach zero. Let N be the smallest integer such that $\min(\mathbf{v}_{N+1}) = 0$. For short, let $(a, b, c) = \mathbf{v}_N$ and, w.l.o.g. assume that $a \leqslant b \leqslant c$. Let $M = \mathbf{FSS}(\mathbf{v}_N)$ (resp. $\mathbf{FSB}(\mathbf{v}_N)$). There are three possibilities for M:

- $M = \mathbf{M}_1^{\mathrm{FS}}$, this case is impossible, since both **FSS** and **FSB** require that $\mathbf{v}_N$ is such that $a < b$ and $a < c$, so that $\mathbf{v}_{N+1} = (a, b - a, c - a)$ has no coordinate equal to zero.
- $M = \mathbf{M}_{123}^{\mathrm{S}}$, in such case, $\mathbf{v}_{N+1} = (a, b, c - a)$, so that $a = b = c$.
- $M = \mathbf{M}_{123}^{\mathrm{B}}$, in such case, $\mathbf{v}_{N+1} = (a, b, c - b)$, so that $b = c$. Moreover, $a = b$ since, otherwise, $a + b > c$, $a \neq b$ and $\mathbf{FSB}(\mathbf{v}_N)$ returns $\mathbf{M}_1^{\mathrm{FS}}$.

The uniqueness of N is obvious since $\mathbf{v}_N = \mathbf{1}$ implies $\min(\mathbf{v}_{N+1}) = 0$ and each coordinate is non-increasing.   □

4.2   Generating a Digital Plane

We now show the main geometrical properties of our construction. First, we show that the points of the patterns B_n and $B_n \cup L_n$ are connected. Since we already know (see Proposition 3.4) that, at the last step, the legs define period vectors that send points of the legs to other points of the legs, the spanning of a pattern by these period vectors is a connected set. Then, we show that B_N, for $N = \text{length}_{\mathbf{FSS}}(\mathbf{v})$ (resp. **FSB**), contains at least one point at each height, which ensures that the spanning is a digital plane.

Theorem 4.2. *Let $\mathbf{v} \in \mathbb{N}_+^3$ be a vector with relatively prime coordinates and let $N = \text{length}_{\mathbf{FSS}}(\mathbf{v})$ (resp. **FSB**). For each $n \in \{0, 1, \ldots, N\}$, the sets B_n and L_n are disjoint, and both B_n and $B_n \cup L_n$ are connected.*

Proof. Let $\mathbf{x} \in L_n$. By definition, there exists $i \in \{1, 2, 3\}$ such that $\mathbf{x} = \mathbf{h}_n + M_1^\top \cdots M_n^\top \mathbf{e}_i$. The height of $M_1^\top \cdots M_n^\top \mathbf{e}_i$ is strictly greater than zero since it is equal to the value of the i-th coordinate of $\mathbf{v}_n$. This implies that $\mathbf{x} \notin B_n$ since $\overline{\mathbf{x}} > \overline{\mathbf{h}}_n$ and $\mathbf{h}_n$ is the highest point of B_n.

In order to show that $B_n \cup L_n$ is connected, it suffices to see that, for all $n \geqslant 0$ and all $i \in \{1, 2, 3\}$,

$$\mathbf{h}_n + M_1^\top \cdots M_n^\top \mathbf{e}_i - \mathbf{e}_i \in B_n. \tag{1}$$

Indeed, Eq. (1) implies that the points of L_n are adjacent to points of B_n. It also implies that, at each step of the recursive construction, the set $B_{n-1} + \mathbf{t}_n$ contains a point, namely $\mathbf{h}_{n-1} + M_1^\top \cdots M_n^\top \mathbf{e}_{\delta_n}$, that is adjacent to a point of B_{n-1}, so that the connectedness of B_{n-1} implies the one of B_n.

By recurrence, for $n = 0$, $\mathbf{h}_0 + \mathbf{e}_i - \mathbf{e}_i = \mathbf{0} \in B_0$. Now, suppose that (1) is true for n, and we show the result for $n + 1$. There are two cases to consider.

– If the action of M_{n+1} does not modify the i-th coordinate of $\mathbf{v}_n$, that is $M_{n+1}^\top \mathbf{e}_i = \mathbf{e}_i$, then

$$\mathbf{h}_{n+1} + M_1^\top \cdots M_{n+1}^\top \mathbf{e}_i - \mathbf{e}_i = \underbrace{\mathbf{h}_n + M_1^\top \cdots M_n^\top \mathbf{e}_i - \mathbf{e}_i}_{\in B_n} + \mathbf{t}_{n+1} \in B_{n+1}.$$

– Otherwise, the action of M_{n+1} on the i-th coordinate of $\mathbf{v}_n$ is to subtract to it the δ_{n+1}-th coordinate, which implies that $M_{n+1}^\top \mathbf{e}_i = \mathbf{e}_i - \mathbf{e}_{\delta_{n+1}}$ and thus,

$$\begin{aligned} &\mathbf{h}_{n+1} + M_1^\top \cdots M_{n+1}^\top \mathbf{e}_i - \mathbf{e}_i \\ &= \mathbf{h}_{n+1} + M_1^\top \cdots M_n^\top \mathbf{e}_i - M_1^\top \cdots M_n^\top \mathbf{e}_{\delta_{n+1}} - \mathbf{e}_i \\ &= \underbrace{\mathbf{h}_n + M_1^\top \cdots M_n^\top \mathbf{e}_i}_{\in B_n} - \underbrace{M_1^\top \cdots M_{n+1}^\top \mathbf{e}_{\delta_{n+1}}}_{= \mathbf{t}_{n+1}} + \mathbf{t}_{n+1}. \end{aligned}$$

For the last equality, note that since M_{n+1} does not modify the δ_{n+1}-th coordinate of $\mathbf{v}_n$, we have $e_{\delta_{n+1}} = M_{n+1}^\top e_{\delta_{n+1}}$.   □

The second important result of the present work claims that the **FSS** and **FSB** GCF-algorithms construct an entire connected digital plane, using translations.

Theorem 4.3. *Let $\mathbf{v} \in \mathbb{N}_+^3$, be a vector with relatively prime coordinates, let $N = \text{length}_{\mathbf{FSS}}(\mathbf{v})$ (resp. $\text{length}_{\mathbf{FSB}}(\mathbf{v})$). For each $n \in \{0, 1, \ldots, N\}$, the set $\{\langle \mathbf{x}, \mathbf{a}_n \rangle \mid \mathbf{x} \in B_n \cup L_n\}$ is an integer interval.*

The proof of Theorem 4.3 relies on a technical lemma, for which we define the following notation: given a finite set $S \subset \mathbb{N}$, let $\text{SUM}(S) = \sum_{s \in S} s$ be the sum of its elements and $\text{PSUM}(S) = \{\sum_{x \in X} x \mid X \subseteq S\}$ be the set of its partial sums.

Lemma 4.4. *Let $x \in \mathbb{N}$ and $S \subset \mathbb{N}$ be such that $\text{PSUM}(S)$ is an integer interval. If $x \leqslant \text{SUM}(S) + 1$, then $\text{PSUM}(S \cup \{x\})$ is an integer interval.*

Proof. Let $a \in \{0, 1, \ldots, \text{SUM}(S) + x\}$. If $a \leqslant \text{SUM}(S)$, then $a \in \text{PSUM}(S) \subseteq \text{PSUM}(S \cup \{x\})$. Otherwise, $0 \leqslant a - x \leqslant \text{SUM}(S)$ and $a - x \in \text{PSUM}(S)$. $\square$

*Proof (of Theorem 4.3 – Sketch). The proof is given for **FSS** algorithm, the case **FSB** being similar.* Let,

$$\mathbb{T}_{n,k} = \{\mathbf{t}_{n-k}, \ldots, \mathbf{t}_n\}, \qquad \overline{\overline{\mathbb{T}}}_{n,k} = \{\langle \mathbf{x}, \mathbf{a}_n \rangle \mid \mathbf{x} \in \mathbb{T}_{n,k}\}.$$

Hence $\overline{\overline{\mathbb{T}}}_{n,0} = \{\mathbf{0}\}$ and its partial sums is an integer interval. Let us suppose that the partial sums of the set $\overline{\overline{\mathbb{T}}}_{n,k}$, for some $k \in \{0, \ldots, n-1\}$, form an integer interval and let us show that the partial sums of the set $\overline{\overline{\mathbb{T}}}_{n,k+1}$ form an integer interval too, using

$$\mathbb{T}_{n,k+1} = \mathbb{T}_{n,k} \cup \{\mathbf{t}_{n-k-1}\}, \qquad \overline{\overline{\mathbb{T}}}_{n,k+1} = \overline{\overline{\mathbb{T}}}_{n,k} \cup \{\langle \mathbf{t}_{n-k-1}, \mathbf{a}_n \rangle\}.$$

According to Lemma 4.4, it is sufficient to show:

$$\langle \mathbf{t}_{n-k-1}, \mathbf{a}_n \rangle \leqslant 1 + \sum_{x \in \overline{\overline{\mathbb{T}}}_{n,k}} x. \tag{2}$$

Remind that $\mathbf{a}_n = M_1^{-1} M_2^{-1} \cdots M_n^{-1} \cdot \mathbf{1}$, or, equivalently, $\mathbf{1} = M_n \ldots M_1 \cdot \mathbf{a}_n$. Let $\mathbf{a}_{n,0} = \mathbf{a}_n$ and $\mathbf{a}_{n,i+1} = M_i \cdot \mathbf{a}_{n,i}$, with $i \in \{0, \ldots, n-1\}$. The *action* of M_i consists in subtracting $\langle \mathbf{t}_i, \mathbf{a}_n \rangle$ to at least one coordinate of $\mathbf{a}_{n,i-1}$ to compute $\mathbf{a}_{n,i}$, and $\langle \mathbf{t}_i, \mathbf{a}_n \rangle$ is the minimal coordinate of $\mathbf{a}_{n,i-1}$. Since $\mathbf{a}_{n,n} = \mathbf{1}$, there exists a subset $J \subseteq \{i+1, \ldots, n\}$ such that $1 = \langle \mathbf{t}_i, \mathbf{a}_n \rangle - \sum_{j \in J} \langle \mathbf{t}_i, \mathbf{a}_n \rangle$, that is:

$$\langle \mathbf{t}_i, \mathbf{a}_n \rangle = 1 + \sum_{j \in J} \langle \mathbf{t}_i, \mathbf{a}_n \rangle \leqslant 1 + \sum_{j=i+1}^{n} \langle \mathbf{t}_i, \mathbf{a}_n \rangle.$$

Setting $i = n - k - 1$, condition (2) holds. $\square$

Now, it becomes natural to investigate the thickness of the digital plane generated.

Theorem 4.5. *Let* $\mathbf{v} \in \mathbb{N}_+^3$, *be a vector with relatively prime coordinates, let* $\mathbf{X}$ *be either* **FSS** *or* **FSB** *and let* $N = \text{length}_{\mathbf{X}}(\mathbf{v})$. *The thickness is bounded:*

$$\left\lfloor \frac{\|\mathbf{v}\|_1}{2} \right\rfloor \leqslant \max\{\langle \mathbf{x}, \mathbf{v} \rangle \mid \mathbf{x} \in B_N \cup L_N\} \leqslant \|\mathbf{v}\|_1 - 2.$$

Proof. The highest point of B_N is, by construction, $h_N = \mathbf{t}_1 + \mathbf{t}_2 + \cdots + \mathbf{t}_N$. Also,

$$\max\{\langle \mathbf{x}, \mathbf{v} \rangle \mid \mathbf{x} \in B_N\} = \sum_{i=1}^{N} \bar{\mathbf{t}}_i,$$

which is, by Property 3.1 (6), the sum of the values subtracted at each step of the execution of the algorithm.

Using the definition of the algorithms, Selmer and Brun subtract each time this value to exactly one coordinate. In such case, $\|\mathbf{v}_n\|_1 = \|\mathbf{v}_{n-1}\|_1 - \bar{\mathbf{t}}_{n-1}$. For Fully Subtractive algorithm, one subtracts at each step the latter value to two different coordinates. In such case, $\|\mathbf{v}_n\|_1 = \|\mathbf{v}_{n-1}\|_1 - 2\bar{\mathbf{t}}_{n-1}$.

This means that if every step of the reduction is Selmer or Brun, then $\|\mathbf{v}\|_1 = 3 + \sum_{i=1}^{N} \bar{\mathbf{t}}_i$. On the other hand, if the reduction only uses the Fully Subtractive matrices, then $\|\mathbf{v}\|_1 = 3 + 2\sum_{i=1}^{N} \bar{\mathbf{t}}_i$.

A mix of both Fully Subtractive and Selmer (resp. Brun) reductions returns a value that lies between the bounds for iterating only **FS** or only **S** (resp. **B**). Finally, by Property 3.1 (7), and Lemma 4.1, the height of the points in L_N is exactly 1 more than the highest point in B_N, showing the result. $\square$

Let us remind that according to Proposition 3.4, the set L_N provides linearly independent vectors allowing us to span the entire connected digital plane $\mathcal{P}(\mathbf{v}, \omega)$, with $\omega = \max\{\langle \mathbf{x}, \mathbf{v} \rangle \mid \mathbf{x} \in B_N \cup L_N\} + 1 < \|\mathbf{v}\|_1$.

5   Conclusion and Further Work

In the present paper, we have provided a process to construct connected digital planes with integer normal vectors. This process is guided by a GCF-algorithm which mixes the Fully Subtractive and Selmer (resp. Brun) GCF-algorithms. The result is an algorithm that recursively builds a digital plane for any rational normal vector. However, although the resulting digital plane is connected, it is not, in general, the thinnest one among the connected digital planes with the same normal vector. More precisely, the more the GCF-algorithm applies the Fully Subtractive reduction, the thinner the digital plane is. The GCF-algorithms **FSS** and **FSB** construct different patterns but we are not able to state that one is better than the other. As a future work, we want to identify more geometric properties of these patterns. Moreover, we hope that new GCF-algorithms will provide patterns with even more properties like the ability to tile a digital plane or to control the anisotropy of the patterns. These properties, among others, would be of interest for practical use in the context of image analysis.

References

1. Arnoux, P., Ito, S.: Pisot substitutions and Rauzy fractals. Bull. Belg. Math. Soc. Simon Stevin **8**(2), 181–207 (2001)
2. Berstel, J., Lauve, A., Reutenauer, C., Saliola, F.V.: Combinatorics on Words: Christoffel Words and Repetitions in Words. CRM Monograph Series, vol. 27. AMS, Providence, RI (2009)
3. Berthé, V., Fernique, T.: Brun expansions of stepped surfaces. Discrete Math. **311**(7), 521–543 (2011)
4. Berthé, V., Domenjoud, E., Jamet, D., Provençal, X.: Fully subtractive algorithm, tribonacci numeration and connectedness of discrete planes. Research Institute for Mathematical Sciences, Lecture note Kokyuroku Bessatu B46, pp. 159–174 (2014)
5. Berthé, V., Jamet, D., Jolivet, T., Provençal, X.: Critical connectedness of thin arithmetical discrete planes. In: Gonzalez-Diaz, R., Jimenez, M.-J., Medrano, B. (eds.) DGCI 2013. LNCS, vol. 7749, pp. 107–118. Springer, Heidelberg (2013)
6. Berthé, V., Jamet, D., Jolivet, T., Provençal, X.: Critical connectedness of thin arithmetical discrete planes (2013). http://arxiv.org/abs/1312.7820
7. Berthé, V., Labbé, S.: Uniformly balanced words with linear complexity and prescribed letter frequencies. In: WORDS, pp. 44–52 (2011)
8. Brimkov, V., Coeurjolly, D., Klette, R.: Digital planarity–a review. Discrete Appl. Math. **155**(4), 468–495 (2007)
9. Domenjoud, E., Jamet, D., Toutant, J.-L.: On the connecting thickness of arithmetical discrete planes. In: Brlek, S., Reutenauer, C., Provençal, X. (eds.) DGCI 2009. LNCS, vol. 5810, pp. 362–372. Springer, Heidelberg (2009)
10. Domenjoud, E., Provençal, X., Vuillon, L.: Facet connectedness of discrete hyperplanes with zero intercept: the general case. In: Barcucci, E., Frosini, A., Rinaldi, S. (eds.) DGCI 2014. LNCS, vol. 8668, pp. 1–12. Springer, Heidelberg (2014)
11. Domenjoud, E., Vuillon, L.: Geometric palindromic closure. Uniform Distrib. Theory **7**(2), 109–140 (2012)
12. Fernique, T.: Generation and recognition of digital planes using multi-dimensional continued fractions. Pattern Recogn. **42**(10), 2229–2238 (2009)
13. Fokkink, R., Kraaikamp, C., Nakada, H.: On schweiger's problems on fully subtractive algorithms. Isr. J. Math. **186**(1), 285–296 (2011)
14. Jolivet, T.: Combinatorics of Pisot substitutions. Ph.D. thesis, Université Paris Diderot, University of Turku (2013)
15. Klette, R., Rosenfeld, A.: Digital straightness - a review. Discrete Appl. Math. **139**(1–3), 197–230 (2004)
16. Labbé, S.: 3-dimensional continued fraction algorithms cheat sheets (2015). http://arxiv.org/abs/1511.08399
17. Labbé, S., Reutenauer, C.: A d-dimensional extension of christoffel words. Discrete Comput. Geom. **54**(1), 152–181 (2015)
18. Schweiger, F.: Multidimensional Continued Fractions. Oxford Science Publications, Oxford University Press, Oxford (2000)