Graph encoding of 2D-gon tilings.

Frédéric Chavanon, Matthieu Latapy, Michel Morvan, Laurent Vuillon

Laboratoire de l'Informatique du Parallélisme

École Normale Supérieure de Lyon
Unité Mixte de Recherche CNRS-INRIA-ENS LYON n° 5668

Graph Encoding of 2D-gon Tilings

Frédéric Chavanon, Matthieu Latapy,
Michel Morvan, Laurent Vuillon

Septembre 2003

Research Report N° 2003-43

École Normale Supérieure de Lyon, 46 Allée d'Italie, 69364 Lyon Cedex 07, France
Téléphone : +33(0)4.72.72.80.37 Fax : +33(0)4.72.72.80.80 Adresse électronique : `lip@ens-lyon.fr`

Graph Encoding of 2D-gon Tilings

Frédéric Chavanon, Matthieu Latapy, Michel Morvan, Laurent Vuillon

Septembre 2003

Abstract

2D-gons tilings with parallelograms are the main model used in physics to study quasicrystals, and they are also important in combinatorics for the study of aperiodic structures. In this paper, we study the graph induced by the adjacency relation between tiles. This relation can been used to encode simply and efficiently 2D-gon tilings for algorithmic manipulation. We show for example how it can be used to sample random 2D-gon tilings.

Keywords: tilings, quasicrystals, combinatorics

Résumé

Les pavages de 2D-gones parparallélogrammes sont un des principaux modèles de quasicristaux utilisés en physique, ainsi qu'en combinatoire pour l'étude des structures apériodiques. Dans ce papier nous étudions le graphe induit par les relations d'adjacence entre tuiles. Cette relation est utilisée afin de coder les pavages de 2D-gones de façon simple et utilisable pour des manipulations algorithmiques. Nous montrons par exemple comment ce codage peut être exploité pour la génération aléatoire de pavages de 2D-gones.

Mots-clés: pavages, quasicristaux, combinatoire

Graph Encoding of 2D-gon Tilings

Frédéric Chavanon [1], Matthieu Latapy [2, 3],
Michel Morvan [1] and Laurent Vuillon [3]

Abstract : 2D-gons tilings with parallelograms are the main model used in physics to study quasicrystals, and they are also important in combinatorics for the study of aperiodic structures. In this paper, we study the graph induced by the adjacency relation between tiles. This relation can been used to encode simply and efficiently 2D-gon tilings for algorithmic manipulation. We show for example how it can be used to sample random 2D-gon tilings.

1 Introduction

A tiling can be defined as a partition of a given region of an affine space. More classically, one considers a finite set of shapes, called *prototiles*, and a region of an affine space. The tiling problem is then to decide whether this region can be tiled, i.e. covered by translated copies of prototiles, without gaps or overlappings between them. If this is possible, the region is *tilable*, and a solution is called a *tiling* of the region. The translated copies of the prototiles are the *tiles* of the tiling. If the region to tile is the whole plane, this problem has been shown to be undecidable by Berger [Ber66], which was the first important incursion of tilings in computer science.

In this paper, we are concerned with tilings of $2D$-gons with parallelograms. A $2D$-gon is an hexagon when D=3, an octagon when D=4, a decagon when D=5, etc. Such a region can always be tiled with parallelograms. $2D$-gon tilings by parallelograms appear in physics as a model for quasicrystals [Des97] and aperiodic structures [Sen95]. They are also used to encode several combinatorial problems [Eln97, Lat00], and have been studied from many points of view [Lat00, RGZ94, Bai99]. In particular, they are strongly related to the oriented matroid theory, since the Bohne-Dress theorem proves the equivalence of $2D$-gon tilings with a class of matroids [RGZ94, BVS$^+$99].

These tilings cannot be easily manipulated by a program when one uses the geometric definitions. Some efficient solutions arise from the oriented matroids side [BFF00, CFG00]. We propose here another solution from graph theory, which has the advantage of giving very simple algorithms and to introduce some interesting questions about graphs related to 2D-gon tilings. The aim of this paper is to study the graph induced by the adjacency of tiles, and to find the minimal amount of information to add to this graph in order to provide an effective notion of dual graph of a tiling. We will first present the tilings more formally, and define the adjacency graph we will use. Then we will obtain a one-to-one correspondence between a class of graphs and $2D$-gon tilings by introducing the notion of *graph with origins*, and we give an algorithm which builds the $2D$-gon tiling corresponding to a given graph with origins. We will finally see how the flip operation can be defined on the graph, which makes it possible to sample random tilings of $2D$-gons. Let us emphasize on the fact that we

[1]LIP, umr 5668 CNRS-INRIA ENS Lyon 46, allée d'Italie 69364 Lyon cedex 07, France. (frederic.chavanon, michel.morvan)@ens-lyon.fr

[2]INRIA Rocquencourt. BP 105. 78153 Le Chesnay Cedex, France. Matthieu.Latapy@inria.fr

[3]LIAFA, Université Paris 7 2, place Jussieu 75005 Paris, France. latapy@liafa.jussieu.fr

have two aims in this study : give some properties of the adjacency graph of $2D$-gon tilings, which is a fundamental object on which very few information is known, and provide a very simple and efficient way to algorithmically manipulate 2D-gon tilings.

2 Preliminaries

Given two vectors v and v', we will say that $v < v'$ in the *natural order* if the angle between $(1,0)$ and v_i is smaller than the angle between $(1,0)$ and v_{i+1}. Let $V = \{v_1, v_2, ..., v_D\}$ be a family of D pairwise non-colinear positive vectors of the plane. We suppose that for all integer i, $v_i < v_{i+1}$ in the natural order. Let $M = \{m_1, m_2, ..., m_D\}$ be a family of D positive integers. The integer m_i is called the *multiplicity* of v_i. The $2D$-gon $P = (V, M)$ associated to V and M is the region of the affine plane defined by :

$$\left\{ \sum_{i=1}^{D} \lambda_i v_i, 0 \leq \lambda_i \leq m_i, m_i \in M, v_i \in V \right\}$$

There exists many equivalent definitions for these objects. For example, a $2D$-gon can be viewed as the projection of a hypercube of dimension D onto the plane. See [Zie95] for more details. For D=2, the $2D$-gons are parallelograms ; for D=3, hexagons ; for D=4, octagons ; for D=5, decagons ; etc. See Figure 1 for an illustration.

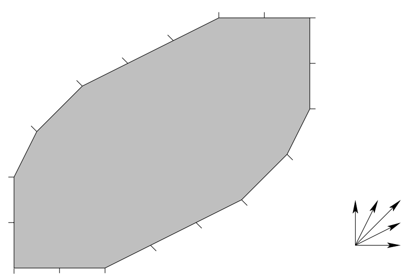

FIG. 1 – A $2D$-gon and its vectors, associated to $M = \{2, 3, 1, 1, 2\}$. This is a decagon with sides of length 2,3,1,1,2,2,3,1,1,2.

Given a $2D$-gon $P = (V, M)$, a prototile of P is a $2D$-gon built using only 2 vectors in V, each of them with multiplicity 1. Therefore, each prototile of P is a parallelogram defined by two vectors in V, and we will make no distinction between the prototile viewed as an area and the pair of the indexes of the vectors in V which define it.

Finally, a tiling T of a $2D$-gon $P = (V, M)$ is a set of tiles (i.e. translated copies of the prototiles) which cover exactly P and such that there is no overlapping between tiles. Therefore, T is a set of couples, their first component being the pair of vectors which defines the prototile, the second one being a translation. The translations used in $2D$-gon tilings can always be written as a linear combination of vectors in V : $t = \sum_i t_i v_i$, t_i being an integer between 0 and m_i. Therefore, t will be described by the D-dimension vector $(t_1, ..., t_D)$. Two tilings T and T' of two $2D$-gons P and P' are said to be *equivalent* if $T = T'$, where T and T' are viewed as sets of couples.
Consider for example the three tilings in Figure 2. From left to right, they are described by :

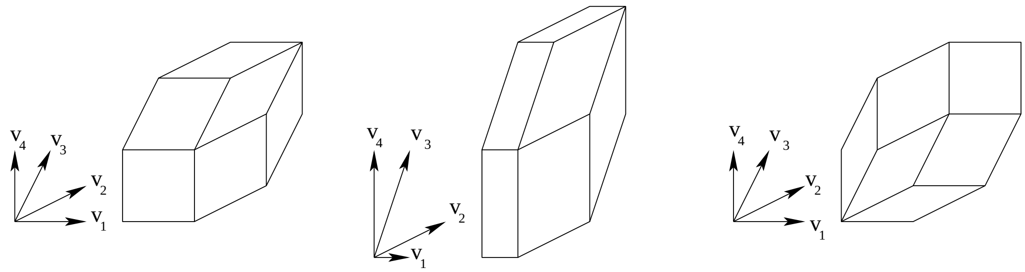

FIG. 2 – Three tilings of $2D$-gons, namely T_1, T_2 and T_3 from left to right ; T_1 and T_2 are *equivalent*, whereas T_3 is equivalent to none of the others.

$$T_1 = \left\{ \begin{array}{l} (\{1,4\},(0,0,0,0)),(\{2,4\},(1,0,0,0)),(\{3,4\},(1,1,0,0)),(\{2,3\},(1,0,0,1)), \\ (\{1,3\},(0,0,0,1)),(\{1,2\},(0,0,1,1)) \end{array} \right\}$$
$$T_2 = \left\{ \begin{array}{l} (\{1,2\},(0,0,1,1)),(\{1,3\},(0,0,0,1)),(\{1,4\},(0,0,0,0)),(\{2,3\},(1,0,0,1)), \\ (\{2,4\},(1,0,0,0)),(\{3,4\},(1,1,0,0)) \end{array} \right\}$$
$$T_3 = \left\{ \begin{array}{l} (\{1,2\},(0,0,0,0)),(\{1,3\},(0,1,0,0)),(\{1,4\},(0,1,1,0)),(\{2,3\},(0,0,0,0)), \\ (\{2,4\},(0,0,1,0)),(\{3,4\},(0,0,0,0)) \end{array} \right\}$$

Therefore, tilings T_1 and T_2 are equivalent, while T_1 and T_3 are not.

Let P be a $2D$-gon, and T be a tiling of P. The i-th *de Bruijn family* of T is the set of all the tiles in T which are built with the vector v_i. Moreover, each family can be decomposed into *de Bruijn lines* : the j-th de Bruijn line of the i-th family is the set of tiles built with v_i which have $j-1$ as the i-th component of their translation. Continuing with our example of Figure 2, we obtain that the first line of the second de Bruijn family is equal to $\{(\{2,4\},(1,0,0,0)),(\{2,3\},(1,0,0,1)),((2,1),(1,0,1,1))\}$. For practical convenience, we will also say that the j-th line of the i-th family is the α-th line of the tiling where $\alpha = \sum_{k=1}^{i-1} m_k + j$, m_k being the multiplicity of the k-th family. We also define $f(i)$ as the index of the vector associated to the i-th line, i.e. $f(i)$ is the number of the line's family. See Figure 3 for example. Notice that two lines in the same family never have a tile in common, whereas two lines in two different families always have exactly one tile in common. Moreover, each line divides the tiling into two disjoint parts. We will use these classical properties [dB81] in the following.

Before entering in the core of the paper, we need a few more notations, which we introduce now.

Definition 1 *Let $P = (V, M)$ be a $2D$-gon. For all k, $1 \leq k \leq D$ we define the k-th* side *of P as the set of points :*

$$\left\{ \sum_{i=1}^{D} \lambda_i v_i, \forall i < k : \lambda_i = m_i, \forall i > k : \lambda_i = 0, \textit{and } 0 \leq \lambda_k \leq m_i \right\}$$

Likewise, we define the $(k+D)$-th side of P as :

$$\left\{ \sum_{i=1}^{D} \lambda_i v_i, \forall i > k : \lambda_i = m_i, \forall i < k : \lambda_i = 0, \textit{and } 0 \leq \lambda_k \leq m_i \right\}$$

Moreover, the hull *of P, denoted by $H(P)$, is the union of all the sides of P. We will also say that a tile t is on the i-th side of P if one of the sides of t is included in the i-th side of P.*

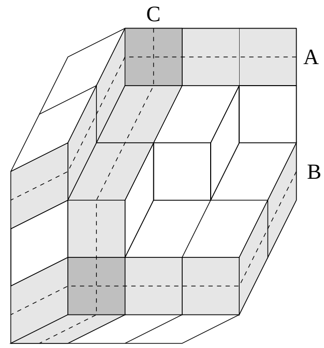

FIG. 3 – A $2D$-gon tiling and three de Bruijn lines (a line is a set of tiles crossed by a dotted line). A and B are in the same de Bruijn family. A is the third line of the fourth family, and B the second line of the same family. C is the first line of the first family. According to our notations, A is the 9-th line of the tiling, B is the 7-th, and C is the first. Notice that C crosses exactly once A and B, whereas A and B do not cross.

We can now introduce the notion of adjacency graph associated to a tiling, which will be the main object of our study.

Definition 2 (adjacency graph of a tiling) *Let T be a tiling of a given $2D$-gon $P = (V, M)$, and let $n = |T|$ be the number of tiles of this tiling. Let $\pi : T \to \{1, ..., n\}$ be a labelling of the tiles of T. The adjacency graph of T is the undirected graph $A(T) = (V_T, E_T)$ where $V_T = \{\pi(t), t \in T\}$ and $\{\pi(t), \pi(t')\} \in E_T$ if and only if t and t' have one side in common in T. See Figure 4 for an example.*

The adjacency graphs of $2D$-gon tilings will be our main object of interest in the rest of this paper. We will see that they encode much information on the tiling. However, the fact that two tilings have the same adjacency graph does not imply that they are equivalent : for example, one can verify that the tilings T_1 and T_3 in Figure 2 have the same adjacency graph. In order to obtain a one-to-one correspondence between a set of graphs and the set of tilings of a $2D$-gon, we introduce now the *de Bruijn graph*.

Definition 3 (de Bruijn graph of a tiling) *Let $A = (V, E)$ be the adjacency graph of a tiling T of a $2D$-gon P, and π be the mapping of the tiles of T to the vertices of A. The* de Bruijn graph *$A' = (V, E, \lambda, \nu)$ is a graph with labelled vertices and with a distinguished vertex ν. The label $\lambda(t)$ of $t \in V$ is the pair of integers $\{i, j\}$ such that the two de Bruijn lines which contain the tile $\pi^{-1}(t)$ are the $i - th$ and $j - th$. The vertex ν, called the* origin *of the graph, is associated to the tile with translation vector $(0, ..., 0)$ which is on the $2D$-th side of P. See Figure 4 for an example.*

Theorem 1 *Given the de Bruijn graph of a tiling T, Algorithm 1 constructs a tiling equivalent to T in time $O(n)$, where n is the number of tiles of T, i.e. the number of vertices of the graph.*

Proof : The idea of the algorithm is to start with the origin of the graph, and then make a breadth-first search which makes it possible to compute the tile associated to each vertex, i.e. the couple of vectors which describes the prototile, and the translation vector.

Let us consider a vertex v labelled with $\{i, j\}$, which means that it corresponds to a tile $t = (\tau, \text{trans})$ in the i-th de Bruijn line and the j-th one. Recall that $f(x)$ is the number of the

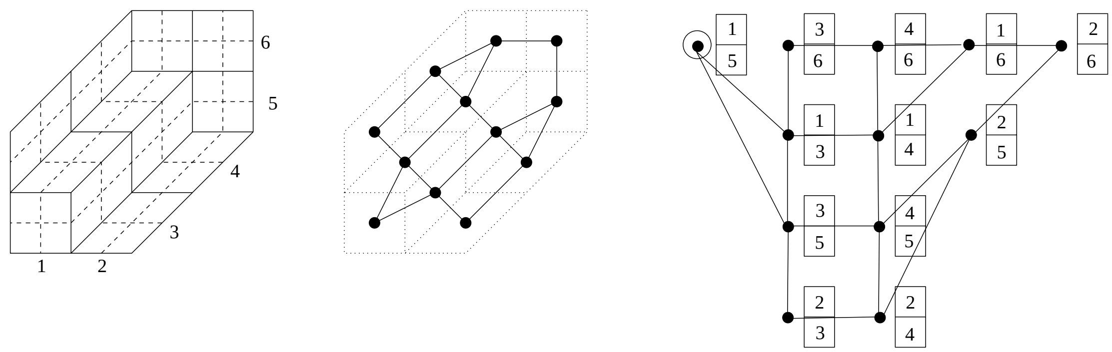

FIG. 4 – A tiling, its adjacency graph and its de Bruijn graph.

de Bruijn family of the line x. Then the prototile τ is defined by $f(i)$ and $f(j)$. The function f is easy to compute, since two de Bruijn lines i and j are in the same family if and only if they do not cross each other, i.e. if there is no vertex labelled $\{i, j\}$ in the de Bruijn graph. Therefore, we can easily find all the lines which belong to the same family, and so we can find the de Bruijn families. Finally, we obtain the prototile τ.

The next point is to compute the translation vectors. Each of them is deduced from the translation vector of a previously marked vertex. Since we start with a vertex with translation vector $(0, ..., 0)$ (the origin), and since we visit the vertices in a breadth-first order, a neighbor v' of v visited after v has a translation vector componentwise greater than or equal to the one of v. Suppose that we have already computed the tile $t = (\tau, \text{trans})$ which corresponds to v, and consider $t' = (\tau', \text{trans'})$ which corresponds to v'. As discussed above, we already have $t = \{i, j\}$ and $t' = \{j, k\}$ and trans' $\geq$ trans. The six cases illustrated in Figure 5 can occur. They lead to two possibilities :

- if $i > j > k$ (see Figure 5.a) or $k > j > i$ (see Figure 5.b), then trans' $=$ trans
- in the other cases (see Figure 5.c,d,e,f), the x-th component of the translation vector, where x is the label that t and t' have in common, has to be increased by one.

These remarks lead directly to Algorithm 1, and since we visit each vertex twice (once to compute $f(x)$ for all x, and once in the main loop), its complexity is $O(n)$, where n is the number of vertices.

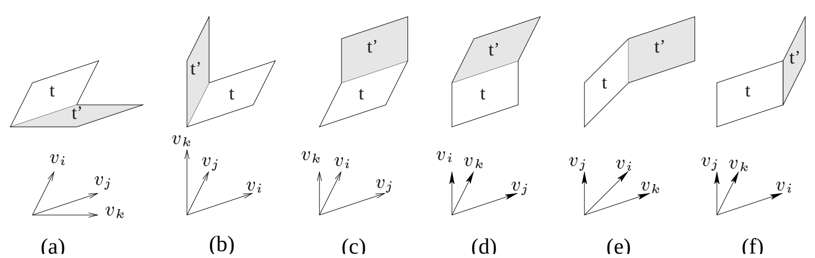

FIG. 5 – The six possible positions of the two tiles t and t' during the computation of the translation vectors.

□

Algorithm 1: Construction of a tiling from its de Bruijn graph.

Input: $G = (V, E, \lambda, \nu)$, the de Bruijn graph of a tiling T.

Output: A tiling equivalent to T, given by a list of $(tile, translation)$.

```
begin
    Let {i, j} = λ(ν) ;
    Set all the vertices as unmarked;
    resu ← {({f(i), f(j)}, (0, ..., 0))};
    current ← {(ν, (0, 0, ..., 0))};
    Mark ν ;
    while current ≠ ∅ ;
    do
        foreach τ = (v, trans) in current do
            foreach unmarked vertex v' such that (v, v') ∈ E ;
            do
                Let {i, j} be the label of v, and {j, k} be the label of v';
                Let trans' be a copy of trans;
                if not (f(i) > f(j) > f(k) or f(k) > f(j) > f(i)) then
                    Increase the f(i)-th component of trans' by one;
                resu ← resu ∪ {({f(j), f(k)}, trans')};
                current ← current ∪ {(v, trans')};
                Mark v';
            current ← current \ {v'};
    Return (resu);
end
```

This result shows that all the information contained in a $2D$-gon tiling is encoded in its de Bruijn graph. However, we will show that the de Bruijn graph contains much more information than really needed to construct the tiling. Actually, we will show that the adjacency graph contains almost all the information we need. Indeed, it suffices to add two marks to the adjacency graph of T to be able to reconstruct the tiling T. This leads to the definition of the graph with origins of a tiling T.

Definition 4 (graph with origins of a tiling) *Let T be a tiling of a $2D$-gon, $A = (V, E)$ its adjacency graph, and π the mapping of the tiles to the vertices. The* graph with origins *associated to T is $G = (V, E, v_1, v_2)$, where v_1 and v_2 are two vertices in V called the* origins *of G and defined as follows. Let t_1 be the tile of T on the $2D$-th side of P with translation vector $(0, ..., 0)$. Let t_2 be the tile on a side of P with translation vector $(0, ..., 1, 1)$ if t_1 is also on the $(2D-1)$-th side of P, else the tile on a side of P with translation vector $(0, ..., 0, 1)$. Then, $v_1 = \pi(t_1)$ and $v_2 = \pi(t_2)$. See Figure 6 for an example.*

Notice that the adjunction of the two origins makes it possible to distinguish two different tilings which have isomorphic adjacency graphs, as shown for example in Figure 6. We will show in the following that this is always true : the correspondence between the graphs with origins we defined and the $2D$-gon tilings is one-to-one.

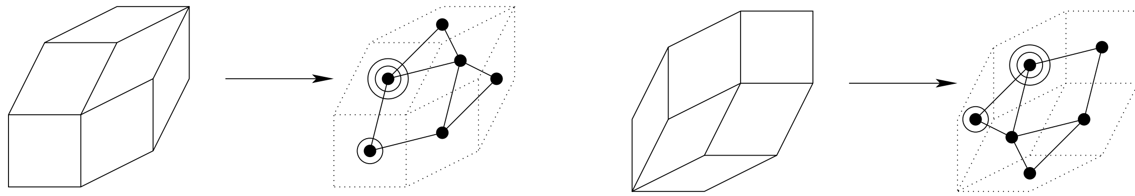

FIG. 6 – Two tilings and their graphs with origins. Notice that, if one removes the origins, then the two graphs are isomorphic.

3 Duality

In this section, we give an algorithm which computes the de Bruijn graph of a tiling from its graph with origins. This correspondence is one-to-one, therefore, together with Algorithm 1 and Theorem 1 it shows that the graphs we introduced can be considered as *dual* of the considered tilings, despite the fact that they are very close to adjacency graph (they only have two additional marks). Our algorithm has complexity $O(n \cdot m)$, where n is the number of vertices of the graph, or equivalently the number of tiles of the tiling, and m is the sum of the multiplicities used to define the $2D$-gon.

In order to build the algorithm and prove its correctness, we will first prove some properties linking tilings of $2D$-gons and their adjacency graphs. In particular, some special sub-structures, namely *borders* and fans, will play a very important role. We introduce them now, and prove some of their basic properties.

Definition 5 (border of a tiling) *Let T be a tiling of a $2D$-gon P and π be the mapping of the tiles to the vertices of its adjacency graph. Let $C \subseteq T$ be the set of tiles of T which have at least one point in $H(P)$. We define the* border *of T, denoted by $B(T) = (\pi(C), E)$, as follows : $(\pi(t), \pi(t')) \in E$ if and only if t and t' have one side in common and if this common side has at least one point in $H(P)$. Notice that this is a subgraph of the adjacency graph of T, but it is not the subgraph* induced *by C (some edges are missing). See Figure 7.*

Definition 6 (border of a graph) *Let G be a graph. The border of G, denoted by $\mathcal{B}(G)$, is the unique shortest self-avoiding cycle in G which contains all the vertices of G of degree at most 3, if it exists. See Figure 7.*

Definition 7 (fan) *Given a tiling T of a $2D$-gon P, a* fan *F of T is a k-tuple $(f_1, f_2, .., f_k)$ of tiles in T such that for all i between 1 and k :*

- *f_i and f_{i+1} have a side in common,*
- *$\cap_{i=1}^{k} f_i$ is a point of $H(P)$,*
- *f_1 and f_k have one side on the hull of the $2D$-gon.*

The tiles f_1 and f_k are called the extremities *of F, and the integer $k-2$ is called the* size *of F. Moreover, we say that a fan F belongs to the $i-th$ side of P if one of its extremities has a side included in the $i-th$ side of P.*

The first step of algorithm to construct a tiling equivalent to a tiling T starting from the graph with origins of T will be to construct its border. We will first show that the adjacency

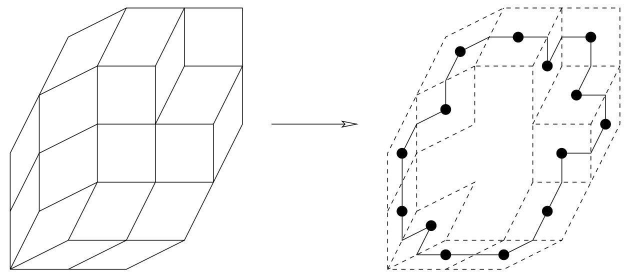

FIG. 7 – Left : a tiling. Right : its border and the border of its adjacency graph. As one may guess from this drawing, we will see below (Theorem 3) that the border of the tiling is nothing but the border of its adjacency graph.

graph of T always has a border (Theorem 2 below), and then we will show that the border of T is nothing but its adjacency graph (Theorem 3). We will finally give an algorithm to compute the border of this graph (Theorem 4), which completes the first step of the construct of T from its graph with origins.

Notation : Let $F = (f_1, ..., f_k)$ be a fan of a $2D$-gon tiling T. For all i, since f_i and f_{i+1} have exactly one side in common, there is a de Bruijn line which contains both f_i and f_{i+1}, a de Bruijn line which contains f_i and not f_{i+1}, and one de Bruijn line which contains f_{i+1} and not f_i. We denote by l_i the line which contains f_i and not f_{i+1}, and by $\mathcal{L}(F)$ the set of the l_i lines for all i.

Lemma 1 *Let T be a tiling of a $2D$-gon, and t_1, t_2 be two tiles of T having exactly one point in common and having one side included in a given side of the $2D$-gon. Let $A = (V, E)$ be the adjacency graph of T and π be the mapping of the tiles to the vertices. The unique shortest path between $\pi(t_1)$ and $\pi(t_2)$ in the adjacency graph of T is $\pi(t_1) = \pi(f_1), \pi(f_2), ..., \pi(f_k) = \pi(t_2)$, where $F = (f_1, ..., f_k)$ is a fan of T. See Figure 8.*

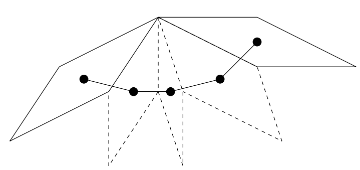

FIG. 8 – A fan and the unique shortest path between its extremities (Lemma 1).

Proof : Going from t_1 to t_2, one clearly has to cross each of the lines of $\mathcal{L}(F)$, and so the path described in the claim, which is of length exactly $k-1$, is a shortest path.

Suppose now this shortest path is not unique, and let p' be another shortest path between t_1 and t_2. This path contains at least one tile which is not a tile of F. Let t be the first tile in p' such that all the tile before t in p' are in F, and let f_j be the tile just before t in p'. Then t is in the de Bruijn line l_j. The part $f_1, f_2, ..., f_j, t$ crosses j de Bruijn lines and contains $j+1$ tiles. Since it still has to cross all the de Bruijn lines $l_{j+1}, ..., l_k$, it must have length at least k. Therefore p' is not a shortest path. $\square$

Lemma 2 *Let $A = (V, E)$ be the adjacency graph of a tiling T of a $2D$-gon P and π be the mapping of the tiles to the vertices. Let t be a tile of T on a side of P, and let $t' \neq t$ be a tile of T on a side of P such that $dist(\pi(t), \pi(t'))$ is minimal in A. Then, there is a fan $F = (f_1, ..., f_k)$ in T with $f_1 = t$ and $f_k = t'$.*

Proof : Let us consider a shortest path from $\pi(t)$ to $\pi(t')$, and suppose there is not a fan between t and t' in T. This implies that there is a tile t'' in T such that t'' is on the hull of P and there is a fan between t and t' in T. Consider the lines in $\mathcal{L}(F)$. As in Lemma 1, the path from t to t' must cross each of these lines, and so it is at least as long as shortest path from t to t''. Actually, it is longer since there is also a fan between t' and a tile t''' which is between t and t' (see Figure 9). Therefore, we reach a contradiction which proves that there must be a fan between t and t'. □

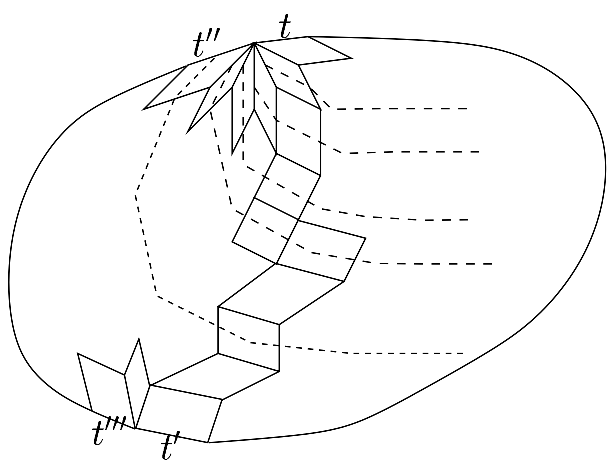

FIG. 9 – The de Bruijn line associated to t, and the path from t to t' crossing each line issued from the fan.

Lemma 3 *Let $A = (V, E)$ be the adjacency graph of a tiling T of a $2D$-gon P and π be the mapping of the tiles to the vertices. Let $v = \pi(t)$ be a vertex, and $v' = \pi(t')$ be the closest vertex to v such that $v, v' \in \mathcal{B}(A)$. Let $v'' = \pi(t'') \in \mathcal{B}(A) \backslash \{v'\}$ be the closest vertex to v such that $dist(v, v'') \leq dist(v', v'')$. Then, there is a fan $F = (f_1, ..., f_k)$ in T with $f_1 = t$ and $f_k = t''$.*

Proof : There is at least a de Bruijn line starting from t, which divides the tiling into two parts. Two cases may occur :

- t' and t'' are not in the same part. Then we can prove the claim with the same argument as in the proof of Lemma 2.
- t' and t'' are in the same part. From Lemma 2 there is a fan F' from t to t', and the path from t'' to t has to cross all the de Bruijn lines in $\mathcal{L}(F')$, and so it is at least as long as the shortest path from t to t', which contradicts the hypotheses.

□

Theorem 2 *Let T be a tiling of a $2D$-gon $P = (V, M)$ with $M \neq \{1, 1, n\}$. The adjacency graph of T has a border.*

Proof : Let A be the adjacency graph of T. Suppose there is no border, and let c be the vertex which appears twice in $\mathcal{B}(A)$, and t_c such that $c = \pi(t_c)$ (see Figure 10). There exist four tiles $\{t_i\}_{i\leq 4}$ of the border which are closest to t_c in terms of the path length. Then there is no border tile between t_1 and t_2, and respectively between t_3 and t_4, and the tiles have respectively one corner in common. Three cases occur :

1. t_c has two opposite corners in $H(P)$. It is exactly the exception of the theorem, and there is no border avoiding t_c.
2. t_c has one corner in $H(P)$. According to Lemma 2, t_c is either on the shortest path between t_1 and t_2, or between t_3 and t_4. Suppose it is between t_1 and t_2. Then there is a path between t_3 and t_4 that avoids t_c, and from Lemma 2 it is shorter than the path including t_c.
3. t_c has no corner in $H(P)$, and then, from Lemma 2, we see that there are two paths between t_1 and t_2 and respectively t_3 and t_4 that are shorter than the current ones and avoid t_c. □

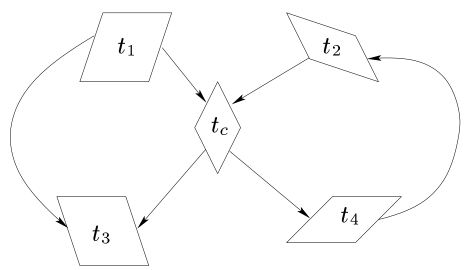

FIG. 10 – A cycle crossing on t_c.

Theorem 3 *The border of a tiling and the border of its adjacency graph are isomorphic.*

Proof : Let A be the adjacency graph of a tiling T, π be the mapping of the tiles to the vertices, and let t and t' be two tiles of $B(T)$. If t and t' are adjacent (in $B(T)$), then t and t' are adjacent in T, and there is no set of tiles $t_1, t_2, ..., t_k$ of adjacent tiles in $B(T)$ such that $t, t_1, .., t_k, t'$ is a path in $B(T)$. Then t and t' are adjacent in T, and $\pi(t)$ and $\pi(t')$ are adjacent in A, and then in $\mathcal{B}(A)$, because they both have degree at most 3.

Suppose now that two vertices v and v' are adjacent in $\mathcal{B}(A)$. Then they are in the shortest cycle containing all the vertices with degree at most 3, and are adjacent in A. This means that there is no path $v_1, v_2, ..., v_k$ in A composed of vertices of degree at most 3 leading from v to v'. Then $\pi^{-1}(v)$ and $\pi^{-1}(v')$ are adjacent in T, and there is no path between them composed of tiles on the hull of T. Then these two tiles are adjacent in $B(T)$. Therefore the adjacency relations in $\mathcal{B}(A)$ and $B(T)$ are the same, and we only have to show that $\pi(\mathcal{V}) = V$ (i.e that the set of tiles of $\mathcal{B}(A)$ is the projection of the set of tiles of $B(T)$) :

1. Let $\alpha \in V$ with $\pi^{-1}(\alpha) \notin \mathcal{V}$. Then, α has degree 4, it has no corner on the hull of the $2D$-gon, and is not on the shortest path between two adjacent tiles in the hull of the $2D$-gon, which is in contradiction with Lemma 2. Therefore $V \subseteq \pi(\mathcal{V})$
2. Let $\beta \in \mathcal{V}$ with $\pi(\beta) \notin V$. Then, β is a tile which has one corner on the hull of the $2D$-gon and $\pi(\beta)$ is not on the shortest path between two vertices associated with two

adjacent tiles in the hull of the $2D$-gon. Again, this is in contradiction with Lemma 2. Therefore $\pi(\mathcal{V}) \subseteq V$. $\square$

We have now all the preliminary results necessary to write an algorithm which constructs the border of the adjacency graph of a $2D$-gon tiling (Algorithm 2), which, from Theorem 3, is equivalent to construct the border of the tiling.

Theorem 4 *Given the adjacency graph of a $2D$-gon tiling, Algorithm 2 computes its border in time $O(n)$ where n is the number of vertices of the graph.*

Proof : Each vertex with degree less than 4 is in $\mathcal{B}(G)$ and it has two neighbors in $\mathcal{B}(G)$. Therefore, each vertex v must be associated two vertices, namely the two closest ones v' and v'' for which $dist(v, v') = dist(v', v)$ and $dist(v, v'') = dist(v'', v)$, and such that v is between them in the border, i.e. $dist(v, v'') > dist(v', v'')$ or $dist(v', v'') > dist(v, v'')$. Therefore, Algorithm 2 is a greedy algorithm which :

- starts with the origin v_1 of the graph and computes its closest neighbor
- for the current vertex v (which has a previously computed neighbor v'), computes its closest neighbor v'' such that $dist(v, v'') > dist(v', v'')$. From Lemma 3, this vertex is the next one in the border.

The basic complexity of this greedy algorithm is $O(n \cdot m)$, where n is the number of vertices, because it is just visiting all the vertices with degree at most 3, and builds a breadth first search in the rest of the graph, where every vertex has degree at most 4. then, starting with one origin, it only has to follow the distance 1 vertices until there is none. Then it has to find the closest vertex for which the distance in the two directions is the same, and go on. This can also be done in time $O(m \cdot n)$. $\square$

We will now show that, when one knows the border of the graph with origins of a $2D$-gon tiling T, then one can construct the de Bruijn lines of T by computing shortest paths in the graph. Indeed, the two following lemmas show that there is an equivalence between unique shortest paths in an adjacency graph and de Bruijn lines in the corresponding tiling : firstly, there is a unique shortest path between the two extremities of a de Bruijn line (Lemma 4), and moreover if two tiles are extremities of different de Bruijn lines then there is not a unique shortest path between them.

Lemma 4 *Let A be the adjacency graph of a tiling of a $2D$-gon, and π the mapping of the tiles to the vertices. Let $v = \pi(t)$ and $v' = \pi(t')$ be two vertices of A such that t and t' are the extremities of a given de Bruijn line L. There exists a unique shortest path in A between v and v' : $v = \pi(t) = \pi(t_1), \pi(t_2), ..., \pi(t_k) = \pi(t') = v'$ with $L = \{t_1, t_2, ..., t_k\}$.*

Proof : The line L crosses each de Bruijn line of the tiling which does not belong to its de Bruijn family. Every de Bruijn line separates the graph into two parts, and therefore each de Bruijn line crossed by L divides the graph in two parts $\mathcal{P}_a$ and $\mathcal{P}_b$, with $a \in \mathcal{P}_a$ and $b \in \mathcal{P}_b$. Therefore, every path from a to b has to cross each of the lines crossed by L, and therefore it must be at least as long as L.

Suppose now the shortest path between a and b is not unique, and let p be another shortest path between a and b. p and L have a common prefix (maybe reduced to a). Let c be the last vertex which is in p and L, and let d be the first vertex of p which is not in L. d does not belong to L, and therefore it belongs to the same line as c, say l_c. Therefore, if l_n is the de

Bruijn line crossed by L just after l_c, then d is on the same side of l_n as c. All the lines between c and b have to be crossed both by p and L, but at this point, L has crossed l_c, whereas p has not, and p must be at least one vertex longer than L. $\square$

Lemma 5 *Let T be a tiling of a $2D$-gon, t and t' be two tiles of T, with t on the side i and t' on the side $m+i$ of the $2D$-gon (where m is the sum of the multiplicities). Let $A(T)$ be its adjacency graph, and π the mapping of the tiles to the vertices. If t and t' are the extremities of two different de Bruijn lines, then there are at least two shortest paths between $\pi(t)$ and $\pi(t')$ in $A(T)$. See Figure 11.*

Proof : t and t' being on the sides i and $m+i$ of the $2D$-gon, they are in the i-th de Bruijn family. Let l_t be the line of this family containing t, and $l_{t'}$ be the line containing t'. Let now l_j be the line crossing l_t at the tile t, and l_k be the line crossing $l_{t'}$ at the tile t'. Let now γ be the tile on which l_t and l_k cross, and δ be the tile on which l_β and l_j cross. Call now l_1 the path starting at t, following l_t until the tile γ, and then following l_k until the tile t', and call l_2 the path starting at t which follows l_j until δ, and then follows $l_{t'}$ until the tile t'. Let LdB be the set of de Bruijn lines crossed by l_1. Each element in LdB is crossed by l_1 either between t and γ, or between γ and t'. Each of these two parts of l_1 is a piece os de Bruijn line, therefore each element of LdB cuts it exactly once. Moreover, suppose $lb \in LdB$ crosses l_j between t and γ. Then lb crosses $l_{t'}$ between γ and the tile adjacent to t'. The line $l_{t'}$ divides the tiling into two parts, each of which contains one of the extremities of $l_{t'}$. Therefore lb cannot cross $l_{t'}$ between γ and t'. Symmetrically, l_2 has the same properties. Each of them crosses exactly once all the lines of LdB, therefore their length is minimal, and there are two paths with minimal length between t and t'. See Figure 11. $\square$

Since we now have all the material needed to compute the border of the adjacency graph of a tiling, and since from this we can obtain the de Bruijn lines of the tiling by shortest paths computations, we can give an algorithm to compute the de Bruijn graph of a tiling from its adjacency graph :

Algorithm 2: Construction of the border of the adjacency graph of a $2D$-gon tiling.

Input: $G = (V, E, v_1, v_2)$ the graph with origins of a $2D$-gon tiling.

Output: The border of G, $\mathcal{B}(G)$, as an ordered list of vertices.

begin

Let B be the set of vertices in V of degree< 4;
Let v' be an element of B such that $dist(v_1, v')$ is minimal;
$\mathcal{B} \leftarrow$ shortest path from v_1 to v';
repeat
  $v \leftarrow$ last element of B;
  $\nu \leftarrow$ element just before v in B;
  Let B' be a copy of B;
  repeat
    Let v' be an element of B' such that $dist(v, v')$ is minimal;
    Remove v' from B';
  until $dist(v, v') \leq dist(\nu, v')$;
  Add the shortest path from v to v' at the end of $\mathcal{B}$;
until $v' = v_1$;
if *the second element of $\mathcal{B}$ with degree < 4 is not v_2* **then**
  Reverse $\mathcal{B}$;

Return($\mathcal{B}$);

end

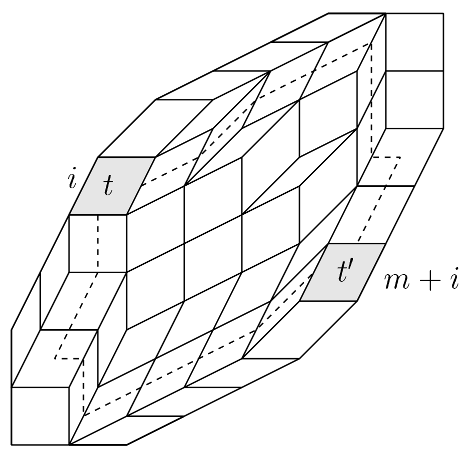

FIG. 11 – As announced in Lemma 5, there are two shortest paths between t and t'.

Algorithm 3: Construction of the de Bruijn graph of a $2D$-gon tiling from its graph with origins.

```
Input: G = (V, E, v_1, v_2) the graph with origins of a 2D-gon tiling T and B(G) its
       border.
Output: G' = (V, E, λ, ν), the de Bruijn graph of T.
begin
    i ← 1;
    foreach x ∈ B(G) do
        if d^o(x) < 4 then
            B[i] ← x, i ← i + 1, N ← N + 1;
            if d^o(x) = 2 then
                B[i] ← x, i ← i + 1, N ← N + 1;
    l ← 1;
    for α = 1 to N/2 do
        β ← α + N/2;
        build all the shortest paths between B[α] and B[β];
        if there is one path then
            foreach vertex v of the path do
                λ(v) ← λ(v) ∪ {l};
            l ← l + 1;
        else
            β ← β + 1;
    Return (V, E, λ, v_1);
end
```

Theorem 5 *Given the graph with origins of a $2D$-gon tiling T and its border, Algorithm 3 computes the de Bruijn graph of T in time $O(n \cdot m)$ where n is the number of vertices of the graph and m is the sum of all the multiplicities which define the $2D$-gon.*

Proof : The first part of the algorithm constructs a table B which contains the vertices of the border having at least one side in $H(P)$, and the vertices with degree 2 appearing twice. We then have a walk on the hull, presented side by side with the associated tiles. It is then easy to find the indexes of the tiles that are extremities of a same de Bruijn line.

Theorem 4 gives a practical way of rebuilding de Bruijn lines. The algorithm has complexity $O(n \cdot m)$, because it is building shortest paths (all of them being at most m tiles long) between two vertices in a graph with n vertices.

The part corresponding to more than one path between t and t' corresponds to the multiplicities more than 1. We refer again to Lemma 5. In the rebuilding of a de Bruijn line, we examine the border two vertices at a time. Two cases occur :

1. if these two vertices are extremities of the same de Bruijn line, then there is one unique shortest path which is exactly the de Bruijn line.
2. if these two vertices are not extremities of the same de Bruijn line, they are extremities of two different de Bruijn lines of the same family. Then there are at least two shortest

paths. Then we only have to keep one vertex and go on with another at the other extremity (namely the next in the border). As long as there are two paths, we go through the lines of the de Bruijn family. When at last we find only one shortest path, we have found the other extremity, and then we have the number of lines of the de Bruijn family, and we can build all the family, going backwards in the list of visited vertices. □

Starting from the graph with origins of a $2D$-gon tiling T, it is now clear that one can construct a tiling equivalent to T by computing the border of the graph with Algorithm 2, then compute the de Bruijn graph with Algorithm 3, and finally obtain the tiling using Algorithm 1. Therefore, we can finally combine Theorems 1, 2 and 5 to obtain :

Theorem 6 *Given the graph with origins of a $2D$-gon tiling T, there is an algorithm which constructs a $2D$-gon tiling equivalent to T in time $O(m \cdot n)$, where n is its number of tiles, and m the sum of the multiplicities of the vectors used to define the $2D$-gon.*

This result not only gives an efficient and simple way to encode and manipulate 2D-gon tilings ; it also clarifies the relation that exists between the adjacency graph of a tiling and the tiling itself. In particular, it proves that, despite the fact that there is no one-to-one correspondence between adjacency graphs and tilings, the adjacency graph contains almost all the information on the tiling.

4 An application : random tilings

Tilings of $2D$-gons are an important model of quasicrystals in physics. In this context, it is very important to be able to sample random tilings, which helps the study of the entropy of the quasicrystal [BDMW02]. The sampling uses the key notion of *flip* : given a $2D$-gon tiling, one may rearrange locally three tiles (which form an hexagon) in order to obtain a new tiling of the same $2D$-gon (see Figure 12).

This enables the random generation of tilings of a $2D$-gon : it is shown in [Eln97, Ken93] that one can obtain all the tilings of a $2D$-gon from a given one by iterating the flip operation. When one wants to obtain a random tiling, one then has to choose a particular tiling and then iterate the flip operation *until the obtained tiling can be considered as random*. This notion of *when* one can stop the process is central when one wants to sample random tilings *with the uniform distribution*. It is possible to sample perfectly random tilings of hexagons because of the distributive lattice structure of the set of all the tilings [Pro98]. This technique can no longer be used for octagon, but a recent study explains how long the process has to be continued in order to be as close as one may want of the uniform distribution [Des01]. For the other $2D$-gons, *i.e.* when $D > 4$, there are no known results [BDMW02].

Therefore, when one wants to sample a random tiling of a given $2D$-gon P, the only solution is to construct a particular tiling of P and iterate the flip operation. To acheive this, one can use the graphs encodings we proposed above : the flip operation can be encoded on the graph, as shown in Figure 13. The vertices which correspond to the tiles to flip form a triangle in the graph, and conversly, all the triangles in the graph correspond to a possible flip in the tiling. Moreover, the transformation on the graph is a local rearrangement of vertices. This makes it possible to implement the flip operation very efficiently and so to iterate it a very high number of times. We show in Figure 14 a random tilings of a decagon obtained this way.

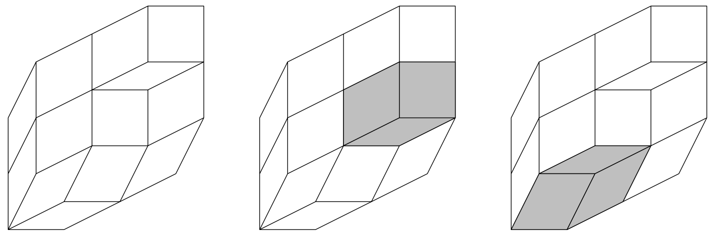

FIG. 12 – A tiling of an octagon ($D = 4$) (left) and two other tilings of the same octagon obtained from the first one by a flip (the shaded tiles are the ones which moved during the flip).

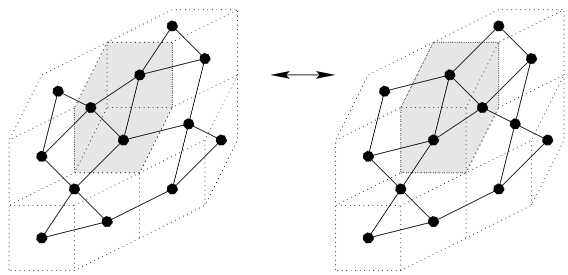

FIG. 13 – A flip on a $2D$-gon, and on its graph.

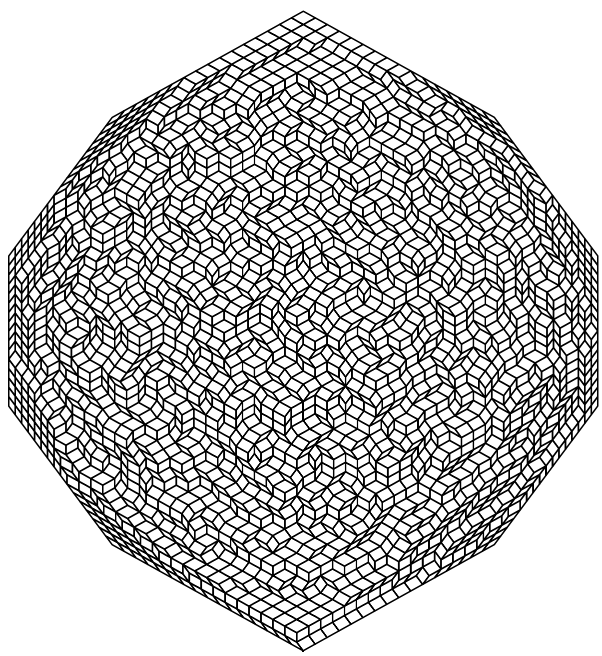

FIG. 14 – a random tiling of a $5 \to 2$ tiling with side size 16, obtained after 100 million flips.

5 Perspectives

The algorithmic study of tilings of $2D$-gons is only at its beginning, and many open problems still exist. We cited the problem of knowing how many flips have to be done in order to obtain random tilings with a distribution *close* to the uniform distribution. Another important area is the generation of all the tilings of a $2D$-gon, and their enumeration. The encodings with graphs may be used to study these problems. For example, one may obtain a characterization of which graphs are the graphs associated to a $2D$-gon tiling : these graphs are planar, the degree of each vertex is at most four, and they may have many other properties which could help in generating and counting them.

Moreover, $2D$-gons are a special class (the dimension 2 case) of a very important class of objects, namely *zonotopes* [Zie95]. These objects can be viewed as generalizations of $2D$-gons in higher dimensions, and they play an important role in combinatorics and physics. They are also strongly related to oriented matroid theory [RGZ94]. Many studies already deal with these objects, but their algorithmical manipulation is still a problem, while it would help a lot in verifying conjectures, compute special tilings, and compute some statistics over them. The results presented here may be extended to this more general case, leading to other classes of graphs with interesting properties. Notice however that this generalization is not obvious, since our proofs deeply use properties related to the dimension 2. It is well known in zonotopes theory that there is a gap of complexity between 2-dimensional zonotopes ($2D$-gons) and 3-dimensional ones [Zie95].

Références

[Bai99] Guy David Bailey. Coherence and enumeration of tilings of 3-zonotopes. *Discrete & Computational Geometry*, 22 :119–147, 1999.

[BDMW02] F. Bailly, N. Destainville, R. Mosseri, and M. Widom. Arctic octahedron in three-dimensional rhombus tilings and related integer solid partitions. 2002.

[Ber66] R. Berger. The undecidability of the domino problem. *Memoirs of the American Mathematical Society*, 66, 1966.

[BFF00] Eric Babson, Lukas Finschi, and Komei Fukuda. Cocircuit graphs and efficient orientation reconstruction in oriented matroids. to appear in European Journal of Combinatorics, 2000.

[BVS+99] A. Björner, M. Las Vergnas, B. Sturmfels, N. White, and G. M. Ziegler. *Oriented Matroids*, volume 46 of *Encyclopedia of Mathematics*. Cambridge University Press, 1999. Second Edition.

[CFG00] R. Codovil, K. Fukuda, and A. Guedes de Oliveira. On the cocircuit-graph of an oriented matroid. *Discrete and Computational Geometry*, 24 :257–265, 2000.

[dB81] N.G. de Bruijn. Dualization of multigrids. *J. Phys. France*, C(47) :3–9, 1981.

[Des97] Nicolas Destainville. *Entropie configurationnelle des pavages aléatoires et des membranes dirigées*. PhD thesis, University Paris VI, 1997.

[Des01] Nicolas Destainville. Mixing times of plane random rhombus tilings. In *Proceedings of the Conference DM-CCG : Discrete Models : Combinatorics, Computation, and Geometry*, pages 1–22, 2001.

[Eln97] Serge Elnitsky. Rhombic tilings of polygons and classes of reduced words in Coxeter groups. *Journal of Combinatorial Theory*, 77 :193–221, 1997.

[Ken93] Richard Kenyon. Tiling a polygon with parallelograms. *Algorithmica*, 9 :382–397, 1993.

[Lat00] Matthieu Latapy. Generalized integer partitions, tilings of zonotopes and lattices. In A.A. Mikhalev D. Krob and E.V. Mikhalev, editors, *Proceedings of the 12-th international conference Formal Power Series and Algebraic Combinatorics (FPSAC'00)*, pages 256–267. Springer, June 2000. Preprint available at `http://www.liafa.jussieu.fr/~latapy/`.

[Pro98] James Propp. Generating random elements of finite distributive lattices. *Electronic Journal of Combinatorics*, 4, 1998.

[RGZ94] Jürgen Richter-Gebert and Günter Ziegler. Zonotopal tilings and the Bohne-Dress theorem. *Contemporary Mathematics*, 178 :211–232, 1994.

[Sen95] Marjorie Senechal. *Quasicrystals and Geometry*. Cambridge University Press, 1995.

[Zie95] Günter Ziegler. *Lectures on Polytopes*. Graduate Texts in Mathematics. Springer-Verlag, 1995.