Approximation of Digital Surfaces by a Hierarchical Set of Planar Patches

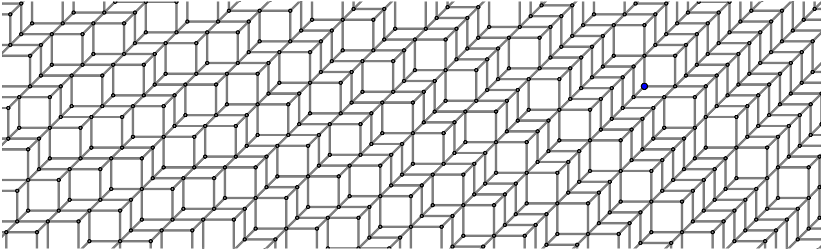
Jocelyn Meyron and Tristan Roussillon

Université de Lyon, INSA Lyon, LIRIS, France

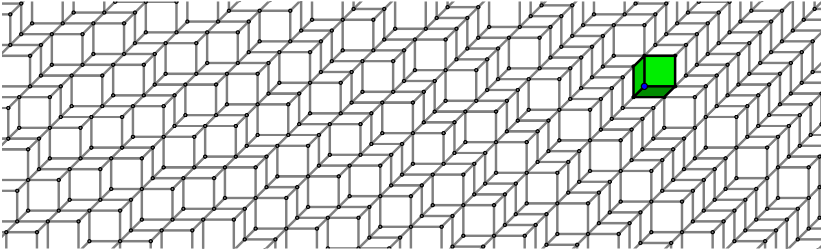
DGMM, 25-27/10/2022

PARADIS (ANR-18-CE23-0007-01) research grant

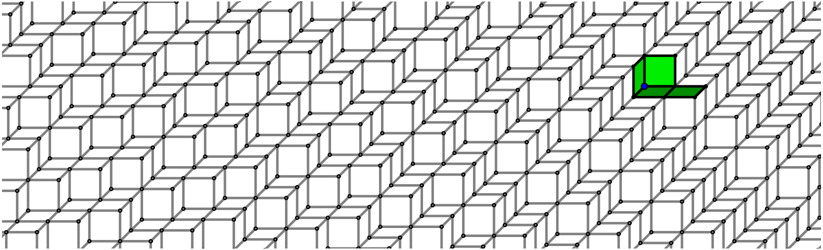
Overview



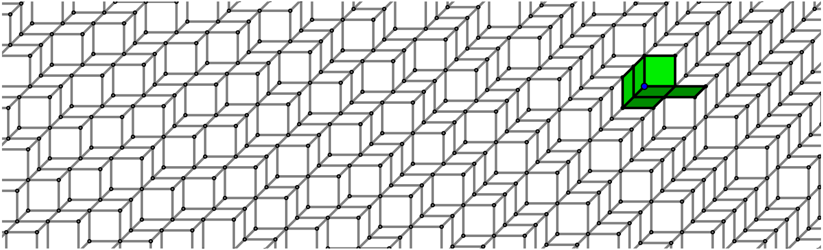
Overview



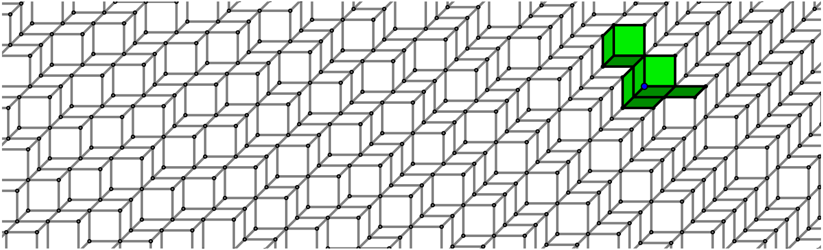
Overview



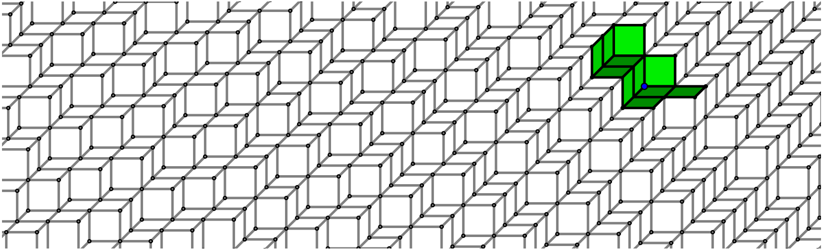
Overview



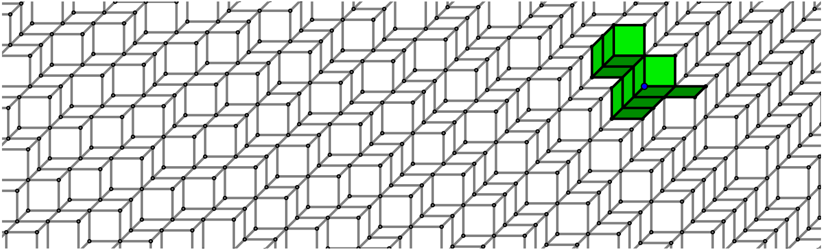
Overview



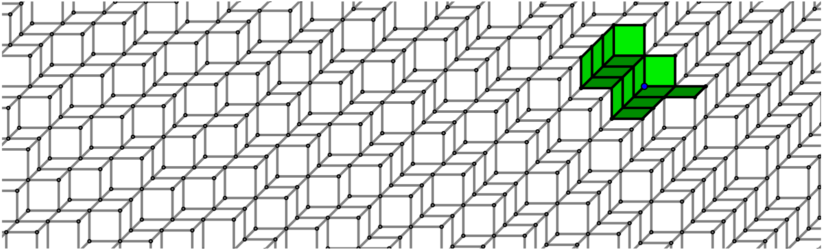
Overview



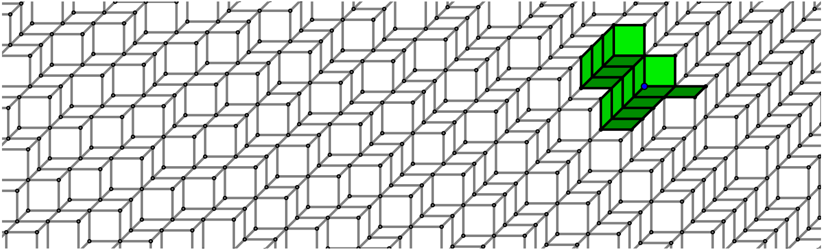
Overview



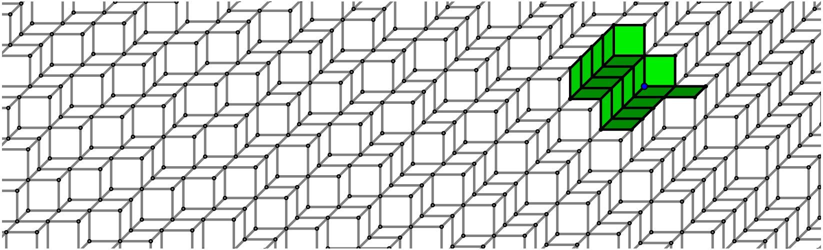
Overview



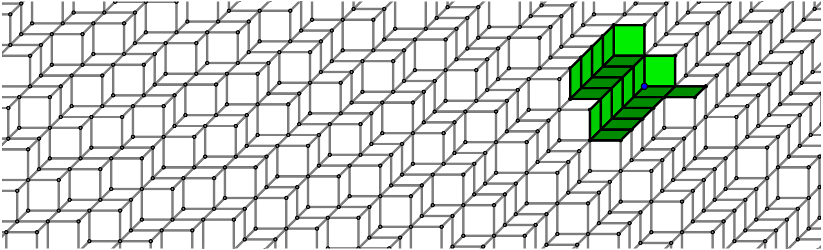
Overview



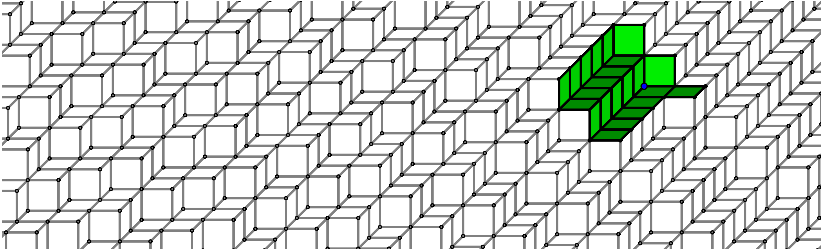
Overview



Overview



Overview



Outline

1. 3D Extension of the Euclidean Algorithm and Plane-Probing Algorithms

2. $E_1^*(\cdot)$, a Way of Generating Planar Patches

No unique 3D extension to the Euclidean algorithm

Assuming $1 \leq a \leq b \leq c$:

- *Brun*: $(a, b, c)^T \to (a, b, c - b)^T$;
- *Selmer*: $(a, b, c)^T \to (a, b, c - a)^T$;
- *Farey*: $(a, b, c)^T \to (a, b - a, c)^T$;
- *Fully-Subtractive*: $(a, b, c)^T \to (a, b - a, c - a)^T$;
- *Poincaré*: $(a, b, c)^T \to (a, b - a, c - b)^T$.
- ...

$\Rightarrow$ ambiguities or null coordinate in case of ties,
with, e.g., $\begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix}$, $\begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix}$.

A class of generalized Euclidean algorithms

Definition of a set of unimodular matrices

$$\mathcal{U} := \left\{ \begin{array}{ccc} \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix} \\ \\ \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} & \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix} \end{array} \right\}$$

Algorithm

Given $\mathbf{v} \in \mathbb{N}^3$, with coprime coordinates,

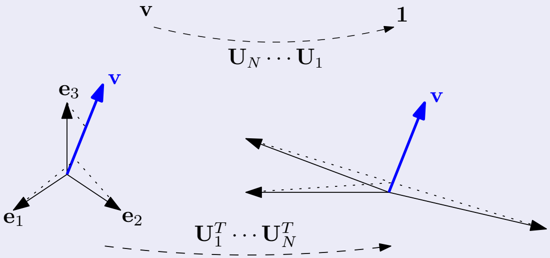
- while $\mathbf{v} \neq \mathbf{1}$,
- choose $\mathbf{U} \in \mathcal{U}$ such that $\mathbf{U}\mathbf{v} \in \mathbb{N}^3$ *(several possibilities)*
- and do $\mathbf{v} \leftarrow \mathbf{U}\mathbf{v}$.

A matrix sequence

Termination, complexity and output

- The number of steps is $N \leq \|\mathbf{v}\|_1 - 3$,
- The matrices used as numbered from $\mathbf{U}_1$ to $\mathbf{U}_N$.

Geometrical interpretation

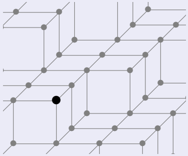


Using a set-membership predicate instead of $\mathbf{v}$

Digital plane

Given $\mathbf{v} \in \mathbb{N}^3$, with coprime coordinates,
and an offset $\mu \in \mathbb{Z}$,

$$\mathbf{P}_{\mathbf{v},\mu} := \{\mathbf{x} \in \mathbb{Z}^3 \mid \mu \leq \mathbf{v}^T\mathbf{x} < \mu + \|\mathbf{v}\|_1\}.$$



Interesting fact

$$\begin{aligned}\mathbf{U}_n \cdots \mathbf{U}_1\mathbf{v} \in \mathbb{N}^3 &\Leftrightarrow \forall i,\ \mathbf{e}_i^T(\mathbf{U}_n \cdots \mathbf{U}_1\mathbf{v}) \geq 1\\ &\Leftrightarrow \forall i,\ \mathbf{v}^T(\mathbf{U}_1^T \cdots \mathbf{U}_n^T\mathbf{e}_i) \geq 1\\ &\Leftrightarrow \forall i,\ \mathbf{U}_1^T \cdots \mathbf{U}_n^T\mathbf{e}_i \notin \mathbf{P}_{\mathbf{v},-\|\mathbf{v}\|_1+1}\end{aligned}$$

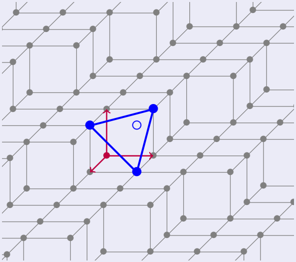
Plane-probing Algorithms

Definition

Given a digital plane **P** and a starting point $\mathbf{p} \in \mathbf{P}$, a plane-probing algorithm computes the normal vector of **P** by sparsely probing it with the predicate "is $\mathbf{x} \in \mathbf{P}$?".

State of the art

- Upward-oriented tetrahedron. No guarantee that it stays close to the starting point. [L.P.R.,2016]
- Downward-oriented tetrahedron with a fixed apex (= triangle).
  - H, R [L.P.R.,2017], R^1 [R.L.,2019]
  - implemented in DGtal (dgtal.org)



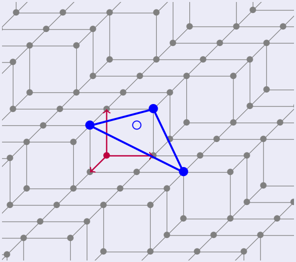
Plane-probing Algorithms

Definition

Given a digital plane **P** and a starting point $\mathbf{p} \in \mathbf{P}$, a plane-probing algorithm computes the normal vector of **P** by sparsely probing it with the predicate "is $\mathbf{x} \in \mathbf{P}$?".

State of the art

- Upward-oriented tetrahedron. No guarantee that it stays close to the starting point. [L.P.R.,2016]
- Downward-oriented tetrahedron with a fixed apex (= triangle).
  - H, R [L.P.R.,2017], R^1 [R.L.,2019]
  - implemented in DGtal (dgtal.org)



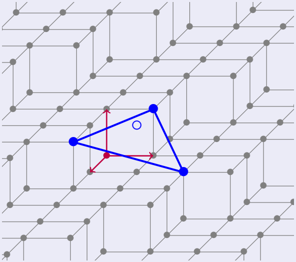
Plane-probing Algorithms

Definition

Given a digital plane **P** and a starting point $\mathbf{p} \in \mathbf{P}$, a plane-probing algorithm computes the normal vector of **P** by sparsely probing it with the predicate "is $\mathbf{x} \in \mathbf{P}$?".

State of the art

- Upward-oriented tetrahedron. No guarantee that it stays close to the starting point. [L.P.R.,2016]
- Downward-oriented tetrahedron with a fixed apex (= triangle).
  - H, R [L.P.R.,2017], R^1 [R.L.,2019]
  - implemented in DGtal (dgtal.org)



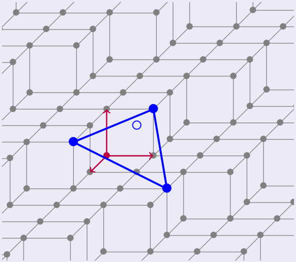
Plane-probing Algorithms

Definition

Given a digital plane $\mathbf{P}$ and a starting point $\mathbf{p} \in \mathbf{P}$, a plane-probing algorithm computes the normal vector of $\mathbf{P}$ by sparsely probing it with the predicate "is $\mathbf{x} \in \mathbf{P}$?".

State of the art

- Upward-oriented tetrahedron. No guarantee that it stays close to the starting point. [L.P.R.,2016]
- Downward-oriented tetrahedron with a fixed apex (= triangle).
  - H, R [L.P.R.,2017], R^1 [R.L.,2019]
  - implemented in DGtal (`dgtal.org`)



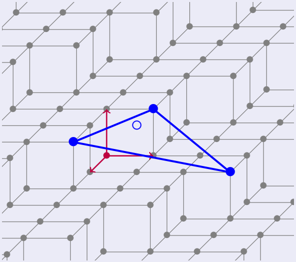
Plane-probing Algorithms

Definition

Given a digital plane **P** and a starting point $\mathbf{p} \in \mathbf{P}$, a plane-probing algorithm computes the normal vector of **P** by sparsely probing it with the predicate "is $\mathbf{x} \in \mathbf{P}$?".

State of the art

- Upward-oriented tetrahedron. No guarantee that it stays close to the starting point. [L.P.R.,2016]
- Downward-oriented tetrahedron with a fixed apex (= triangle).
  - H, R [L.P.R.,2017], R^1 [R.L.,2019]
  - implemented in DGtal (dgtal.org)



Main Features of Plane-Probing Algorithms

They are generalizations of the Euclidean algorithm...

- Output: a matrix sequence that *reduces* the normal $\mathbf{v}$ of $\mathbf{P}$, i.e.,

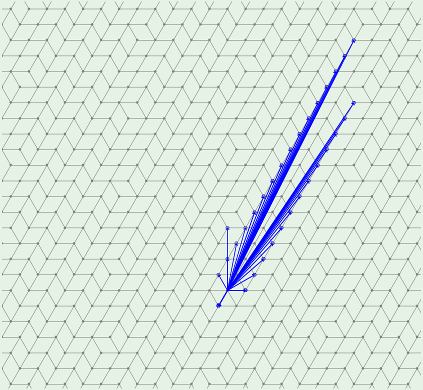
$$\mathbf{U}_1^{-1} \cdots \mathbf{U}_N^{-1} \mathbf{1} = \mathbf{v}.$$

...but with very specific features

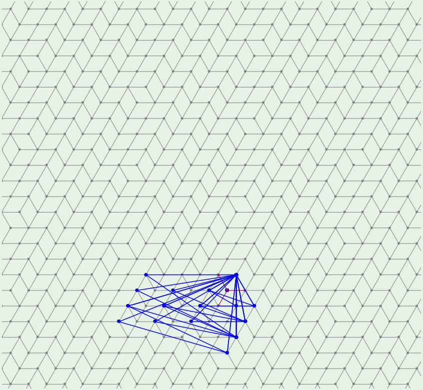
- Input: a predicate "is $\mathbf{x}$ in $\mathbf{P}$?"
- At every step $n \in \{1, \ldots, N\}$,
  - $\mathbf{U}_n$ is chosen from $\mathcal{U}$ (for H) or from a *larger set* (for R, R^1),
  - by a *geometrical criterion*, depending on $\mathbf{U}_1, \ldots, \mathbf{U}_{n-1}$ for $n \geq 2$.

$\Rightarrow$ the column vectors of $\mathbf{U}_1^T \cdots \mathbf{U}_N^T$ are usually shorter with plane-probing algorithms than with other generalizations.

Selmer vs H for $\mathbf{v} = \begin{pmatrix} 1 \\ 13 \\ 17 \end{pmatrix}$



(a) Selmer (dmax $\approx$ 15)



(b) H (dmax $\approx$ 5)

Generation of planar patches

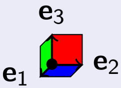
State of the art

- Based on a digitization model,
- Based on 3D generalizations of the Euclidean algorithm:
  - union and translation of point sets,
    e.g., [D.V.,2012], [J.L.P.,2016],
  - $E_1^*(\cdot)$: rules that replace square faces by sets of square faces,
    e.g., [A.I.,2001], [B.F.,2011].

Square faces and stepped planes

Three types of face ($\approx$ geometrical alphabet)

$\forall \mathbf{x} \in \mathbb{Z}^3$, $(\mathbf{x}, 1^*)$, $(\mathbf{x}, 2^*)$, $(\mathbf{x}, 3^*)$ are as follows:



Stepped plane

Given $\mathbf{v} \in \mathbb{N}^3$, with coprime coordinates,

$$\Pi_{\mathbf{v}} := \{(\mathbf{x}, i^*) \mid \mathbf{x} \in \mathbb{Z}^3, 0 \leq \mathbf{v}^T\mathbf{x} < \mathbf{v}^T\mathbf{e}_i\}.$$

The vertices of the faces of $\Pi_{\mathbf{v}}$ are the points of $\mathbf{P}_{\mathbf{v},0}$.

$E_1^*(\cdot)$

Definition

$$E_1^*(\sigma)(\mathbf{x}, i^*) := \bigcup_{i \cdot s \text{ is a suffix of } \sigma(j)} (\mathbf{M}_\sigma^{-1}(\mathbf{x} + l(s)), j^*)$$

It preserves stepped planes for any unimodular substitution

$$E_1^*(\sigma)(\Pi_{\mathbf{v}}) = \Pi_{\mathbf{M}_\sigma^T \mathbf{v}}.$$

NB:

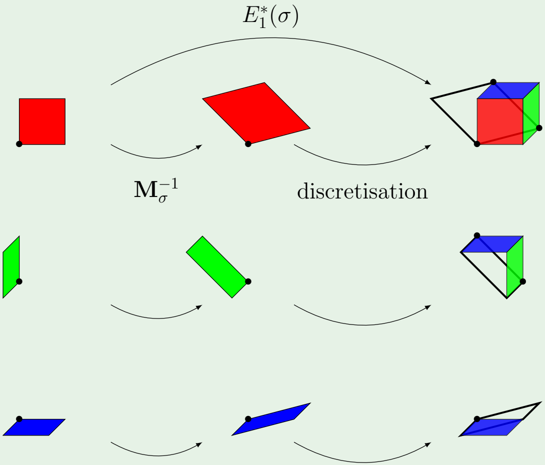
- σ is a *substitution* over $\{1, 2, 3\}$
- $l : \{1, 2, 3\}^* \to \mathbb{N}^3$ is the *Parikh mapping*
- $\mathbf{M}_\sigma$ is the *incidence matrix* of σ.

Geometrical illustration of $E_1^*(\sigma)$ with an example

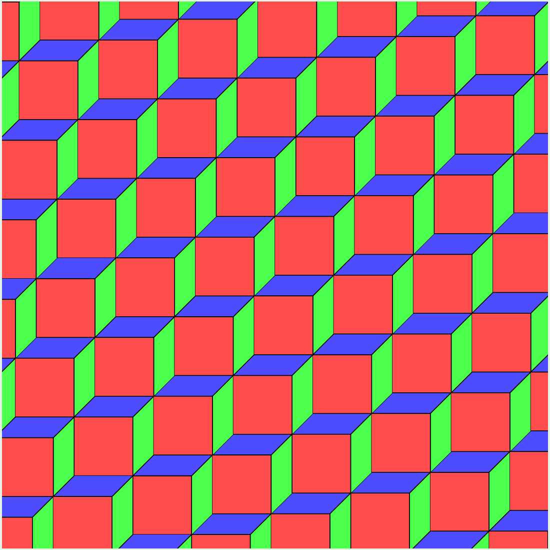
$$\sigma : \begin{cases} 1 & \mapsto 1 \\ 2 & \mapsto 12 \\ 3 & \mapsto 123 \end{cases}$$

$$\mathbf{M}_\sigma = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$

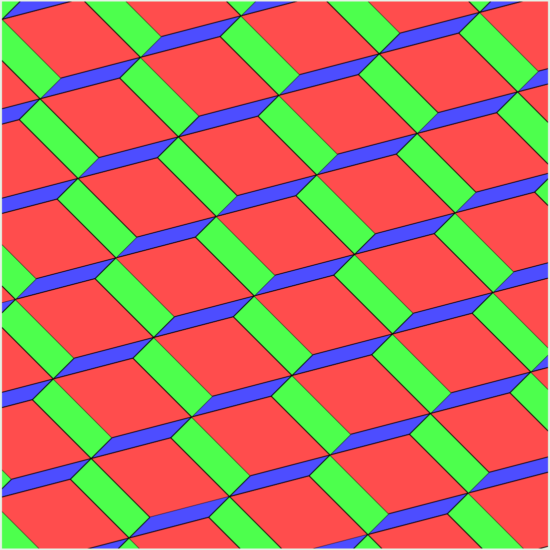
$$\mathbf{M}_\sigma^{-1} = \begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{pmatrix}$$



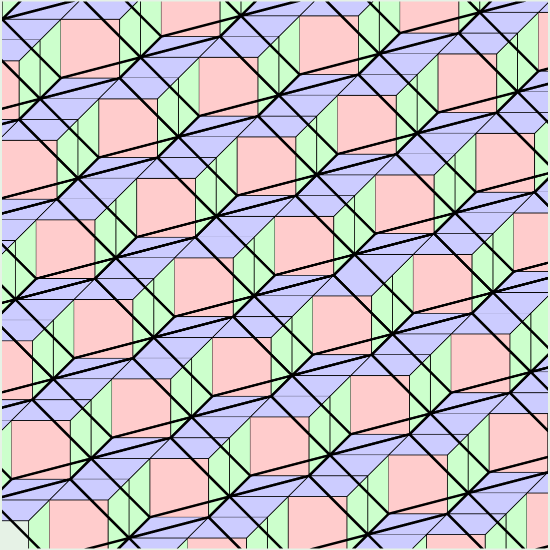
Geometrical illustration of $E_1^*(\sigma)$ with an example



Geometrical illustration of $E_1^*(\sigma)$ with an example



Geometrical illustration of $E_1^*(\sigma)$ with an example



Method overview

Input

- $\mathcal{W}_0$, the lower part of the unit cube: 
- a matrix sequence $(\mathbf{U}_n)_{1 \leq n \leq N}$ that reduces a normal vector $\mathbf{v}$

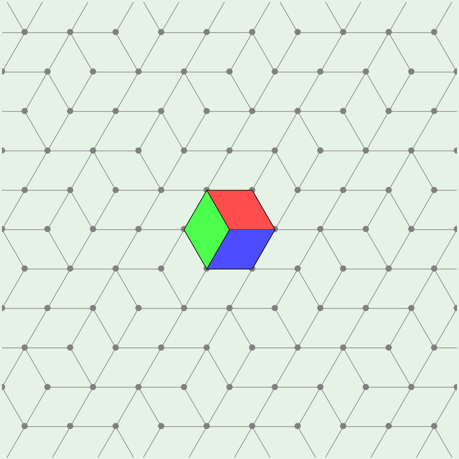
$\forall n \in \{1, \ldots, N\}$, choose a substitution σ_n such that $\mathbf{M}_{\sigma_n}^{-1} = \mathbf{U}_n^T$

- 2 possible choices for every matrix $\mathbf{U}_n^T \in \mathcal{U}$,
- one can use a convention ($\forall i \in \{1, 2, 3\}$, $\sigma_n(i)$ starts with i).

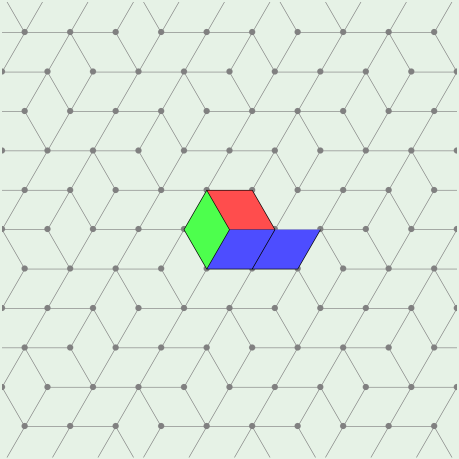
Output

Patterns $(\mathcal{W}_n)_{1 \leq n \leq N}$, images of $\mathcal{W}_0$ by $E_1^*(\sigma_n \circ \cdots \circ \sigma_1)$.

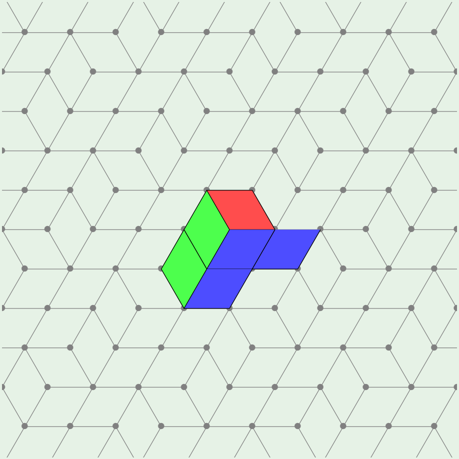
Incremental generation for $\mathbf{v} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}$



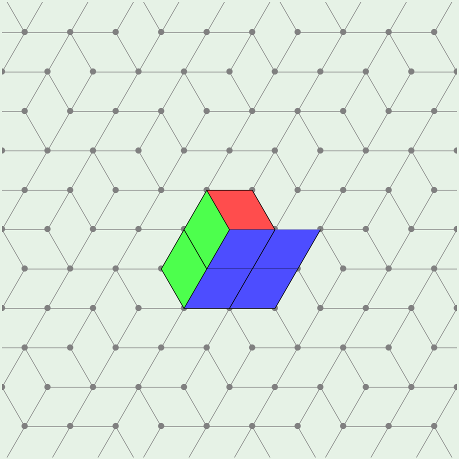
Incremental generation for $\mathbf{v} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}$



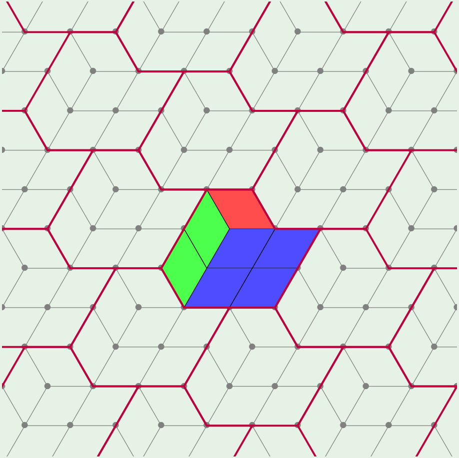
Incremental generation for $\mathbf{v} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}$



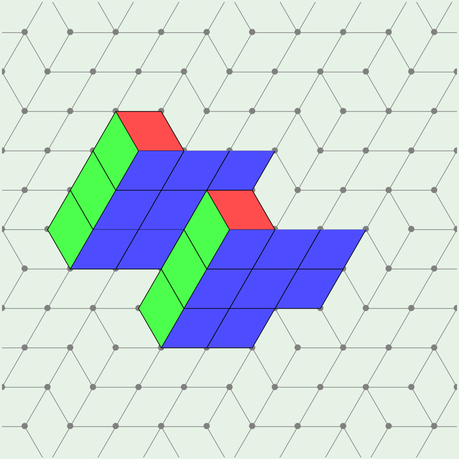
Incremental generation for $\mathbf{v} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}$



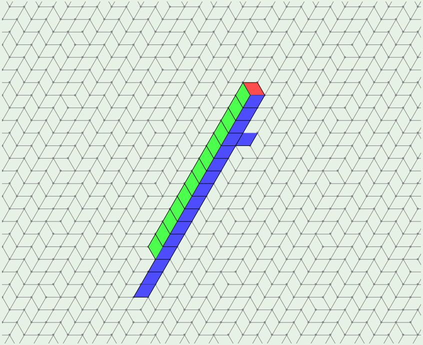
Incremental generation for $\mathbf{v} = \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix}$



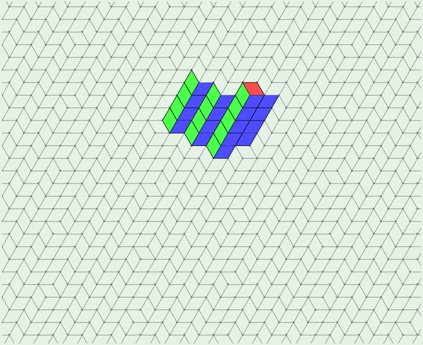
Another example, $\mathbf{v} = \begin{pmatrix} 2 \\ 6 \\ 15 \end{pmatrix}$



Selmer vs H for $\mathbf{v} = \begin{pmatrix} 1 \\ 13 \\ 17 \end{pmatrix}$

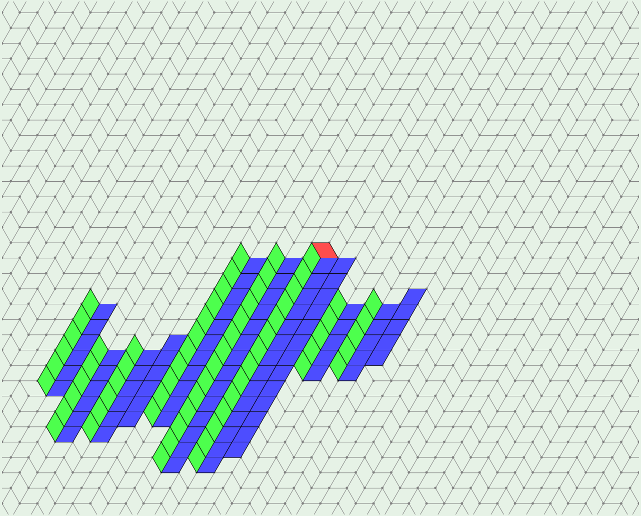


(c) Selmer (dmax $\approx$ 16)

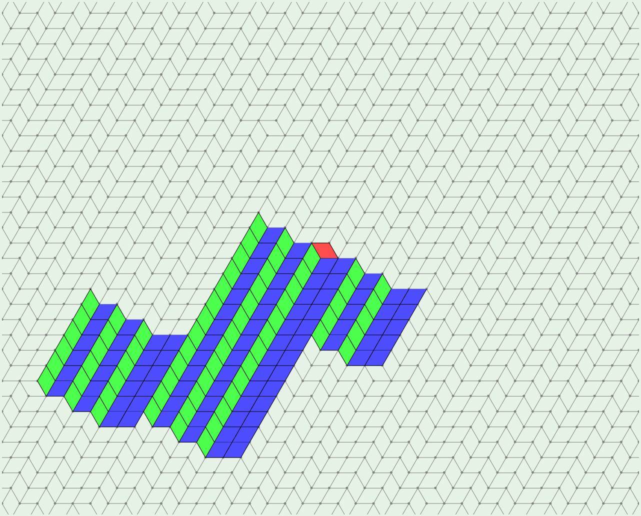


(d) H (dmax $\approx$ 6)

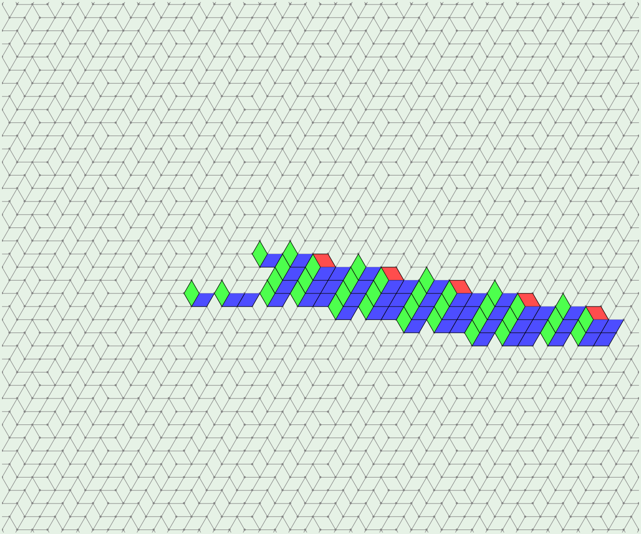
Substitions matter, $\mathbf{v} = \begin{pmatrix} 1 \\ 67 \\ 91 \end{pmatrix}$



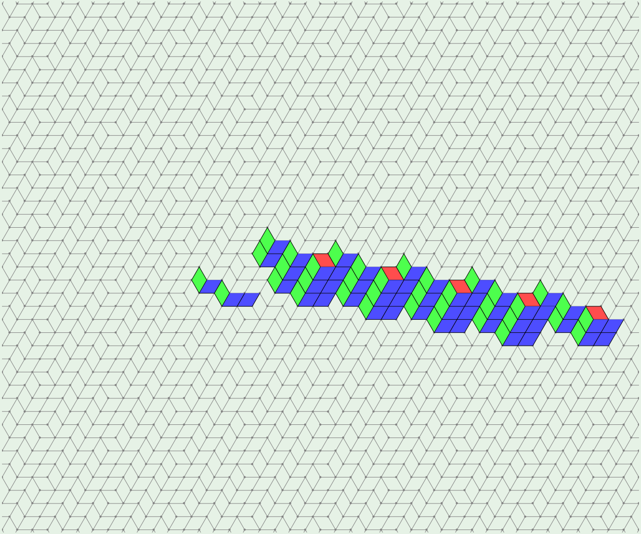
Substitions matter, $\mathbf{v} = \begin{pmatrix} 1 \\ 67 \\ 91 \end{pmatrix}$



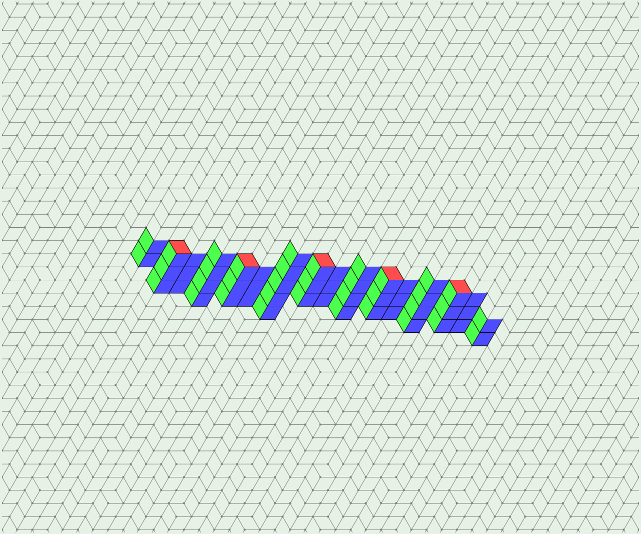
Topological issues, $\mathbf{v} = \begin{pmatrix} 5 \\ 36 \\ 51 \end{pmatrix}$



Topological issues, $\mathbf{v} = \begin{pmatrix} 5 \\ 36 \\ 51 \end{pmatrix}$



Topological issues, $\mathbf{v} = \begin{pmatrix} 5 \\ 36 \\ 51 \end{pmatrix}$



Main features of the generated patterns

Theoretical results

- inclusion: $\forall n < N, \mathcal{W}_n \subset \mathcal{W}_N \subset \Pi_{\mathbf{v}}$.
- periodicity: $\mathcal{W}_N$ periodically generates $\Pi_{\mathbf{v}}$ with:

$$\mathbf{M}_{\sigma_1}^{-1} \cdots \mathbf{M}_{\sigma_N}^{-1}(\mathbf{e}_2 - \mathbf{e}_1) \text{ and } \mathbf{M}_{\sigma_1}^{-1} \cdots \mathbf{M}_{\sigma_N}^{-1}(\mathbf{e}_3 - \mathbf{e}_2),$$

- minimality: $\mathcal{W}_N$ has $\mathbf{e}_i^T \mathbf{v}$ faces of type $i \in \{1, 2, 3\}$.

Experimental results

(+) compact and close to $\mathcal{W}_0$ when using a plane-probing algorithm,

(−) not necessarily connected...

Conclusion and Perspectives

Conclusion

- Link between a 3D generalization of the Euclidean algorithm and plane-probing algorithms
- (both produce a matrix sequence that reduces a given vector).
- Use E_1^* to produce a pattern from a matrix sequence:
  - theoretical properties: inclusion, periodicity, minimality,
  - observation: compact when using a plane-probing algorithm.

Perspectives

- short-term: measure how far from the initial faces the patterns are.
- middle-term: find a way of making them necessarily connected.
- long-term: use pattern generation for the analysis of digital surfaces.

References

Plane-probing

[L.P.R.,2016] J-O. Lachaud, X. Provençal, T. Roussillon, An output-sensitive algorithm to compute the normal vector of a digital plane. *Theor. Comput. Sci.*, 624:73–88, 2016.

[L.P.R.,2017] J-O. Lachaud, X. Provençal, T. Roussillon, Two Plane-Probing Algorithms for the Computation of the Normal Vector to a Digital Plane. *J. Math. Imaging Vis.*, 59(1):23–39, 2017.

[R.L.,2019] T. Roussillon, J-O. Lachaud, An efficient and quasi linear worst-case time algorithm for digital plane recognition, *DGCI'19*, LNCS, vol. 11414, p.380–393, 2019.

[L.M.R.,2020] J-O. Lachaud, J. Meyron, T. Roussillon, An Optimized Framework for Plane-Probing Algorithms, *J. Math. Imaging Vis.*, 62(5):718–736, 2020.

Pattern generation

[A.I.,2001] P. Arnoux, S. Ito, Pisot substitutions and Rauzy fractals. *Bull. Belg. Math. Soc. Simon Stevin*, 8(2):181–208,2001.

[B.F.,2011] V. Berthé, T. Fernique, Brun expansions of stepped surfaces, *Discret. Math.*, 311(7):521–543, 2011.

[D.V.,2012] E. Domenjoud, L. Vuillon, Geometric Palindromic Closure. *Unif. Distrib. Theory*, 7(2):109–140, 2012.

[J.L.P.,2016] D. Jamet, N. Lafrenière, X. Provençal, Generation of digital planes using generalized continued-fractions algorithms, *DGCI'16*, LNCS, vol. 9647, p.45–56, 2016.