Digital convexity and digital planarity, global and local perspectives

Jacques-Olivier Lachaud[1]

[1]Lab. of Mathematics, University Savoie Mont Blanc

May 6th, 2019
Meeting on Tomography and Applications
Poltitecnico di Milano

Collaborators

Maximal DSS

- F. de Vieilleville
- F. Feschet
- A. Vialard

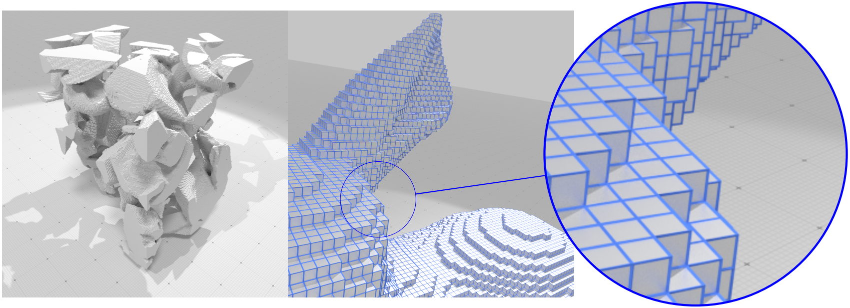
2D convexity

- S. Brlek
- X. Provençal
- C. Reutenauer

Plane probing

- X. Provençal
- T. Roussillon

Why digital convexity ?



- no (infinitesimal) differential geometry for digital shapes
- convexity: a fundamental tool to analyze the geometry of shapes
- identifies convex/concave/flat/saddle regions
- gives locally its piecewise linear geometry
- facets give normal estimations

Digital convexity and digital planarity, global and local perspectives

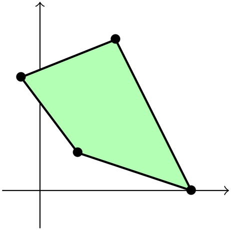
Digital convexity: 2D case

3D digital convexity and digital plane recognition

Local plane probing algorithms

Convex polytopes and polyhedra

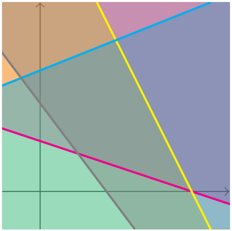
convex polytope



convex combination of vertices

$\Leftrightarrow$

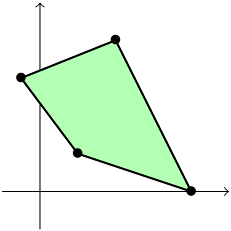
bounded convex polyhedron



finite intersection of half-planes

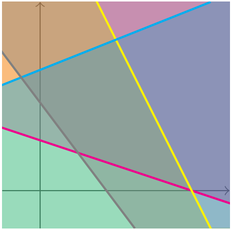
Convex polytopes and polyhedra

convex polytope



$\Leftrightarrow$

bounded convex polyhedron



convex combination of vertices

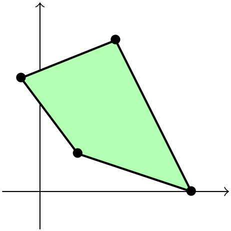
finite intersection of half-planes

Property

Convexity implies (arc)-connectedness.

Convex polytopes and polyhedra

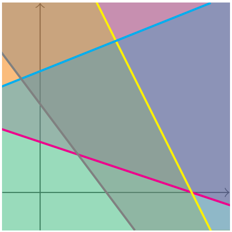
convex polytope



convex combination of vertices

$\Longleftrightarrow$

bounded convex polyhedron



finite intersection of half-planes

Property

Convexity implies (arc)-connectedness.

Global shape view

every diagonal lies inside the shape

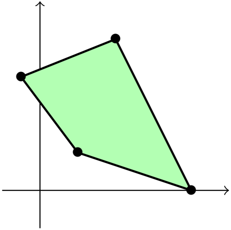
$\Longleftrightarrow$

Local boundary view

planar facets with dihedral angle $\leq \pi$

Convex polytopes and polyhedra

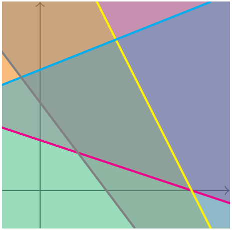
convex polytope



convex combination of vertices

$\Leftrightarrow$

bounded convex polyhedron



finite intersection of half-planes

Link number of vertices and facets

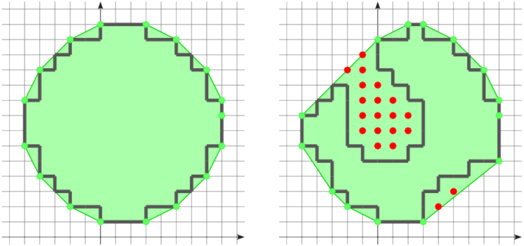
dimension	# vertices	# half-planes
2	v	v
3	v	$\leq 2v - 4$
d	v	$\leq O(v^{\lfloor d/2 \rfloor})$

Reciprocally, determining if v vertices are enough to represent a polyhedron with m facets is hard (vertex counting problem, PP-complete).

Digital convexity (global shape view)

Definition (Digital convexity in d-D)

Digital set $S \subset \mathbb{Z}^d$ is convex iff $\mathrm{Conv}(S) \cap \mathbb{Z}^d = S$.

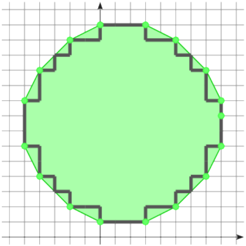


convex not convex

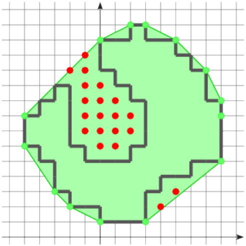
Digital convexity (global shape view)

Definition (Digital convexity in d-D)

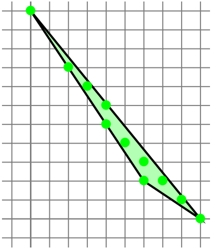
Digital set $S \subset \mathbb{Z}^d$ is convex iff $\mathrm{Conv}(S) \cap \mathbb{Z}^d = S$.



convex



not convex



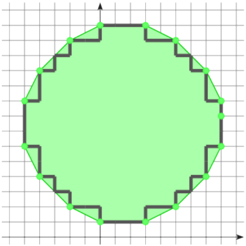
convex
(not connected)

Unfortunately, $d \geq 2$, digital convexity does not imply digital connectedness

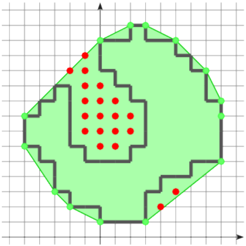
Digital convexity (global shape view)

Definition (Digital convexity in d-D)

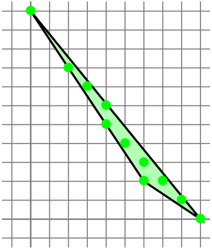
Digital set $S \subset \mathbb{Z}^d$ is convex iff $\mathrm{Conv}(S) \cap \mathbb{Z}^d = S$.



convex



not convex



convex
(not connected)

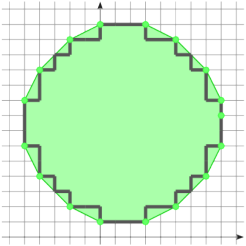
Digital convexity test in $\mathbb{Z}^2$

Best algorithm in $O(n + h \log r)$, $n = \mathrm{Card}(S)$, $h =$ nb output edges, $r = \mathrm{diam}(S)$ [Crombez, da Fonseca, Gerard 2019]

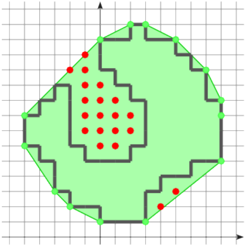
Digital convexity (global shape view)

Definition (Digital convexity in d-D)

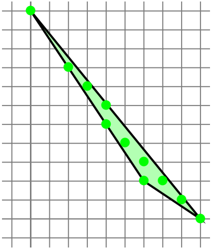
Digital set $S \subset \mathbb{Z}^d$ is convex iff $\mathrm{Conv}(S) \cap \mathbb{Z}^d = S$.



convex



not convex



convex
(not connected)

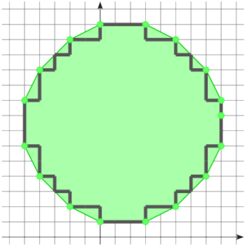
Non connectedness

No correct definition of digital shape boundary, useless for local geometric analysis

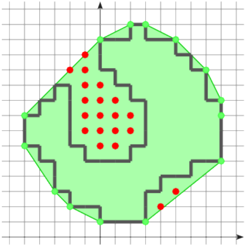
Digital convexity in 2D (global shape view)

Definition ((Usual) digital convexity in 2-D)

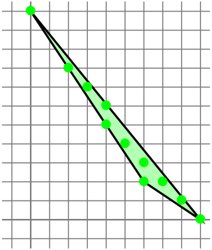
Digital set $S \subset \mathbb{Z}^2$ is convex iff $\mathrm{Conv}(S) \cap \mathbb{Z}^2 = S$ and S 4-connected.



convex



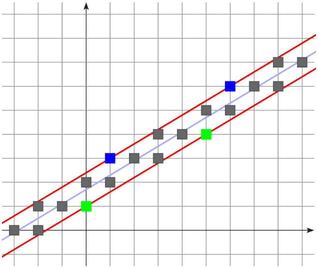
not convex



not convex

- many equivalent definitions: straight segment convexity, triangle convexity, . . . [Minsky, Papert 88], [Kim, Rosenfeld 83], [Hübler, Klette, Voss], . . .
- convexity test or convex hull in $O(n)$,
- digitally convex set have 4-connected boundary.

2D digital straightness, i.e. what is planar facet ?



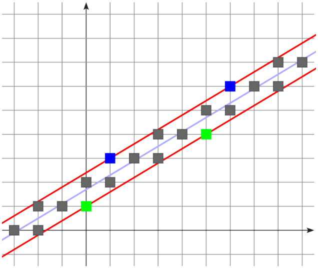
$-12 \leq 3x - 5y < -4$

Standard line [Reveillès 91],[Kovalevsky 90]

$\mu \leq ax - by < \mu + |a| + |b|$

- for $(x, y) \in \mathbb{Z}^2$
- slope $\frac{a}{b}$, shift μ
- 4-connected path in $\mathbb{Z}^2$

2D digital straightness, i.e. what is planar facet ?

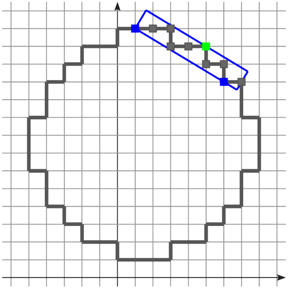


$-12 \leq 3x - 5y < -4$

Standard line [Reveillès 91],[Kovalevsky 90]

$\mu \leq ax - by < \mu + |a| + |b|$

- for $(x, y) \in \mathbb{Z}^2$
- slope $\frac{a}{b}$, shift μ
- 4-connected path in $\mathbb{Z}^2$



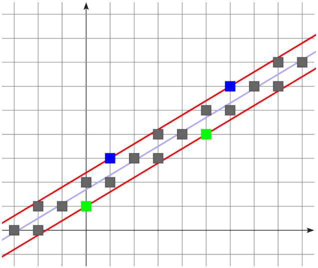
Digital Straight Segment (DSS)

Connected subset of standard line

Maximal DSS

Inextensible DSS on a 4-connected contour C

2D digital straightness, i.e. what is planar facet ?

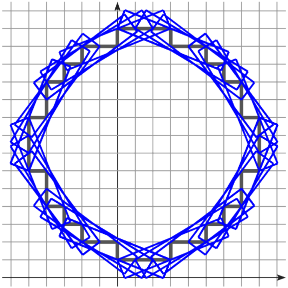


$-12 \leq 3x - 5y < -4$

Standard line [Reveillès 91],[Kovalevsky 90]

$\mu \leq ax - by < \mu + |a| + |b|$

- for $(x, y) \in \mathbb{Z}^2$
- slope $\frac{a}{b}$, shift μ
- 4-connected path in $\mathbb{Z}^2$



Digital Straight Segment (DSS)

Connected subset of standard line

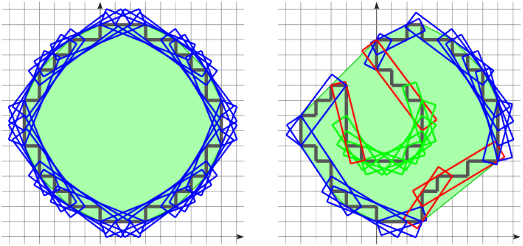
Maximal DSS

Inextensible DSS on a 4-connected contour C

Tangential cover

Sequence of maximal DSS along C [Feschet, Tougne, 99]

Digital convexity and maximal DSS (local boundary view)

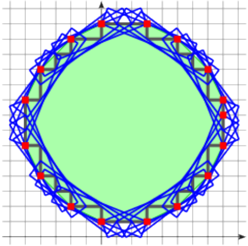


Theorem ([Debled-Rennesson,Reiter-Doerksen 04])

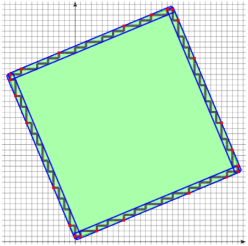
A 4-connected subset $S \subset \mathbb{Z}^2$ is digitally convex, iff the directions of its maximal DSS are monotonous along $\mathrm{Bd}(S)$.

- can split a digital contour into convex and concave parts, separated by a flat inflexion zone,
- when $S = X \cap \mathbb{Z}^2$ has an inflexion zone, X is not convex (around)
- convexity test in $O(m)$, $m = \mathrm{Card}(\mathrm{Bd}(S))$, $m \ll \mathrm{Card}(S) = n$

Number of vertices and number of maximal DSS



$n_{MS} = 24, v = 16$



$n_{MS} = 4, v = 24$

Theorem ([de Vieilleville, L., Feschet 07])

If X is a compact convex shape with C^3 boundary, h a digitization step, then

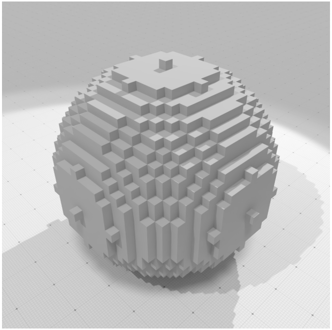
$$\frac{v(\Gamma_h)}{\Theta(\log \frac{1}{h})} \leq n_{MS}(\mathrm{Bd}(\Gamma_h)) \leq 3v(\Gamma_h), \qquad \text{avec} \quad \Gamma_h = \left(\frac{1}{h} \cdot X\right) \cap \mathbb{Z}^2.$$

Digital convexity and digital planarity, global and local perspectives

Digital convexity in 3D (global shape view)

Definition (digital convexity in 3-D)

Digital set $S \subset \mathbb{Z}^3$ is convex iff $\mathrm{Conv}(S) \cap \mathbb{Z}^3 = S$ and S 6-connected.

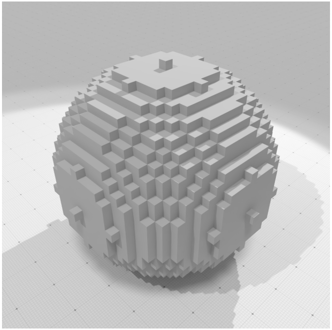


convex

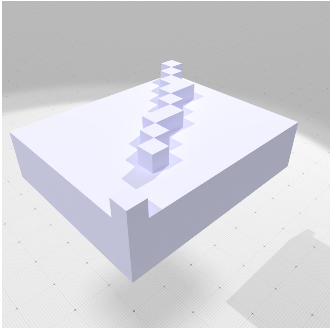
Digital convexity in 3D (global shape view)

Definition (digital convexity in 3-D)

Digital set $S \subset \mathbb{Z}^3$ is convex iff $\mathrm{Conv}(S) \cap \mathbb{Z}^3 = S$ and S 6-connected.



convex

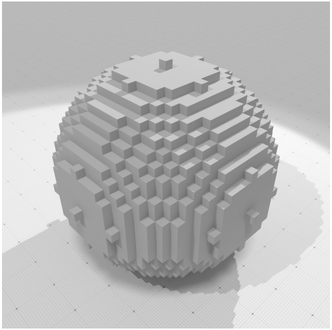


convex !

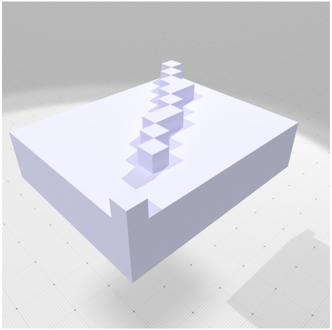
Digital convexity in 3D (global shape view)

Definition (digital convexity in 3-D)

Digital set $S \subset \mathbb{Z}^3$ is convex iff $\mathrm{Conv}(S) \cap \mathbb{Z}^3 = S$ and S 6-connected.



convex

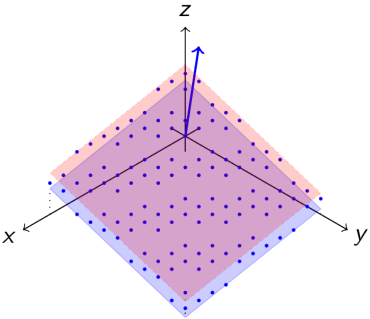


convex !

No clear definition due to connectedness issues.

3D digital straightness, i.e. what is a planar facet ?

(Naive) Arithmetic plane



[Forchhammer 89], [Reveillès 91]

Standard digital plane is:

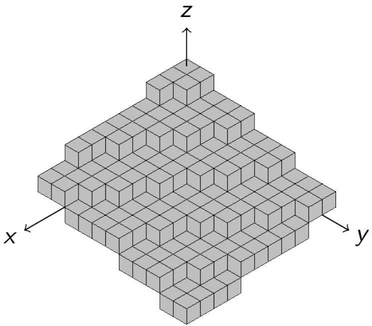
$$\mathbf{P}(\mathbf{N}, \mu) = \{\mathbf{x} \in \mathbb{Z}^3 \mid \mu \leq \langle \mathbf{N}, \mathbf{x} \rangle < \mu + \|\mathbf{N}\|_1\}$$

where

- $\mathbf{N}$ is the normal vector.
- μ is the shift.

3D digital straightness, i.e. what is a planar facet ?

(Naive) Arithmetic plane



[Forchhammer 89], [Reveillès 91]

Standard digital plane is:

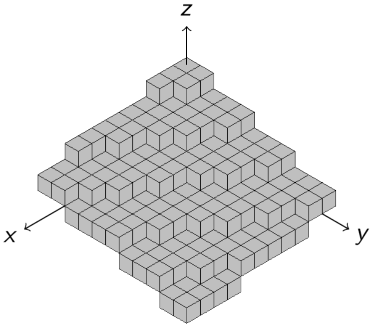
$$\mathbf{P}(\mathbf{N},\mu) = \{\mathbf{x} \in \mathbb{Z}^3 \mid \mu \leq \langle \mathbf{N}, \mathbf{x} \rangle < \mu + \|\mathbf{N}\|_1\}$$

where

- $\mathbf{N}$ is the normal vector.
- μ is the shift.

3D digital straightness, i.e. what is a planar facet ?

(Naive) Arithmetic plane



[Forchhammer 89], [Reveillès 91]

Standard digital plane is:

$$\mathbf{P}(\mathbf{N}, \mu) = \{\mathbf{x} \in \mathbb{Z}^3 \mid \mu \leq \langle \mathbf{N}, \mathbf{x} \rangle < \mu + \|\mathbf{N}\|_1\}$$

where

- $\mathbf{N}$ is the normal vector.
- μ is the shift.

Digital Plane Segment (DPS)

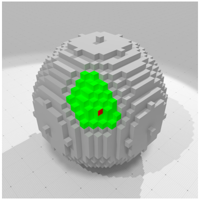
Any connected subset of a standard plane.

- DPS recognition: given a subset $T \subset \mathbb{Z}^3$, tells if T is a DPS and its characteristics $\mathbf{N}, \mu$
- many algorithms [Charrier,Buzer 08] [Gérard *et al* 05], [Veelaert 94], [Brimkov, Dantchev 05], . . .

Tangential cover in 3D ?

Facets = inextensible pieces of planes ?

Can we define facets of S as inextensible connected pieces of standard planes along $\mathrm{Bd}(S)$?



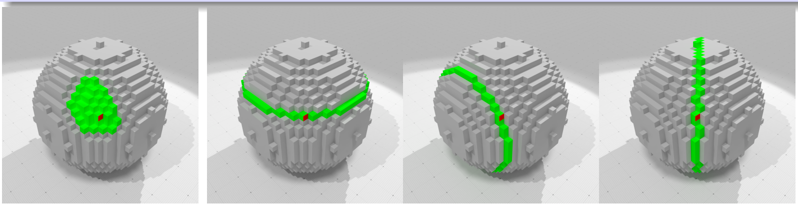
Tangential cover in 3D ?

Facets = inextensible pieces of planes ?

Can we define facets of S as inextensible connected pieces of standard planes along $\mathrm{Bd}(S)$?

Contrarily to 2D, maximal pieces of planes along $\mathrm{Bd}(S)$ are **not tangent**.

- there are a lot of inextensible DPS
- most of them are meaningless



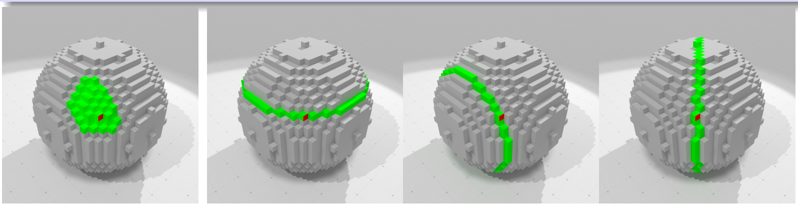
Tangential cover in 3D ?

Facets = inextensible pieces of planes ?

Can we define facets of S as inextensible connected pieces of standard planes along $\mathrm{Bd}(S)$?

Contrarily to 2D, maximal pieces of planes along $\mathrm{Bd}(S)$ are **not tangent**.

- there are a lot of inextensible DPS
- most of them are meaningless



- greedy methods to isolate meaningful ones: [Klette, Sun, Coeurjolly, Sivignon, Kenmochi, Provot, Debled-Rennesson, Charrier, L., . . .]

Digital convexity and digital planarity, global and local perspectives

Digital convexity: 2D case

3D digital convexity and digital plane recognition

Local plane probing algorithms

Probing algorithms (local boundary view)

Main difficulty of planar facet identification

Given object S, the problem is not to decide whether a subset $T \subset S$ is planar, but to determine local meaningful subsets (T_i), i.e. the "most tangent ones".

Probing algorithms

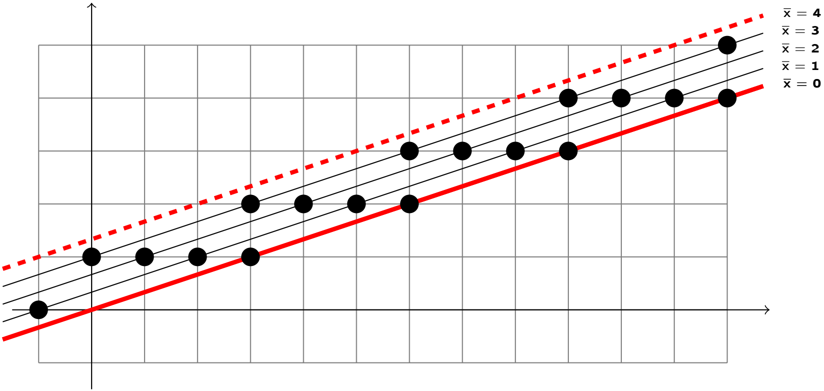
- **Input :** predicate $\mathcal{P}(\mathbf{x}) :=$ "is $\mathbf{x}$ in Object S", where $S \subset \mathbb{Z}^3$
- given a starting "corner", decides on-the-fly which points to probe
- and output a basis of the local planar geometry

Upward-oriented frame algorithm of [L., Provençal, Roussillon 2016]

- Starting "corner" is any trivial frame included in S
- if S is a standard plane or half-plane, outputs the exact normal $\mathbf{N}$ of S in time $O(\|\mathbf{N}\|_1 \log \|\mathbf{N}\|_1)$

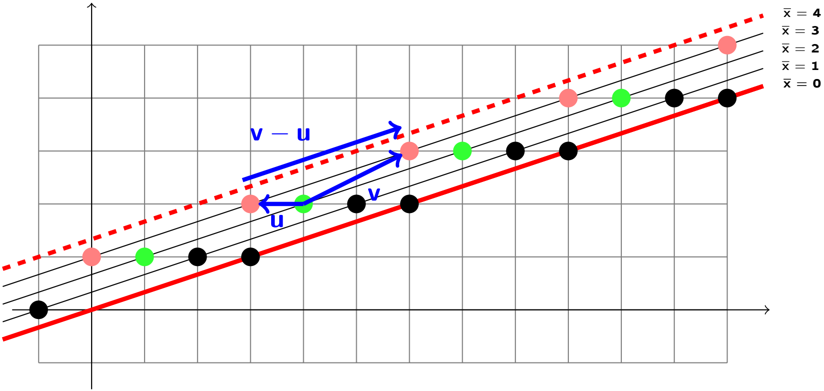
Digital straight line structure

- Notation : $\bar{\mathbf{x}} = \langle \mathbf{N}, \mathbf{x} \rangle$ is the height of point $\mathbf{x}$,
- line with slope $(-3, 1)$ and shift 0 : $\{\mathbf{x} \in \mathbb{Z}^2 \mid 0 \leq \bar{\mathbf{x}} < 4\}$,



Digital straight line structure

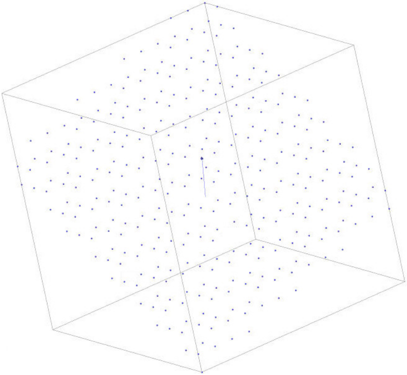
- Notation : $\overline{\mathbf{x}} = \langle \mathbf{N}, \mathbf{x} \rangle$ is the height of point $\mathbf{x}$,
- line with slope $(-3, 1)$ and shift 0 : $\{\mathbf{x} \in \mathbb{Z}^2 \mid 0 \leq \overline{\mathbf{x}} < 4\}$,

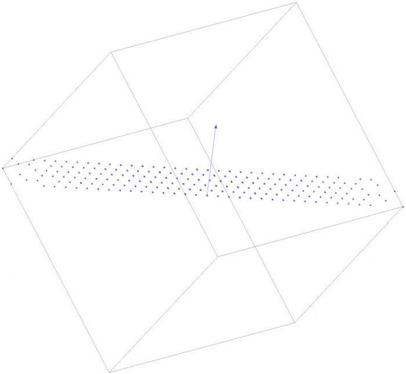


- Bezout vectors : $\overline{\mathbf{u}} = \overline{\mathbf{v}} = 1$,
- if $\det\left(\begin{bmatrix} \mathbf{u} \\ \mathbf{v} \end{bmatrix}\right) = 1$ then $\mathbf{v} - \mathbf{u}$ is a basis of $\{\mathbf{x} \in \mathbb{Z}^2 \mid \overline{\mathbf{x}} = 0\}$.

Digital plane structure

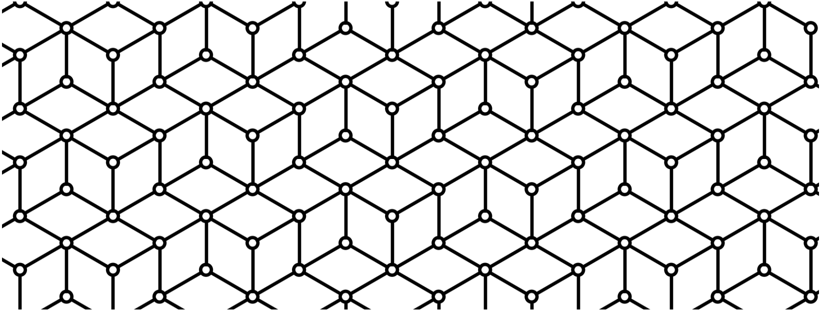
$$\mathbf{N} = (1, 2, 3), \quad \mathbf{P}(\mathbf{N}, 0) = \{\mathbf{x} \in \mathbb{Z}^3 \mid 0 \leq \bar{\mathbf{x}} < \|\mathbf{N}\|_1\}$$





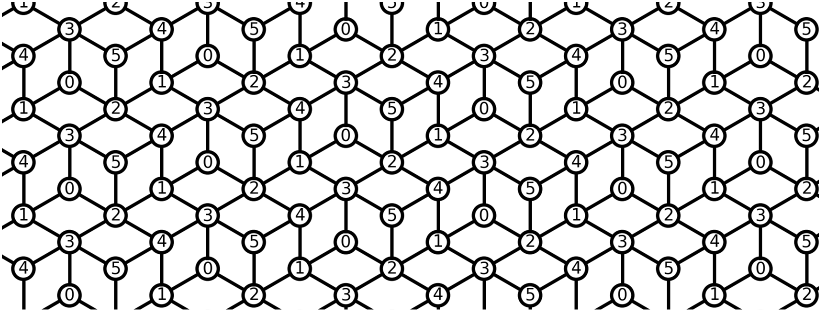
Digital plane structure

$$\mathbf{N} = (1,2,3), \quad \mathbf{P}(\mathbf{N},0) = \{\mathbf{x} \in \mathbb{Z}^3 \mid 0 \leq \overline{\mathbf{x}} < \|\mathbf{N}\|_1\}$$



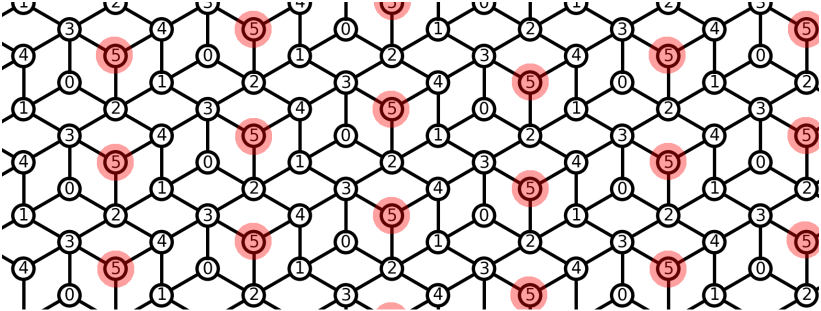
Digital plane structure

$$\mathbf{N} = (1,2,3), \quad \mathbf{P}(\mathbf{N},0) = \{\mathbf{x} \in \mathbb{Z}^3 \mid 0 \leq \overline{\mathbf{x}} < \|\mathbf{N}\|_1\}$$



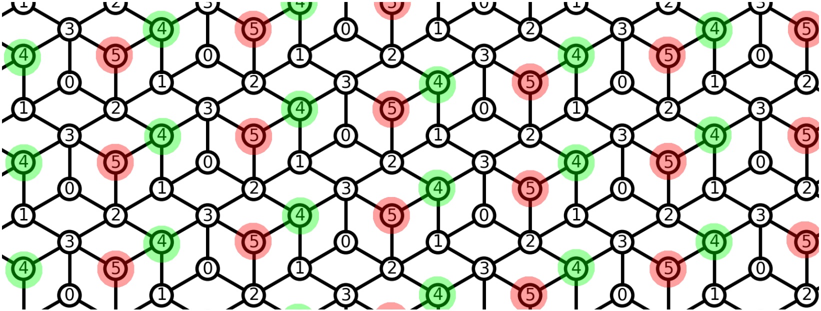
Digital plane structure

$$\mathbf{N} = (1, 2, 3), \quad \mathbf{P}(\mathbf{N}, 0) = \{\mathbf{x} \in \mathbb{Z}^3 \mid 0 \leq \overline{\mathbf{x}} < \|\mathbf{N}\|_1\}$$



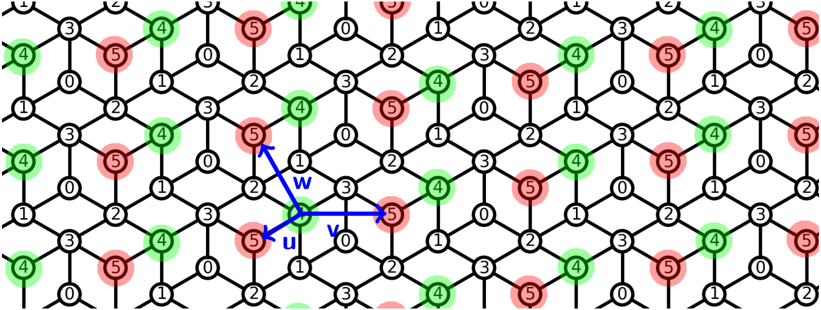
Digital plane structure

$$\mathbf{N} = (1,2,3), \quad \mathbf{P}(\mathbf{N},0) = \{\mathbf{x} \in \mathbb{Z}^3 \mid 0 \leq \overline{\mathbf{x}} < \|\mathbf{N}\|_1\}$$



Digital plane structure

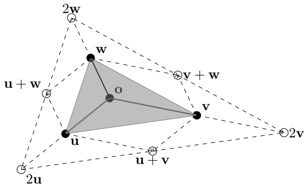
$$\mathbf{N} = (1,2,3), \quad \mathbf{P}(\mathbf{N},0) = \{\mathbf{x} \in \mathbb{Z}^3 \mid 0 \leq \overline{\mathbf{x}} < \|\mathbf{N}\|_1\}$$



If $\overline{\mathbf{u}} = \overline{\mathbf{v}} = \overline{\mathbf{w}} = 1$ (Bezout vectors) and $\det\left(\begin{bmatrix}\mathbf{u}\\\mathbf{v}\\\mathbf{w}\end{bmatrix}\right) = 1$ then

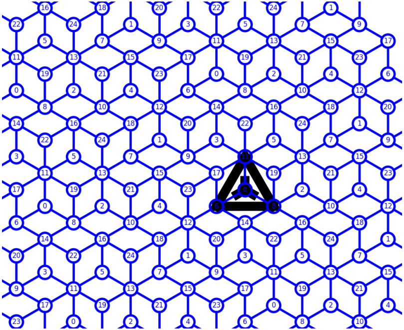
- $(\mathbf{v}-\mathbf{u})$ and $(\mathbf{w}-\mathbf{u})$ form a basis of $\{\mathbf{x} \in \mathbb{Z}^3 \mid \overline{\mathbf{x}} = 0\}$,
- $(\mathbf{v}-\mathbf{u}) \times (\mathbf{w}-\mathbf{u}) = \pm\mathbf{N}$

Idea of Upward-oriented frame algorithm [LPR2016]



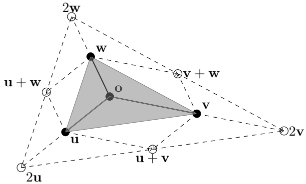
Update progressively an initial trivial basis **o**, **u**, **v**, **w** by probing neighbor points . . . and sometimes further points

- $\mathbf{N} = (6, 8, 11)$,
- Opérations :

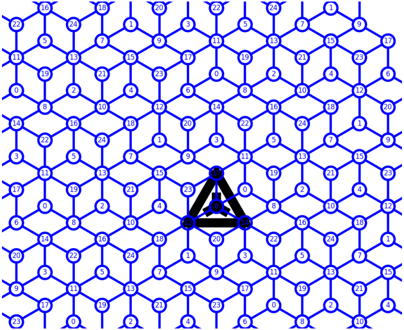


Idea of Upward-oriented frame algorithm [LPR2016]

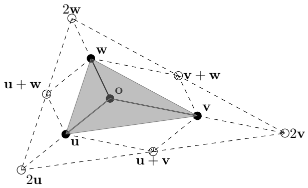
Update progressively an initial trivial basis $\mathbf{o}, \mathbf{u}, \mathbf{v}, \mathbf{w}$ by probing neighbor points . . . and sometimes further points



- $\mathbf{N} = (6, 8, 11)$,
- Opérations :
  - ▷ translation $\mathbf{o}' \leftarrow \mathbf{o} + \mathbf{u}$

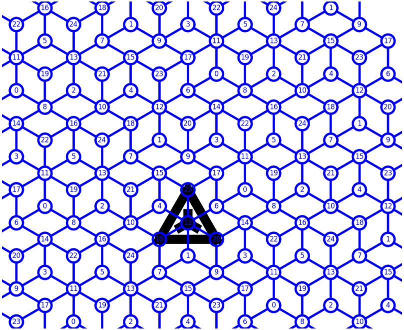


Idea of Upward-oriented frame algorithm [LPR2016]



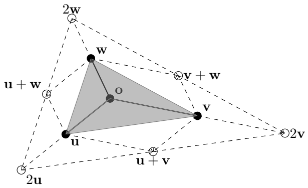
Update progressively an initial trivial basis $\mathbf{o}, \mathbf{u}, \mathbf{v}, \mathbf{w}$ by probing neighbor points ... and sometimes further points

- $\mathbf{N} = (6, 8, 11)$,
- Opérations :
  - ▷ translation $\mathbf{o}' \leftarrow \mathbf{o} + \mathbf{u}$
  - ▷ translation $\mathbf{o}' \leftarrow \mathbf{o} + \mathbf{u}$

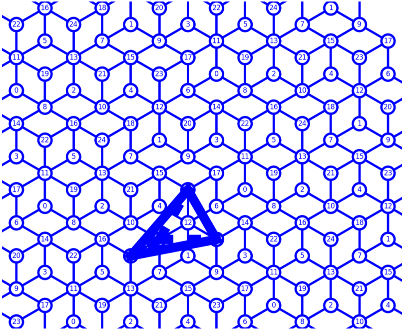


Idea of Upward-oriented frame algorithm [LPR2016]

Update progressively an initial trivial basis $\mathbf{o}, \mathbf{u}, \mathbf{v}, \mathbf{w}$ by probing neighbor points … and sometimes further points

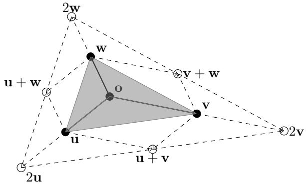


- $\mathbf{N} = (6, 8, 11)$,
- Opérations :
  - ▷ translation $\mathbf{o}' \leftarrow \mathbf{o} + \mathbf{u}$
  - ▷ translation $\mathbf{o}' \leftarrow \mathbf{o} + \mathbf{u}$
  - ▷ Brun $\begin{cases} \mathbf{v}' \leftarrow \mathbf{v} - \mathbf{u} \\ \mathbf{w}' \leftarrow \mathbf{w} - \mathbf{u} \end{cases}$

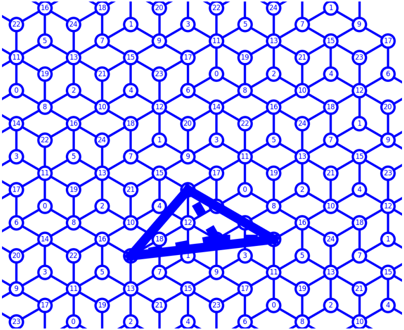


Idea of Upward-oriented frame algorithm [LPR2016]

Update progressively an initial trivial basis **o**, **u**, **v**, **w** by probing neighbor points . . . and sometimes further points

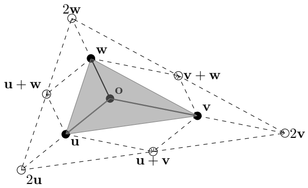


- $\mathbf{N} = (6, 8, 11)$,
- Opérations :
  - ▷ translation $\mathbf{o}' \leftarrow \mathbf{o} + \mathbf{u}$
  - ▷ translation $\mathbf{o}' \leftarrow \mathbf{o} + \mathbf{u}$
  - ▷ Brun $\begin{cases} \mathbf{v}' \leftarrow \mathbf{v} - \mathbf{u} \\ \mathbf{w}' \leftarrow \mathbf{w} - \mathbf{u} \end{cases}$
  - ▷ Brun $\begin{cases} \mathbf{u}' \leftarrow \mathbf{u} - \mathbf{v} \\ \mathbf{w}' \leftarrow \mathbf{w} - \mathbf{v} \end{cases}$

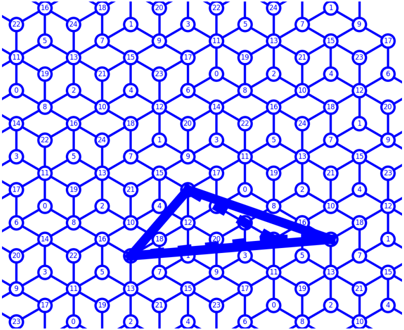


Idea of Upward-oriented frame algorithm [LPR2016]

Update progressively an initial trivial basis $\mathbf{o}, \mathbf{u}, \mathbf{v}, \mathbf{w}$ by probing neighbor points ... and sometimes further points

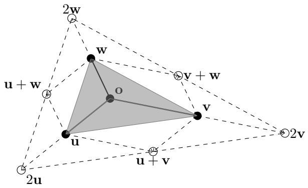


- $\mathbf{N} = (6, 8, 11)$,
- Opérations :
  - ▷ translation $\mathbf{o}' \leftarrow \mathbf{o} + \mathbf{u}$
  - ▷ translation $\mathbf{o}' \leftarrow \mathbf{o} + \mathbf{u}$
  - ▷ Brun $\begin{cases} \mathbf{v}' \leftarrow \mathbf{v} - \mathbf{u} \\ \mathbf{w}' \leftarrow \mathbf{w} - \mathbf{u} \end{cases}$
  - ▷ Brun $\begin{cases} \mathbf{u}' \leftarrow \mathbf{u} - \mathbf{v} \\ \mathbf{w}' \leftarrow \mathbf{w} - \mathbf{v} \end{cases}$
  - ▷ Brun $\begin{cases} \mathbf{u}' \leftarrow \mathbf{u} - \mathbf{v} \\ \mathbf{w}' \leftarrow \mathbf{w} - \mathbf{v} \end{cases}$

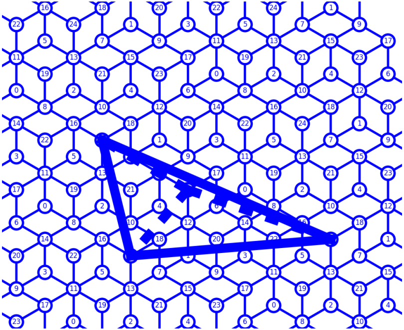


Idea of Upward-oriented frame algorithm [LPR2016]

Update progressively an initial trivial basis **o**, **u**, **v**, **w** by probing neighbor points . . . and sometimes further points



- $\mathbf{N} = (6, 8, 11)$,
- Opérations :
  - ▷ translation $\mathbf{o}' \leftarrow \mathbf{o} + \mathbf{u}$
  - ▷ translation $\mathbf{o}' \leftarrow \mathbf{o} + \mathbf{u}$
  - ▷ Brun $\begin{cases} \mathbf{v}' \leftarrow \mathbf{v} - \mathbf{u} \\ \mathbf{w}' \leftarrow \mathbf{w} - \mathbf{u} \end{cases}$
  - ▷ Brun $\begin{cases} \mathbf{u}' \leftarrow \mathbf{u} - \mathbf{v} \\ \mathbf{w}' \leftarrow \mathbf{w} - \mathbf{v} \end{cases}$
  - ▷ Brun $\begin{cases} \mathbf{u}' \leftarrow \mathbf{u} - \mathbf{v} \\ \mathbf{w}' \leftarrow \mathbf{w} - \mathbf{v} \end{cases}$
  - ▷ Brun $\begin{cases} \mathbf{u}' \leftarrow \mathbf{u} - \mathbf{w} \\ \mathbf{v}' \leftarrow \mathbf{v} - \mathbf{w} \end{cases}$



Another probing algorithm

Upward-oriented frame algorithm [L., Provençal, Roussillon 2016]

- starting "corner" is any trivial frame included in S
- if S is a standard plane or half-plane, outputs the exact normal $\mathbf{N}$ of S in time $O(\|\mathbf{N}\|_1 \log \|\mathbf{N}\|_1)$
- but no control over the frame displacement

Another probing algorithm

Upward-oriented frame algorithm [L., Provençal, Roussillon 2016]

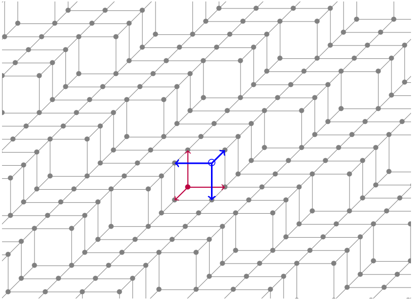
- starting "corner" is any trivial frame included in S
- if S is a standard plane or half-plane, outputs the exact normal $\mathbf{N}$ of S in time $O(\|\mathbf{N}\|_1 \log \|\mathbf{N}\|_1)$
- but no control over the frame displacement

Downward-oriented algorithms [L., Provençal, Roussillon 2017, 2019]

- starting "corner" is a reentrant corner of $\mathrm{Bd}(S)$
- origin is **immutable**
- if S is a standard plane and origin Bezout point, outputs the exact normal $\mathbf{N}$ of S in time $O(\|\mathbf{N}\|_1)$
- variants: H-, R- and R^1-algorithms

A common procedure for all algorithms

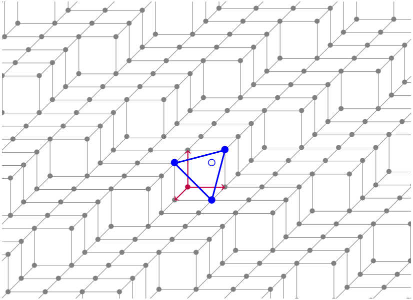
We are given a predicate $\mathcal{P}$:
"is $\mathbf{x} \in$ Object?".



A common procedure for all algorithms

We are given a predicate $\mathcal{P}$:
"is $\mathbf{x} \in$ Object?".

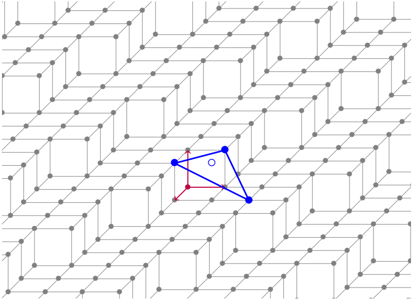
- start with a triangle T in a reentrant corner $\mathbf{N}(T) = (1, 1, 1)$



A common procedure for all algorithms

We are given a predicate $\mathcal{P}$:
"is $\mathbf{x} \in$ Object?".

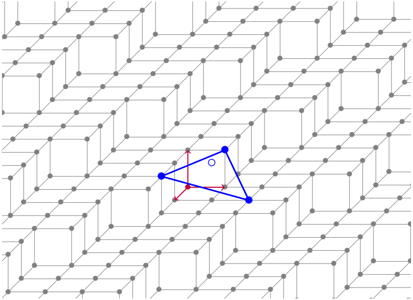
- start with a triangle T in a reentrant corner $\mathbf{N}(T) = (1, 1, 1)$
- update one vertex



A common procedure for all algorithms

We are given a predicate $\mathcal{P}$:
"is $\mathbf{x} \in$ Object?".

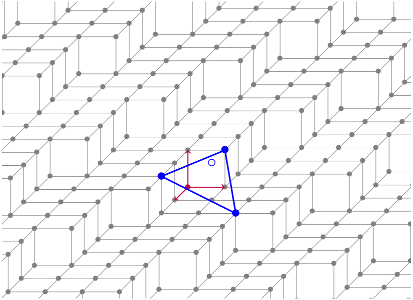
- start with a triangle T in a reentrant corner $\mathbf{N}(T) = (1, 1, 1)$
- update one vertex
- reapeat until $\mathbf{N}(T) = \mathbf{N}$ (for a deep enough corner)



A common procedure for all algorithms

We are given a predicate $\mathcal{P}$:
"is $\mathbf{x} \in$ Object?".

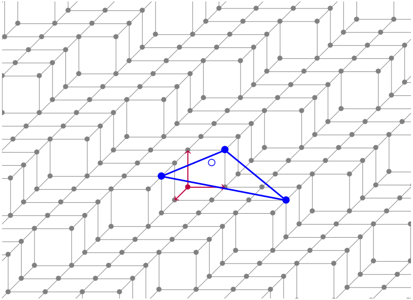
- start with a triangle T in a reentrant corner $\mathbf{N}(T) = (1, 1, 1)$
- update one vertex
- reapeat until $\mathbf{N}(T) = \mathbf{N}$ (for a deep enough corner)



A common procedure for all algorithms

We are given a predicate $\mathcal{P}$:
"is $\mathbf{x} \in$ Object?".

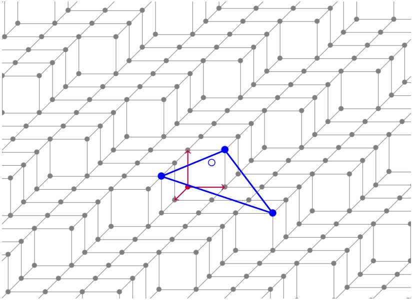
- start with a triangle T in a reentrant corner $\mathbf{N}(T) = (1, 1, 1)$
- update one vertex
- reapeat until $\mathbf{N}(T) = \mathbf{N}$ (for a deep enough corner)



A common procedure for all algorithms

We are given a predicate $\mathcal{P}$:
"is $\mathbf{x} \in$ Object?".

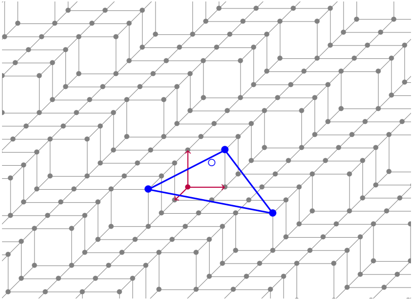
- start with a triangle T in a reentrant corner $\mathbf{N}(T) = (1,1,1)$
- update one vertex
- reapeat until $\mathbf{N}(T) = \mathbf{N}$ (for a deep enough corner)



A common procedure for all algorithms

We are given a predicate $\mathcal{P}$:
"is $\mathbf{x} \in$ Object?".

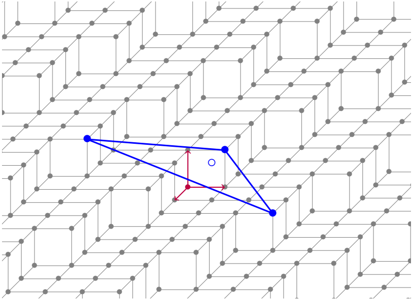
- start with a triangle T in a reentrant corner $\mathbf{N}(T) = (1, 1, 1)$
- update one vertex
- reapeat until $\mathbf{N}(T) = \mathbf{N}$ (for a deep enough corner)



A common procedure for all algorithms

We are given a predicate $\mathcal{P}$:
"is $\mathbf{x} \in$ Object?".

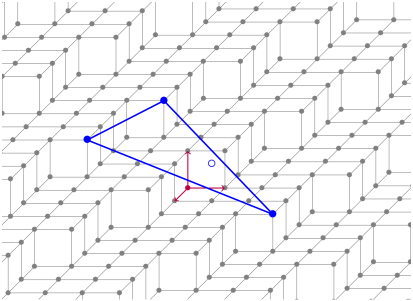
- start with a triangle T in a reentrant corner $\mathbf{N}(T) = (1, 1, 1)$
- update one vertex
- reapeat until $\mathbf{N}(T) = \mathbf{N}$ (for a deep enough corner)



A common procedure for all algorithms

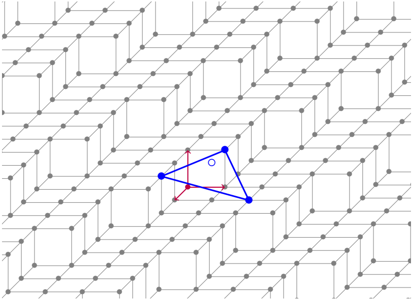
We are given a predicate $\mathcal{P}$:
"is $\mathbf{x} \in$ Object?".

- start with a triangle T in a reentrant corner $\mathbf{N}(T) = (1, 1, 1)$
- update one vertex
- reapeat until $\mathbf{N}(T) = \mathbf{N}$ (for a deep enough corner)
- at each step, vectors ∘ to T) form an unimodular matrix



Update procedure

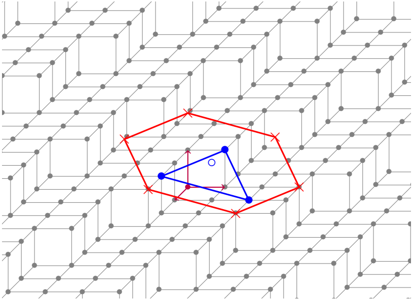
At a given step:



Update procedure

At a given step:

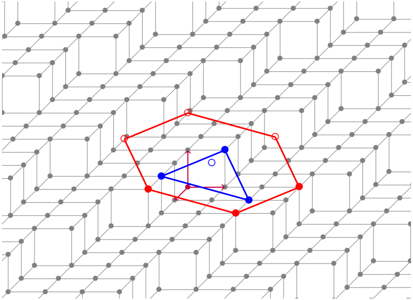
- consider a candidate set S



Update procedure

At a given step:

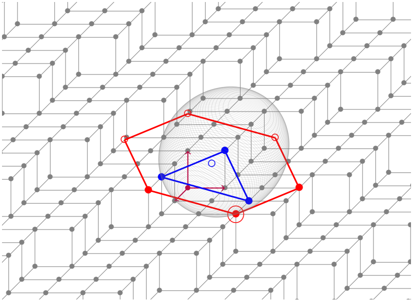
- consider a candidate set S
- filter S through $\mathcal{P}$



Update procedure

At a given step:

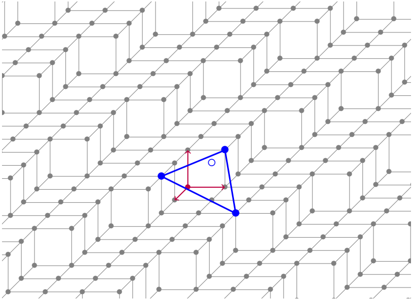
- consider a candidate set S
- filter S through $\mathcal{P}$
- select a *closest* point $s^\star$: the circumsphere of $T \cup s^\star$ doesn't contain any other



Update procedure

At a given step:

- consider a candidate set S
- filter S through $\mathcal{P}$
- select a *closest* point $s^\star$: the circumsphere of $T \cup s^\star$ doesn't contain any other
- update T with this point



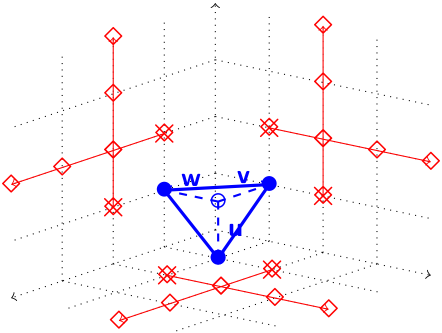
Difference between algorithms

Each algorithm considers a distinct candidate set:

S_H ($\times$): 6 Hexagon vertices

S_R ($\diamond$): 6 Rays (which are infinite)

S_{R^1} ($\diamond$): 6 Hexagon vertices + 1 Ray



Difference between algorithms

Each algorithm considers a distinct candidate set:

S_H ($\times$): 6 Hexagon vertices

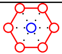
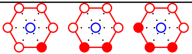
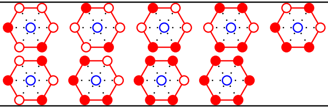
S_R ($\diamond$): 6 Rays (which are infinite)

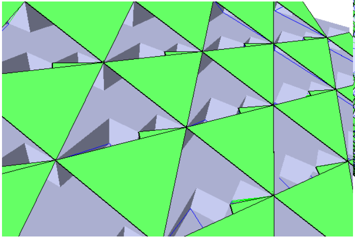
S_{R^1} ($\diamond$): 6 Hexagon vertices + 1 Ray

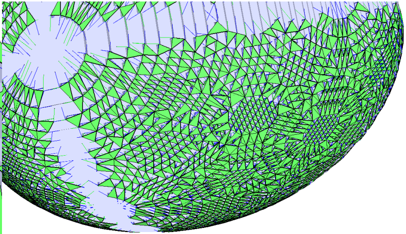
algorithm	complexity	observed	reduced basis	local	output
Upward algo	$O(\omega \log \omega)$	$\log \omega$	6%	no	**N**
H-algo	$O(\omega \log \omega)$	$\log \omega$	99.99%	yes	**N** if origin is Bezout point
R-algo	$O(\omega \log \omega)$	$\log \omega$	100%	yes	
R^1-algo	$O(\omega)$	$\log \omega$	100%	yes	

if $\omega = \|\mathbf{N}\|_1$.

What about arbitrary digital shape ?

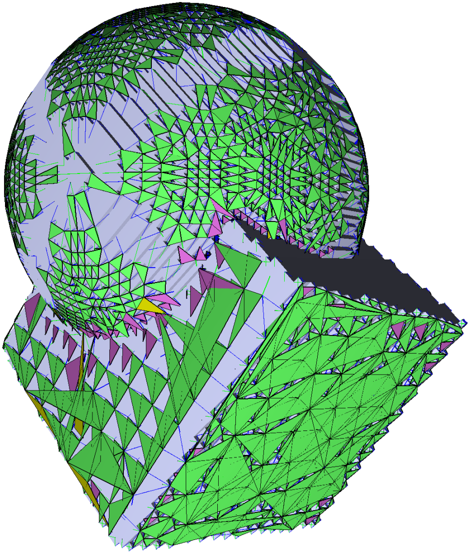
H-neighborhood configurations	Stop	Local planarity
	yes	convex or planar (green)
	no	(still probing)
	yes	non-convex (magenta)





Digital shape analysis

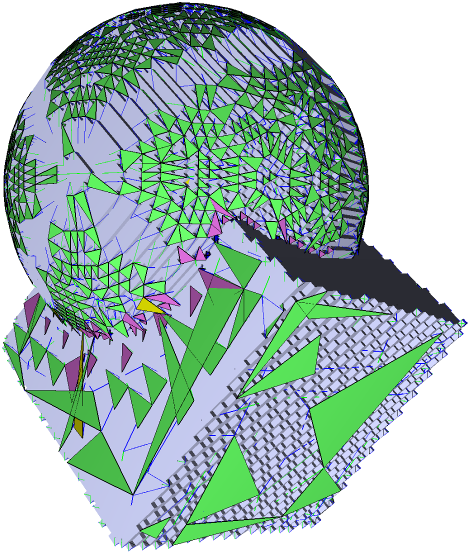
- convex or planar
- non convex
- points under triangle not planar



Digital shape analysis

- convex or planar
- non convex
- points under triangle not planar

Patterns "included" into other patterns are removed

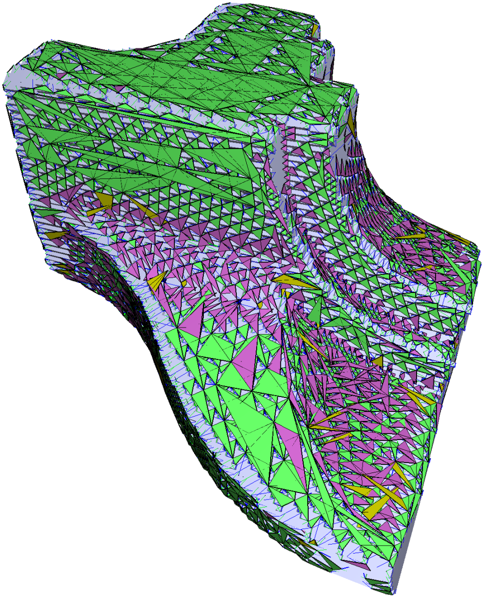


Digital shape analysis

convex or planar

non convex

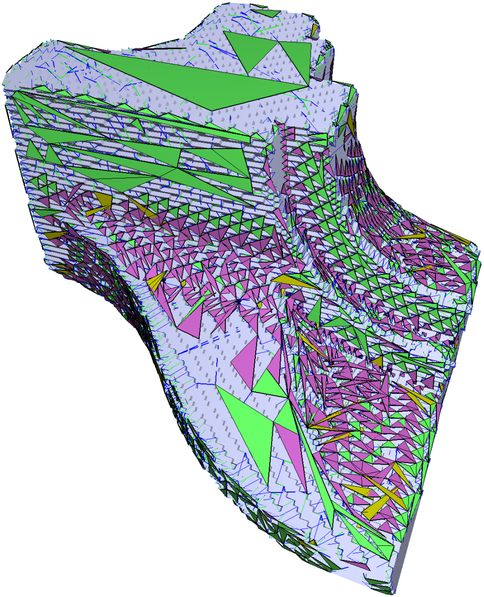
points under
triangle not planar



Digital shape analysis

- convex or planar
- non convex
- points under triangle not planar

Patterns "included" into other patterns are removed



Conclusion

To conclude

- digital straightness give local approaches to convexity
- convexity tests, inflexion zones, tangent/normal estimations
- 3d digital convexity leaves open questions
- plane probing algorithms identify planar subsets along shape boundaries
- local geometric analysis: convex, concave, saddle + tangent/normal
- quasi linear algorithms (since normal vectors have bounded norm)

Open questions

- link number of meaningful DPS wrt number of vertices
- complete piecewise linear reconstruction of digital shapes
- consistent definition of digital convexity in nD, $n \geq 3$