Parameter-free analysis of digital surfaces with plane-probing algorithms

Advisors: David Coeurjolly, Tristan Roussillon
PhD student: Jui-Ting Lu

Université de Lyon, INSA Lyon, LIRIS, France

21 December 2023

Outline of talk

1. Introduction
   - Context
   - Plane-probing algorithms
2. L-algorithm
   - L-Neighborhood
   - Complexity
   - Delaunay Property
3. Normal vector estimation
   - Estimation and evaluation
   - Convergence on selected surfels
   - Comparing plane-probing with other methods
4. Conclusion

Table of Contents

1 Introduction
- Context
- Plane-probing algorithms

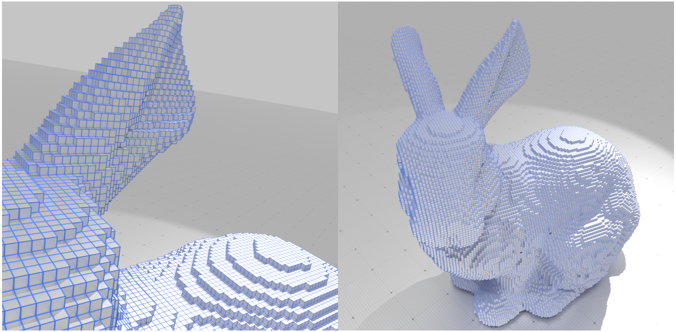
2 L-algorithm
- L-Neighborhood
- Complexity
- Delaunay Property

3 Normal vector estimation
- Estimation and evaluation
- Convergence on selected surfels
- Comparing plane-probing with other methods

4 Conclusion

Digital volume

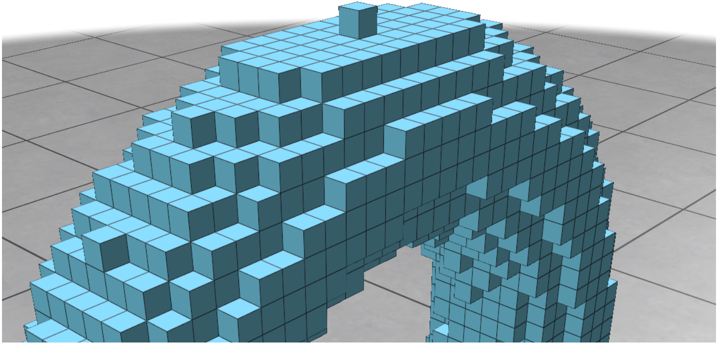
A *digital volume* can be defined as a collection of identical small cubes that share the sames axes of direction



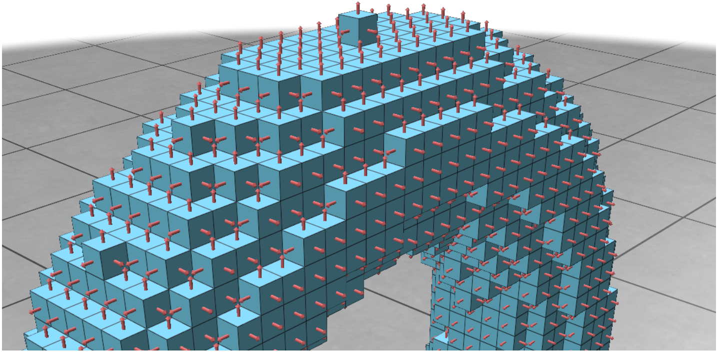
- Efficient spatial data structures.
- Calculation using integers.

Digital surface

We consider *digital surfaces* as boundary of digital volumes.

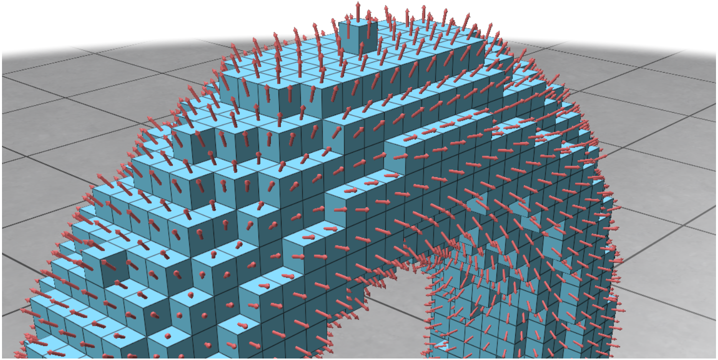


Characteristics of digital surfaces



The normal vector of a surfel always aligns with one of the axes.

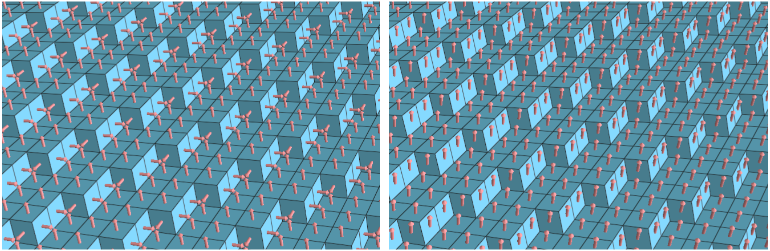
Geometry processing on digital surface



Objective: analyze digital surfaces [1].
For example: estimate normal vectors.

[1] ANR project: PARameter-free Analysis of DIgital Surfaces

Special case: digital plane



Digital plane

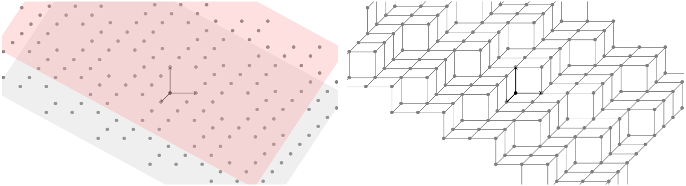
Definition

Given a normal vector $\mathbf{N} \in \mathbb{Z}^3 \setminus \{\mathbf{0}\}$ and a shift vector $\mu \in \mathbb{Z}$, a standard and rational *digital plane* [2] is an infinite digital set:

$$\mathbf{P}_{\mu,\mathbf{N}} := \{\mathbf{x} \in \mathbb{Z}^3 \mid \mu \leq \mathbf{x} \cdot \mathbf{N} < \mu + \|\mathbf{N}\|_1\}. \quad (1)$$

Here, we suppose that $\mu = 0$ and $\mathbf{N} \in \mathbb{N}^3 \setminus \{\mathbf{0}\}$.

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



[2] J-P. Reveillès. "Géométrie Discrète, calculs en nombres entiers et algorithmique". Thèse d'Etat. Université Louis Pasteur, 1991

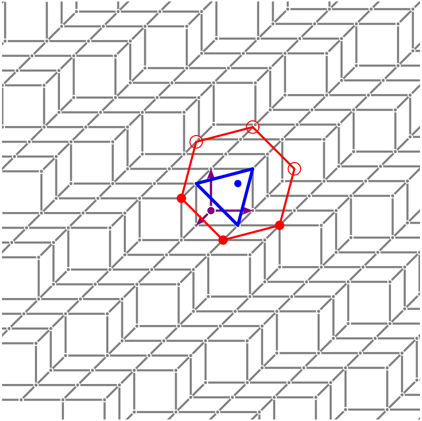
Plane-probing algorithm

Given a digital plane $\mathbf{P}$ and a starting point $\mathbf{p} \in \mathbf{P}$, a plane-probing algorithm[3] computes the normal vector of $\mathbf{P}$ by sparsely probing it with the predicate "is $\mathbf{x} \in \mathbf{P}$" ?

[3] Jacques-Olivier Lachaud, Xavier Provençal, and Tristan Roussillon. "Computation of the normal vector to a digital plane by sampling signicant points". In: *19th IAPR International Conference on Discrete Geometry for Computer Imagery*. Nantes, France, Apr. 2016.

An example of plane-probing algorithm on a plane

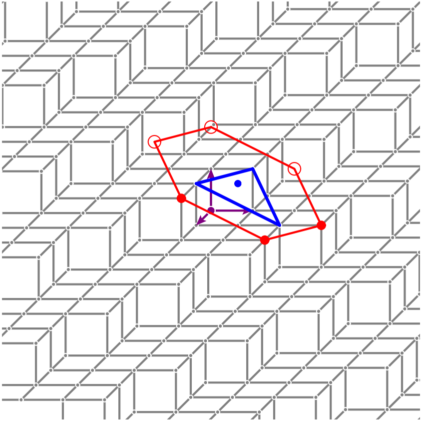
Normal vector: $(1, 2, 5)$



- A predicate **InPlane**: $x \in \mathbf{P}$?
- At each iteration:
  - Consider a **neighborhood** $\mathcal{N}$.
  - Choose a point from $\mathcal{N} \cap \mathbf{P}$.
  - Update one vertex of the base triangle.
- Termination: $\mathcal{N} \cap \mathbf{P} = \emptyset$, with $N(\mathbf{T}) = \mathbf{N}$.

An example of plane-probing algorithm on a plane

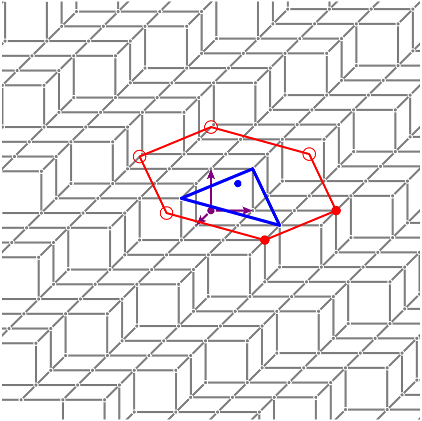
Normal vector: $(1, 2, 5)$



- A predicate **InPlane**: $x \in \mathbf{P}$?
- At each iteration:
  - Consider a **neighborhood** $\mathcal{N}$.
  - Choose a point from $\mathcal{N} \cap \mathbf{P}$.
  - Update one vertex of the base triangle.
- Termination: $\mathcal{N} \cap \mathbf{P} = \emptyset$, with $N(\mathbf{T}) = \mathbf{N}$.

An example of plane-probing algorithm on a plane

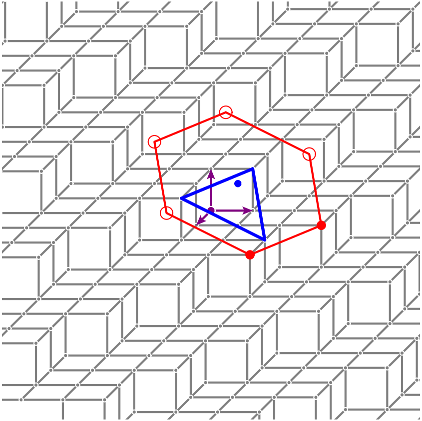
Normal vector: $(1, 2, 5)$



- A predicate **InPlane**: $x \in \mathbf{P}$?
- At each iteration:
  - Consider a **neighborhood** $\mathcal{N}$.
  - Choose a point from $\mathcal{N} \cap \mathbf{P}$.
  - Update one vertex of the base triangle.
- Termination: $\mathcal{N} \cap \mathbf{P} = \emptyset$, with $N(\mathbf{T}) = \mathbf{N}$.

An example of plane-probing algorithm on a plane

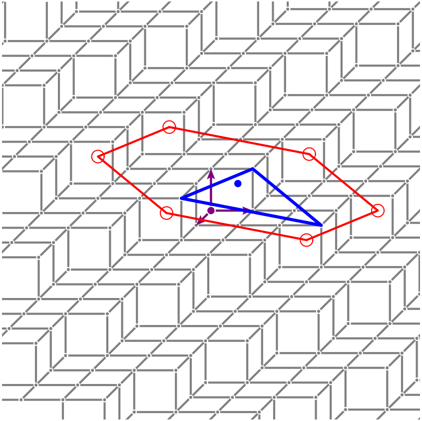
Normal vector: $(1, 2, 5)$



- A predicate **InPlane**: $x \in \mathbf{P}$?
- At each iteration:
  - Consider a **neighborhood** $\mathcal{N}$.
  - Choose a point from $\mathcal{N} \cap \mathbf{P}$.
  - Update one vertex of the base triangle.
- Termination: $\mathcal{N} \cap \mathbf{P} = \emptyset$, with $N(\mathbf{T}) = \mathbf{N}$.

An example of plane-probing algorithm on a plane

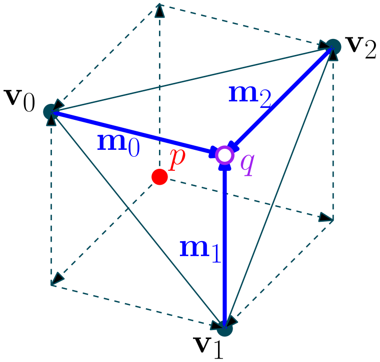
Normal vector: $(1, 2, 5)$



- A predicate **InPlane**: $x \in \mathbf{P}$?
- At each iteration:
  - Consider a **neighborhood** $\mathcal{N}$.
  - Choose a point from $\mathcal{N} \cap \mathbf{P}$.
  - Update one vertex of the base triangle.
- Termination: $\mathcal{N} \cap \mathbf{P} = \emptyset$, with $N(\mathbf{T}) = \mathbf{N}$.

Initialization

- Given $\mathbf{p} \in \mathbf{P}$
- Define $\mathbf{T}^{(0)} := (\mathbf{v}_k^{(0)})_{k\{0,1,2\}}$ such that $(\forall k)$ $\mathbf{v}_k^{(0)} := \mathbf{p} + \mathbf{e}_k + \mathbf{e}_{k+1} \in \mathbf{P}$, where $(\mathbf{e}_0, \mathbf{e}_1, \mathbf{e}_2)$ is the canonical basis.
- $\mathbf{q} := \mathbf{p} + \mathbf{e}_0 + \mathbf{e}_1 + \mathbf{e}_2 \notin \mathbf{P}$.
- $\forall k, \mathbf{m}_k^{(i)} = \mathbf{q} - \mathbf{v}_k^{(i)}$.



Neighborhood of existing plane-probing algorithms

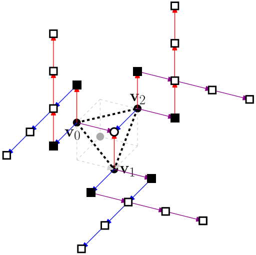
Let Σ be the set of all possible permutation of $\{0, 1, 2\}$.

- The H-algorithm explores a hexagonal neighborhood,

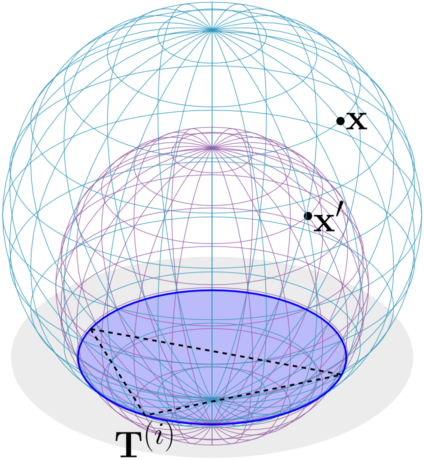
$$\mathcal{N}_H = \left\{ \mathbf{v}^{(i)}_{\sigma(0)} + \mathbf{m}^{(i)}_{\sigma(1)} \mid \sigma \in \Sigma \right\} \tag{3}$$

- The R-algorithm explores a neighborhood that includes 6 rays,

$$\mathcal{N}_R = \left\{ \mathbf{v}^{(i)}_{\sigma(0)} + \mathbf{m}^{(i)}_{\sigma(1)} + \lambda \mathbf{m}^{(i)}_{\sigma(2)} \mid \lambda \in \mathbb{N}; \sigma \in \Sigma \right\} \tag{4}$$



We favor a point that, with the three vertices of the triangle, defines a sphere that occupies the smallest area above the triangle.



We say $\mathbf{x}'$ is closer than $\mathbf{x}$ and we note $\mathbf{x}' \leq_{\mathbf{T}} \mathbf{x}$.

Plane-probing algorithms on digital planes

Input : The predicate $\mathrm{InPlane} :=$ "Is a point $\mathbf{x} \in \mathbf{P}$?", a point $\mathbf{p} \in \mathbf{P}$ and the type of neighborhood $\mathcal{N} \in \{\mathcal{N}_H, \mathcal{N}_R\}$

Output : A normal vector $\hat{\mathbf{N}}$

Function *normalEstimation()*:

$\mathbf{q} \leftarrow \mathbf{p} + (1,1,1)$; $\mathbf{T}^{(0)} \equiv (\mathbf{v}_k^{(0)})_{k \in \{0,1,2\}} \leftarrow (\mathbf{q} - \mathbf{e}_k)_{k \in \{0,1,2\}}$; $i \leftarrow 0$;

while $\mathcal{N}^{(i)} \cap \{\mathbf{x} \mid \mathrm{InPlane}(\mathbf{x})\} \neq \emptyset$ **do**

Let (k, α, β) be such that, for all $\mathbf{y} \in \mathcal{N}^{(i)} \cap \{\mathbf{x} \mid \mathrm{InPlane}(\mathbf{x})\}$, $\mathbf{v}_k^{(i)} + \alpha \mathbf{m}_{k+1}^{(i)} + \beta \mathbf{m}_{k+2}^{(i)} \leq_{\mathbf{T}^{(i)}} \mathbf{y}$,

$\mathbf{v}_k^{(i+1)} \leftarrow \mathbf{v}_k^{(i)} + \alpha \mathbf{m}_{k+1}^{(i)} + \beta \mathbf{m}_{k+2}^{(i)}$,

$\forall l \in \{0,1,2\} \setminus k,\ \mathbf{v}_l^{(i+1)} \leftarrow \mathbf{v}_l^{(i)}$ // $\mathbf{T}^{(i)}$ updated to $\mathbf{T}^{(i+1)}$

$i \leftarrow i + 1$;

$B \leftarrow \{\mathbf{v}_0^{(i)} - \mathbf{v}_1^{(i)}, \mathbf{v}_1^{(i)} - \mathbf{v}_2^{(i)}, \mathbf{v}_2^{(i)} - \mathbf{v}_0^{(i)}\}$;

Let $\mathbf{b}_1$ and $\mathbf{b}_2$ be the shortest and second shortest vectors of B ;

return $\mathbf{b}_1 \times \mathbf{b}_2$

Termination [4]

Invariants

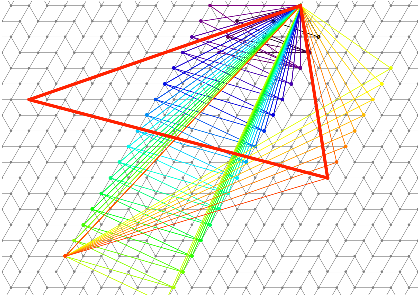
- Height of $\mathbf{m}_k$: For all $i \in \{0, \ldots, n\}$, $\forall k \in \mathbb{Z}/3\mathbb{Z}$, $\mathbf{m}_k^{(i)} \cdot \mathbf{N} > 0$.
- Unimodularity: For all $i \in \{0, \ldots, n\}$, $\det(\mathbf{m}_0^{(i)}, \mathbf{m}_1^{(i)}, \mathbf{m}_2^{(i)}) = 1$.

Termination

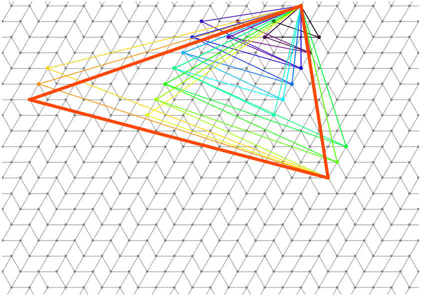
- The number of iterations is $O(\|\mathbf{N}\|_1)$.
- If $\mathbf{p} \cdot \mathbf{N} = 0$, the normal vector of the last triangle $\mathbf{T}^{(n)}$ is equal to $\mathbf{N}$.

[4] Jacques-Olivier Lachaud, X. Provençal, and Tristan Roussillon. "Two Plane-Probing Algorithms for the Computation of the Normal Vector to a Digital Plane". In: *J. Math. Imaging Vis.* 59.1 (2017), pp. 23–39

Locality



(a) H-algorithm



(b) R-algorithm

Existing plane-probing algorithms

We note R1 an optimized version of R.

Comparison among H, R and R1

	H	R	R1
Neighborhood size	$\lvert\mathcal{N}_H\rvert = 6$	$\lvert\mathcal{N}_R\rvert > 6$	$\lvert\mathcal{N}_{R1}\rvert = \lvert\mathcal{N}_R\rvert$
Complexity	$O(\lVert\mathbf{N}\rVert_1)$	$O(\lVert\mathbf{N}\rVert_1 \log \lVert\mathbf{N}\rVert_1)$	$O(\lVert\mathbf{N}\rVert_1)$
The last triangle is always not obtuse	✗	✓$_{exp.}$	✓$_{exp.}$

Table of Contents

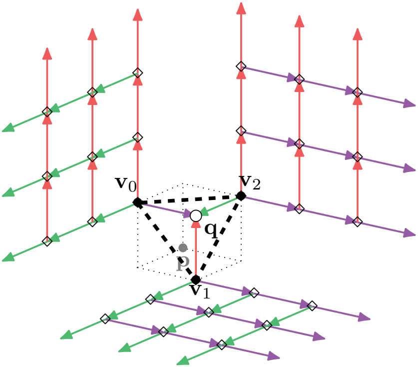
1 Introduction
- Context
- Plane-probing algorithms

2 L-algorithm
- L-Neighborhood
- Complexity
- Delaunay Property

3 Normal vector estimation
- Estimation and evaluation
- Convergence on selected surfels
- Comparing plane-probing with other methods

4 Conclusion

Neighborhood

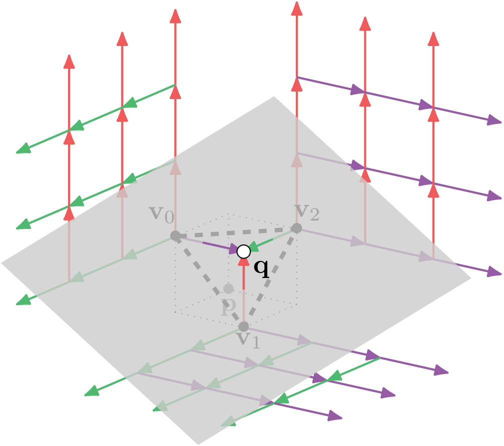


$$\mathcal{N}_L = \left\{ \mathbf{v}_{\sigma(0)}^{(i)} + \alpha \mathbf{m}_{\sigma(1)}^{(i)} + \beta \mathbf{m}_{\sigma(2)}^{(i)} \mid \alpha, \beta \in \mathbb{N}; \sigma \in \Sigma \right\} \quad (5)$$

Lemma

The set $\mathcal{N}_L \cap \mathbf{P}$ is finite.

Idea: For all $i \in \{0, \ldots, n\}$, $\forall k \in \mathbb{Z}/3\mathbb{Z}$, $\mathbf{m}_k^{(i)} \cdot \mathbf{N} > 0$.



Plane-probing algorithms on digital planes

Input : The predicate $\mathrm{InPlane} :=$ "Is a point $\mathbf{x} \in \mathbf{P}$?", a point $\mathbf{p} \in \mathbf{P}$ and the type of neighborhood $\mathcal{N} \in \{\mathcal{N}_H, \mathcal{N}_R, \mathcal{N}_L\}$

Output : A normal vector $\hat{\mathbf{N}}$

Function *normalEstimation()*:

- $\mathbf{q} \leftarrow \mathbf{p} + (1,1,1)$; $\mathbf{T}^{(0)} \equiv (\mathbf{v}_k^{(0)})_{k \in \{0,1,2\}} \leftarrow (\mathbf{q} - \mathbf{e}_k)_{k \in \{0,1,2\}}$; $i \leftarrow 0$;
- **while** $\mathcal{N}^{(i)} \cap \{\mathbf{x} \mid \mathrm{InPlane}(\mathbf{x})\} \neq \emptyset$ **do**
  - Let (k, α, β) be such that, for all $\mathbf{y} \in \mathcal{N}^{(i)} \cap \{\mathbf{x} \mid \mathrm{InPlane}(\mathbf{x})\}$, $\mathbf{v}_k^{(i)} + \alpha \mathbf{m}_{k+1}^{(i)} + \beta \mathbf{m}_{k+2}^{(i)} \leq_{\mathbf{T}^{(i)}} \mathbf{y}$,
  - $\mathbf{v}_k^{(i+1)} \leftarrow \mathbf{v}_k^{(i)} + \alpha \mathbf{m}_{k+1}^{(i)} + \beta \mathbf{m}_{k+2}^{(i)}$,
  - $\forall l \in \{0,1,2\} \setminus k,\ \mathbf{v}_l^{(i+1)} \leftarrow \mathbf{v}_l^{(i)}$ `// T^(i) updated to T^(i+1)`
  - $i \leftarrow i + 1$;
- $B \leftarrow \{\mathbf{v}_0^{(i)} - \mathbf{v}_1^{(i)}, \mathbf{v}_1^{(i)} - \mathbf{v}_2^{(i)}, \mathbf{v}_2^{(i)} - \mathbf{v}_0^{(i)}\}$;
- Let $\mathbf{b}_1$ and $\mathbf{b}_2$ be the shortest and second shortest vectors of B ;
- **return** $\mathbf{b}_1 \times \mathbf{b}_2$

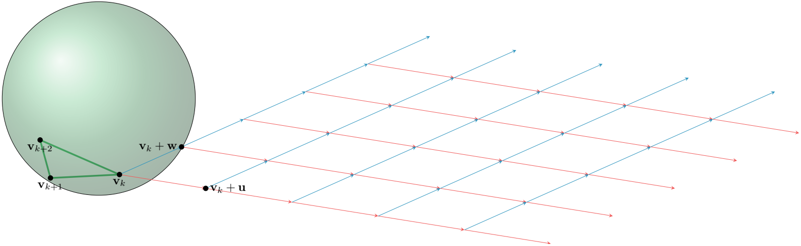
Comparison among H, R1 and L

	H	R1	L
Neighborhood size	$\lvert\mathcal{N}_H\rvert = 6$	$\lvert\mathcal{N}_R\rvert > 6$	$\lvert\mathcal{N}_L\rvert > \lvert\mathcal{N}_R\rvert$
Complexity	$O(\lVert\mathbf{N}\rVert_1)$	$O(\lVert\mathbf{N}\rVert_1)$	?
The last triangle is always not obtuse	✗	✓$_{exp.}$	?

Reduction of candidate set

Lemma

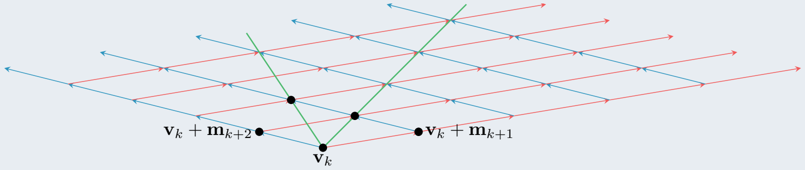
For all $k \in \mathbb{Z}/3\mathbb{Z}$, let Λ_k be the set $\{\mathbf{v}_k + \alpha\mathbf{u} + \beta\mathbf{w} \mid (\alpha, \beta) \in \mathbb{N}^2 \setminus \{0\}\}$, where $\mathbf{u}$, $\mathbf{w}$ are any two non-zero vectors of $\mathbb{Z}^3$ such that $\mathbf{v}_k + \mathbf{u}, \mathbf{v}_k + \mathbf{w} \in \mathcal{H}_+$. If $\mathbf{u} \cdot \mathbf{w} \geq 0$, we have either $\mathbf{v}_k + \mathbf{u} \leq_{\mathbf{T}} \mathbf{x}$ for all $\mathbf{x} \in \Lambda_k$ or $\mathbf{v}_k + \mathbf{w} \leq_{\mathbf{T}} \mathbf{x}$ for all $\mathbf{x} \in \Lambda_k$.



Reduction of candidate set

Smaller candidate set

We partition the neighborhood into sectors with acute angle. Instead of exploring an infinite lattice, the L-algoithm only needs to look into a finite set of representative points (•) of these sections.



Complexity

Complexity for the search of updated point

The number of calls to predicate $\mathrm{InPlane}$ at each iteration is in $O(\log(\|\mathbf{N}\|_1)$. [5]

[5] Jui-Ting Lu, Tristan Roussillon, and David Coeurjolly. "A New Lattice-based Plane-probing Algorithm". en. In: *Second International Conference on Discrete Geometry and Mathematical Morphology (DGMM 2022)*. Ed. by Etienne Baudrier et al. Lecture Notes in Computer Science. Springer Verlag, 2022

Complexity

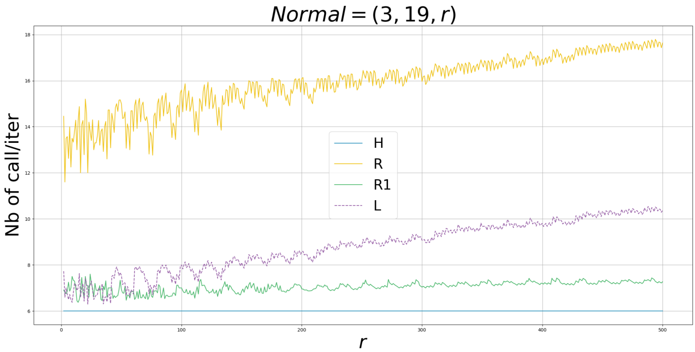


Figure: Number of calls to predicate **per iteration** for normal vectors of form $\{(3, 19, r), 1 \leq r \leq 500\}$.

Complexity

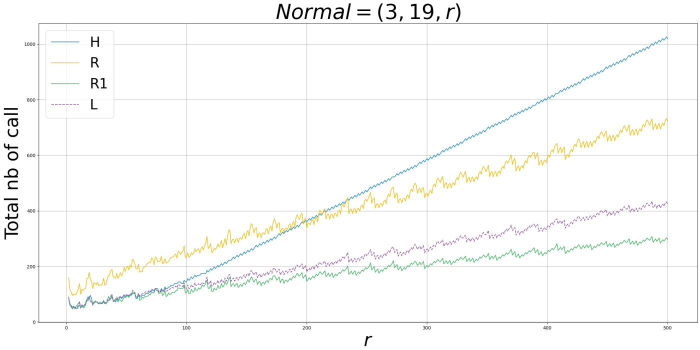
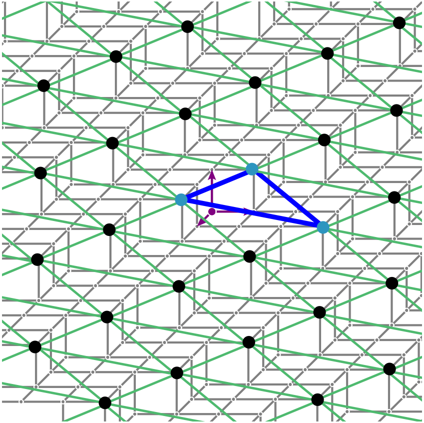


Figure: Number of calls to predicate **in total** for normal vectors of form $\{(3, 19, r), 1 \leq r \leq 500\}$.

Comparison among H, R1 and L

	H	R1	L
Neighborhood size	$\lvert\mathcal{N}_H\rvert = 6$	$\lvert\mathcal{N}_R\rvert > 6$	$\lvert\mathcal{N}_L\rvert > \lvert\mathcal{N}_R\rvert$
Complexity	$O(\lVert\mathbf{N}\rVert_1)$	$O(\lVert\mathbf{N}\rVert_1)$	$O(\lVert\mathbf{N}\rVert_1 \log \lVert\mathbf{N}\rVert_1)$
The last triangle is always not obtuse	✗	✓$_{exp.}$	?

Delaunay triangulation



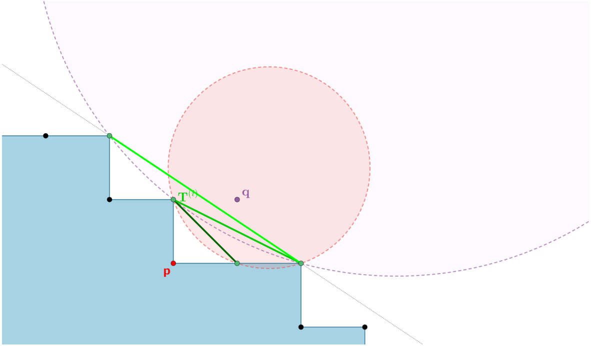
When the last triangle is non-obtuse, it belongs to the Delaunay triangulation of the lattice $\{\mathbf{x} \in \mathbb{Z}^3 \mid \mathbf{x} \cdot \mathbf{N} = \|\mathbf{N}\|_1 - 1\}$.

The Delaunay Property

For all $i \in \{0, \ldots, n\}$, let $\mathcal{B}^{(i)}$ be the ball uniquely determined by the four distinct points of two consecutive triangles.

Theorem: Delaunay property for L-algorithm

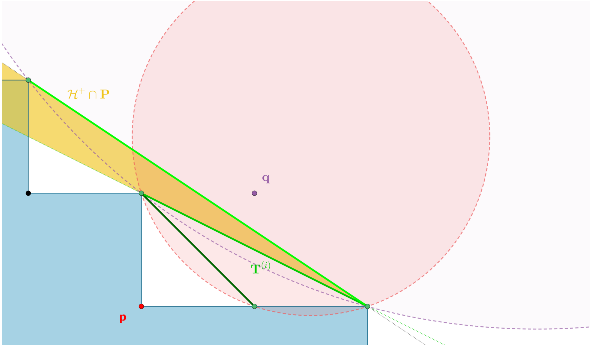
For all $i \in \{0, \ldots, n\}$, the ball $\mathcal{B}^{(i)}$ does not contain any point of $\mathbf{P}$ in its interior.



Let $\mathcal{H}_+^{(i)}$ be the half-space delimited by $\mathbf{T}^{(i)}$ that includes $\mathbf{q}$.

Lemma

For all $i \in \{0, \dots, n-1\}$, if the interior of $\mathcal{B}^{(i)}$ contains no point of $\mathbf{P}$, then the interior of $\mathcal{B}^{(i+1)}$ contains no point of $\mathbf{P} \cap \mathcal{H}_+^{(i)}$.



Outline of the proof

Since $(\mathbf{m}_0, \mathbf{m}_1, \mathbf{m}_2)$ is unimodular, we can represent all point of $\mathbb{Z}^3$ as

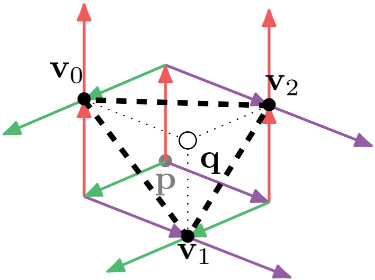
$$\mathbf{x} = \mathbf{p} + \sum_k c_k \mathbf{m}_k.$$

Points that lie in the plane that passes $\mathbf{T}^{(i)}$ satisfy:

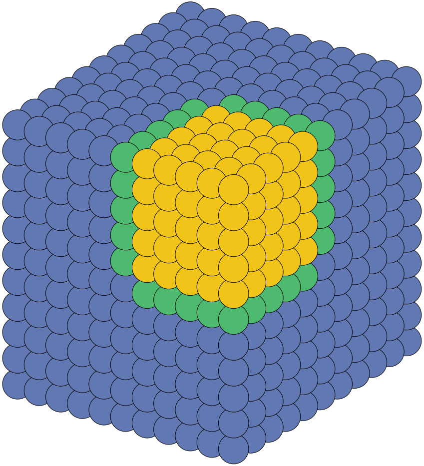
$$\sum_k c_k = 2.$$

Then, points that lies in $\mathcal{H}_+^{(i)}$ satisfy:

$$\sum_k c_k \geq 2.$$

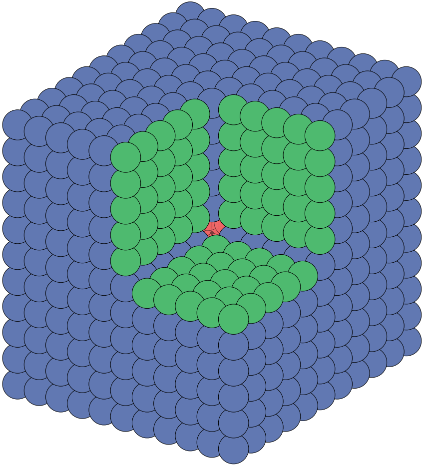


$\{\mathbf{p} + \sum_k c_k \mathbf{m}_k\} \sim \mathbb{Z}^3$
can be divided into:



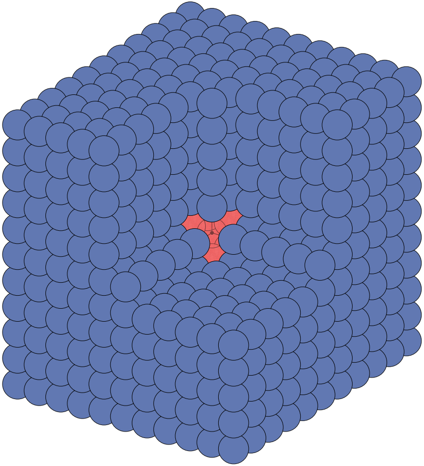
1. the points are not in the plane so not considered by the lemma.
2. these points are exactly the ones probed in the L-algorithm
3. these points are farthest than specific green points.
4. the red one is discarded because none of its points lie in $\mathcal{H}_+^{(i)}$

$\{\mathbf{p} + \sum_k c_k \mathbf{m}_k\} \sim \mathbb{Z}^3$
can be divided into:



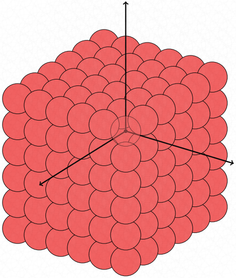
1. the points are not in the plane so not considered by the lemma.
2. these points are exactly the ones probed in the L-algorithm
3. these points are farthest than specific green points.
4. the red one is discarded because none of its points lie in $\mathcal{H}_+^{(i)}$

$\{\mathbf{p} + \sum_k c_k \mathbf{m}_k\} \sim \mathbb{Z}^3$
can be divided into:

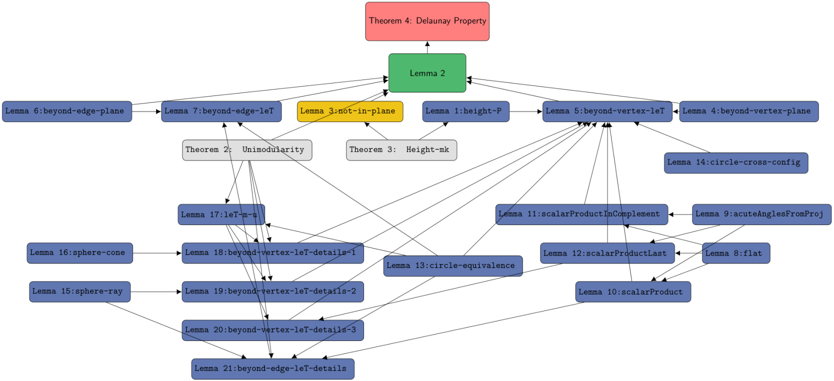


1. the points are not in the plane so not considered by the lemma.
2. these points are exactly the ones probed in the L-algorithm
3. these points are farthest than specific green points.
4. the red one is discarded because none of its points lie in $\mathcal{H}_+^{(i)}$

$\{\mathbf{p} + \sum_k c_k \mathbf{m}_k\} \sim \mathbb{Z}^3$
can be divided into:

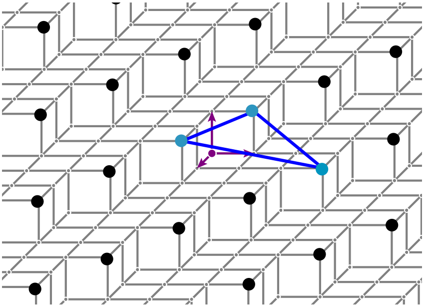


1. the points are not in the plane so not considered by the lemma.
2. these points are exactly the ones probed in the L-algorithm
3. these points are farthest than specific green points.
4. the red one is discarded because none of its points lie in $\mathcal{H}_+^{(i)}$


Theorem 4: Delaunay Property
Lemma 2
Lemma 6:beyond-edge-plane
Lemma 7:beyond-edge-leT
Lemma 3:not-in-plane
Lemma 1:height-P
Lemma 5:beyond-vertex-leT
Lemma 4:beyond-vertex-plane
Theorem 2: Unimodularity
Theorem 3: Height-mk
Lemma 14:circle-cross-config
Lemma 17:leT-m-u
Lemma 11:scalarProductInComplement
Lemma 9:acuteAnglesFromProj
Lemma 16:sphere-cone
Lemma 18:beyond-vertex-leT-details-1
Lemma 13:circle-equivalence
Lemma 12:scalarProductLast
Lemma 8:flat
Lemma 15:sphere-ray
Lemma 19:beyond-vertex-leT-details-2
Lemma 10:scalarProduct
Lemma 20:beyond-vertex-leT-details-3
Lemma 21:beyond-edge-leT-details

Consequence 1: Last triangle

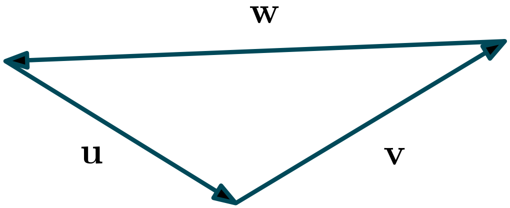
The last triangle returned by the L-algorithm is always acute or straight.



Consequence 2: Minimal basis

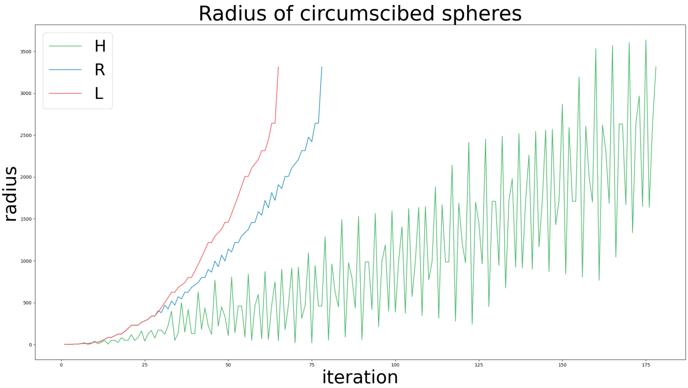
Let L be a rank-two integral lattice. A basis $(\mathbf{x}, \mathbf{y})$ of L is minimal if and only if $\|\mathbf{x}\|_2, \|\mathbf{y}\|_2 \leq \|\mathbf{x} - \mathbf{y}\|_2 \leq \|\mathbf{x} + \mathbf{y}\|_2$, where $\| \cdot \|_2$ denotes the Euclidean norm.

The two shortest edges of the final triangle form a minimal basis of the lattice $\{\mathbf{x} \in \mathbb{Z}^3 \mid \mathbf{x} \cdot \mathbf{N} = \|\mathbf{N}\|_1 - 1\}$.



Consequence 3: Increasing ball radius

$$\mathbf{N} = (198, 195, 193)$$



Evolution of basis of the tetrahedron $(\mathbf{N} = (2, 5, 156))$

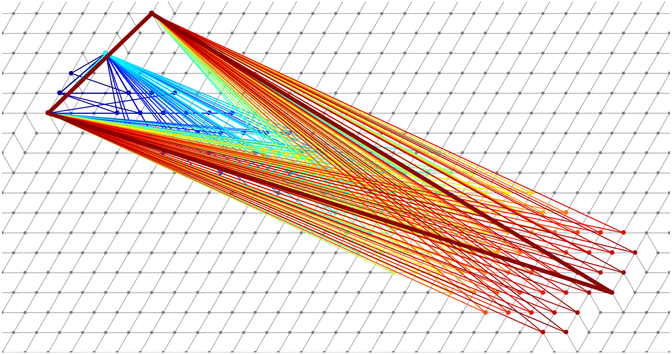


Figure: H-algorithm

Evolution of basis of the tetrahedron $(\mathbf{N} = (2, 5, 156))$

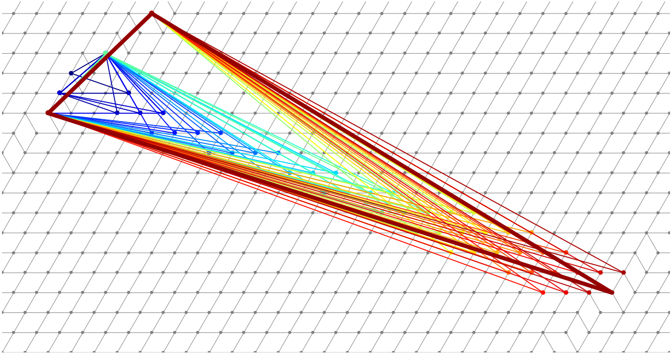


Figure: R-algorithm

Evolution of basis of the tetrahedron ($\mathbf{N} = (2, 5, 156)$)

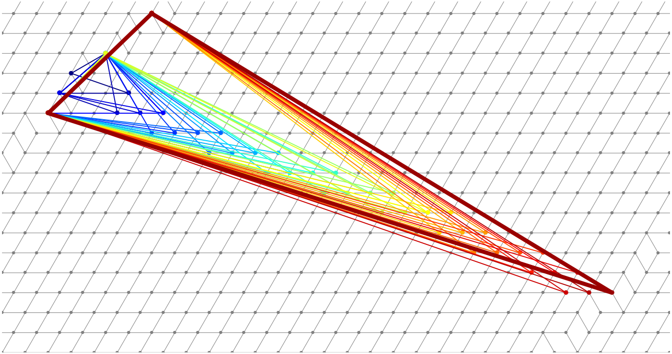


Figure: L-algorithm

Summary

Comparison among H, R1 and L

	H	R1	L
Neighborhood size	$\lvert\mathcal{N}_H\rvert = 6$	$\lvert\mathcal{N}_R\rvert > 6$	$\lvert\mathcal{N}_L\rvert > \lvert\mathcal{N}_R\rvert$
Complexity	$O(\lVert\mathbf{N}\rVert_1)$	$O(\lVert\mathbf{N}\rVert_1)$	$O(\lVert\mathbf{N}\rVert_1 \log \lVert\mathbf{N}\rVert_1)$
The Delaunay property	✗	✗	✓
The last triangle is always not obtuse	✗	✓ $_{exp.}$	✓

Table of Contents

1 Introduction
- Context
- Plane-probing algorithms

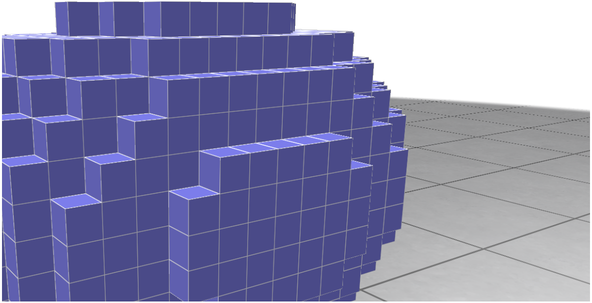
2 L-algorithm
- L-Neighborhood
- Complexity
- Delaunay Property

3 Normal vector estimation
- Estimation and evaluation
- Convergence on selected surfels
- Comparing plane-probing with other methods

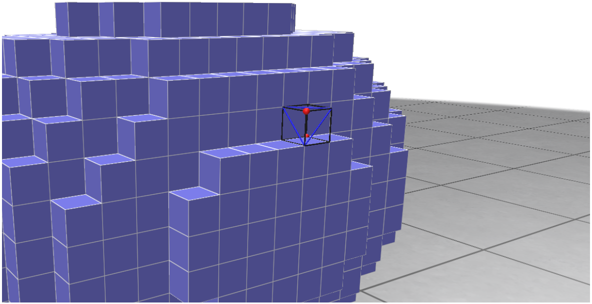
4 Conclusion

What happens when launching a plane-probing algorithm on a digital surface ?

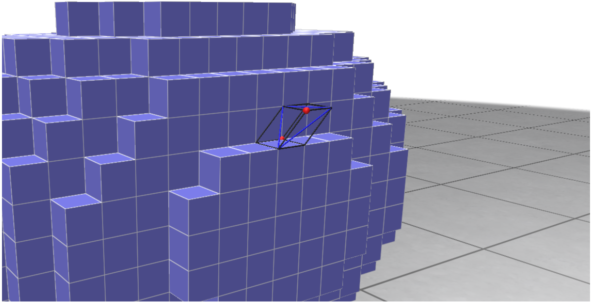
Parallepiped-based plane-probing algorithm



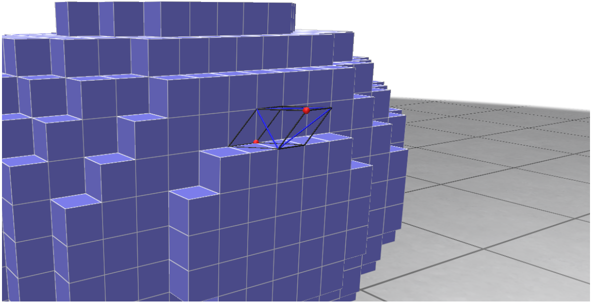
Parallepiped-based plane-probing algorithm



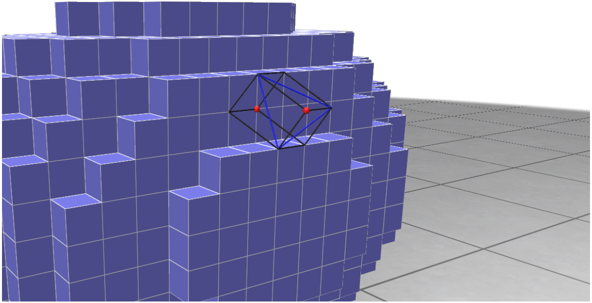
Parallepiped-based plane-probing algorithm



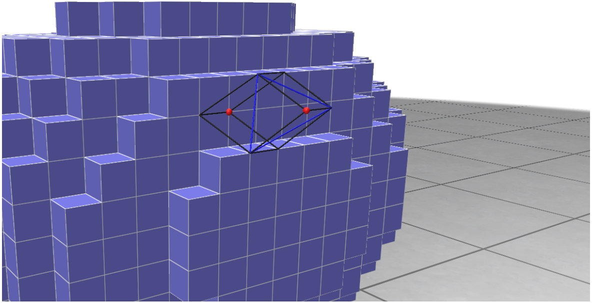
Parallepiped-based plane-probing algorithm



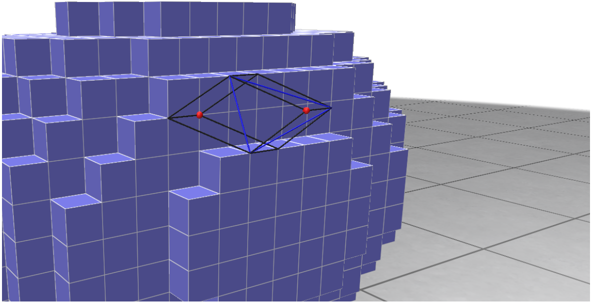
Parallepiped-based plane-probing algorithm



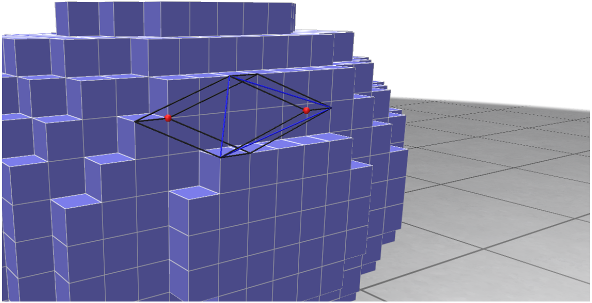
Parallepiped-based plane-probing algorithm



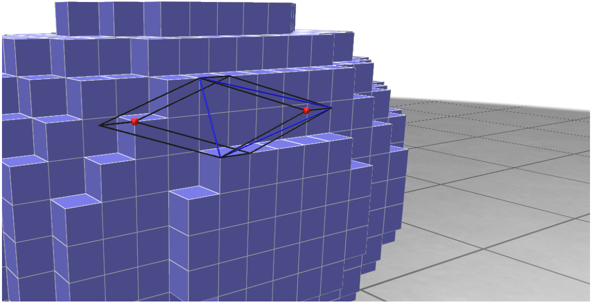
Parallepiped-based plane-probing algorithm



Parallepiped-based plane-probing algorithm



Parallepiped-based plane-probing algorithm



Plane-probing algorithms on digital surfaces

Input : The predicate $\mathrm{InSurface} :=$ "Is a point $\mathbf{x} \in \mathbf{S}$?" , an octant $\mathbf{s} \in \{\pm1, \pm1, \pm1\}$, a point $\mathbf{p} \in \mathbf{S}$ and the type of neighborhood $\mathcal{N} \in \{\mathcal{N}_H, \mathcal{N}_R, \mathcal{N}_L\}$

Output : A normal vector $\hat{\mathbf{N}}$.

```
Function normalEstimation():
    q ← p + s ; T^(0) ≡ (v_k^(0))_{k∈Z/3Z} ← (q − e_k)_{k∈Z/3Z} ;
    i ← 0 ;
    while N^(i) ∩ {x | InSurface(x)} ≠ ∅ do
        compute T^(i+1), updated copy of T^(i) ;
        i ← i + 1 ;
    B ← {v_0^(i) − v_1^(i), v_1^(i) − v_2^(i), v_2^(i) − v_0^(i)} ;
    Let b_1 and b_2 be the shortest and second shortest vectors of B ;
    return b_1 × b_2
```

$$\mathbf{q} \leftarrow \mathbf{p} + \mathbf{s} \; ; \; \mathbf{T}^{(0)} \equiv (\mathbf{v}_k^{(0)})_{k \in \mathbb{Z}/3\mathbb{Z}} \leftarrow (\mathbf{q} - \mathbf{e}_k)_{k \in \mathbb{Z}/3\mathbb{Z}}$$

$$B \leftarrow \{\mathbf{v}_0^{(i)} - \mathbf{v}_1^{(i)}, \mathbf{v}_1^{(i)} - \mathbf{v}_2^{(i)}, \mathbf{v}_2^{(i)} - \mathbf{v}_0^{(i)}\}$$

Plane-probing algorithms on digital surfaces

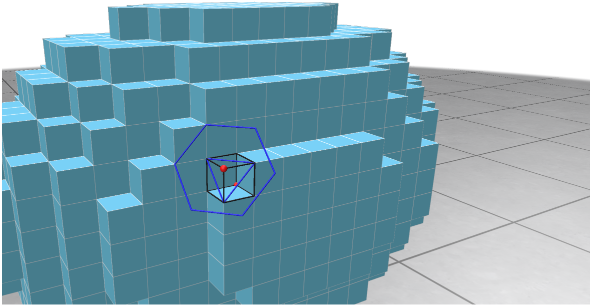
Input : The predicate NotAbove, an octant $\mathbf{s} \in \{\pm 1, \pm 1, \pm 1\}$, a point $\mathbf{p} \in \mathbf{S}$ and the type of neighborhood $\mathcal{N} \in \{\mathcal{N}_H, \mathcal{N}_R, \mathcal{N}_L\}$

Output : A normal vector $\hat{\mathbf{N}}$.

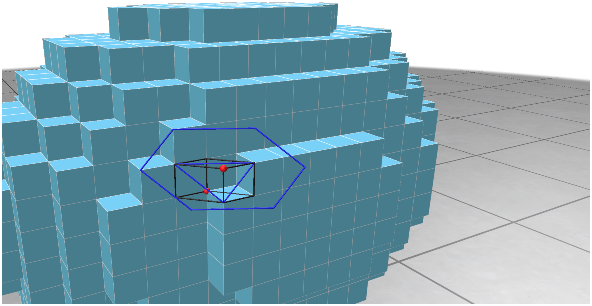
Function *normalEstimation()*:

- $\mathbf{q} \leftarrow \mathbf{p} + \mathbf{s}$; $\mathbf{T}^{(0)} \equiv (\mathbf{v}_k^{(0)})_{k \in \mathbb{Z}/3\mathbb{Z}} \leftarrow (\mathbf{q} - \mathbf{e}_k)_{k \in \mathbb{Z}/3\mathbb{Z}}$;
- $i \leftarrow 0$;
- **while** $\mathcal{N}^{(i)} \cap \{\mathbf{x} \mid \mathrm{NotAbove}(\mathbf{x})\} \neq \emptyset$ **do**
  - compute $\mathbf{T}^{(i+1)}$, updated copy of $\mathbf{T}^{(i)}$;
  - **if** $\mathrm{Card}(\{\mathbf{x} \in \mathbf{\Pi}^{(i)} \mid \mathrm{NotAbove}(\mathbf{x})\}) < 4$ **then**
    - compute $\mathbf{q}^{(i+1)}$, translated copy of $\mathbf{q}^{(i)}$;
  - $i \leftarrow i + 1$;
- $B \leftarrow \{\mathbf{v}_0^{(i)} - \mathbf{v}_1^{(i)}, \mathbf{v}_1^{(i)} - \mathbf{v}_2^{(i)}, \mathbf{v}_2^{(i)} - \mathbf{v}_0^{(i)}\}$;
- Let $\mathbf{b}_1$ and $\mathbf{b}_2$ be the shortest and second shortest vectors of B ;
- **return** $\mathbf{b}_1 \times \mathbf{b}_2$

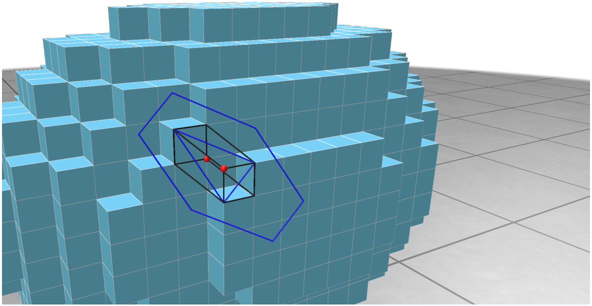
Example with fixed q



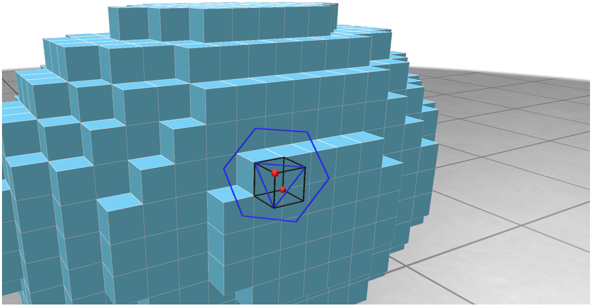
Example with fixed q



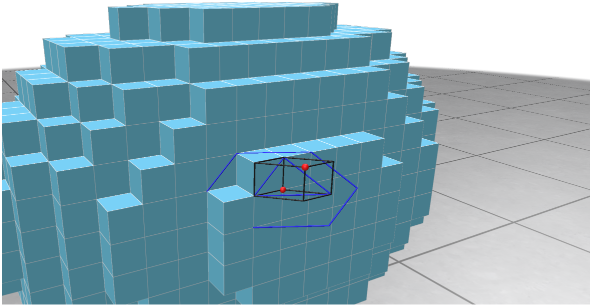
Example with fixed q



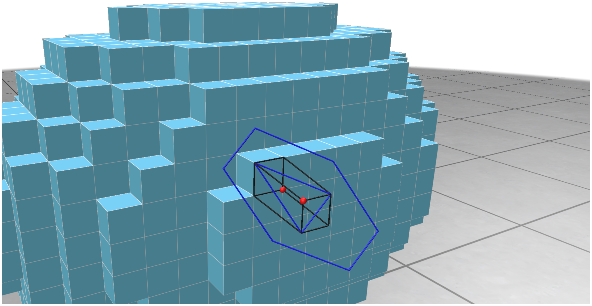
Example with moving q



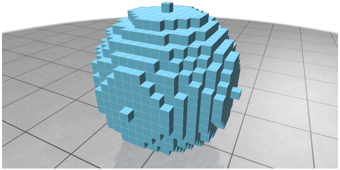
Example with moving $\mathbf{q}$



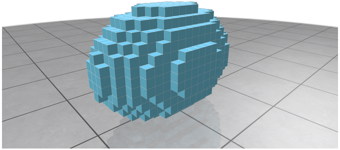
Example with moving q



Equations of Euclidean shapes for experimental evaluation



(a) sphere9



(b) ellipsoid

Figure: Visualization when $h = 1$

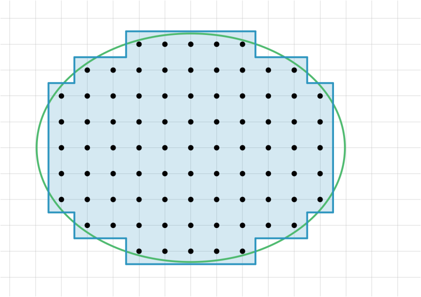
Shape	DGtal name	Equation
Sphere	sphere9	$x^2 + y^2 + z^2 - 9^2 = 0$
Ellipse	ellipse	$90 - 3x^2 - 2y^2 - z^2 = 0$

Gauss digitization

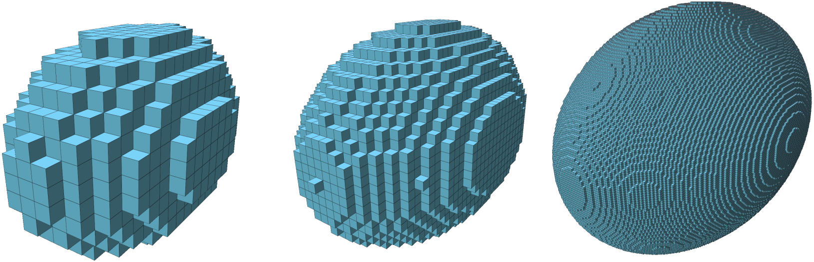
Definition

Let $X \subset \mathbb{R}^3$ be a compact simply connected volume. The *Gauss digitization* of X with grid step $h > 0$, denoted by $\mathbf{DS}(X)_h$, is the set of all discete points $p \in h(\mathbb{Z}^3)$ lying in X. In other words,

$$\mathbf{DS}(X)_h = X \cap h(\mathbb{Z}^3). \qquad (6)$$

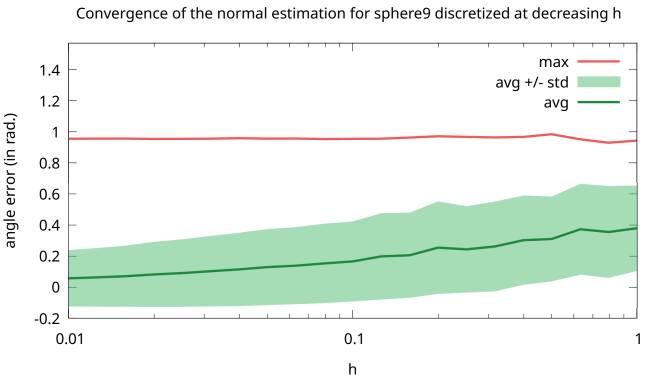


Multigrid convergence

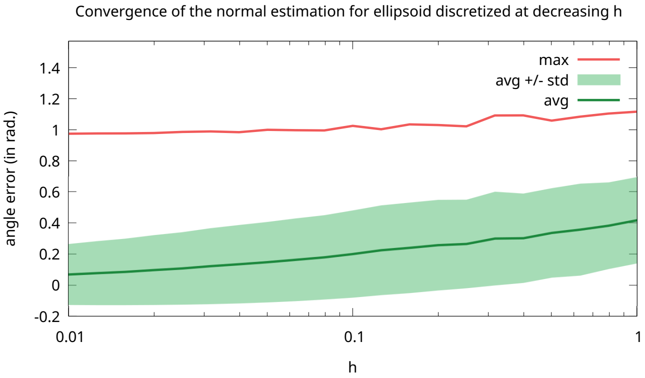


(a) $h = 1$ (b) $h = 0.5$ (c) $h = 0.1$

Plane-probing algorithm



Plane-probing algorithm



Selection of starting surfels

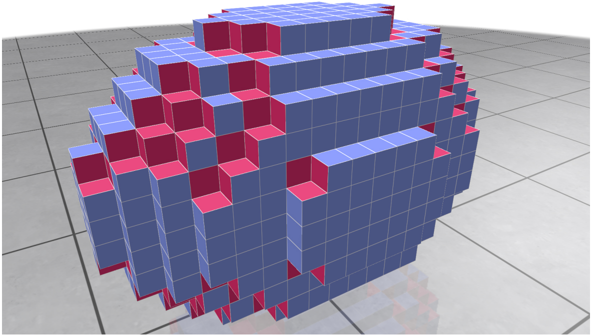
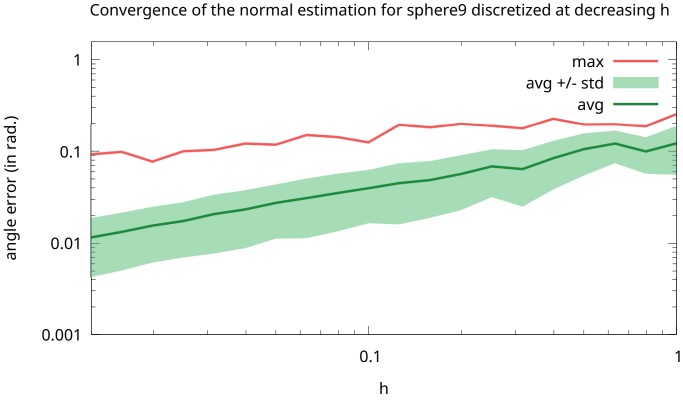


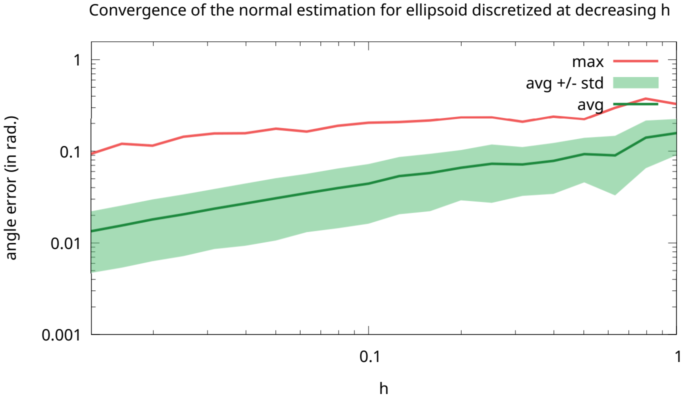
Figure: Illustration of the reentrant corners (red) on a digital ellipsoid.

1. A reentrant corner exists in the initial parallelepiped (unique octant).
2. The point $\mathbf{q}^{(i)}$ does not translate throughout iterations.

Multigrid convergence on selected surfels



Multigrid convergence on selected surfels



Digital methods for normal estimations

There are three methods that are multigrid convergent:

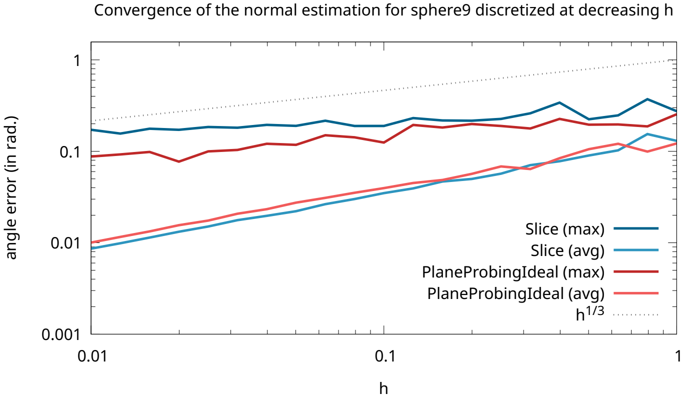
- *Slice method*[6] computes the cross product of two 2D tangent vectors of two perpendicular slices.
- *Voronoi covariance measure*[7] is a covariance tensor that requires two parameters: offset radius and kernel radius.
- *Integral invariant*[8] compute integrals on the intersection between the shape and the convolution kernel, that has a specific radius.

[6] Lachaud and Vialard, "Geometric Measures on Arbitrary Dimensional Digital Surfaces".

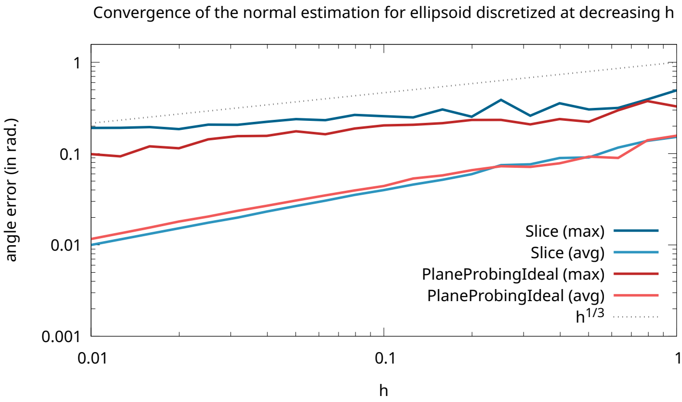
[7] Cuel et al., "Robust Geometry Estimation Using the Generalized Voronoi Covariance Measure".

[8] Levallois, Coeurjolly, and Lachaud, "Parameter-free and Multigrid Convergent Digital Curvature Estimators".

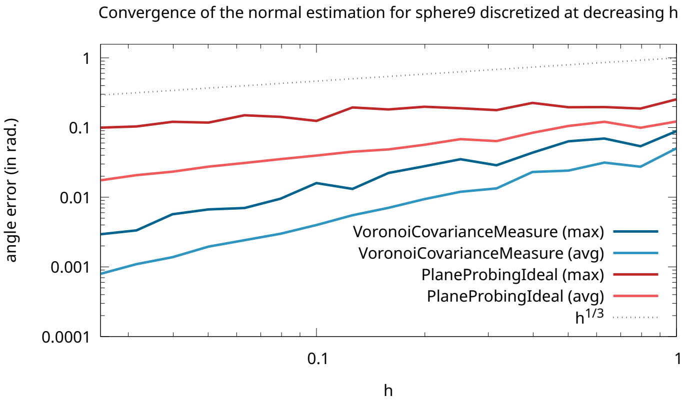
Slice method



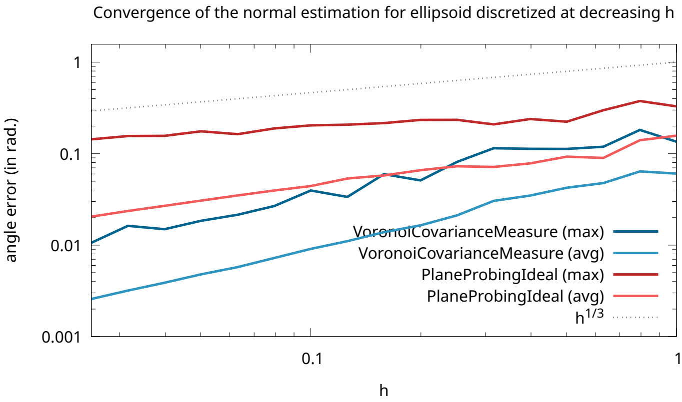
Slice method



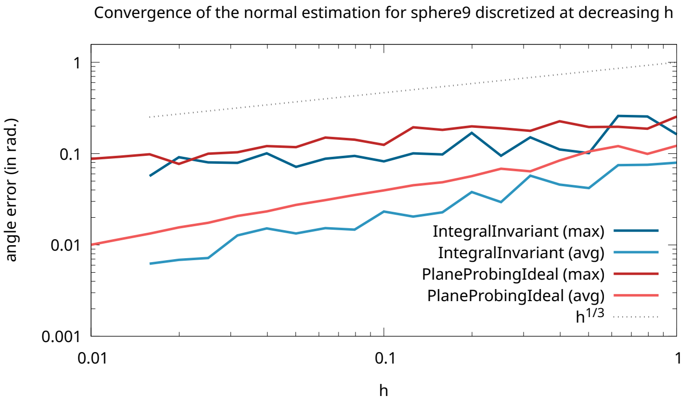
Voronoi Covariance Measure (VCM)



Voronoi Covariance Measure (VCM)



Integral Invariant



Integral Invariant

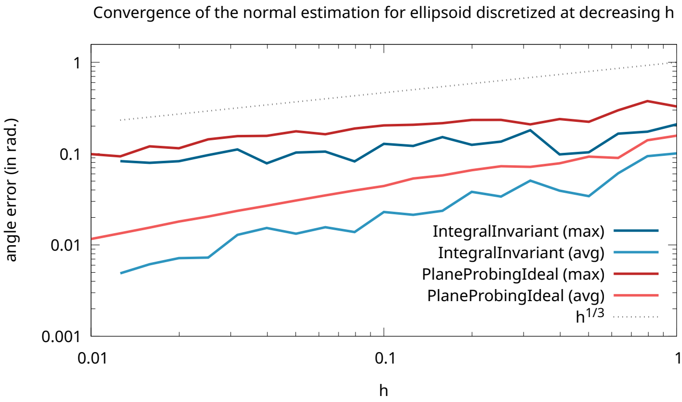


Table of Contents

1 Introduction
- Context
- Plane-probing algorithms

2 L-algorithm
- L-Neighborhood
- Complexity
- Delaunay Property

3 Normal vector estimation
- Estimation and evaluation
- Convergence on selected surfels
- Comparing plane-probing with other methods

4 Conclusion

A new plane-probing algorithm

We introduce a new plane-probing algorithm, the L-algorithm.

- If $\mathbf{p} \cdot \mathbf{N} = 0$, it return the exact normal vector and stops in $O(\|\mathbf{N}\|_1)$ steps.
- It requires theoretically $O(\|\mathbf{N}\|_1 \log \|\mathbf{N}\|_1)$ calls to predicate in total.

Geometrical properties:

- The Delaunay property: the interior of $\mathcal{B}^{(i)}$ does not include any point of $\mathbf{P}$.
- The sequence of radius of the balls $\mathcal{B}^{(i)}$ is non decreasing.
- Always returns a non obtuse triangle (experimentally verified for R-algorithm).

Perspectives for plane-probing algorithms

- Understand why R-algorithm also returns a non obtuse triangle.
- Improve the complexity of the L-algorithm.
- Other approaches, such as: select multiple points to update at each iteration.

Normal vector estimation

A parallelepipedic version of the L-algorithm can be applied for normal vector estimation on digital surfaces:

- The current method does not converge on all surfels
- On some selected surfels, there is convergence.

Future work:

- Provide a proof for the multigrid convergence on selected surfels:
- Identify the surfels where there is no convergence.
- Study the estimation of normal vectors on other shapes (for example, non convex).