Gift-Wrapping Based Preimage Computation Algorithm

Yan Gerard[1], Fabien Feschet[1], and David Coeurjolly[2]

[1] Univ. Clermont 1, LAIC, Campus des Cézeaux,63172 Aubière, France
{gerard,feschet}@laic.u-clermont1.fr
[2] LIRIS, Univ. Lyon 1, Bât Nautibus, 69622 Villeurbanne Cedex, France
david.coeurjolly@liris.cnrs.fr

Abstract. The aim of the paper is to define an algorithm for computing preimages - roughly the sets of naive digital planes containing a finite subset S of $\mathbb{Z}^3$. The method is based on theoretical results: the preimage is a polytope that vertices can be decomposed in three subsets, the upper vertices, the lower vertices and the intermediary ones (equatorial). We provide a geometrical understanding (as facets on S or $S \ominus S$) of each kind of vertices. These properties are used to compute the preimage by gift-wrapping some regions of the convex hull of S or of $S \ominus S \cup \{(0,0,1)\}$.

1 Introduction

Digital straightness is an important concept in computer vision. In dimension two, for nearly half a century many digital straight line characterizations have been proposed with interactions with many fields such as arithmetic or theory of words (refer to [1] for a survey on digital straight line). A convenient framework is to consider the set of Euclidean straight lines, so-called *preimage* whose digitizations contain the input set of pixels. Based on a parametrization of digital straight lines, the preimage can be simply defined and correspond to a convex polygon in a given parameter space [2,3,4]. An important result is that such a preimage has got an important arithmetical structure that limits to four the number of vertices. This result is useful for a better understanding of this simple digital object and thus to design efficient digital straight line recognition algorithms.

In dimension 3, the same approach leads to define the notion of digital plane (see [5] or a survey) and of preimage [6,7,8]. The preimage becomes however more complex than a single polytope with 4 vertices. Questions about its arithmetical structure are open: even if it is a convex polyhedron in the digital plane parameter space, we have some difficulties to clearly understand its arithmetical structure and to bound the number of its vertices [9,10].

In this paper, we focus on a geometrical interpretation of preimage vertices and facets in order to design a fast preimage computation algorithm. The proposed algorithm is based on the computation of a surface of the convex hull of S and of the chords set $S \ominus S \cup \{(0,0,1)\}$ (as already used in [11] for recognition).

D. Coeurjolly et al. (Eds.): DGCI 2008, LNCS 4992, pp. 310–321, 2008.

The first sections are dedicated to the preliminaries and to the analyse of the preimage geometry. Then Section 4 details the proposed algorithm. Finally, Section 5 presents some experiments.

2 Preliminaries

2.1 Digital Planes and Digital Plane Recognition Problems

We start by defining naive digital planes according to J.P Reveilles definition[12]. We recall that for any vector $x \in \mathbb{R}^d$, its uniform norm is $||x||_\infty = \max\{|x_i|/1 \leq i \leq d\}$.

Definition 1. *A naive digital plane is a subset of* $\mathbb{Z}^3$ *characterized by a double inequality* $h \leq ax + by + cz < h + ||(a,b,c)||_\infty$ *where the normal vector* $(a,b,c) \in \mathbb{R}^3$ *is different from* $(0,0,0)$ *and where* $h \in \mathbb{R}$.

For a survey on digital plane characterization and alternative definitions, see [5].

According to the value of $||(a,b,c)||_\infty$, there exist three classes of naive digital planes obtained by rotation of the coordinates. In the following, we focus our attention on the special case where $||(a,b,c)||_\infty = |c|$. We introduce therefore the notion of z-slice which is related to naive digital planes with $|a| \leq |c|$ and $|b| \leq |c|$ while x-slices and y-slices are the equivalent objects corresponding to naive digital planes verifying $|a| = ||(a,b,c)||_\infty$ and $|b| = ||(a,b,c)||_\infty$ respectively.

Definition 2. *A z-slice is a subset of* $\mathbb{Z}^3$ *characterized by a double inequality* $h \leq ax+by+cz < h+|c|$ *where c is in* $\mathbb{R}^\star$ *and where a, b and h are real numbers.*

If $|c|$ is greater than $|a|$ and $|b|$, the z-slice of double-inequality $h \leq ax+by+cz < h + |c|$ is a naive digital plane. Otherwise (if $|c| < |a|$ or $|c| < |b|$), the z-slice of double-inequality $h \leq ax + by + cz < h + |c|$ is a subset of the naive digital plane $h \leq ax+by+cz < h+||(a,b,c)||_\infty$. Thus in any case, a z-slice is contained in a naive digital plane and conversely, a naive digital plane is either an x-slice, either a y-slice or a z-slice. It means obviously that naive digital planes recognition as well as generalized preimage computation can be decomposed into x-slices, y-slices and z-slices recognition. Since these three problems only differ by a rotation of the coordinates, we focus our attention on the problem of recognition of z-slices.

2.2 Digital Plane Recognition Problems

We now state the problems that we shall address in this framework:

Problem 1. Input: a finite subset $S \subset \mathbb{Z}^3$

- P-Exi: Does there exist a z-slice containing S?
- P-One: Provide a z-slice containing S
- P-All: Provide a description of all z-slices containing S

As z-slices are described by double-inequalities $h \leq ax + by + cz < h + |c|$, the question is to find a, b, c and h. This inequation is almost linear. The only non-linear term is the absolute value $|c|$. Let us reduce the problem to the case $c = 1$.

We first notice that the set of possible $(a, b, c, h) \in \mathbb{R}^4$ is a positive cone since for any solution (a, b, c, h) of inequalities and any positive real $\lambda \in \mathbb{R}^{+\star}$, the point $(\lambda a, \lambda b, \lambda c, \lambda h)$ is also a solution. This comes from the homogeneity of the inequalities. Hence, instead of computing the whole cone of solutions, it is more convenient to reduce its computation to a section. As c is assumed different from 0, we can take the sections by the hyperplanes $c = 1$ and $c = -1$. It is clear that the whole set of solutions is characterized by its two sections.

We also notice that the solutions for a direction (a, b, c) and the solutions for direction $(-a, -b, -c)$ are closely related. By denoting respectively h_{min} and h_{max} the minimum and maximum of the finite set of values $\{ax_i + by_i + cz_i/(x_i, y_i, z_i) \in S\}$ (S is finite), the set S belongs to the z-slice $h \leq ax+by+cz < h+|c|$ iff $h \in]h_{max} - |c|, h_{min}]$ while it belongs to the z-slice $h \leq -ax-by-cz < h + |c|$ with a symmetric normal $(-a, -b, -c)$ iff $h \in] - h_{min} - |c|, -h_{max}]$. It means that one interval can be easily obtained from the other. Thus we can reduce the computations of possible parameters h to vectors (a, b, c) with a positive c (and then with $c = 1$).

2.3 The Preimage Polyotope

According to previous remarks, problems P-Exi, P-One and P-All can all be reduced to the case $c = 1$. Thus these problems can be reduced to a system of linear inequalities $h \leq ax_i + by_i + z_i < h + 1$ for $(x_i, y_i, z_i) \in S$. Classical Linear Programming algorithms such for instance simplex method can be used for solving the problems P-Exi and P-One but we can notice that usually, they do not provide the whole set of solutions. Thus the problem P-All does not enter in their framework of application.

Definition 3. *The z-preimage of a finite set S is the 3-dimensional set of values $(a, b, h) \in \mathbb{R}^3$ verifying for any point $(x_i, y_i, z_i) \in S$ the double-inequality $h \leq ax_i + by_i + z_i < h + 1$.*

We can as well define x-preimage and y-preimage. Usually, z-preimage will simply be called *preimage*. It is a convex set since it is obtained by intersection of finitely many open and closed half-spaces. If we assume that the preimage is not empty, we notice that its interior is neither empty with the consequence that its vertices and faces are also the vertices and faces of its closure. Hence, under this assumption, we can work with the polyhedron defined by non-strict inequalities $h \leq ax_i + by_i + z_i \leq h + 1$ without changing the vertices and faces of the preimage.

If we consider the 3D space of solutions (a, b, h), we can notice that the half-spaces $h \leq ax_i + by_i + z_i$ are directed downwards while the open half-spaces $ax_i + by_i + z_i \leq h + 1$ are directed upwards. It leads to introduce the upper bound of the preimage of S according to direction h and its lower bound. We denote them respectively $preimage^+(S)$ and $preimage^-(S)$. They meet in a curve that we denote $preimage^0(S)$ and call the *equator* of the preimage (Fig 1). Due to orientation considerations, the faces of $preimage^-(S)$ are among the inequalities $ax_i + by_i + z_i \leq h + 1$ while the faces of $preimage^+(S)$ are of the form $h \leq ax_i + by_i + z_i$.

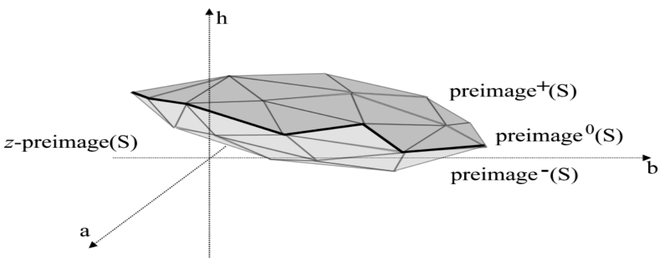

Fig. 1. We decompose the boundary of $preimage(S)$ in its upper bound $preimage^+(S)$, its lower bound $preimage^-(S)$ and its equator $preimage^0(S)$ (the points that vertical projection are on the boundary of the vertical projection of $preimage(S)$ (according to h))

In the following, we assume that the preimage is not empty and that the points of S do not belong to a vertical plane. We can eliminate this degenerated case since it can be replaced in a 2D framework. With these conditions, the preimage of S is a polytope (and the equator is closed) which can be described either by providing its vertices or a minimal set of linear constraints.

3 Geometry of the Digital Plane Preimage

Our algorithm of preimage computation requires to put in relation the preimage elements (vertices and faces) with the geometry of S. This structural analysis was initiated in [9] in which a similar result of Proposition 2 is given. This result is reformulated and improved by new properties using a uniform formalism.

3.1 The Vertices of $preimage^+(S)$ and $preimage^-(S)$

Let us consider a vertex (a', b', h') of the upper boundary $preimage^+(S)$ of the preimage of S. The vertex (a', b', h') satisfies the inequalities $h' \leq a'x + b'y + z$ (and even $h' \leq a'x+b'y+z \leq h'+1$) for any point $(x, y, z) \in S$ and at least three independent inequalities are equalities. It follows that there exist three affinely independent points (x_i, y_i, z_i), (x_j, y_j, z_j) and (x_k, y_k, z_k) in S such that we have exactly $h' = a'x_i + b'y_i + z_i$, $h' = a'x_j + b'y_j + z_j$ and $h' = a'x_k + b'y_k + z_k$. It means that $h' = a'x + b'y + z$ is the affine plane of a facet of the lower boundary of the convex hull of S. Conversely, let us take a facet $h' = a'x + b'y + z$ of the lower boundary of the convex hull of S verifying the complementary condition that for any $(x, y, z) \in S$: $h' \leq a'x + b'y + z \leq h' + 1$. There exist three affinely independent points (x_i, y_i, z_i), (x_j, y_j, z_j) and (x_k, y_k, z_k) on this facet. They verify $h' = a'x_i + b'y_i + z_i$, $h' = a'x_j + b'y_j + z_j$ and $h' = a'x_k + b'y_k + z_k$. All the points (a, b, h) of the preimage verify $h \leq ax_i + by_i + z_i$, $h \leq ax_j + by_j + z_j$ and $h \leq ax_k + by_k + z_k$. In the parameter space (a, b, h), these three conditions define a cone of vertex (a', b', h') containing the preimage. As (a', b', h') is at least in preimage of S (due to the complementary condition), it is a vertex of the preimage.

Proposition 1. *Let S be a finite subset of $\mathbb{Z}^3$ which is not contained by any vertical plane. A point (a, b, h) is a vertex of $preimage^+(S)$ if and only if the affine plane $ax + by + z = h$ contains a facet of the lower boundary of the convex hull of S and we have for any point $(x_i, y_i, z_i) \in S$: $h \leq ax_i + by_i + z_i \leq h + 1$.*

An equivalent proposition can be given for the vertices of $preimage^-(S)$: a point (a', b', h') is a vertex of $preimage^-(S)$ if and only if the affine plane $a'x+b'y+z = h' + 1$ contains a facet of the upper boundary of the convex hull of S and we have for any point $(x_i, y_i, z_i) \in S$: $h' \leq a'x_i + b'y_i + z_i \leq h' + 1$. It means that the vertices of $preimage^+(S)$ and $preimage^-(S)$ are both derived from a subset of facets of the convex hull of S. These two subsets of facets (if non empty) can be introduced as the *caps* of S.

Definition 4. *The upper (resp. lower) cap of S is the set of points (x_i, y_i, z_i) of the upper (resp. lower) boundary of the convex hull of S for which there exist $(a, b, h) \in \mathbb{R}^3$ with $h + 1 = ax_i + by_i + z_i$ (resp. $h = ax_i + by_i + z_i$) and $h \leq ax + by + z \leq h + 1$ for all points (x, y, z) of S (see Fig 2).*

Two cases are possible: the upper (lower) cap contains facets and according to Proposition 1, these facets are one to one with the vertices of $preimage^-(S)$ ($preimage^+(S)$) -the correspondence is described in Fig 2- or there is no facet in the upper (lower) cap (it is reduced to points or edges) and $preimage^-(S)$ ($preimage^+(S)$) has no vertex (the vertices are equatorial). Authors of [9] notice that the caps usually contain exactly one facet with the consequence that there is one non-equatorial vertex in $preimage^-(S)$ and another one in $preimage^+(S)$. They prove that this case occurs necessarily under some assumptions on the input sets (large enough to contain "leaning points").

3.2 The Vertices of $preimage^0(S)$: Chords and Visibility Cone

The situation on the equator of the preimage is different because a vertex (a, b, h) of the equator is either the intersection of two faces $h \leq ax_i + by_i + z_i$ of $preimage^+(S)$ and one face $ax_j + by_j + z_j \leq h + 1$ of $preimage^-(S)$, either the intersection of one face $h \leq ax_i + by_i + z_i$ of $preimage^+(S)$ and two faces $ax_j + by_j + z_j \leq h+1$ of $preimage^-(S)$. We thus introduce a new tool to describe them differently: the notion of *chords set*.

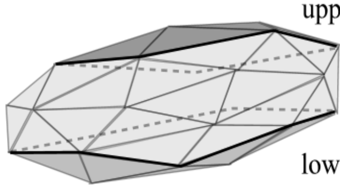

Fig. 2. The upper cap of the convex hull is made of the points of upper convex hull of S contained by a plane $h+1 = ax+by+z$ that translation by $(0,0,-1)$ is under S. The lower cap contains the points of the lower convex hull of S contained by a plane $h = ax+by+z$ that translation by $(0,0,+1)$ is above S. The support planes $h = ax+by+z$ of facets of the upper cap provide the vertices $(a,b,h-1)$ of $preimage^{-}(S)$ while the facets $h = ax+by+z$ of the lower cap provide the vertices (a,b,h) of $preimage^{+}(S)$.

Definition 5. *The set of the differences* $\{x' - x/x \in S, x' \in S\}$ *is called the chords set of* S *and we denote it* $S \ominus S$.

The interest of the chords set comes from next lemma.

Lemma 1. *Given* (a,b)*, there exist a real* h *such that the finite set* S *belongs to the z-slice* $h \leq ax+by+z \leq h+1$ *if and only if there exists a real* $h' \leq 1$ *such that the plane* $ax+by+z = h'$ *separates* $(0,0,1)$ *from the chords set of* S *(equality is permitted).*

Proof. We denote again $[h_{min}, h_{max}]$ the range of the values $ax_i + by_i + z_i$ for (x_i, y_i, z_i) in S. The first proposition means exactly that $h_{max} - h_{min} \leq 1$ while the second one means that any pair of indices i and j, the difference between $ax_i + by_i + z_i$ and $ax_j + by_j + z_j$ is ≤ 1. By taking the index i providing h_{min} and the index j providing the value h_{max}, the equivalence is easy to obtain. □

Lemma 1 means more generally that the chords set of S contains all the information necessary to compute the projection of the preimage of S on the plane of coordinates a and b: a point (a,b) is in the projection of the preimage of S if and only if there exist a plane of normal direction $(a,b,1)$ separating the chords set $S \ominus S$ from $(0,0,1)$. This last proposition makes the link between the projection of the preimage (according to direction h) and the set of planes separating $S \ominus S$ from a point. Thus it can be useful to recall that the set of directions of the planes separating a set S' from a point P can be given by the cone of visibility of S' from P. More precisely, the directions (a,b,c) of the planes separating S' from P are convex combination of the normal directions of the faces of the visibility cone. In the present framework lemma 1 means that the directions $(a,b,1)$ of the preimage of S (or namely the projection of the preimage on the plane (a,b)) are convex combinations of the directions $(a,b,1)$ of the faces of the cone of visibility of $S \ominus S$ from $(0,0,1)$ (see Fig 3). It means that the cone of visibility of $S \ominus S$ from $(0,0,1)$ provides directly the vertices of the projection of the preimage on the plane (a,b): a face of the cone of visibility with equation

$ax + by + z = 1$ provides the vertex (a, b) of the projection of the preimage of S on the plane (a, b).

To pass from the vertices of the projection of the preimage on the plane (a, b) to the ones of the equator $preimage^0(S)$, we just need to pick up the point (a, b) by computing the unique value h such that $h \leq ax_i + by_i + z_i \leq h + 1$ for any $(x_i, y_i, z_i) \in S$. Thus we obtain easily the vertices of the equator $preimage^0(S)$ of the preimage of S.

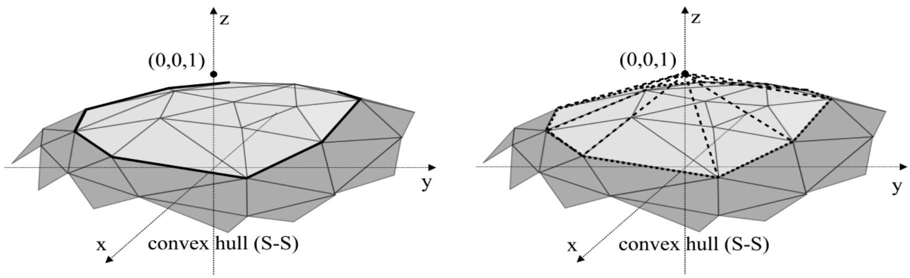

Fig. 3. The cone of visibility of the convex hull of $S \ominus S$ from $(0, 0, 1)$. The normal direction $(a, b, 1)$ of a face of the cone provides a vertex (a, b) of the projection of the preimage on plane (a, b). By picking it with height h (the unique value verifying $h \leq ax_i + by_i + z_i \leq h + 1$ for any point (x_i, y_i, z_i) of S), we obtain the vertices of the equator $preimage^0(S)$ of the preimage of S.

3.3 The Faces of the Preimage

After the geometrical description of the vertices of the preimage of S, next step is the characterization of its faces. It is obvious that the faces of $preimage^+(S)$ are among the inequalities $h \leq ax_i + by_i + z_i$ with $(x_i, y_i, z_i) \in S$ while the faces of $preimage^-(S)$ are among the inequalities $ax_j + by_j + z_j \leq h + 1$ with $(x_j, y_j, z_j) \in S$. The question is to know which points of S define the faces of $preimage^+(S)$ and which ones define the faces of $preimage^-(S)$.

Proposition 2. *The points (x_i, y_i, z_i) and (x_j, y_j, z_j) providing the upper faces $h \leq ax_i + by_i + z_i$ and lower faces $ax_j + by_j + z_j \leq h + 1$ of the polyhedron preimage of S are respectively some vertices of the lower and upper caps of S.*

Proposition 2 does not mean that all vertices of the caps provide some faces of the preimage. Let (a', b', h') be the center of an upper face $h \leq ax_i + by_i + z_i$ of the preimage, the point (x_i, y_i, z_i) is unique in S to verify $h' = a'x_i + b'y_i + z_i$ ($h' < a'x_i + b'_i + z_i \leq h' + 1$ for all other points of S). Hence, (x_i, y_i, z_i) is in the lower cap but it means more: a point (x_i, y_i, z_i) without any (a', b', h') satisfying $h' = a'x_i + b'y_i + z_i$ and $h' < a'x_i + b'_i + z_i \leq h' + 1$ for all other points of S can not provide a face of the preimage. This necessary condition is even sufficient but we do not know if there exist points in the lower cap which do not satisfy it!

4 The Recognition Algorithm

4.1 Sketch of the Algorithm

The task of the paper is to provide an efficient algorithm to solve problem P-All. We recall that problems P-Exi and P-One are already solved efficiently (quasi-linear time) by linear programming or other algorithms derived from Computational Geometry [11]. Problem P-All consists in computing the whole polyhedron of solutions. The algorithm is based on the computation of its vertices :

- a function *upperVertices* computes the vertices of $preimage^+(S)$ (according to proposition 1, they correspond to the faces of the lower cap of S);

- a second function *lowerVertices* computes the vertices of $preimage^-(S)$ by using their correspondence with the vertices to the upper cap of S;

- a third function *equatorVertices* is in charge of the computation of the vertices of the equator by using the cone of visibility of $S \ominus S$ from $(0, 0, 1)$. Instead of working with the real cone, we compute the facets of the convex hull of $(S \ominus S) \cup \{(0, 0, 1)\}$ containing vertex $(0, 0, 1)$.

The common point of each function is to explore a set of facets, the facets of the upper and lower caps for functions *upperVertices* or *lowerVertices* and the facets of the convex hull of $(S \ominus S) \cup \{(0, 0, 1)\}$ containing $(0, 0, 1)$ for *equatorVertices*. This exploration can be done according to a gift-wrapping principle [13,14]: starting from a facet F of the convex hull, we compute the neighboring facet F' according to a given edge e of F (thus F and F' share e).

4.2 Initialization

We start the computation with any algorithm solving P-One. If there exists no solution, the preimage is empty and we stop. Otherwise, the algorithm provides a double inequality $h \leq ax + by + z < h + 1$ satisfied by all points (x, y, z) of S. Normal direction $(a, b, 1)$ provides three initial points:

- we compute the point $(x_i, y_i, z_i) \in S$ realizing the minimum of $ax + by + z$ in S. We have $ax + by + z = h_i$. It is straightforward that all points (x, y, z) of S verify $h_i \leq ax + by + z \leq h_i + 1$. This proves that (x_i, y_i, z_i) is in the lower cap;

- we compute the point $(x_j, y_j, z_j) \in S$ realizing the maximum of $ax + by + z$ in S. We have $ax + by + z = h_j$. As previously, (x_j, y_j, z_j) is in the upper cap because all points of S verify $h_j - 1 \leq ax + by + z < h_j$;

- the point $(x_j - x_i, y_j - y_i, z_j - z_i)$ is a vertex of the convex hull of $S \ominus S$. It belongs to the plane $ax + by + z = h_j - h_i$ above $S \ominus S$ and cutting the vertical axis between 0 and $(0, 0, 1)$.

4.3 From a Starting Point to a Starting Facet

We have a starting point and we explore the facets that contain it until finding a satisfying one.

- We explore the facets of the convex hull of S with vertex (x_i, y_i, z_i). We can do it by turning around (x_i, y_i, z_i) by gift-wrapping until finding a facet in the lower cap of S. It is also possible that no facet adjacent with (x_i, y_i, z_i) belongs to the lower cap: it simply means that $preimage^+(S)$ has no vertex and closes this computation.

- We explore the facets of the convex hull of S with vertex (x_j, y_j, z_j) by gift-wrapping. If we do not find any facet of the upper cap of S, it ends the computation ($preimage^-(S)$ has no vertex).

- We search a facet of the convex hull of $(S \ominus S) \cup \{(0,0,1)\}$ having $(0,0,1)$ as vertex but at this point, we have only a vertex $(x_j - x_i, y_j - y_i, z_j - z_i)$ of the convex hull of $S \ominus S$ (Fig 4). We can decompose the convex hull of $S \ominus S$ between the facets which are destroyed when we add $(0,0,1)$ to $S \ominus S$ (region A) and the ones which are preserved (region B). By construction, the point $(x_j - x_i, y_j - y_i, z_j - z_i)$ is in region A or at least on its boundary. The challenge is to go from this point in region A in the direction of region B until finding the boundary. In this goal, we compute a first facet containing the vertex $(x_j - x_i, y_j - y_i, z_j - z_i)$ and cross a "line" of facets until an edge of the boundary (the choice of the edge that we share is made in order to advance in a given direction so to avoid loops). We go from a facet to next one by gift-wrapping. Although the number of points in $S \ominus S$ is the square of the cardinality of S, this computation can be done $O(card(S))$ time (see next Sect. 4.4). We end this computation with an edge of the boundary of region A: it remains only to add the vertex $(0,0,1)$ to have a facet of the convex hull of $(S \ominus S) \cup \{(0,0,1)\}$

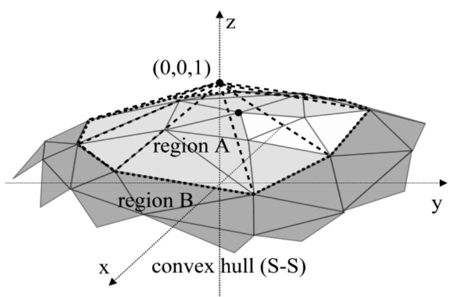

Fig. 4. Region A is made of the facets of the convex hull of $S \ominus S$ which disappear when we add the point $(0,0,1)$. The remaining facets are in region B. The equator of the preimage is given by the visibility cone of $S \ominus S$ from $(0,0,1)$. The facets of this cone are bounding the two regions. We can go from region A to region B by using a gift-wrapping process following a given direction, as drawn for instance in white.

4.4 Gift-Wrapping on the Chords Set $S \ominus S$

Gift-wrapping on a set of cardinality n takes $O(n)$ time. Given a face F of the convex hull of S and one of its edges e, we find the other facet sharing e by projecting the points of S in a plane according to direction e. To determine the new vertex to associate with e, we research the point providing the maximum angle with the projection of F. Thus the time necessary for this computation is linear in n (find the maximum of n angles through the ratio of the coordinates).

For the computation of the vertices of the equator, we use a gift-wrapping procedure working on the chords set $S \ominus S$ that cardinality can be quadratic. For thousands of points, if we still use a procedure linear in the number of points, the cost could be important. We can however improve the research of the new vertex by reducing the number of angles that should be compared. As drawn in Fig 5, the edge issued from a vertex $u - v$ in the convex hull of $S \ominus S$ is necessarily equivalent with an edge issued from u or from v. It allows to work directly on S: we compute the point of S that angle with u or v is maximal. It reduces the number of angles to compare from n^2 to $2n$ with the consequence that gift-wrapping on the chords set remains linear.

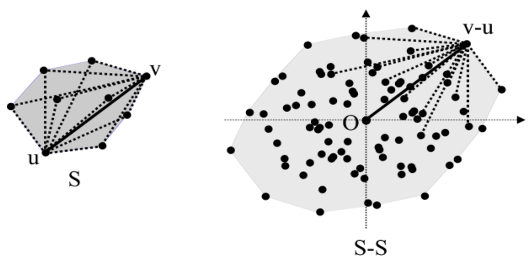

Fig. 5. If we consider the edges issued from a vertex $v - u$ of the chords set $S \ominus S$ (S is here in 2D), they are both equivalent either to an edge issued from u, either to an edge issued from v in the convex hull of S

4.5 Last Step: From a Facet to the Others

The exploration of the whole upper and lower caps from an initial facet is a standard procedure of Computational Geometry. We just have to stop the exploration in a direction as soon as next facet does not satisfy the condition to be in the considered cap. Each lower facet $h' = a'x + b'y + z$ or upper facet $a'x + b'y + z = h' + 1$ computed this way provides a vertex (a, b, h) of $preimage^+(S)$ or $preimage^-(S)$. We have however to take care that they are not on the equator by verifying that there does not exist two points (x_i, y_i, z_i) and (x_j, y_j, z_j) of S with $a'(x_j - x_i) + b'(y_j - y_i) + z_j - z_i = 1$.

In the case of the equator, we start from a facet of the convex hull of $(S \ominus S) \cup \{(0, 0, 1)\}$ containing $(0, 0, 1)$ and we just have to turn around $(0, 0, 1)$ to obtain the whole cone of visibility of $S \ominus S$ from $(0, 0, 1)$. The faces $a'x + b'y + z = 1$ of this cone provide the values (a', b') of the vertices of the equator. It remains to compute h as the minimum of $a'x + b'y + z$ for (x, y, z) in S.

4.6 Complexity Analysis

The advantage of Gift-Wrapping algorithms is that they are output-sensitive. The complexity of computation of each new facet is linear [13]. It provides a theoretical complexity in $O(nh)$ where n is the cardinality of the set S and h the number of vertices of the preimage.

5 Experiments

In this part, we provide some experiments about the behavior of our algorithm with respect to the number n of voxels in the set S. To do this, we have used the random generator of the TC18 challenge. We have generated around 6000 tests with various sizes in the coordinates of the voxels. The number of voxels in each test was also randomly generated. We have segmented the results into classes corresponding to set size that varies from 0 to 1000, from 1000 to 2000 and so on. For each class, we have computed the minimum, the maximum and the mean value of the execution time. All those experiments have been plotted on Fig. 6.

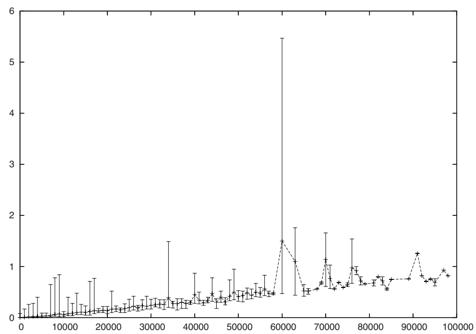

Fig. 6. Variation of the execution time (vertically in seconds) in function of the number of voxels

It is clear that both dependence on n and h are visible in the graph. For instance, the running time globally increases linearly with n but when complex instances are found the h factor leads to an increase in the running time. The maximal time obtained around 60000 voxels comes probably from a bug in our current implementation. We are currently studying this problem.

To compare to already available codes, we have used another set of around 4000 voxels not used in previous experiments. The implementation of Sivignon [15] took 9.78 seconds, the lrs code [16] took 66.76 seconds and our new implementation took 0.56 seconds on the same computer. This behavior was confirmed during all our experiments where our new method is at least 10 times faster than other preimage computation algorithms.

6 Conclusion

In this article, we have precisely studied the preimage set of digital planar sets. We have decomposed the vertices of the preimage into three sets and we have presented algorithms for compute them. We have provided an implementation of this algorithm as well as experiments showing that our new method is fast in comparison with other available algorithm. We plan to test our algorithm in the TC18 Challenge and our implementation will be shortly available on the TC18 pages (`http://www.cb.uu.se/~tc18/`).

References

1. Rosenfeld, A., Klette, R.: Digital straightness. In: Int. Workshop on Combinatorial Image Analysis. Electronic Notes in Theoretical Computer Science, vol. 46, Elsevier Science Publishers, Amsterdam (2001)
2. Dorst, L., Smeulders, A.N.M.: Discrete representation of straight lines. IEEE Trans. on Pattern Analysis and Machine Intelligence 6, 450–463 (1984)
3. McIlroy, M.D.: A note on discrete representation of lines. AT&T Technical Journal 64(2), 481–490 (1985)
4. Lindenbaum, M., Bruckstein, A.: On recursive, $O(n)$ partitioning of a digitized curve into digital straigth segments. IEEE Trans. on Pattern Analysis and Machine Intelligence 15(9), 949–953 (1993)
5. Brimkov, V., Coeurjolly, D., Klette, R.: Digital planarity - a review. Discrete Applied Mathematics (2006)
6. Veelaert, P.: Geometric constructions in the digital plane. Journal of Mathematical Imaging and Vision 11, 99–118 (1999)
7. Vittone, J., Chassery, J.M.: Recognition of digital naive planes and polyhedization. In: Nyström, I., Sanniti di Baja, G., Borgefors, G. (eds.) DGCI 2000. LNCS, vol. 1953, pp. 296–307. Springer, Heidelberg (2000)
8. Dexet, M., Andres, E.: A generalized preimage for the standard and supercover digital hyperplane recognition. In: Kuba, A., Nyúl, L.G., Palágyi, K. (eds.) DGCI 2006. LNCS, vol. 4245, pp. 639–650. Springer, Heidelberg (2006)
9. Coeurjolly, D., Sivignon, I., Dupont, F., Feschet, F., Chassery, J.M.: On digital plane preimage structure. Discrete Applied Mathematics 151(1–3), 78–92 (2005)
10. Coeurjolly, D., Brimkov, V.: Computational aspects of digital plane and hyperplane recognition. In: Reulke, R., Eckardt, U., Flach, B., Knauer, U., Polthier, K. (eds.) IWCIA 2006. LNCS, vol. 4040, Springer, Heidelberg (2006)
11. Gérard, Y., Debled-Rennesson, I., Zimmermann, P.: An elementary digital plane recognition algorithm. Discrete Applied Mathematics 151, 169–183 (2005)
12. Reveillès, J.P.: Géométrie discrète, calcul en nombres entiers et algorithmique. Thèse d'etat, Université Louis Pasteur, Strasbourg, France (1991)
13. Preparata, F.P., Shamos, M.I.: Computational Geometry : An Introduction. Springer, Heidelberg (1985)
14. de Berg, M., Schwarzkopf, O., van Kreveld, M., Overmars, M.: Computational Geometry: Algorithms and Applications. Springer, Heidelberg (2000)
15. Sivignon, I.: De la caractérisation des primitives à reconstruction polyhèdrique de surfaces en géométrie discrète. PhD thesis, Laboratoire des Images et des Signaux, INPG, Grenoble (2004)
16. Avis, D.: lrs: A Revised Implementation of the Reverse Search Vertex Enumeration Algorithm (1999), `http://cgm.cs.mcgill.ca/~avis/C/lrs.html`