



**HAL**  
open science

# Computational aspects of Digital Plane and Hyperplane Recognition

David Coeurjolly, Valentin Brimkov

► **To cite this version:**

David Coeurjolly, Valentin Brimkov. Computational aspects of Digital Plane and Hyperplane Recognition. 11th International Workshop on Combinatorial Image Analysis, Jun 2006, Berlin, Germany. pp.191–204. hal-00185163

**HAL Id: hal-00185163**

**<https://hal.science/hal-00185163v1>**

Submitted on 5 Nov 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Computational aspects of Digital Plane and Hyperplane Recognition

David Coeurjolly<sup>1</sup> and Valentin Brimkov<sup>2</sup>

<sup>1</sup> Laboratoire LIRIS - CNRS UMR 5205

Universit, Claude Bernard Lyon1

43 Bd du 11 novembre 1918

Villeurbanne, France

dcoeurjo@liris.cnrs.fr

<sup>2</sup> Mathematics Department

Buffalo State College

State University of New York

Buffalo, NY 14222, USA

brimkove@buffalostate.edu

**Abstract.** In these note we review some basic approaches and algorithms for discrete plane/hyperplane recognition. We present, analyze, and compare related theoretical and experimental results and discuss on the possibilities for creating algorithms with higher efficiency.

## 1 Introduction

In discrete geometry various definitions and properties of linear structures – such as straight lines, planes, or hyperplanes – have been proposed. On this basis, computational efficient analytic characterizations of these objects have been obtained. In many applications one considers a reverse problem: given a set of pixels or (hyper)voxels, decide if it is a portion of a discrete line or (hyper)plane. For this, a recognition algorithm is needed.

In dimension two, the arithmetic structure of discrete straight lines (DSL) has been exploited to design efficient algorithms, as both their asymptotic computational cost and practical efficiency have been studied (see [31] for a survey on the matter). In higher dimension, similar arithmetic structures still exist in digital planes and hyperplanes. However, to solve a recognition problem, one usually adapts algorithms from linear programming (LP) [13, 14, 19, 27] or computational geometry (CG) [23, 24, 35]. As a rule, there is a gap between the theoretical time complexity bounds obtained for linear programming or computational geometry problems and the practical efficiency of these algorithms when applied to discrete objects. Indeed, one can observe that the existing time complexity bounds are not tight when experimental analysis is performed.

In this presentation we review certain basic approaches for digital plane and hyperplane recognition and consider related computational aspects from both theoretical and practical point of view. In particular, we compare approaches and results related to computational geometry on one hand and integer linear programming on the other.

## 2 Approaches to Defining Digital Planarity

Throughout the paper we will refer to a set of integer points  $S = \{p^1, p^2, \dots, p^m\}$ . In this section we review three basic approaches for defining digital planes and hyperplanes. Chronologically the first one is the following.

Consider a Euclidean hyperplane  $\Gamma$  defined by  $\gamma_0 + \sum_{i=1}^n \gamma_i x_i = 0$  with  $\{\gamma_i\} \in \mathbb{R}$ ,  $|\gamma_i| \leq |\gamma_n|$  for  $1 \leq i < n$  and  $|\gamma_n| > 0$  (the axis  $x_n$  is called the major axis of the plane, see below). Let  $p = (p_1, \dots, p_n)$  be the intersection point of  $\Gamma$  and the straight line defined by  $x_1 = r_1, x_2 = r_2, \dots, x_{n-1} = r_{n-1}$  with  $r_i \in \mathbb{Z}$ ,  $1 \leq i \leq n$ . The grid point  $P = (r_1, r_2, \dots, r_n) \in \mathbb{Z}^n$  with  $p_n - \frac{1}{2} < r_n \leq p_n + \frac{1}{2}$  is the *digital image* of  $p$  with respect to  $\Gamma$ . We have the following definition.

**Definition 1 (Digital hyperplane [35]).**  $S \subset \mathbb{Z}^n$  is a digital hyperplane iff there exists an Euclidean hyperplane  $H$  such that each grid point  $P$  of  $S$  is the digital image of a point  $p \in H$ .

Another approach is based on the following definition.

**Definition 2 (Digital flatness [36]).** Let  $S \subset \mathbb{Z}^n$ .  $S$  is called flat iff there exist  $n + 1$  real numbers  $\gamma_0, \dots, \gamma_n$  such that:

1.  $\max\{|\gamma_0|, \dots, |\gamma_n|\} = 1$ ;
2. every point  $P = (P_1, \dots, P_n) \in S$  satisfies the condition

$$-\frac{1}{2} < \gamma_0 + \sum_{i=1}^n \gamma_i P_i \leq \frac{1}{2}. \quad (1)$$

In [36] Veelaert proves that a discrete set satisfying Definition 1 is flat. Note that Veelaert's definition is more anisotropic since there is no constraints on the Euclidean hyperplane orientation (i.e., the conditions  $|\gamma_i| \leq |\gamma_n|$  for  $1 \leq i < n$  and  $|\gamma_n| > 0$  in Definition 1).

Another way to define a digital hyperplane is the following.

**Definition 3 (Discrete analytic hyperplane [3]).** A discrete analytic hyperplane  $P$  with coefficients  $(a_1, a_2, \dots, a_n, b) \in \mathbb{Z}^{n+1}$ ,  $\gcd(a_1, \dots, a_n) = 1$ , and thickness  $\omega \in \mathbb{N}^*$  is given by:

$$P(a_1, a_2, \dots, a_n, b) = \{(x_1, x_2, \dots, x_n) \in \mathbb{Z}^n \mid 0 \leq b + \sum_{i=1}^n a_i x_i < \omega\} \quad (2)$$

Thus a set  $S \subset \mathbb{Z}^n$  is a subset of a discrete analytic hyperplane iff there exists a vector  $(a_1, a_2, \dots, a_n, b) \in \mathbb{Z}^{n+1}$  with  $\gcd(a_1, \dots, a_n) = 1$  and  $\omega \in \mathbb{N}^*$ , such that  $S \subset P(a_1, a_2, \dots, a_n, b)$ .

This last definition is more general than the previous one since it allows to control the *thickness* of the set of grid point (see [3]). In what follows, we consider a class of discrete analytic hyperplanes, called *naive*, which have thickness  $\omega = \max_{i=1..n} \{a_i\}$ .

If the coefficients of the Euclidean plane in Definition 1 are rational numbers, then the digital hyperplane obtained by that definition is a discrete analytic hyperplane according to Definition 3 as well. Note that in plane recognition problems we usually have to consider finite subsets of grid-points. Hence, the case of irrational coefficients handled by Definitions 1 and 2 may not occur.

In the following, we will use the abbreviation DHP for a digital (or discrete) analytic hyperplane, and DHPS (Digital Hyperplane Segment) for a finite subset of grid points belonging to a DHP. In dimension three, we will denote digital planes (resp. digital plane segments) by DP (resp. DPS).

We conclude this section with one more technical notion to be used in the sequel.

**Definition 4 (Major axis and DHP base).** *Let  $S$  be either a digital hyperplane or a naive discrete analytic hyperplane. Then there exists a major axis  $x_j$  such that the projection  $\bar{S}$  of  $S$  along  $x_j$  onto the plane  $x_j = 0$  is a one-to-one and onto mapping. The  $(n - 1)$ -dimensional set  $\bar{S}$  is called the base of the DHP.*

It was proved in [10] that every digital plane has a major axis. For instance, in Definition 1 the major axis is  $x_n$ , in Definition 2 it is  $x_j$  provided that  $|\gamma_j| = 1$ , while in Definition 2 the major axis is  $x_j$  whenever  $\omega = |a_j|$ .

### 3 Survey on DHP Recognition Algorithms

In this section we review algorithmic solutions for the discrete (hyper)plane recognition problem that can be stated as follows: Decide if a given finite set  $S \subset \mathbb{Z}^n$  is a DHPS. Furthermore, if the answer to this last question is positive, we would like to determine the DHP parameters.

We can distinguish two basic classes of recognition algorithms: ones based on computational geometry techniques and ones using linear programming.

#### 3.1 Computational Geometry Algorithms

The first algorithms of this kind have been proposed in [23, 24] for DPS (*i.e.* for dimension  $n = 3$ ). Let us define a *support* of a point set  $S$  to be a Euclidean plane such that all points from  $S$  lie on the same side of the plane. Then we have the following theorem.

**Theorem 1.** [23] *Let  $S \subset \mathbb{Z}^3$ .  $S$  is a DPS iff there is a support  $H$  of  $S$ , such that the distance between the points from  $S$  and  $H$  is less than 1.*

This theorem has been stated for  $n = 3$  but it trivially extends to higher dimensions.

To find such a support plane, Kim considered the faces of the convex hull  $CH(S)$  of  $S$ . He stated that  $S$  is a DPS iff there is a face of  $CH(S)$  that induces a support plane satisfying the distance criterion of Theorem 1. In [23] and [24] Kim proposed several algorithms to discover such a face. The last statement,

however, turned out to be wrong: as shown in [18, 12], the support of  $S$  can be defined by an edge of  $CH(S)$ . In such a case,  $S$  can be a DPS and there may be no face of its convex hull satisfying the distance criterion.

Another geometric approach to recognize DHPS is based on point set separability. We have the following theorem.

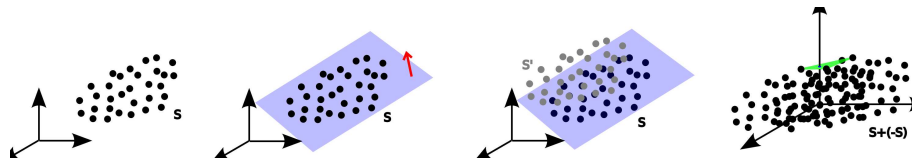
**Theorem 2.** [35] *Let  $S \subset \mathbb{Z}^n$ .  $S$  is a DHPS iff there exists an Euclidean hyperplane  $H$  that separates  $S$  from  $S'$ , where  $S'$  is obtained by a translation of  $S$  at distance 1 along the major axis  $x_n$  of  $H$ .*

Thus the recognition problem is reduced to a separability test for two sets of grid-points. In [35], two algorithms are detailed. One of them uses linear programming (see below). The other is based on computation of convex hull and polytopes intersection [29]. Specifically,  $S$  can be separated from  $S'$  by a plane iff  $CH(S) \cap CH(S') = \emptyset$ . Thus first  $CH(S)$  and  $CH(S')$  are found, then their intersection is computed. Convex hulls computation takes  $O(m \cdot \log m + m^{\lfloor n/2 \rfloor})$  time, while the polytopes intersection can be found in  $O(2^n \cdot m^{2^{n-2}} \cdot \log m)$  (see [35]).

Finally, a class of algorithms are based on the notion of *thickness* of  $S$  (defined for  $n = 3$ ). The thickness can be linked to the distance criterion proposed in Theorem 1. To this end, let us define the *chords set* of  $S$  as the set  $\{P - P' \mid P, P' \in S\}$  [22]. Without loss of generality, suppose that the major axis of  $S$  is  $x_3$ . Then the geometric thickness of  $S$  is the  $x_3$  coordinate of the intersection point of the convex hull of the chords set of  $S$  and the ray defined by  $x_3$  and the origin  $O$  where  $x_3 \geq 0$ . The following theorem holds.

**Theorem 3.** [22] *Let  $S \subset \mathbb{Z}^3$ .  $S$  is a DPS iff its geometric thickness is less than 1.*

These definitions and results have been proposed for dimension  $n = 3$ , but admit easy generalizations. Some of the above-mentioned computational geometry algorithms are illustrated in Figure 1.



**Fig. 1.** Illustration of Computational Geometry recognition algorithms. *From left to right:* The input set of grid points  $S$ , recognition using a support, recognition using the separability test, and recognition based on the thickness of the chords set.

### 3.2 Linear Programming and Integer Linear Programming Algorithms

Since DHP definitions are based on inequalities, the aim of these recognition techniques is to directly solve a linear inequality system using tools from Linear Programming or Integer Linear Programming (ILP). Applying Definition 2 (resp. Definition 3), we can associate to each grid-point  $P$  of  $S$  two inequalities with  $n+1$  unknowns  $\{\gamma_0, \dots, \gamma_n\}$  in  $\mathbb{R}$  (resp.  $n+1$  unknowns  $\{a_1, \dots, a_n, b\}$  in  $\mathbb{Z}$ ). For example using Definition 2, for each grid point  $P \in S$  we have the inequalities  $\gamma_0 - \frac{1}{2} + \sum_{i=1}^n \gamma_i P_i \leq 0$  and  $\gamma_0 + \frac{1}{2} + \sum_{i=1}^n \gamma_i P_i > 0$ . Using LP or ILP algorithms to these linear inequality systems, we can thus decide if the set  $S$  is a DHPS or not.

If we consider a specific orientation of the DHPS, *i.e.* if we restrict our attention to DHPS with a major axis  $x_n$ , the dimension can be reduced from  $n+1$  to  $n$  by considering the unknowns  $\{\gamma_0/\gamma_n, \dots, \gamma_{n-1}/\gamma_n, 1\}$  or  $\{a_1/a_n, \dots, a_{n-1}/a_n, 1, b/a_n\}$ . Note that for the last system the unknowns are in  $\mathbb{Q}$ . Hence, for each grid point  $P \in S$  we have, for example, the constraints:

$$\Gamma_0 - \frac{1}{2} + P_n + \sum_{i=1}^{n-1} \Gamma_i P_i \leq 0 \quad \text{and} \quad \Gamma_0 + \frac{1}{2} + P_n + \sum_{i=1}^{n-1} \Gamma_i P_i > 0. \quad (3)$$

For solving such kind of linear programs one can take advantage of the rich arsenal of available linear programming algorithms (see, e.g., [32]). Optimal theoretical algorithms exist to decide if a set of grid points is a DHPS using Megiddo's theorem:

**Theorem 4 ([27]).** *Given a LP problem with  $m$  linear inequalities in  $\mathbb{R}^n$  where  $n$  is fixed, an algorithm exists to solve the problem in  $O(m)$  time.*

Unfortunately, the above bound  $O(m)$  includes an implicit factor that is exponential in  $n$  (but is a constant when  $n$  is fixed). Considering the DHPS recognition problem based on ILP problems, complexity results can be founded in [13].

**Algorithms Based on Preimage Computation** For the sake of clarity we will present separately the algorithms based on preimage computation, although these methods are deeply linked to those using linear programming.

**Definition 5 (DHP preimage).** *Given a DHP (resp. DHPS)  $S$ , its preimage is the set of Euclidean hyperplanes whose digitization coincides with/contains  $S$ .*

Note that for each DHP definition from Section 2 a digitization scheme can be specified. So in what follows, whenever a digitization scheme is considered, we will suppose that it is the one related to the particular DHP definition adopted.

Basically, the preimage is nothing but the feasible region of the LP inequality system associated with a set of grid points. If we suppose that the major axis is known ( $x_n$ , for example), the preimage is an  $n$ -dimensional polytope, possibly

unbounded, whose vertices have rational coordinates. Obviously, if the preimage associated with a set  $S$  is empty, then  $S$  is *not* a DHPS. Note that if the major axis and the sign of each  $\{T_i\}$  are known (*i.e.* if one knows the global orientation of the DHPS), the preimage polytope is initialized using the unit hypercube since  $(T_0, \dots, T_{n-1}) \in [0, 1]^n$ . A simple incremental DHPS recognition algorithm is given next. Considering  $n$ -dimensional DHPS, the step 6 in Algorithm 1 may have high computational cost due the combinatorial aspects of  $n$ -dimensional polytopes.

However, if  $n = 3$ , several efficient DPS recognition algorithms have been proposed [38, 15]. Indeed, if we denote by  $E$  the number of preimage vertices, the intersection of a Euclidean plane and a convex polyhedron can be computed in  $O(E)$  time. It is not hard to realize that in this case the computational cost of the above algorithm is  $O(m \cdot E)$ . Moreover, the algorithm admits an online implementation which runs in time  $O(E)$ .

---

**Algorithm 1** DHPS preimage based recognition

---

```

1: Let  $S$  be a set of grid points
2: Preimage initialization
3: for each grid point  $P$  of  $S$  do
4:   Let  $C_1$  and  $C_2$  be the two linear constraints of dimension  $n - 1$  associated to  $P$ 
   using Equation (3)
5:   for each constraint  $C$  in  $\{C_1, C_2\}$  do
6:     Update the preimage cutting the polytope with the oriented hyperplane  $C$ 
7:     if the preimage is empty then
8:       Stop the recognition,  $S$  is not a DHPS
9:     end if
10:  end for
11: end for

```

---

To have a tight computational cost bound, we need to bound  $E$  by the number of grid points  $m$ . Several approaches have been proposed to solve this problem for  $n = 3$  (see [17]). It is not trivial, however. In dimension 3, we can easily see that  $E$  is bounded by the number of vertices of  $CH(S)$ . This can be proved using either the construction of the feasible region based on dual transformation proposed in [29] or simply by observing that the preimage corresponds to the set of Euclidean planes separating  $CH(S)$  and  $CH(S')$  in view of Theorem 2. In this case extremal Euclidean planes are defined by one or two grid points in  $CH(S)$  and two or one grid point in  $CH(S')$ . Hence, the number of such extremal planes is bounded by  $O(|CH(S)|)$ . Finally, for  $n = 3$ , a bound on  $|CH(S)|$  is also a bound on  $E$ .

In the experimental results presented in Section 5.3,  $E$  is always less than  $|CH(S)|$  whatever the dimension. Further developments on that point is an important challenge since it is directly linked to the efficiency of Algorithm 1.

### 3.3 Other Algorithms

In this section we present some other algorithms that are not based on results from computational geometry or linear programming. The first one has been proposed by Veelaert [37]. It recognizes a DPS whose base is a *strip*, that is a 2-D set of the form  $\{(x, y) \mid r \leq x \leq s\}$  with  $r, s \in \mathbb{Z}$ . The recognition process is based on the concept of *evenness* of a DHP [36].

**Definition 6 ([36, 37]).**  $S \subset \mathbb{Z}^n$  is called even iff its projection along the  $x_n$  axis is a one-to-one mapping, and for every quadruple  $(A, B, C, D)$  of points in  $S$  such that  $A_{x_n=0} - B_{x_n=0} = C_{x_n=0} - D_{x_n=0}$ , it holds  $|(A_{x_n} - B_{x_n}) - (C_{x_n} - D_{x_n})| \leq 1$ .

This characterization is related to the Rosenfeld's DSL chord property [30] as well as to the Kim's DP chordal triangle property [23]. Veelaert showed that a strip of voxels  $S$  is a subset of a digital plane iff  $S$  is even [37]. (Note however that such an equivalence does not hold for arbitrary dimension.) This result implies an  $O(m^2)$  algorithm for DPS recognition.

Another approach is based on the parametrization of DP by least-squares fits. It is proved in [25] that, given a rectangular base DPS  $S$ , there is a one-to-one correspondence between least-squares plane fit parameters and the coefficient of the DPS  $S$ . Although these authors' goal was just to give a finite parametrization of DPS, the result can be used to obtain a simple recognition algorithm that consists of two stages:

1. Given an input set  $S$ , first the least-squares plane fit is computed, which provides the coefficients of the relevant Euclidean plane  $H$ . This stage takes  $O(m)$  time since the fitting problem is of linear complexity.
2. Then it is verified if the digitization of the plane  $H$  obtained in Stage 1 coincides with  $S$ . If it is so, then  $S$  is a DPS that is precisely the digitization of  $H$ . Otherwise,  $S$  is not a DPS. It is shown that such a verification can be done in  $O(m)$  time.

It is easy to show that the following holds.

**Theorem 5.** *The least-squares plane fit algorithm solves correctly the rectangular base DPS recognition problem.*

*Proof.* The proof is straightforward: if the two sets coincide, then  $S$  is a DPS by Definition 1 or 2. Conversely, if  $S$  is a DPS, then because of the one-to-one correspondence between least-squares fit parameters and DPS coefficients, there exists a unique least-squares plane fitting whose digitization coincides with  $S$ .  $\square$

Although the algorithm described above is restricted to DPS's with rectangular bases, generalizations to other base shapes as well as to higher dimension seem possible (see, e.g. [25] for a recent related results).



Description	Sources	Dimension	Time	Online time	Remark
Convex hull width	[23, 24]	3	$O(m^2)$	-	
Convex hull separability	[35]	n fixed	$O(m \cdot \log m + m^{\lfloor n/2 \rfloor} + 2^n \cdot m^{2^n - 2n})$	-	
Point set width	[22]	3	$O(m^7)$	yes	
Fourier-Motzkin Algorithm	[19]	3	n.a.	-	
Direct Linear Programming	[27]	n fixed	$O(m)$	$O(1)$ [14]	
Separability test based on LP	[35]	n fixed	$O(m)$	$O(1)$ [14]	
Integer Linear Programming	[13]	n fixed	$O(m \cdot \log D)^1$	-	
Arithmetic Preimage	[38]	3	$O(m^3 \cdot \log m)$	$O()$	
Arithmetic Preimage	[15]	3	$O(m \cdot \log^2 m)$	-	
Preimage		3	$O(m \cdot E)$	$O(E)$	
Evenness property	[37]	3	$O(m^2)$		strip base DPS
Least-squares fits	[25]	3	$O(m)$	no	rectangular base DPS
Arithmetic recognition	[28]	3		-	rectangular base DPS

**Table 1.** Survey of DPS and DHPS recognition techniques.

## 4 Efficiency of DHP Recognition Algorithms

Table 1 summarizes data about some basic DHPS recognition algorithms. One can see that optimal algorithms exist to recognize DHPS in any fixed dimension. However, these are only theoretical since no their implementation is available. In fact, only few algorithms have been implemented for  $n = 3$ . Among these are some algorithms based on point set thickness arguments [22], Fourier-Motzkin elimination [19, 20], direct LP [26], and preimage computation [38, 34].

Considering the asymptotic bounds on these algorithms' complexity, one can observe that there is a gap between those bounds and the estimations of algorithms' efficiency obtained through experiments. To illustrate this point, consider, for instance, the point set thickness algorithm from [22]. Its theoretic computational cost is  $O(m^7)$  whereas in practice it features near linear time complexity when  $m$  increases.

Among the algorithms presented in Table 1, we will focus on the preimage based techniques. We have two serious arguments to do so. First, these are online algorithms and their incremental computational cost depends on the DPS specific characteristics (see below). Moreover, preimage computation provides a complete description of all Euclidean hyperplanes that satisfy the DP definitions. These two features appear to be important requirements in various applications, such as discrete surface segmentation into DPS and reversible polyhedrization of binary objects [20, 26, 33, 34, 16].

In the next section, we consider some results from number theory, theory of lattice polytopes, and integer linear programming in order to obtain time complexity bounds for DHPS recognition algorithms based on preimage computation. More precisely, we focus on bounds on the size of the convex hull of  $S$  that allows us to obtain bounds on the number of preimage vertices (see Section 3.2).

## 5 Towards Obtaining Tight Bounds on the Computational Cost of DP and DHP Recognition

### 5.1 Integer Programming and Associated Lattice Polytopes

Consider the integer linear programming problem (ILP) [32]:

$$\max cx \tag{4}$$

$$Ax \leq b \tag{5}$$

$$x \in \mathbb{Z}^n \tag{6}$$

with  $A = (a_{ij}) \in \mathbb{Z}^{m \times n}$ ,  $b = (b_i) \in \mathbb{Z}^m$  and  $c = (c_j) \in \mathbb{Z}^n$ . The special case when  $m = 1$  and all coefficients as well as the solution components are nonnegative is known as a *knapsack problem*.

It is well-known [21] that both ILP and KP are NP-complete, i.e., it is unlikely to have polynomial algorithms for their solution. Despite of this, many results have been obtained to characterize the set of grid-points in the feasible region of system (5) [32, 7, 39, 6].

Let us denote by  $\mathcal{P}$  the convex polytope defined by Equation (5) and by  $P$  the convex hull of  $\mathcal{P} \cap \mathbb{Z}^n$ . Further, let  $N(A, b)$  be the set of vertices of  $P$  and  $|N(A, b)|$  its cardinality. Various upper and lower bounds on  $|N(A, b)|$  have been found (see, for instance, [39] for a recent survey). In particular, we have the following theorem.

**Theorem 6.** *Let  $A = (a_{ij}) \in \mathbb{Z}^{m \times n}$ ,  $b = (b_i) \in \mathbb{Z}^m$ ,  $c = (c_j) \in \mathbb{Z}^n$  and  $\alpha = \max\{|a_{ij}|, i = 1, \dots, m, j = 1, \dots, n\}$ . Then*

$$|N(A, b)| \leq c_n m^{\lfloor n/2 \rfloor} \log^{n-1}(1 + \alpha) \tag{7}$$

where  $\alpha$  is an upper bound on the largest (by absolute value) coefficient in the ILP formulation and  $c_n$  is a quantity depending only on  $n$ .

Moreover, upper bound (6) is tight, i.e., there is a class of matrices  $A$  and vectors  $b$  for which

$$|N(A, b)| \geq c'_n m^{\lfloor n/2 \rfloor} \log^{n-1}(\alpha)$$

where  $c'_n$  is a constant depending only on  $n$ .

Similar tight bounds hold for the number of the knapsack polytope vertices.

Now let  $P$  be a non-empty  $n$ -dimensional lattice polytope and  $f_k(P)$  the number of its  $k$ -dimensional facets for  $0 \leq k < n$ . In particular,  $f_0$  is the number of vertices of  $P$ . Then the following theorem holds.

**Theorem 7.** [8]

$$f_k \leq c_n (\text{Vol } P)^{\frac{n-1}{n+1}} \quad (8)$$

where  $c_n$  is a quantity depending only on  $n$  and  $(\text{Vol } P)$  the volume of  $P$ .

In dimension 2, the above result can be linked to the maximal number of edges  $e(N)$  of a convex digital polygon included into an  $N \times N$ -grid [5, 2]:

$$e(N) = \frac{12}{(4\pi^2)^{1/3}} N^{2/3} + O(N^{1/3} \log(N)). \quad (9)$$

**Theorem 8.** [8] *Let  $K \in C(D)$  where  $C(D)$  is the family of convex bodies with  $C^2$  boundary and radius of curvature at every point and every direction between  $1/D$  and  $D$ ,  $D \geq 1$ . Let  $\bar{K} = \text{conv}(K \cap \mathbb{Z}^n)$ . If the diameter of  $K$  is enough large, then for every  $n \geq 2$  there are constants  $c_1(n)$  and  $c_2(n)$  such that for all  $k \in \{0, 1, \dots, n-1\}$ ,*

$$c_n d^n \frac{n-1}{n+1} \leq f_k(\bar{K}) \leq c'_n d^n \frac{n-1}{n+1}. \quad (10)$$

In the following, we use these results to give bounds on the size of the convex hull and the preimage of  $S$ .

## 5.2 Application to DPS Recognition

In the following, we suppose that  $S$  is parametrized using the discrete analytical hyperplane definition (see Definition 3). Hence, let  $\alpha = \max\{|a_1|, \dots, |a_n|\}$ . Obviously, diverse parametrizations exist for a DHPS. However, in a recognition process it is common to consider a parametrization that minimizes  $\alpha$  and this parameter is usually bounded by the diameter of  $S$  (*i.e.* bounded by  $O(N)$  in our framework).

**Theorem 9.** 1. *Let  $S$  be a DHPS with a hyper-rectangular base. Then*

$$|CH(S)| \leq c_n \log^{n-1}(1 + \alpha); \quad (11)$$

2. *Alternatively,*

$$|CH(S)| \leq c_n N^{\frac{(n-1)^2}{n+1}}; \quad (12)$$

where  $c, c_n$  are some quantities depending only on  $n$ .

3. *If  $S$  is a DPS with digitally convex base containing the origin, then*

$$|CH(S)| \leq cN^{\frac{2}{3}} \log^2(1 + \max\{N, \alpha\}). \quad (13)$$

*Proof.* We first proof Equation (11). Given DHPS  $S$  with a hyper-rectangular base, we can construct  $CH(S)$  using  $2 \cdot n + 2$  linear constraints in dimension  $n$ . First,  $2 \cdot n$  constraints are necessary to define the hyper-rectangular base (all these are given by  $\{0 \leq x_j \leq b_j\}$  for  $0 \leq j \leq n-1$  and  $b_i \in \mathbb{N}$ ). Moreover, two additional constraints are needed to encode the two parallel Euclidean leaning

hyperplanes associated with the DHPS. Finally, we can construct a matrix  $A = (a_{ij}) \in \mathbb{Z}^{(2n+2) \times n}$  with  $\max\{|a_{ij}|, i = 1, \dots, m, j = 1, \dots, n\} = \alpha$  whose size depends only on  $n$ . Hence, using Theorem 6 with  $m = 2 \cdot n + 2$ , we obtain Equation (11).

To prove Equation (12), it suffices to observe that  $S$  is included in an  $N^n$  grid, therefore  $\text{Vol } CH(S)$  is bounded by  $O(N^{n-1})$ . Indeed, the thickness of  $CH(S)$  is necessarily lower than 1. Finally, using Theorem 7, we obtain the result stated.

Similarly, to prove Equation (13), we observe that if  $S$  is a DPS with a digitally convex base, then the base can be encoded by at most  $O(N^{\frac{2}{3}})$  2-D linear constraints (one constraint per edge of the 2-D convex hull of the base using Equation (9)). Furthermore, since  $S$  is included in an  $N \times N \times N$  grid, the coefficients of the linear constraints are bounded by  $N$ . Finally, using the two constraints that define the 3-D discrete plane,  $CH(S)$  can be represented by  $O(N^{\frac{2}{3}})$  constraints overall. Then Equation (13) follows from Theorem 6.  $\square$

Note that for dimension three, the bound for a DHPS with a hyper-rectangular base conforms to earlier results about rectangular base DPS presented in [17]. Combining geometric and number-theoretic approaches, the author also proves that, under some assumptions, the preimage has at most  $O(\log N)$  facets.

Through some experiments presented in the next section, we quantify the differences between the size of the convex hull of  $S$  and its preimage.

### 5.3 Experimental Results

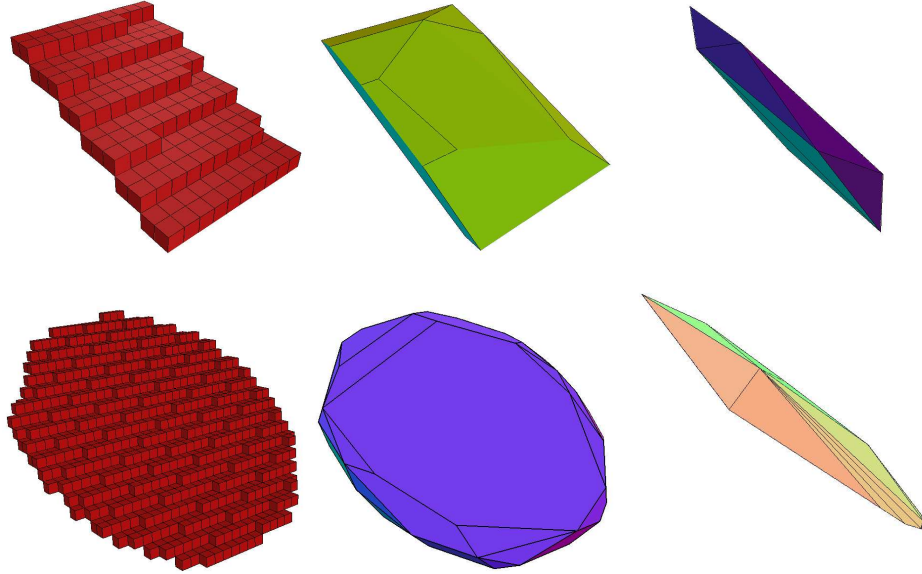
To evaluate the size of both the convex hull and the preimage of a DHPS, we have utilized a specific experimental framework. First, we need a DHPS random generator. Since no uniform random generator of DHPS is available, we created one that conforms to the following natural scheme:

Without loss of generality let us fix the major axis to be  $x_n$ . Then:

1. Construct the DHPS base in an  $N^{n-1}$ -grid;
2. Use a uniform normal vector generator to obtain the DHPS parameters;
3. *Raise* the base along the  $x_n$  axis using the parameters obtained in stage 2.

To generate the base, we consider two main classes of DHPS. The first is a hyper-rectangular one in which the lengths of the  $(n - 1)$ -rectangle are given by independent random generators. DHPS's from the second class have digitally convex bases. In the latter case we generate a random set of cospherical Euclidean points that belong to the hyperparallelepiped  $[-1, 1]^{n-1}$ . Then the Euclidean convex hull of this set is computed and digitized for a given grid resolution in order to obtain digitally convex bases with *round* shapes. We choose this point distribution since it seems to be close to the worst-case regarding the convex hull size.

Figure 2 gives some examples of randomly generated DHPS's in dimension 3.



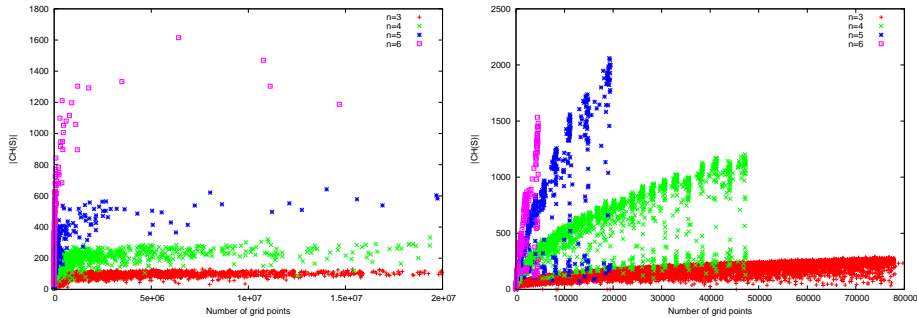
**Fig. 2.** *First row:* a rectangular base DPS, its convex hull and its preimage in the parameter space. *Second row:* a rounded base DPS, its convex hull and its preimage in the parameter space.

To test the theoretical results from Section 5.2, we use the well-known `qhull` program for  $n$ -dimensional convex hulls computation [1, 9]. To evaluate the number of vertices of  $n$ -dimension preimages, we first obtain a system of linear inequalities. For this, we apply Equation (3). Then we make use of the `lrs` software to convert an H-representation (half-space) of a polytope into a V-representation (vertex/ray) or vice versa, as exact arithmetic is used [4].

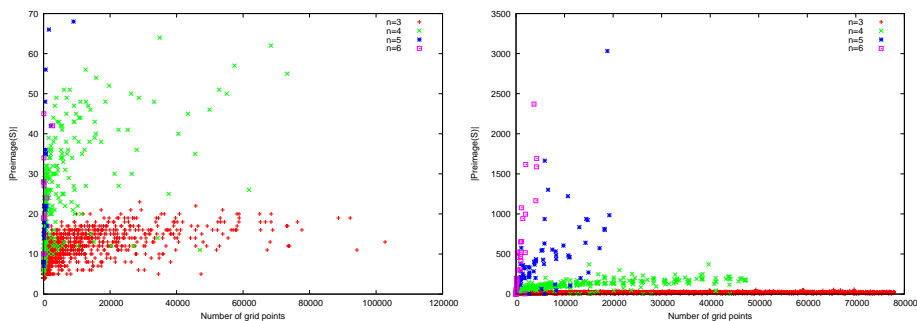
The above algorithm has been designed for problems of arbitrary dimension  $n$ , so it is not surprising that it is outperformed by the incremental Algorithm 1 from Section 3.2 on practical three-dimensional recognition problems.

We present results concerning the two classes of DHPS defined above with increasing  $N$ . In the graphs presented in Figure 3, 4 and 5, x-axis corresponds to the number of grid points given by the DHPS generators and the y-axis corresponds to either  $|CH(S)|$  or  $|Preimage(S)|$  with  $n = \{3, 4, 5, 6\}$ . Figure 3 shows results on  $|CH(S)|$  and Figure 4 the results on  $|Preimage(S)|$ . Finally, Figure 5 details a comparison between  $|CH(S)|$  and  $|Preimage(S)|$  for  $n = 3$ .

As expected in view of Theorem 9, the logarithmic behavior of both  $|CH(S)|$  and  $|Preimage(S)|$  for hyper-rectangular base DHPS clearly appears in the graphs. However, for the other class of DHPS with dimension greater than 3, the experimental results do not show a polynomial behavior of the curves. Further theoretical analysis is thus expected to lead to bounds closer to the experimental framework.



**Fig. 3.** Evaluation of  $|CH(S)|$  on randomly generated DHPS. *Left:* the hyperrectangular base class, and *Right:* the *cospherical* base DHPS class.

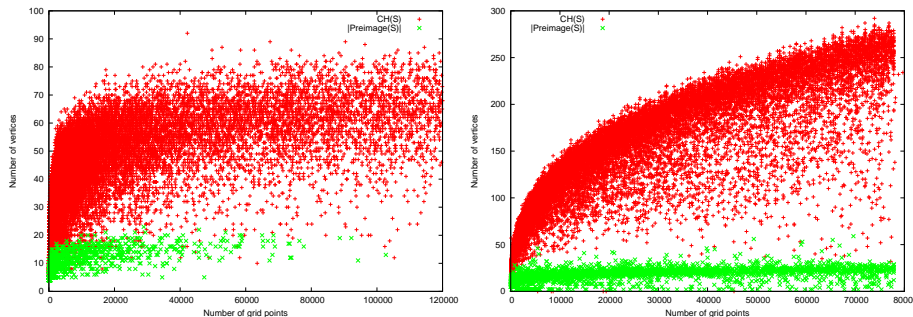


**Fig. 4.** Evaluation of  $|Preimage(S)|$  on randomly generated DHPS. *Left:* the hyperrectangular base class, and *Right:* the *cospherical* base DHPS class.

## 6 Conclusion and Future works

In this article we have first reviewed algorithmic solutions to recognize DHPS whatever the dimension. We have theoretically optimal in time algorithm without efficient implementations and fast algorithms with quite high worst case computational costs. Our analysis suggests that specificities of DHPS recognition problems should be taken into account to obtain tight bounds. Based on results from integer programming and associated lattice polytopes, several theoretical results have been proposed to bound the number of vertices of the convex hull of a DHPS and the number of vertices of its preimage.

One may attempt to solve the recognition problem taking advantage of the theoretical results of Section 5.2. Our experimental results suggest the following conclusions. The direct use of convex hull computation may not be tractable since the design of incremental and output sensitive convex hull algorithms is a difficult task [11]. For dimension  $n = 3$ , our experiments on preimage computation showed that the quantity  $E$  defined in Section 3.2 is indeed quite low whatever the number of grid points. Thus Algorithm 1 appears to be a very simple and practically efficient incremental DPS recognition algorithm.



**Fig. 5.** Comparison between  $|CH(S)|$  and  $|Preimage(S)|$  for  $n = 3$ . *Left* the hyper-rectangular base class and *right*, the *cospherical* base DHPS class.

Several questions related to DHPS recognition are still open. Among these we list the following:

1. Are there tighter bounds on  $|CH(S)|$  in higher dimensions?
2. Can arithmetical structures of DHPS speed up recognition algorithms?

## Acknowledgments

The authors thank Isabelle Sivignon and Raphaëlle Chaine, Laboratoire LIRIS, Université Claude Bernard Lyon 1, for their fruitful comments on this work.

## References

1. Qhull. <http://www.qhull.org/>.
2. D.M. Acketa and J.D. Žunić. On the maximal number of edges of convex digital polygons included into a  $m \times m$ -grid. *Journal of Combinatorial Theory, Serie A*(69):358–368, 1995.
3. E. Andres, R. Acharya, and C. Sibata. Discrete analytical hyperplanes. *Graphical Models and Image Processing*, 59(5):302–309, September 1997.
4. D. Avis. lrs: implementation as a callable library of the reverse search algorithm for vertex enumeration/convex hull problems. <http://cgm.cs.mcgill.ca/~avis/C/lrs.html>.
5. A. Balog and I. Bárány. On the convex hull of the integer points in a disc. In ACM-SIGACT ACM-SIGGRAPH, editor, *Proceedings of the 7th Annual Symposium on Computational Geometry (SCG '91)*, pages 162–165, North Conway, NH, USA, June 1991. ACM Press.
6. I. Barany. Random points, convex bodies, lattices. In *ICM: Proceedings of the International Congress of Mathematicians, 2002*.
7. I. Barany, Howe, and Lovasz. On integer points in polyhedra: A lower bound. *Combinatorica*, 12, 1992.
8. I. Bárány and D. G. Larman. The convex hull of the integer points in a large ball. *Math. Annalen*, 312:167–181, 1998.

9. C. B. Barber, D. P. Dobkin, and H. Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, December 1996.
10. R. P. Barneva, V. E. Brimkov, and P. Nehlig. Thin discrete triangular meshes. *Theoretical Computer Science*, 246(1-2):73–105, 2000.
11. D. Bremner. Incremental convex hull algorithms are not output sensitive. *Discrete & Computational Geometry*, 21(1):57–68, 1999.
12. V. Brimkov, D. Coeurjolly, and R. Klette. Digital planarity - a review. Technical report, Laboratoire LIRIS, Université Claude Bernard Lyon 1, 2004. <http://liris.cnrs.fr/publis/?id=1933>.
13. V. E. Brimkov and S. S. Dantchev. Complexity analysis for digital hyperplane recognition in arbitrary fixed dimension. In *International Conference on Discrete Geometry for Computer Imagery*, pages 287–298, 2005.
14. L. Buzer. A linear incremental algorithm for naive and standard digital lines and planes recognition. *Graphical Models*, 65(1-3):61–76, May 2003.
15. D. Coeurjolly. *Algorithmique et géométrie discrète pour la caractérisation des courbes et des surfaces*. PhD thesis, Université Lumière Lyon 2, Bron, Laboratoire ERIC, dec 2002.
16. D. Coeurjolly, Alexis Guillaume, and I. Sivignon. Reversible discrete volume polyhedrization using marching cubes simplification. In *SPIE Vision Geometry XII*, volume 5300, pages 1–11, San Jose, USA, 2004.
17. D. Coeurjolly, I. Sivignon, F. Dupont, F. Feschet, and J.-M. Chassery. On digital plane preimage structure. *Discrete Applied Mathematics*, 151(1–3):78–92, 2005.
18. I. Debled-Rennesson. *Etude et reconnaissance des droites et plans discrets*. PhD thesis, Université Louis Pasteur, 1995.
19. J. Françon, J.M. Schramm, and M. Tajine. Recognizing arithmetic straight lines and planes. In *6th Discrete Geometry for Computer Imagery*, volume 1176 of *LNCS*, pages 141–150. Springer-Verlag, 1996.
20. J. Françon and L. Papier. Polyhedrization of the boundary of a voxel object. In *8th International Conference in Discrete Geometry for Computer Imagery*, volume 1568 of *LNCS*, pages 425–434. Springer-Verlag, 1999.
21. M. R. Garey and D. S. Johnson. *Computers and intractability; a guide to the theory of NP-completeness*. W.H. Freeman, 1979.
22. Y. Gerard, I. Debled-Rennesson, and P. Zimmermann. An elementary digital plane recognition algorithm. *DAMATH: Discrete Applied Mathematics and Combinatorial Operations Research and Computer Science*, 151, 2005.
23. C. E. Kim. Three-dimensional digital planes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6:639–645, 1984.
24. C. E. Kim and I. Stojmenovic. On the recognition of digital planes in three-dimensional space. *Pattern Recognition Letters*, 12(11):665–669, 1991.
25. R. Klette, I. Stojmenovic, and J. D. Zunic. A parametrization of digital planes by least-squares fits and generalizations. *CVGIP: Graphical Model and Image Processing*, 58(3):295–300, 1996.
26. R. Klette and H. J. Sun. Digital planar segment based polyhedrization for surface area estimation. In *IWVF*, pages 356–366, 2001.
27. N. Megiddo. Linear programming in linear time when the dimension is fixed. *JACM: Journal of the ACM*, 31, 1984.
28. M. M. Mesmoudi. A simplified recognition algorithm of digital planes pieces. In *10th International Conference, Discrete Geometry for Computer Imagery*, volume 2301 of *LNCS*, pages 404–416, Bordeaux, France, april 2002. Springer Verlag.



29. F. P. Preparata and M. I. Shamos. *Computational Geometry : An Introduction*. Springer-Verlag, 1985.
30. A. Rosenfeld. Digital straight lines segments. *IEEE Transactions on Computers*, pages 1264–1369, 1974.
31. A. Rosenfeld and R. Klette. Digital straightness. In *Int. Workshop on Combinatorial Image Analysis*, volume 46 of *Electronic Notes in Theoretical Computer Science*. Elsevier Science Publishers, August 2001.
32. A. Schrijver. *Theory of Linear and Integer Programming*. Wiley and Sons, 1986.
33. I. Sivignon and D. Coeurjolly. From digital plane segmentation to polyhedral representation. In *Theoretical Foundations of Computer Vision "Geometry, Morphology, and Computational Imaging"*, number 2616 in LNCS, Springer-Verlag, pages 356–367, 2003.
34. I. Sivignon, F. Dupont, and J. M. Chassery. Decomposition of a three-dimensional discrete object surface into discrete plane pieces. *Algorithmica*, 38(1):25–43, 2003.
35. I. Stojmenović and R. Tosić. Digitization schemes and the recognition of digital straight lines, hyperplanes and flats in arbitrary dimensions. In *Vision Geometry, contemporary Mathematics Series*, volume 119, pages 197–212, American Mathematical Society, Providence, RI, 1991.
36. P. Veelaert. On the flatness of digital hyperplanes. *Journal of Mathematical Imaging and Vision*, 3:205–221, 1993.
37. P. Veelaert. Digital planarity of rectangular surface segments. *IEEE Pattern Analysis and Machine Intelligence*, 16(6):647–652, jun 1994.
38. J. Vittone and J.-M. Chassery. Recognition of digital naive planes and polyhedization. In *9th Discrete Geometry for Computer Imagery*, volume 1953 of LNCS, pages 296–307. Springer, 2000.
39. N. Y. Zolotykh. On the number of vertices in integer linear programming problems. Technical report, University of Nizhni Novgorod, 2000.