

Introduction to Digital Level Layers

Yan Gérard, Laurent Provot, and Fabien Feschet

Univ. Clermont 1, ISIT, Campus des Cézeaux, 63172 Aubière, France
yan.gerard@u-clermont1.fr, provot.research@gmail.com, feschet@acm.org

Abstract. We introduce the notion of Digital Level Layer, namely the subsets of \mathbb{Z}^d characterized by double-inequalities $h_1 \preceq f(x) \preceq h_2$. The purpose of the paper is first to investigate some theoretical properties of this class of digital primitives according to topological and morphological criteria. The second task is to show that even if we consider functions f of high degree, the computations on Digital Level Layers, for instance the computation of a DLL containing an input set of points, remain linear. It makes this notion suitable for applications, for instance to provide analytical characterizations of digital shapes.

Keywords: Digital Primitives, Cover, Thickness, Linear Programming, Support Vector Machines.

1 About the Words

We propose to introduce in this paper the notion of Digital Level Layers. This name refers to level sets, namely to subsets of a space X characterized by an equation $f(x) = h$ where the variable x is in X and where h is a constant value. Most often the space X is \mathbb{R}^d . Under assumption that the gradient of f is not null, level sets are sub-manifolds of dimension $d-1$, namely hypersurfaces. In this paper we are interested in digital sets, which means that the space X is restricted from \mathbb{R}^d to \mathbb{Z}^d . In order to preserve interesting properties, this restriction of support space requires to relax the constraint $f(x) = h$ to a double-inequality $h_1 \leq f(x) \leq h_2$. It leads to the definition of Digital Level Layers: *Digital* because the space X is \mathbb{Z}^d , *Level* because we use a height function f and *Layers* because we consider the layer between the two level sets $f(x) = h_1$ and $f(x) = h_2$.

Definition 1. A Digital Level Layer –DLL for short– is a subset of points x of \mathbb{Z}^d verifying a double inequality $h_1 \preceq f(x) \preceq h_2$ where f goes from \mathbb{Z}^d to \mathbb{R} , where the symbols \preceq and \preceq denote \leq or $<$ and where h_1, h_2 are real numbers.

2 Why This Kind of Primitives ?

One of the main challenges of digital geometry is to provide digital primitives corresponding to algebraic curves or surfaces in the Euclidean world. There exist several ways to define such objects. The first idea to introduce geometrical digital primitives is to consider the trace on \mathbb{Z}^d of the geometrical primitives of \mathbb{R}^d . It means that a digital straight line would be the trace of a real straight line, a

digital circle the trace of a real circle, namely the set of solutions of multivariate Diophantine equations. Although the resolution of such equations is undecidable, the main reason why they are not used for geometrical purposes is that their visual rendering is completely different from the visual rendering of the Euclidean objects – in fact they have bad properties in terms of digital topology: they have not enough voxels to separate the space, which can be traduced by tunnels. Since this first naive idea is not satisfactory, one can wonder how to define digital primitives that fulfill some good topological and morphological properties. Several answers to this kind of problem have been developed for the last thirty years. We can mention two kinds of approaches: “morphological” and “topological”.

- the morphological approach is to consider an extended notion of trace. A point $x \in \mathbb{Z}^d$ belongs to the trace of the Euclidean surface $S \subset \mathbb{R}^d$ if the hypercube $[x - \epsilon, x + \epsilon]^d$ has a non-empty intersection with S . This extended notion of trace is equivalent to the trace of the Minkowski sum $S + [x - \epsilon, x + \epsilon]^d$. These objects are simply called “covers” [6] (Fig. 1) and with some refinements “supercovers” [5,2] (*supercover* deals with the cases where, if $\epsilon \geq 1/2$, a real point of S can provide several integer points and hence a non minimal thickness). We can also use different “structuring elements” such as Euclidean balls, hypercubes, segments... With the square $[-0.5; 0.5]^2$, the cover of $S + [-0.5; 0.5]^2$ corresponds to the set of pixels centered on the grid points that cross the circle. If we choose a segment $[-0.5; 0.5] \times \{0\}$ or $\{0\} \times [-0.5; 0.5]$ (depending on the region) as structuring element, we have another definition of circle which is also widely used. Generally speaking, it can be hard to provide a general analytic characterization of these primitives but as we will notice in the sequel, local characterizations can be found.
- a topological approach can be used if the Euclidean primitive we want to digitize has an interior and an exterior. Hence a natural way to define the digitization of an implicit curve or surface $f(x) = h$ is for instance to consider the boundary of the trace of the interior integer points ($f(x) < h$). The main advantage of this topological approach is to define objects which are topologically guaranteed (Fig. 1).

Hence is there a reason to introduce another class of primitives? In fact each one of these two classes of digital surfaces has its own drawbacks. The main problem with morphological surfaces is their recognition. As far as we know, no efficient algorithm has been published to recognize ellipsoids for instance, defined according to this approach. The topological surfaces do not suffer from the same problem. Their recognition is a problem of separation. To recognize for instance an ellipsoid, the problem is to find a Euclidean ellipsoid separating the interior points from their complement. This problem is solved by Support Vector Machines [7,15]. There is however another reproach we can do for both approaches: they provide no direct analytical characterization of the voxels of the surface. The main interest of Digital Level Layers is to satisfy this requirement which can be useful in the framework of conversions from raster to vector graphics. In other words, DLL is the third approach, the “analytical” one, to define digital primitives.

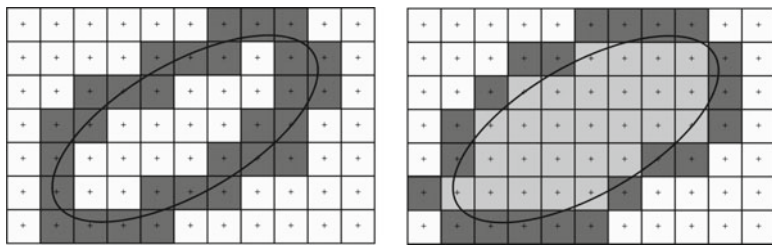


Fig. 1. Digitization's of an ellipse, on the left according to the morphological principle – we take the pixels which cross the Euclidean ellipse – on the right according to the topological principle – we take the difference between the pixels that center is in the interior of the ellipse and their 4-neighborhood

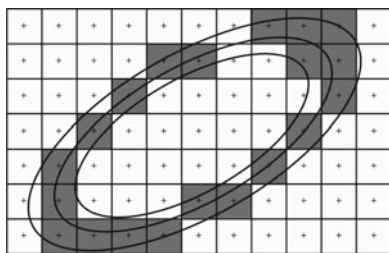


Fig. 2. Digitization of an ellipse as a Digital Level Layer – we take the integer points between two level sets on both sides of the ellipse

The main drawback of DLL as “surfaces” appears on Fig. 2: the width of DLL is not constant all along its points. An idea could be to say: “OK, instead of using a double inequality which can be written $0 < f(x) < h$ where h is a constant, we just have to work with inequalities $0 < f(x) < h(x)$ where $h(x)$ depends linearly on the Euclidean norm of the gradient $\vec{\nabla} f$ namely $h(x) = h_0 \|\vec{\nabla} f(x)\|$ with a constant h_0 ” but the inequality $0 < f(x) < h_0 \|\vec{\nabla} f(x)\|$ can be simply rewritten $0 < \frac{f(x)}{\|\vec{\nabla} f(x)\|} < h_0$. It is just a change of function f and the gradient of the new function $\frac{f(x)}{\|\vec{\nabla} f(x)\|}$ is not guaranteed to have a constant norm: the problem remains!

Is it a sufficient reason to reject DLL? No because in spite of its main drawback, there are a lot of situations where we do not take care of the Euclidean width of the surface. The main question is thus to determine whether Digital Level Layers can be useful. The outline of the paper is to investigate the interest of this notion through two main angles:

- from a theoretical point of view: which theoretical properties can we obtain, with some assumptions on the function f , for instance if f is a polynomial of bounded degree? We are going to consider topological and morphological properties.

- the second important point is to determine the algorithmic tools that can deal with DLL. We focus our attention on the recognition of these families of primitives – with again some assumptions on f . The recognition problem has been deeply investigated in the framework of linear structures and we provide here a new approach based on excluded sets of points. We are going to see how the results obtained in a linear framework can be extended in order to work efficiently with DLL.

3 Theory

The definition of Digital Level Layers is very general, maybe too much, as soon as we make no assumption on the function f . We can even notice that for any finite subset S of \mathbb{Z}^d , we can construct the analytic expression of a function f such that S is the DLL of double-inequality $-1/2 \leq f(x) \leq 1/2$. Hence to say that a given set is a DLL has clearly no interest by itself. The interest of the notion is in the relation between the set S , the class of the function f that appears in the double-inequality $h_1 \preceq f(x) \preceq h_2$ and the value of the difference $h_2 - h_1$.

3.1 Functional and Algebraic DLL

We start by introducing the notion of *functional* DLL.

Definition 2. A Digital Level Layer L is called *functional in direction i* if it contains exactly one point in each line of the lattice \mathbb{Z}^d parallel to the i^{th} axis.

We call these DLL functional because they define clearly a function from \mathbb{Z}^{d-1} to \mathbb{Z} and conversely any function from \mathbb{Z}^{d-1} to \mathbb{Z} can be represented by a functional DLL. Most often, functional DLL are used to digitize real functions f sending \mathbb{Z}^{d-1} or even \mathbb{R}^{d-1} to \mathbb{R} by taking their integer part $\lfloor f \rfloor$ (other convention can be chosen as $\lfloor f + 0.5 \rfloor$, $\lceil f + 0.5 \rceil$ or $\lceil f \rceil$). The functional DLL associated to $\lfloor f \rfloor$ is simply characterized by double-inequality $0 \leq f(x_i)_{1 \leq i \leq d-1} - x_d < 1$. Such functional DLL are also morphological surfaces since we can obtain them as the trace on \mathbb{Z}^d of the sum of the surface $x_d = f(x_i)_{1 \leq i \leq d-1}$ with the vertical segment $] - 1; 0]$ as structuring element.

As we will see in the section devoted to algorithms, digital hyperplanes are important for the recognition of DLL and they are particular cases of these generic class of objects: A digital straight line is a DLL with $f(x) = a_1x_1 + a_2x_2$, a digital plane is a DLL with $f(x) = a_1x_1 + a_2x_2 + a_3x_3$ and more generally digital hyperplanes are simply defined with an affine form $f(x) = \sum_{i=1}^d a_i x_i$ as height function [13]. DLL cover also the case of digital hyperspheres $(R - \frac{1}{2})^2 \leq \sum_{i=1}^d (x_i - a_i)^2 < (R + \frac{1}{2})^2$ (according their definition in [4]) (Fig. 4). All these usual primitives of digital geometry are DLL with a polynomial height function f of low degree. It leads to the introduction of the notions of *algebraic* DLL (Fig. 5).

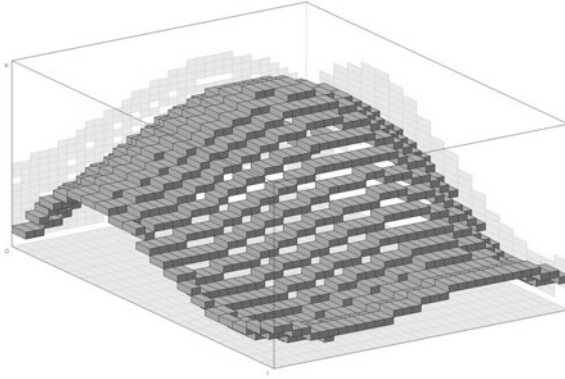


Fig. 3. The functional DLL of double-inequality $4680370559422800 \leq -17758960839495486x + -2015536714762185y + 292165823537172xy + 2304539315667564x^2 + -172062298114380y^2 + -8936152895940xy^2 + -75980632598934x^3 + 1625988388601x^2y + 7065487923165y^3 + 4336388981634600z < 4680370559422800 + 4336388981634600$ namely $z = \lfloor (4680370559422800 + 17758960839495486x + 2015536714762185y - 292165823537172xy - 2304539315667564x^2 + 172062298114380y^2 + 8936152895940xy^2 + 75980632598934x^3 - 1625988388601x^2y^1 - 7065487923165y^3)/4336388981634600 \rfloor$

Definition 3. An algebraic Digital Level Layer is a subset of points x of \mathbb{Z}^d verifying a double inequality $h_1 \preccurlyeq P(x) \preccurlyeq h_2$ where P is a multivariate polynomial of $\mathbb{R}[X_i]_{1 \leq i \leq d}$ and where h_1, h_2 are real numbers.

It follows of course that digital lines, hyperplanes and spheres (for at least one definition) are algebraic DLL.

3.2 Topological Properties

We consider in this section a notion of digital connectedness induced by a norm N of \mathbb{R}^d (two points x and x' of \mathbb{Z}^d are neighbors if their distance $N(x' - x) \leq 1$) or equivalently by a convex neighborhood which can be translated in any point of the lattice \mathbb{Z}^d . The ball of radius 1 is denoted B_1 .

We recall that the dual norm N^* of N is defined by its ball of radius 1 with $B_1^* = \{x \in \mathbb{R}^d / \forall y \in B_1, x \cdot y \leq 1\}$. The balls B_1 and B_1^* are polar from each other [16]. We have for instance $N_\infty^* = N_1$ and conversely $N_1^* = N_\infty$. The connectedness associated to N_∞ and N_1 is respectively denoted $3^d - 1$ and $2d$ -connectedness. The duality of N_∞ and N_1 leads of course to the idea that the $3^d - 1$ and the $2d$ -connectedness from \mathbb{Z}^d are dual.

Connectedness. By considering a digital layer L_δ of double inequality $h - \frac{\delta}{2} \leq f(x) < h + \frac{\delta}{2}$, three domains are considered:

- the two domains of the complement of L_δ in \mathbb{Z}^d denoted L_δ^- and L_δ^+ and verifying respectively $f(x) < h - \frac{\delta}{2}$ and $f(x) \geq h + \frac{\delta}{2}$
- the layer L_δ itself.

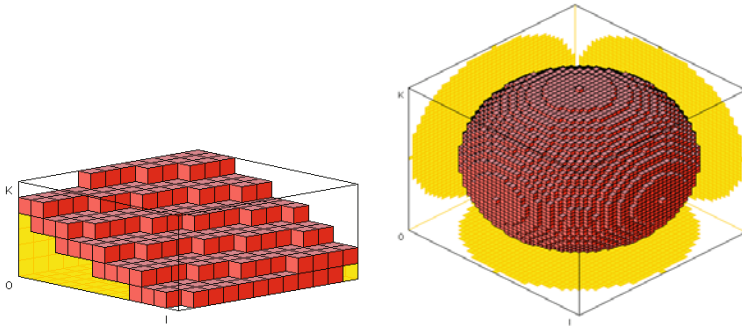


Fig. 4. The digital plane of double-inequality $583 \leq 72x - 28y + 190z \leq 845$ and the digital sphere of double-inequality $19^2 < (x - 20)^2 + (y - 20)^2 + (z - 20)^2 \leq 20^2$

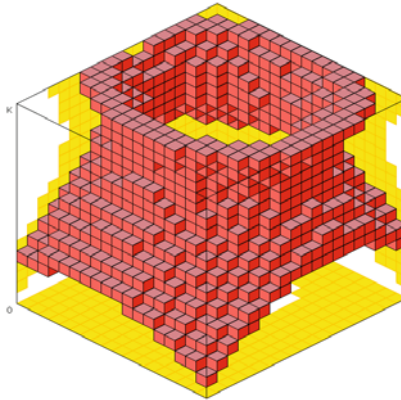


Fig. 5. The algebraic DLL of double-inequality $-233050932002 \leq 9706766352x + 34545685472y - 50669609424z - 1006860048x^2 - 1419723424y^2 + 1813925040z^2 + 346913568xz - 191023296yz \leq -194272584000$

We notice that by increasing the value δ , points are deleted from L^- and L^+ to be added in L . We introduce the set $\Delta = \{\delta \in \mathbb{R}^+ / L_\delta \text{ is connected}\}$. As in a Euclidean framework, the property of Δ to be an interval is related to the existence of local minimum of $|f - h|$ upper than the smaller value δ which provides a connected layer L_δ .

To determine the set Δ is often difficult. Except in the case of the digital straight lines of \mathbb{Z}^2 , namely with functions $f(x) = a_1x_1 + a_2x_2$ (we assume $GCD(a_1, a_2) = 1$) where it is known for a long time that $\Delta = [|a_1| + |a_2|, +\infty[$ according to 4-connectedness and $\Delta = [\max\{|a_1| + |a_2|\}, +\infty[$ according to 8-connectedness, the problem has been longly investigated. In dimension 3 with digital planes of \mathbb{Z}^3 having a rational direction, Δ remains an interval of the form $\Delta = [\min, +\infty[$ that minimal connecting thickness has been expressed in [11]. In the case of digital planes of given irrational normal (a_1, a_2 and a_3 are rationally independent), the lower bound of the connecting thickness has been computed in 2009 in [8].

Separation. With the question of connectedness of L_δ arises also the question of the connectedness between L_δ^- and L_δ^+ , namely the existence of tunnels in L_δ .

Proposition 1. *We assume that f is C^1 and that at least one of the two symbols \preceq or \succcurlyeq of inequality $h - \delta/2 \preceq f(x) \preccurlyeq h + \delta/2$ is strict ($<$). If for every point $x \in \mathbb{R}^d$ we have $N^*(\vec{\nabla} f(x)) \leq \delta$, then L_δ^- and L_δ^+ are separated according to the connectedness associated to the norm N (L_δ is tunnel-free).*

Proof. We assume that for any x in \mathbb{R}^d we have $N^*(\vec{\nabla} f(x)) < \delta$ and that the two points x and x' are adjacent according to N namely that $N(x' - x) \leq 1$. We have $f(x') - f(x) = \int_x^{x'} \vec{\nabla} f(x) \cdot \frac{(x' - x)}{\|x' - x\|_2} ds$ where the path of integration is the segment $[x, x']$. It follows from $N^*(\vec{\nabla} f(x)) \leq \delta$ that $\vec{\nabla} f(x) \cdot (x' - x) \leq \delta N(x' - x)$ or again $\vec{\nabla} f(x) \cdot \frac{x' - x}{\|x' - x\|_2} \leq \delta \frac{N(x' - x)}{\|x' - x\|_2}$. Thus we have $f(x') - f(x) \leq \int_x^{x'} \delta \frac{N(x' - x)}{\|x' - x\|_2} ds$. We obtain $f(x') - f(x) \leq \delta N(x' - x) \leq \delta$. Hence, x and x' cannot be one in L_δ^- and the other in L_δ^+ .

Proposition 1 means that a sufficient condition of separation for the digital line of double-inequality $h - \delta/2 \leq a_1x + a_2y < h + \delta/2$ is $N^*(\vec{\nabla} f(x)) \leq \delta$. We have of course $\vec{\nabla} f(x) = (a_1, a_2)$. It means that a sufficient condition for separability is $N_1^*(a_1; a_2) \leq \delta$ namely $N_\infty(a_1; a_2) \leq \delta$ i.e. $\max(|a_1|; |a_2|) \leq \delta$ for 4-separability. It means that $N_\infty^*(a_1; a_2) \leq \delta$ namely $N_1(a_1; a_2) \leq \delta$ i.e. $|a_1| + |a_2| \leq \delta$ is sufficient to guarantee 8-separability. These two conditions are sufficient to have a tunnel-free line and it is known that they are also necessary [3].

It is the same in the framework of digital planes. The sufficient condition of separability given by Proposition 1 is $\max(|a_1|; |a_2|; |a_3|) \leq \delta$ for 6-connectedness while it is $|a_1| + |a_2| + |a_3| \leq \delta$ for 26-connectedness. In the general framework of digital hyperplanes (double inequality $h - \delta/2 \leq a \cdot x < h + \delta/2$), the Proposition 1 proves the sufficient condition of separability $N_\infty(a) \leq \delta$ for the $2d$ connectedness and $N_1(a) \leq \delta$ for the $3^d - 1$ connectedness. This condition is also necessary [3].

3.3 Morphological Property

The main morphological property of surfaces is their thinness. As noticed previously, the main drawback of DLL is to provide a representation of a Euclidean surface $f(x) = h$ which can be thick in some points and thin elsewhere. It is however not the case of all digital layers, as we can see with digital lines or digital spheres and the fact that some layers are not satisfactory due to their variation of thickness does not mean that it is the case for all DLL.

One of the tools we provide in the next section is precisely a method to compute layers that thickness is locally controlled by forbidden points (which can be chosen arbitrarily or automatically). Even if the layer obtained this way has a variable thickness somewhere, we believe that its local control – with some “control” points – in a region of interest makes DLL an interesting tool for digital geometry applications, and a step towards an efficient vectorization of digital shapes.

3.4 Comparison between DLL and Morphological Surfaces

Morphological surfaces – covers – are defined as the trace on \mathbb{Z}^d of the Minkowski’s sum $S + E$ of a Euclidean surface S with a structuring element E which can be a unit vertical segment $0^{d-1} \times [-0.5; 0.5[$, a unit voxel $[-0.5; 0.5]^d$ or any convex subset of \mathbb{R}^d . The boundary of the sum $S + E$ is a subset of the sum of boundaries $\partial S + \partial E$. Its structure is obtained by translating alternatively some pieces of the boundary of S and some other pieces of the boundary of E . An example is drawn in Fig. 6. We can decompose the ellipse of equation $f(x) = h$ in four pieces according to the quadrant of the normal vector ∇f . A pixel of center x crosses the ellipse if and only if it belongs to a set of equation $f(x - v) \leq h \leq f(x + v)$ where v can be $(-0.5; -0.5)$, $(-0.5; 0.5)$, $(0.5; -0.5)$ or $(0.5; 0.5)$ or to a unit square centered on the four points of extremal coordinates. Without providing the details, it means that the morphological ellipse can be decomposed in four squares and four pieces of curved stripes characterized by inequalities $f(x - v) \leq h \leq f(x + v)$. We can notice that such double-inequality is completely different from the double-inequalities characterizing the DLL. We have here two different functions $f(x - v)$ and $f(x + v)$ translated one from each other, while in DLL we have only one function and it is the constant that differs to define the two bounding surfaces. It follows that DLL are deeply different from morphological surfaces.

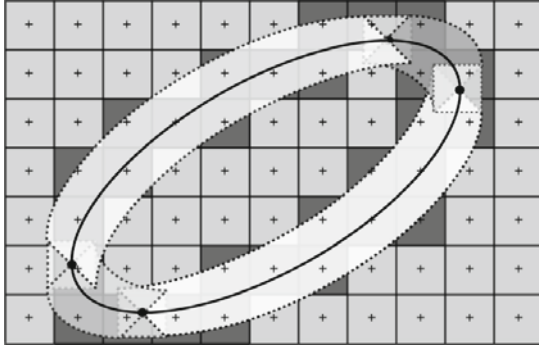


Fig. 6. The cover of an ellipse namely the digital morphological ellipse obtained with a unit square as structuring element. It is made of the integer points belonging to several pieces obtained by translating the initial curve from $(-0.5; -0.5)$, $(-0.5; 0.5)$, $(0.5; -0.5)$, $(0.5; 0.5)$ and to unit squares centered at the extremal points.

4 Algorithms

The main algorithmic challenge with digital primitives is their recognition and the main advantage of DLL on morphological surfaces is that this problem can be tackled efficiently. The important fact is that even with algebraic DLL of degree greater than 1, the recognition problem remains linear. This reduction is known as the “kernel trick”. It has been introduced in 1964 in [1]. Hence

we decompose the section into two parts. The first one is devoted to problems of “linear” computational geometry while the second one is devoted to their application in the framework of general DLL.

4.1 Two Problems of Computational Geometry

Find the thinnest strip containing a given set of points The first problem that we consider is the following:

Problem 1. Input: A finite subset In of \mathbb{R}^d . *Output:* Find the thinnest affine strip $h_1 \leq n \cdot x \leq h_2$ containing In (where $n \cdot x$ denotes the dot product between the normal vector n and the variable point x).

We should precise what we mean by “thin”. A natural choice is to give to the word thin an Euclidean meaning: The Euclidean thickness of the affine strip $h_1 \leq n \cdot x \leq h_2$ is the ratio between $h_2 - h_1$ and the Euclidean norm of the normal vector $\|n\|_2$ but we could of course define the thickness with any other norm. It provides a definition of thickness for any subset of \mathbb{R}^d :

Definition 4. *The thickness of a subset In of \mathbb{R}^d according to a norm $\|\cdot\|$ is the lower bound of the set of thickness of the strips containing In .*

We can even extend the notion of thickness from norms to functions which are just homogeneous. Instead of considering that the thickness of the strip $h_1 \leq n \cdot x \leq h_2$ is the ratio $\frac{h_2 - h_1}{\|n\|}$, it becomes $\frac{h_2 - h_1}{\varphi(n)}$ where φ is a homogeneous function i.e satisfying $\varphi(\lambda n) = \lambda \varphi(n)$. This extension allows us to speak about directional thickness if we consider for instance $\varphi(n) = |n_d|$ where n_d is the last coordinate of n .

Definition 5. *The φ -thickness of a subset In of \mathbb{R}^d is the lower bound of the set of values $\frac{h_2 - h_1}{\varphi(n)}$ where the strip $h_1 \leq n \cdot x \leq h_2$ contains In .*

In the case of a discrete set In , the minimum may be reached for some strips. The question is: given φ , how do we solve Problem 1? For the Euclidean norm ($\varphi(\cdot) = \|\cdot\|_2$), the solution can be computed by the rotating caliper [14] based on the computation of the convex hull. For a directional width ($\varphi(\cdot)$ is the absolute value of a linear form), the problem can be expressed as a linear program (minimize δ subject to $h \leq \sum_{i=1}^{d-1} n_i x_i + x_d \leq h + \delta$) and solved in linear time (if d is fixed) by Megiddo algorithm [12]. We can also solve it efficiently with the chord algorithm [9] or others. If $\varphi(\cdot)$ is a polyhedral norm, namely a norm whose unit ball is a polyhedron, we can compute the φ -thickness as the minimum of all the directional thickness in the directions normal to the faces of the unit polyhedron. We should however notice that in the framework of digital hyperplanes recognition, the problem arises under a different form. Instead of computing the minimal φ -thickness of In , the question is to determine whether this value is smaller than a given maximal thickness:

Problem 2. Input: A finite subset In of \mathbb{R}^d , a homogeneous function $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}$ and a maximal thickness M . *Output:* Find a strip $h_1 \leq n \cdot x \leq h_2$ containing In with a φ -thickness $< M$.

The answer of Problem 1 provides clearly the answer of Problem 2, and usually the algorithms chosen to compute the φ -thickness can be modified so that they stop as soon as the φ -thickness of In becomes greater than M . As known in the framework of Linear Programming, there is usually no great difference in the algorithms computing the minimum of an objective function and the ones determining whether the minimum is greater than a constant value.

Find a strip containing a set of points with two excluded sets of points.

Let us now consider a different problem without using any notion of thickness. It is just based on two sets of forbidden points Up and $Down$. The aim is to put the initial set In in a strip, Up on one side and $Down$ on the other side.

Problem 3. Input: Three finite subsets $In, Up, Down$ of \mathbb{R}^d . *Output:* Find an affine strip $h_1 \leq n.x \leq h_2$ containing In with the points of Up and $Down$ on both sides namely with $h_1 \leq n.x \leq h_2$ for all x in In , with $n.x < h_1$ for $x \in Down$ and with $n.x > h_2$ for x in Up .

Here we have a problem of multiclass separation. These problems have been investigated in the deep SVM literature [7,15] and we can solve Problem 3 efficiently with a variant of the well-known GJK algorithm [10] used for collision detection.

4.2 From Affine Strips to DLL Recognition: The Kernel Trick

In the case of digital straight lines, planes and hyperplanes, the problem of recognition is to determine whether a given set of points In belongs to a digital hyperplane with a fixed maximal thickness. In the extended framework of DLL, the problem arises in the same way. Let us consider a DLL characterized by $h_1 \leq f(x) < h_2$ where the function $f(\cdot) : \mathbb{Z}^d \rightarrow \mathbb{R}$ belongs to the linear space F generated by functions $f_1(\cdot), f_2(\cdot), \dots, f_m(\cdot)$. We define the thickness of the double-inequality $h_1 \leq f(x) < h_2$ from any homogeneous function $\varphi : F \rightarrow \mathbb{R}$.

Definition 6. *The φ -thickness of double inequality $h_1 \leq f(x) \leq h_2$ where f is in the linear space F with a homogeneous real function $\varphi : F \rightarrow \mathbb{R}$ is $\frac{h_2-h_1}{\varphi(f)}$.*

The main drawback of this notion of φ -thickness is to be algebraic. It can have a geometrical meaning, mainly as a directional thickness, to compute for instance functional DLL as in Fig. 3 but in many other cases, this notion of thickness remains mainly algebraic without any possibility to have a direct geometrical understanding of its value. We can however be interested in recognizing thin DLL, at least for the cases where a geometrical meaning holds.

Problem 4. Input: A finite subset In of \mathbb{R}^d , a linear space F of functions $f : \mathbb{Z}^d \rightarrow \mathbb{R}$ generated linearly by $f_1(\cdot), f_2(\cdot), \dots, f_m(\cdot)$, a homogeneous function $\varphi : F \rightarrow \mathbb{R}$ and a maximal thickness M . *Output:* Find a DLL containing In having a double-inequality $h_1 \leq f(x) \leq h_2$ with a φ -thickness $< M$.

This problem consists in finding h_1, h_2 , and coefficients a_1, a_2, \dots, a_m of $f = \sum_{i=1}^m a_i f_i$ with $\frac{h_2-h_1}{\varphi(\sum_{i=1}^m a_i f_i)} < M$ under constraints $h_1 \leq \sum_{i=1}^m a_i f_i(x) \leq h_2$ for

all x in In . If we denote $g(x) = (f_i(x))_{1 \leq i \leq m} \in \mathbb{R}^m$ and n the vector $(a_i)_{1 \leq i \leq m}$, the problem is reduced to solve Problem 2 with the set $g(In)$ as input (in dimension m) and function $\varphi(n) = \varphi(\sum_{i=1}^m a_i f_i)$ as thickness criterion. We can solve it with the rotating caliper if φ is a quadratic norm and by linear programming or chords algorithm if φ is the absolute value of a linear form or a polyhedral norm. As corollary, these tools allow to solve efficiently Problem 4. Such algorithms have been implemented successfully for the directional thickness: Fig. 3 and Fig. 5 have been for instance obtained by using these tools.

As noticed previously, the computation of a DLL of minimal φ -thickness or of φ -thickness $< M$ does not guarantee to have a DLL with a minimal local geometrical thickness. That is the reason why we have introduced Problem 3. Instead of providing the set In as the only input set and to look for a DLL of minimal algebraic thickness containing it – since the notion of φ -thickness has not always the geometrical meaning that we can hope – we suggest to replace it by outliers. The idea is to provide two control sets Up and $Down$ whose points should be bypassed by the DLL. With the kernel trick, we can reduce the computation of a DLL containing In , with the outliers of $Down$ and Up on one and the other side, to Problem 3 with $g(In)$, $g(Up)$ and $g(Down)$ as input sets. A variant of the GJK algorithm can be used to solve it efficiently. This alternative to the recognition of a DLL of minimal thickness allows us to imagine many effective applications and can be particularly interesting to go from raster graphics to vector graphics without dealing with an unsuitable notion of algebraic thickness.

5 Conclusion

From the theoretical point of view, it is rather clear that Digital Level Layers do not have the nice arithmetical properties of digital straight lines or planes. We also have emphasized the main drawback of DLL compared to morphological and topological surfaces: they have a non constant local geometrical thickness. The lack of geometrical meaning of their algebraic thickness makes DLL recognition under the classical approach (Problem 4) not really suitable for applications. It is much more interesting in practice to consider the recognition of DLL by replacing this condition of maximal algebraic thickness by two sets of outliers. With this new approach, which can be solved very efficiently (Problem 3), Digital Level Layers become a promising tool in the framework of vectorization of raster graphics.

References

1. Aizerman, M., Braverman, E., Rozonoer, L.: Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control* 25, 821–837 (1964)
2. Andres, E.: The supercover of an m -flat is a discrete analytical object. *Theoretical Computer Science* 406(1-2), 8–14 (2008)

3. Andres, E., Acharya, R., Sibata, C.: Discrete analytical hyperplanes. *Graphical Model and Image Processing* 59(5), 302–309 (1997)
4. Andres, E., Jacob, M.A.: The discrete analytical hyperspheres. *IEEE Transactions on Visualization and Computer Graphics* 3(1), 75–86 (1997)
5. Brimkov, V.E., Andres, E., Barneva, R.P.: Object discretizations in higher dimensions. *Pattern Recognition Letters* 23(6), 623–636 (2002)
6. Cohen-Or, D., Kaufman, A.E.: Fundamentals of surface voxelization. *Graphical Model and Image Processing* 57(6), 453–461 (1995)
7. Cristianini, N., Shawe-Taylor, J.: *Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, Cambridge (2000)
8. Domenjoud, E., Jamet, D., Toutant, J.L.: On the connecting thickness of arithmetical discrete planes. In: Brlek, S., Reutenauer, C., Provençal, X. (eds.) *DGCI 2009*. LNCS, vol. 5810, pp. 362–372. Springer, Heidelberg (2009)
9. Gérard, Y., Debled-Rennesson, I., Zimmermann, P.: An elementary digital plane recognition algorithm. *Discrete Applied Mathematics* 151, 169–183 (2005)
10. Gilbert, E.G., Johnson, D.W., Keerthi, S.S.: A fast procedure for computing the distance between complex objects in three-dimensional space. *IEEE Journal of Robotics and Automation* 4, 193–203 (1988)
11. Jamet, D., Toutant, J.L.: Minimal arithmetic thickness connecting discrete planes. *Discrete Applied Mathematics* 157(3), 500–509 (2009)
12. Megiddo, N.: Linear programming in linear time when the dimension is fixed. *Journal of the ACM* 31(1), 114–127 (1984)
13. Reveillès, J.P.: *Géométrie discrète, calculs en nombre entiers et algorithmique*. Thèse d'état, Université Louis Pasteur, Strasbourg (1991)
14. Toussaint, G.T.: Solving geometric problems with the rotating calipers. In: *Proceedings of IEEE MELECON 1983*, Athens, Greece (1983)
15. Vapnik, V.: *Statistical Learning Theory*. Wiley, New York (1998)
16. Ziegler, G.M.: *Lectures on polytopes*. Springer, Heidelberg (1995)