Double Dual Approach for Digital Polynomial Curve Segmentation.

Gaelle Largeteau-Skapin, Rita Zrour, and Eric Andres

Université de Poitiers, Laboratoire XLIM, SIC, UMR CNRS 7252, BP 30179, F-86962 Futuroscope Chasseneuil, France
{gaelle.largeteau.skapin,rita.zrour,eric.andres}@univ-poitiers.fr

Abstract. In cite[DGCI] the author proposed a dual space based method to recognize two parameter polynomial implicit curves $C(x,y): \ x^i \times y^j - B \times x^k \times y^l - A = 0$ in digital images. In this dual space, a pixel is associated with convex polygons and the recognition problem is addressed using a line stabbing solutions together with linear programming.
Here we extend the use of this dual space to the segmentation of a digital contour with two parameter functions. The problem is thus the following: given a set of pixel S, wich is the set of two parameter polynomial functions and their definition intervals wich digitization is S.
In this work, the considered digitization model is the simplified 1-Flake model.

Keywords: Digital Curve Recognition, Segmentation, Stabbing, Transversal, High degree function, Flake model.

1 Introduction

Recognition, and more precisely parameter estimation, of geometric primitives has played a central role in the development of various applicative fields such as computer vision. A classical approach for parameter estimation is to work in a parameter space, also called preimage space, which correspond to a dual space approach. The Hough transform is a well known approximate parameter estimation method using a dual space that has been used to recognize primitives that may be imperfect [22, 12]. The basic idea is a voting procedure in a parameter (i.e. dual) space. With the rise of analytical digitization models, dual space approaches have become popular as they allow complete sets of solution parameters rather than just one such solution. Such approaches have been used for recognition [13–15, 26, 16, 6] and fitting methods [8, 7, 23, 31, 30, 25]. The preimages of every pixel of the given set are computed according to the chosen digitization model. Those duals are then intersected to obtain a geometric region that gathers all the parameters that solve the initial problem. The preimage intersection technique works well mainly for the Standard model [2, 3] where the preimage can be easily decomposed into two convex subparts which simplifies their intersection computation. This is not as simple for the Naive model [24] even in the basic case of straight segment recognition.

In this paper, we propose a method based on a double dual space and preimage intersection computation while incrementally scanning the digital points forming the primitive.

We consider the following recognition problem: *Given a set S of digital points and given a digitization model, is there a polynomial curve C defined with two parameters whose digitization according to the chosen model corresponds to S. If such curve exists, provide one or the whole set of parameters that solves the problem.* This paper is a first step toward a generic curve recognition ; we chose to start with the recognition of two parameter polynomial curves which allows us to stay in 2D throughout the whole process. Our main result is the transformation of the problem of finding the parameters A and B of the digitization of an implicit polynomial curve of the form $C(x, y) :\ x^i \times y^j - B \times x^k \times y^l - A = 0$ into a convex line stabbing problem. As digitization model, we consider the 1-Flake digitization model [28] which is related to the Naive digitization model [5].

The remainder of the paper is organized as follows: Section 2 presents the basic definitions and notations. This section also deals with the classical preimage dual used to recognize line segments into digital images using the naive model [16]. Section 3 presents the particularities of this dual applied on two parameter curves of the form $C(x, y) :\ x^i \times y^j - B \times x^k \times y^l - A = 0$. The next section presents the second dual used to transform the non linear problem into a linear one and the line stabbing solutions that solve our problem. The last section presents the implemented algorithm and its results. The paper ends with a conclusion and some perspectives.

2 Preimage definition for digital line recognition in 2D

This section presents the flake digitization model used as well as the preimage definition proposed by M. Dexet [16] and used for straight segment recognition and linear segmentation of digital contours with the Supercover digitization model [4].

2.1 Digitization model

Let us start with some basic notions and notations. Let $\{e_1, \ldots, e_n\}$ denote the canonical basis of the n-dimensional Euclidean vector space. Let $\mathbb{Z}^n$ be the subset of $\mathbb{R}^n$ that consists of all the integer coordinate points. A *digital (resp. Euclidean) point* is an element of $\mathbb{Z}^n$ (resp. $\mathbb{R}^n$). We denote by x_i the i-th coordinate of a point or a vector x, that is its coordinate associated to e_i. A *digital (resp. Euclidean) object* is a set of digital (resp. Euclidean) points. We denote $A \oplus B$ the Minkowksi sum of sets A and B such that $A \oplus B = a + b : a \in A, b \in B$. For all $k \in \{0, \ldots, n-1\}$, two integer points v and w are said to be *k-adjacent* or *k-neighbors*, if for all $i \in \{1, \ldots, n\}$, $|v_i - w_i| \leq 1$ and $\sum_{j=1}^{n} |v_j - w_j| \leq n - k$.

In the 2-dimensional plane, the 0- and 1-neighborhood notations correspond respectively to the classical 8- and 4-neighborhood notations.

Each adjacency relationship can be linked to an adjacency norm introduced by J-L. Toutant [29]. The *k-adjacency norm* $[\cdot]_k, 0 \leq k < n$, is defined as:

$$\forall x \in \mathbb{R}^n, [x]_k = \max\left\{\|x\|_\infty, \frac{\|x\|_1}{n-k}\right\}.$$

It is easy to see that, for v and $w \in \mathbb{Z}^n$ are k-adjacent *iff* $[v-w]_k \leq 1$, hence the name.

Let us denote $\mathcal{B}_{[\cdot]_k}$ be the unit ball (of radius $\frac{1}{2}$) under the norm $[\cdot]_k$. The associated distance is denoted by d_k. It is easy to see that the 0-adjacency norm corresponds to the Chebychev norm and the $(n-1)$-adjacency norm to the Manhattan norm.

Let us now focus on dimensions two. Note however that much of what follows can easily be extended to dimension n [5, 28]. We are considering morphological based digitizations schemes: the digitization of a Euclidean 2D object F is defined by $\mathcal{D}_{[\cdot]}(F) = \left(\mathcal{B}_{[\cdot]} \oplus F\right) \cap \mathbb{Z}^2$, where $[\cdot]$ is a norm and $\mathcal{B}_{[\cdot]}$ denotes its unit ball. See [5] for more details.

The unit ball for the Manhattan norm (1-adjacency norm) is called the Naive ball (cf. left part of Figure 1). The corresponding digitization is called the Naive digitization model [5]. In this paper, we are going to use a model that is related to the Naive model but that is simpler to use in practice. It is based on adjacency flakes. An *adjacency flake* can be described as the union of a finite number of straight segments joining the opposite vertices of a k-adjacency unit ball (see the right part of Figure 2 for the 1-Flake representation).

Definition 1 (1-Adjacency Flake [28, 29]).
Let $0 \leq k < 2$. A 2D k-adjacency flake, F_k is defined by:

$$F_k = \left\{\lambda u : \lambda \in \left[0, \frac{1}{2}\right], u \in \{-1,0,1\}^2, \sum_{i=1}^{2} |u_i| = 2-k\right\}.$$

We are, for what follows, considering a digitization with the 2D adjacency 1-Flake for implicitly defined curves. Let $\mathcal{C}$ be an implicit curve $\mathcal{C} = \left\{p \in \mathbb{R}^2 : C(p) = 0\right\}$ which separates space into one (or several) region(s) where $C(p) < 0$ and one (or several) region(s) where $C(p) > 0$:

Definition 2 (1-Flake Digitization [28, 5, 29]). *The 1-Flake digitization of a 2D curve C is defined by $\mathcal{D}_{F_1}(C) = \left\{v \in \mathbb{Z}^2 : (v \oplus F_1) \cap C \neq \varnothing\right\}$.*

When the curve C verifies a regularity condition (when C is r-regular) [27, 28], then the 1-Flake digitization can be analytically characterized by considering only the vertices of the 1-Flake:

Definition 3 (Simple Analytical 1-Flake Digitization [28]).

$$\mathcal{A}_1(C) = \left\{v \in \mathbb{Z}^2 : \begin{array}{r} min\{C(p) : p \in (v \oplus F_1)\} \leq 0 \\ and\ max\{C(p) : p \in (v \oplus F_1)\} \geq 0 \end{array}\right\}.$$

When the curve verifies the regularity conditions then $\mathcal{A}_1(S) = \mathcal{D}_{F_1}(S)$ and thus verifies all the properties of morphological digitizations and Adjacency flake digitizations. Otherwise there are some differences that may in some cases create topological problems [28]. This is largely compensated by the fact that $\mathcal{A}_1(S)$ is easy to construct while $\mathcal{D}_{F_1}(C)$ may not.

2.2 Generalized preimage definition [16]

The generalized preimage has been used for digital curve segmentation into digital line segments. It is a dual space where a point in the image space is transformed into a line and reversely, a line into a point:

Definition 4. *Let $P(x_1, ..., x_n)$ be a nD Euclidean point. The dual of P noted $\mathcal{D}(P)$ is defined in the $(y_1, ..., y_n)$ space by:*

$$y_n = x_n - \sum_{i=1}^{n-1} y_i \times x_i.$$

Figure 1 presents the preimages for the Naive digitization model.

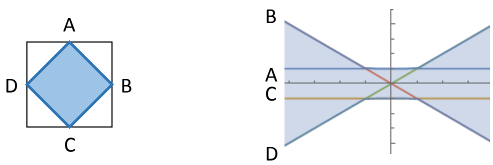

Fig. 1. Generalized preimage for the Naive unity balls centered on $(0, 0)$.

The principle of the curve segmentation method [16] is to incrementally compute the intersection of the preimages of the balls centered on the digital points until it becomes empty. The polygon obtained in the last step before emptyness gathers all the parameter solutions for the point subset (i.e. all the line segment parameters corresponding to those digital points). The algorithm starts again with the last considered point until reaching all the points of the curve. The result is a set of line segments that, when digitizated, form the digital curve. This method has been applied for the Supercover digitization model since, in this case, the preimage is easily decomposed into two convex parts. Those convex parts are then easily intersected (see [16] for more details).

In this paper, we will use the same idea but we are going to adapt it to the Simple Analytical 1-Flake Digitization. We are also going to extend the method to recognize some implicitly defined curves instead of just lines.

3 Preimage property for 2 parameter functions

In this section, we adapt the pixel preimage definition to the following two parameter polynomial implicit curves:

$$C(x, y): \ x^i \times y^j - B \times x^k \times y^l - A = 0.$$

In the following, the preimage space is also called dual 1 to mark the difference with the second dual that is going to be introduced later on.

In the particular cases where, among the given powers i, j, k or l, some are equal to 0, we consider that the corresponding term is removed: for instance the curve $x^i \times y^j - B \times x^k \times y^l - A = 0$, with $i = 1, j = 0, k = 0, l = 1$, is rewritten as: $x - B \times y - A = 0$. The main reason is to avoid the 0^0 issue during the dual computation.

Definition 5. *The dual of the function* $C(x, y): \ x^i \times y^j - B \times x^k \times y^l - A = 0$ *is the point* (A, B)*.*

With this definition, the dual of a curve in the image is a point. Figure 2 presents the preimages of both the Naive and the corresponding simplified 1-Flake models (see $\mathcal{A}_1(C)$ in the previous section). Note that in the Naive preimage there is a parabolic part that does not exist in the simple 1-Flake preimage. It is easy to see that every point in the parabolic part corresponds to curves that scrub the naive ball without fully intersecting it. These curve are discarded with the simple 1-Flake digitization model.

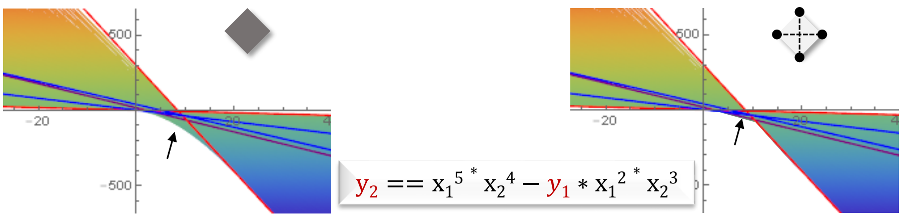

Fig. 2. Preimage of the Naive(Left) and 1-Flake (Right) representations of the 2D point $(1, 2)$ with $i = 5$, $j = 4$, $k = 2$, $l = 3$. The vertices preimages are emphasized.

In the curve segmentation algorithm into digital straight line segments algorithm for the Supercover model presented in [16, 4], the solutions are given by intersecting the preimages of every $[\cdot]$ball centered on the points of the contour. In the Supercover case, the non-convex preimage can be simply split in two convex parts that lie in a set half-space. The intersections are simple to compute. In our case, to compute a recognition solution we have to intersect non-convex complex objects which cannot be simplified as easily as for the Supercover case.

The parabolic parts in the preimage induced by scrubing curves for the Naive model add to the intersection computation complexity.

The geometrical properties of the preimage in some particular cases may allow us to divide the preimage into convex subparts but the intersection of the preimages remains difficult to compute. To solve these drawbacks, we will firstly only consider the 1-Flake digitization model. Secondly, we propose to use a second dual space with the classical linear duality (a point gives a line, a line gives a point). As we will see in the next section this second dual space will simplify the intersection computation.

4 Second Dual space: linearization of the recognition problem

Let us note that the starting point is an image with pixel curve. Each pixel is then identified to its digital center and then with the 1-Flake digitization model, a 1-Flake is centered on each digital point. For the sake of simplicity, we are sometimes speaking about pixels rather then the 1-Flake centered on the digital point of the curve. The second dual space allows to transform the non-convex preimage of a pixel (1-Flake) into a convex quadrilateral. In this second dual space, a line that stabs a quadrilateral corresponds to a point inside the first preimage which in turn corresponds to a function that crosses the pixel in the initial image. The lines that stab all the quadrilaterals associated to every curve pixels in the initial image correspond to two parameter implicit polynomial curves which digitization is the given set of pixels.

4.1 Second dual definition

Definition 6 (Second Dual). *A point (A, B) in the dual 1 space corresponds to the line $z_2 = A \times z_1 + B$ in the dual 2 space. The line $A = -z_1 \times B + z_2$ in the dual 1 space corresponds to the point (A, B) in the dual 2 space.*

From Definition 6, we can define the second dual of the 1-Flake centered on a digital point directly:

Property 1 (1-Flake second dual). The quadrilateral that is the second preimage of the 1-Flake centered on the point $P(x, y)$ considering the function F of the form $x^i \times y^j - B \times x^k \times y^l - A = 0$ is defined by the four points which correspond to the four vertices of the 1-Flake centered on P:
$Dual2(x + \frac{1}{2}, y) = \big((x + \frac{1}{2})^k \times y^l, (x + \frac{1}{2})^i \times y^j\big)$,
$Dual2(x - \frac{1}{2}, y) = \big((x - \frac{1}{2})^k \times y^l, (x - \frac{1}{2})^i \times y^j\big)$,
$Dual2(x, y + \frac{1}{2}) = \big(x^k \times (y + \frac{1}{2})^l, x^i \times (y + \frac{1}{2})^j\big)$,
$Dual2(x, y - \frac{1}{2}) = \big(x^k \times (y - \frac{1}{2})^l, x^i \times (y - \frac{1}{2})^j\big)$.

Recall that function $x^i \times y^j - B \times x^k \times y^l - A = 0$ with $i = 1, j = 0, k = 0, l = 1$ is rewritten in the following way : $x - B \times y - A = 0$. The second dual definition is affected in the same way if 0 powers are involved.

Property 2. In the general case, the dual of a pixel is a quadrilateral. In some cases, the quadrilateral can collapse to a triangle, to an edge and even to a single point. This depends on the parity of i, j, k and l and if one or both coordinates of the point are equal to zero.

The dual of any pixel centered on $(x \neq 0, y \neq 0)$ is a quadrilateral whatever the parity of the powers. The dual of the pixel $(0,0)$ collapses to the point $(0,0)$ whenever all the powers are not zero. The dual of the point $(0,0)$ collapses to the edge $[(0,(1/2)^{2u}),((1/2)^{2v},0)]$ for $i = 0, j = 2u, k = 2v, l = 0$ with u, v positive integers.

Property 3. Between one and four points can have the same dual quadrilateral depending on the parity of i, j, k and l.

We can see in Figure 3 the Naive ball for the point $(1,2)$ in the image space, the preimage of its associated 1-Flake in the first dual space and finally the corresponding quadrilateral in the second dual space.

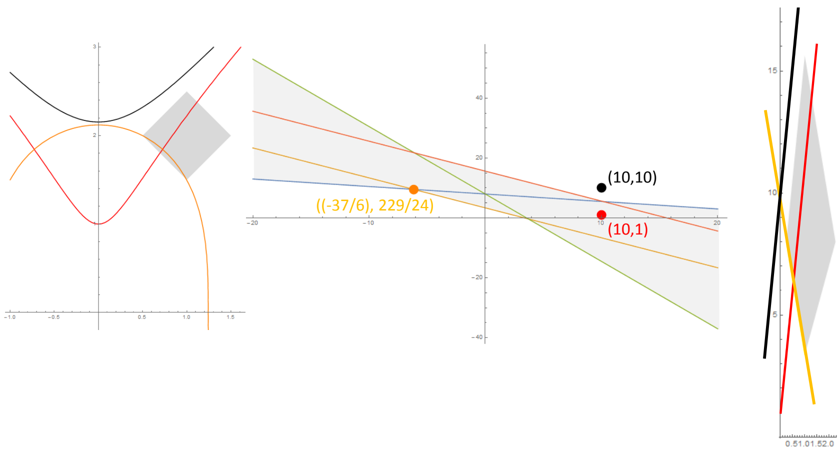

Fig. 3. The primal space (Image), the first dual and second dual for the 1-Flake associated with point $(1,2)$ considering the curve $y^3 - B \times x^2 - A = 0$. The emphasized curves are $y^3 - 10 * x^2 - 1 = 0$, $y^3 - (-37/6) * x^2 - 229/24 = 0$ and $y^3 - 10 * x^2 - 10 = 0$.

In Figure 3 we have highlighted three elements:

- The yellow point corresponds to the intersection of two limit lines of the preimage which also corresponds to an edge in the second preimage.
- The red point is in the middle of the preimage and corresponds to a line that crosses the second preimage and therefore to a function which digitization contains the pixel.
- The black point is out of the preimage and corresponds to a line that avoids the second preimage quadrilateral and thus to a function that does not crosses the pixel.

A line that stabs all the quadrilaterals in the second dual space corresponds to a curve function whose 1-Flake digitization contains all the given pixels. Finding such line(s) is a well know convex polygon transversal problem. In the next section we study some line stabbing solutions.

In Figure 4, we show some examples of how the second dual transformation affects the $\mathcal{Z}^2$ plane depending on the tuple (i, j, k, l). This also illustrate the property 3: when j is even the dual of the half plane $y <= 0$ is superimposed with the dual of the half plane $y >= 0$. The same occurs when k is even for the $x <= 0$ and $x >= 0$ half planes.

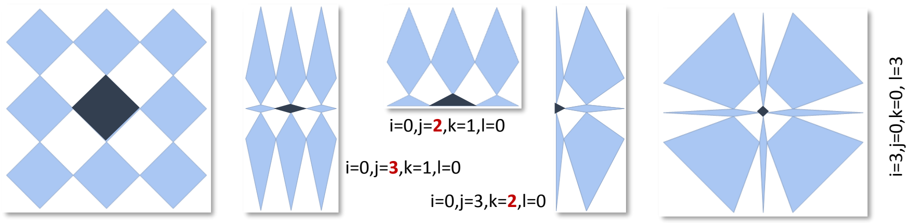

Fig. 4. Second dual for several n-uplets (i, j, k, l). The second dual of the pixel $(0, 0)$ (colored black) is emphasized in each case.

5 Recognition algorithm for 2 parameter digital function and results

This section presents a state of the art on the transversal or line stabbing problems and then it explains the chosen solution to solve them.

5.1 State of the art transversal/line stabbing

Let S be a set of geometric objects in R^d. A $k-$transversal to the set S is a $k-$dimensional flat that intersects each object of the set S. This problem is known as the stabbing problem or transversal. Transversal or stabbing have received significant attention in the 80s. Many researchers in the field of computational geometry studied the problem [18, 17, 9, 21, 19, 10]. Edelsbrunner in [18] presented an algorithm for determining a line that intersects n given line segments in $O(n \log n)$ time. Avis and Wenger [11] presented another algorithm for finding a line that also intersects n line segments in $O(n \log n)$ time complexity. Others algorithms were also proposed for finding a line intersecting simple geometric objects such as circles [9]. As for finding a line stabbing convex polygons, an algorithm for stabbing non intersecting convex sets in the plane was proposed in [19]. In [10] a plane stabber in R^d for a set of m polyhedra with a total of N edges was proposed in $O(N^{d-1}m)$ and in [1] a line stabbing for convex polyhedra in 3D is proposed. For more details about stabbing, a comprehensive survey on geometric transversal theory can be found in [20].

5.2 Chosen algorithm

In this work we are interested in finding a line that stabs all the convex quadrilaterals in the second dual space which corresponds to a curve function in the primal space whose digitization contains all the given pixels. This problem is a problem of line stabbing convex polygons. We decided to apply the method proposed in [17] by M. Doskas to solve our stabbing problem because of its simplicity and the possibility to be extended to 3D and then to higher dimension using the extension proposed in [10]. It should be noted that this method is not optimal in 2D (1-transversal) since the algorithm in [9] solves this problem in $O(nlog(n)\alpha(n))$ where $\alpha(n)$ is the slow growing inverse Ackermann's function. However as said we are interested in future works in increasing the number of parameters of our polynomial function which explains our choice of this algorithm.

The idea behind the algorithm is based on the lemma that for a given convex polygon P and a given $a \in R$, there are exactly two supporting lines for P having as slope a, one upper and one lower. As a result of this lemma in order to find if a line of slope a stabs every polygon of a set, one should determine the contact vertices for the supporting lines with slope a for each polygon and then the appropriate range of displacement c. Finally from the intersection of these ranges the stabbing line $y = ax + c$, if it exists, is determined. This does not yield a finite algorithm however one can make it finite by the subdivision of the slope into regions where, in each region, it can be determined quickly if a line with a slope in that range exists. The overall complexity of the algorithm is $O(Nm + N \log N)$ where m is the number of polygons and N the total number of edges; in our case since each polygons has 4 edges the total number of edges is $4m$ so the overall complexity is $O(m^2)$.

5.3 Results

The second dual transformation and the line stabbing algorithm have been implemented using Mathematica. The algorithm starts by computing the second dual of each digital point. After this computation, duplicates (duplicates occurs if we have even powers in the function) of each point are deleted. The shortened list of second duals is then sent to the line stabbing procedure. This second algorithm starts with the analysis of the second duals: it separates quadrilateral duals from the degenerated ones and handles them differently in order to build a matrix. This matrix, that gathers all the slope constraints, is then solved using linear programming.

Figure 5 presents the recognition results of the implicit function $y^2 - B \times x^2 - A = 0$ on a digital circle. The emphasized pixels are the ones kept for the recognition, the gray ones are those that are deleted since they have the same dual as one of the former points. This allows to compute the solution with only 5 of the 20 starting quadrilaterals and allows thus to decrease the computation time. The actual computation time is less than a second for this example (0.2184014 s).

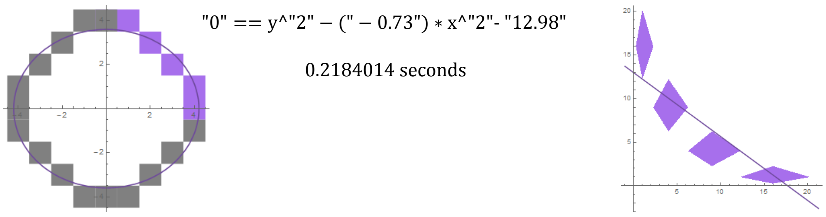

Fig. 5. Recognition of the implicit function $0 = y^2 - B \times x^2 - A$. A solution found is $A = 12.98$ and $B = -0.73$.

With the same number of starting pixels and the function $0 = x^1 \times y^3 - B \times x^2 \times y^1 - A$, our algorithm took less than a second to provide the solution even if, in this case, all the duals are distinct (see Figure 6).

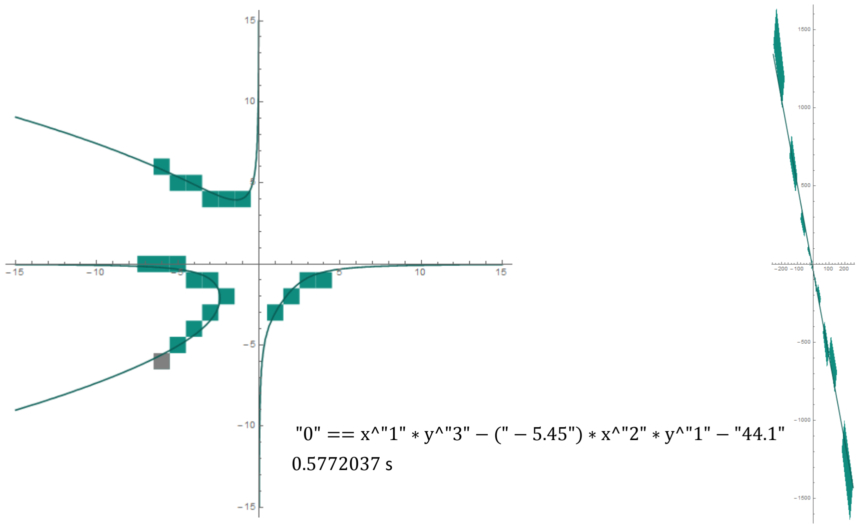

Fig. 6. Recognition of the implicit function $0 = x^1 \times y^3 - B \times x^2 \times y^1 - A$. The found solution is $A = -44.1$ and $B = -5.45$.

6 Conclusion and Perspective

In this paper, we have proposed an algorithm for the segmentation of any digital form using generic polynomial function of any degree with two parameters. The definitions and experiments have been proposed for the 1-Flake model however the same work can be directly extended to the 0-Flake digitization model. The two parameter constraints allows to stay in 2D for all the dual spaces: the dimension of our dual space corresponds to the number of parameters.

A direct extension of this work is the recognition of polynomial function having more than two parameters leading to work in nD for the second dual space. The main point to be investigated will be the existence of an efficient hyperplane stabbing solutions in any dimension. In [10] there is a hyperplane stabbing algorithm in higher dimension which finds the hyperplane that stabs each and every polyhedra of the set. We also want to extend the recognition problem to fitting type problem. In case a line that stabs every polygon does not exists, we will search the line that maximizes the number of stabbed polygons. Another perpective is to use this linear second dual to efficiently perform digital algorithms such as the incremental digitization of such functions.

References

1. P. K. Agarwal. On stabbling lines for convex polyhedra in 3d. *Computational Geometry*, 4(4):177 – 189, 1994.
2. E. Andres. Defining discrete objects for polygonalization: The standard model. In *DGCI, Bordeaux, France*, volume 2301 of *LNCS*, pages 313–325. Springer, 2002.
3. E. Andres. Discrete linear objects in dimension n: the standard model. *Graphical Models*, 65(1-3):92 – 111, 2003.
4. E. Andres. The supercover of an m-flat is a discrete analytical object. *Theor. Comput. Sci.*, 406(1-2):8–14, 2008.
5. E. Andres. Digital analytical geometry: How do i define a digital analytical object? In *Combinatorial Image Analysis: International Workshop, IWCIA 2015, Kolkata, India, November 24-27, 2015*, volume 9448 of *LNCS*, pages 3–17. Springer International Publishing, 2015.
6. E. Andres, G. Largeteau-Skapin, and M. Rodríguez. Generalized perpendicular bisector and exhaustive discrete circle recognition. *Graphical Models*, 73(6):354–364, 2011.
7. E. Andres, G. Largeteau-Skapin, R. Zrour, A. Sugimoto, and Y. Kenmochi. Optimal consensus set and preimage of 4-connected circles in a noisy environment. In *21st International Conference on Pattern Recognition (ICPR 2012)*, pages 3774–3777. IEEE Xplore, 2012.
8. E. Andres and T. Roussillon. *Analytical Description of Digital Circles*, pages 235–246. Springer Berlin Heidelberg, 2011.
9. Mikhail Atallah and Chanderjit Bajaj. Efficient algorithms for common transversals. *Information Processing Letters*, 25(2):87 – 91, 1987.
10. D. Avis and M. Doskas. Algorithms for high dimensional stabbing problems. *Discrete Applied Mathematics*, 27(1):39 – 48, 1990.
11. D. Avis and R. Wenger. Algorithms for line transversals in space. In *Proceedings of the 3rd ACM Conference on Computational Geometry*, pages 300–307, 1987.

12. D.H. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111 – 122, 1981.
13. R. Breton, I. Sivignon, F. Dupont, and E. Andres. Towards an invertible euclidean reconstruction of a discrete object. In *Discrete Geometry for Computer Imagery*, volume 2886 of *LNCS*, pages 246–256, Napoli, Italy, Nov. 2003. Springer Verlag.
14. R. Breton, I. Sivignon, F. Dupont, and E. Andres. Towards an invertible euclidean reconstruction of a discrete object. In *11th DGCI Naples (Italy)*, volume 2886 of *LNCS*, pages 246–256. Springer, 2003.
15. D. Coeurjolly, Y. Gerard, J.P. Reveillès, and L. Tougne. An elementary algorithm for digital arc segmentation. *Discrete Applied Mathematics*, 139(1-3):31–50, 2004.
16. M. Dexet and E. Andres. A generalized preimage for the digital analytical hyperplane recognition. *Discrete Applied Mathematics*, 157(3):476–489, 2009.
17. M. Doskas. Various stabbing problems in computational geometry. Master's thesis, School of Computer Science, McGill University, Montréal, 1987.
18. H. Edelsbrunner, H. A. Maurer, F. P. Preparata, A. L. Rosenberg, E. Welzl, and D. Wood. Stabbing line segments. *BIT Numerical Mathematics*, 22(3):274–281, 1982.
19. H. Edelsbrunner and M. Sharir. The maximum number of ways to stab n convex nonintersecting sets in the plane is $2n-2$. *Discrete and computational geometry*, 5(1):35–42, 1990.
20. J. E. Goodman, R. Pollack, and R. Wenger. New trends in discrete and computational geometry (j. pach, ed.). *Geometric transversal theory*, pages 163–198, 1993.
21. J. Hershberger. Finding the upper envelope of n line segments in o(n log n) time. *Information Processing Letters*, 33(4):169–174, 1989.
22. P.V.C. Hough. Method and means for recognizing complex patterns, Dec 1962.
23. G. Largeteau-Skapin, R. Zrour, and E. Andres. $O(n^3 \log n)$ time complexity for the optimal consensus set computation for 4-connected digital circles. In *17-th International Conference on Discrete Geometry for Computer Imagery (DGCI13)*, volume 7749, pages 241–252, Sevilla, Spain, 2013. LNCS, Springer Verlag.
24. J.-P. Reveillès. *Calcul en Nombres Entiers et Algorithmique*. PhD thesis, Université Louis Pasteur, Strasbourg, France, 1991.
25. F. Sekiya and A. Sugimoto. Discrete polynomial curve fitting guaranteeing inclusion wise maximality. In *The 13th Asian Conference on Computer Vision (ACCV'16), Taipei, Taiwan*, 2016.
26. I. Sivignon, R. Breton, F. Dupont, and E. Andres. Discrete analytical curve reconstruction without patches. *Image and Vision Computing*, 23(2):191–202, 2005.
27. P. Stelldinger and U. Köthe. Towards a general sampling theory for shape preservation. *Image and Vision Computing*, 23(2):237–248, 2005.
28. J.L. Toutant, E. Andres, G. Largeteau-Skapin, and R. Zrour. Implicit digital surfaces in arbitrary dimensions. In *18th DGCI, Siena (Italy)*, volume 8668 of *LNCS*, pages 332–343. Springer, 2014.
29. J.L. Toutant, E. Andres, and T. Roussillon. Digital circles, spheres and hyperspheres: From morphological models to analytical characterizations and topological properties. *Discrete Applied Mathematics*, 161(16-17):2662–2677, 2013.
30. R. Zrour, Y. Kenmochi, H. Talbot, L. Buzer, Y. Hamam, I. Shimizu, and A. Sugimoto. Optimal consensus set for digital line and plane fitting. *International Journal of Imaging Systems and Technology*, 21(1):45–57, 2011.
31. R. Zrour, G. Largeteau-Skapin, and A. Andres. Optimal consensus set for nd fixed width annulus fitting. In *17th International Workshop on Combinatorial Image Analysis (IWCIA15)*, volume 9448, pages 101–114. Springer Verlag, LNCS, 2015.