Optimal covering of a straight line applied to discrete convexity

Jean-Marc Chassery, Isabelle Sivignon

gipsa-lab, CNRS, UMR 5216, F-38420, France
jean-marc.chassery@gipsa-lab.grenoble-inp.fr
isabelle.sivignon@gipsa-lab.grenoble-inp.fr

Abstract. The relation between a straight line and its digitization as a digital straight line is often expressed using a notion of proximity. In this contribution, we consider the covering of the straight line by a set of balls centered on the digital straight line pixels. We prove that the optimal radius of the balls is strictly less than one, and can be expressed as a function of the slope of the straight line. This property is used to define discrete convexity in concordance with previous works on convexity.

1 Introduction

From the seminal work of Sklansky [12], discrete convexity has been the subject of many studies with a common objective: transcribe the Euclidean definition in the digital space.

In the eighties, a previous work introduced ε-convexity using covering of connected sets by balls [1]. In the case of convex shapes, it was shown in [1] that the value ε can be written as $\frac{p}{p+1}$ (lower than 1) where p is a parameter computed from the edges of the convex hull of the shape. However, the value of this parameter p was given algorithmically but not analytically. In this paper, using the arithmetical definition of a digital straight line [9, 3], we prove that ε can be expressed exactly as a function of the characteristics of the digital straight line supporting the edges of the convex hull of the shape.

After classical definitions of digital straight line and its characteristics, we establish the property of covering a straight line by balls centered exclusively on digital straight line pixels. The following section applies that property to the case of discrete convexity and we present the algorithm for discrete convex hull computation working only on digital space.

2 Preliminary definitions

Let $\mathcal{L}$ be an Euclidian straight line in $\mathbb{R}^2$ given by the equation $ax - by + \mu = 0$, with a, b, μ in $\mathbb{Z}$, and $\gcd(a, b) = 1$. In the following, we also assume without loss of generality that $0 \leq a \leq b$. All other cases are symmetrical. Such a straight line may be considered as supporting a linear contour of a shape, such that all the points of the shape are on the same side of the straight line.

Let us now consider the Object Boundary Quantization of $\mathcal{L}$ on the isothetic regular grid. It is given by the set of pixels (x, y) such that $x \in \mathbb{Z}$ and $y = \lfloor \frac{-ax-\mu}{b} \rfloor$ (see Figure 1).

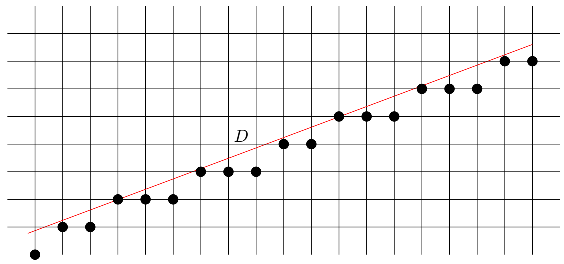


Fig. 1. A straight line $\mathcal{L}$ of equation $3x - 8y + \mu = 0$ and its OBQ digitization analytically given by the digital straight line of equation $0 \leq 3x - 8y + \mu < 8$.

It is well known that this set of digital points denoted by L is a simple 8-connected digital straight line (DSL) that can be defined by the diophantine equation [9, 3] $0 \leq ax - by + \mu < \max(|a|, |b|) = b$.

The slope of L is $\frac{a}{b}$, μ is the shift at origin. The remainder of L for a given digital point (x, y) is the value $ax - by + \mu$.

Two particular straight lines are associated to this DSL L:

– the upper leaning line given by equation $ax - by + \mu = 0$ and,
– the lower leaning line given by equation $ax - by + \mu = b - 1$.

The digital points lying on these lines are similarly called leaning points (see Figure 2). Upper leaning points have remainder value 0 while lower leaning points have remainder value $b - 1$ (see [2] for more details).

Since a, b, μ are integers and $\mathcal{L}$ is digitized with OBQ, $\mathcal{L}$ is identified as the upper leaning line of L.

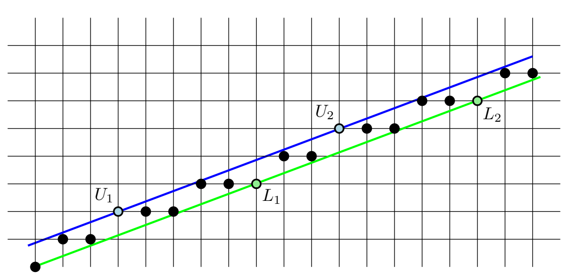


Fig. 2. Digital straight line of equation $0 \leq 3x - 8y + \mu < 8$ with upper (lower) leaning points U_i (L_i).

3 Optimal covering of a straight line

3.1 Setting the problem

The objective is to cover the straight line $\mathcal{L}$ with closed balls centered on the points of the DSL L. Moreover, the union of these balls shall not contain any other digital point. These balls are defined according to the L_∞ metric such that the ball of radius ε is defined by $B(P,\varepsilon) = \{q | d_\infty(P,q) \leq \varepsilon\}$. In our framework, we assume that the radius ε is the same for all the points of L (see Figure 3).

In the general case, setting ε to $\frac{1}{2}$ is not sufficient to cover the straight line $\mathcal{L}$ except for the special cases of horizontal, vertical and diagonal lines. We can notice the following elementary property:

Property 1. The band delimitated by the 2 leaning lines (upper and lower) has a vertical thickness of $\frac{b-1}{b}$.

The proof is straightforward from the equations of the leaning lines. Figure 3 illustrates the covering of a straight line of slope $\frac{a}{b}$ by balls of radius $\frac{b-1}{b}$. We remark that this radius is not sufficient since parts of the straight line remain uncovered.

To analyze this covering, we proceed by successive couples of adjacent pixels. It is easy to verify that the vertical distance between a pixel of remainder r of the DSL and the line $\mathcal{L}$ is equal to r/b, with r varying from 0 to $b-1$. The value $\frac{b-1}{b}$ is obtained for a lower leaning point. Let us consider two successive pixels of L such that the first one is a lower leaning point. Then we notice that the union of the balls of radius $\frac{b-1}{b}$ centered on these points leaves a part of $\mathcal{L}$ uncovered (see Figure 3). So the minimum ε for the line $\mathcal{L}$ to be covered is greater than $\frac{b-1}{b}$.

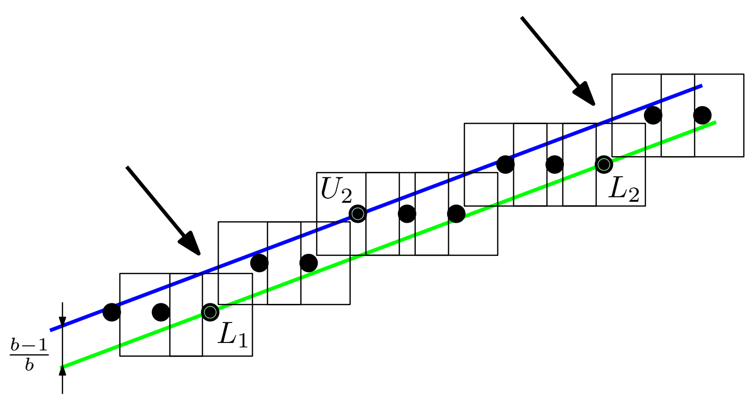


Fig. 3. Balls of radius $\varepsilon = \frac{b-1}{b}$ centered on the DSL points do not cover $\mathcal{L}$: uncovered regions are indicated by arrows.

In the following, we denote by $P_0, P_1, \ldots, P_n$ the ordered sequence (increasing abscissae) of pixels of the DSL L. For each pixel P_i we can define as $proj(P_i)$

the vertical projection of P_i on the straight line $\mathcal{L}$. Consequently, the Euclidean straight line $\mathcal{L}$ may be partitionned as the union of subsegments $[proj(P_i), proj(P_{i+1})]$.

Property 2. If for every pair of successive pixels P_i and P_{i+1}, $B(P_i,\varepsilon) \cup B(P_{i+1},\varepsilon)$ covers the straight segment $[proj(P_i), proj(P_{i+1})]$ then the straight line $\mathcal{L}$ is covered by the union of the balls $B(P_i,\varepsilon)$.

As said before, ε must be greater than $\frac{b-1}{b}$, and it is easy to see that it must also be strictly lower than 1, otherwise new digital points are included in the union of balls.

3.2 Optimal covering

The following theorem defines the optimal value of ε as a function of the straight line parameters a and b.

Theorem 1. *Let $\mathcal{L}$ a straight line of equation $ax-by+\mu=0$ and L its digitization with the OBQ scheme. In the context of the previous definitions, the union of balls $B(P_i,\varepsilon)$ centered on pixels of the DSL L with radius $\varepsilon = \max\left(\frac{1}{2}, \frac{|a|+|b|-1}{|a|+|b|}\right)$ covers the straight line $\mathcal{L}$. This set doesn't contain any other digital pixels excepted those of the DSL.*

Proof. We suppose that $0 \le a \le b$. First of all, if the parameters of $\mathcal{L}$ are $(0,1,\mu)$ (horizontal straight line) or $(1,1,\mu)$ (diagonal straight line), the optimal value of ε is trivially equal to $\frac{1}{2}$. Otherwise, b is greater than or equal to 2.

The distance between a point P_i and its projection $proj(P_i)$ (see above) is equal to $\frac{r}{b}$ if the remainder of P_i is equal to r.

In the case "P_i and P_{i+1} 4-connected", if r is the remainder of P_i then $r+a$ is the remainder of P_{i+1}. Consequently, the distance between P_{i+1} and its projection $proj(P_{i+1})$ is equal to $\frac{r+a}{b}$. We have the following inequalities:

- $\frac{b-1}{b} > \frac{r}{b}$ (P_i belongs to L),
- $\frac{b-1}{b} \ge \frac{r+a}{b}$ (P_{i+1} belongs to L),
- $2\frac{b-1}{b} \ge 1$ (since $b \ge 2$).

With the first two inequalities, we deduce that $proj(P_i)$ and $proj(P_{i+1})$ are covered by $B(P_i, \frac{b-1}{b}) \cup B(P_{i+1}, \frac{b-1}{b})$. The third inequality ensures that $B(P_i, \frac{b-1}{b})$ and $B(P_{i+1}, \frac{b-1}{b})$ overlap. We can conclude that the union of the two balls $B(P_i, \frac{b-1}{b}) \cup B(P_{i+1}, \frac{b-1}{b})$ covers the segment $[proj(P_i), proj(P_{i+1})]$ (see Figure 4).

In the case "P_i and P_{i+1} 8-connected", according to the Figure 5(b), we introduce the point Q belonging to the straight line $\mathcal{L}$ sita at equal distance (L_∞ norm) from P_i and P_{i+1}. The point Q is on $\mathcal{L}$ and shall belong to the two balls $B(P_i,\varepsilon)$ and $B(P_{i+1},\varepsilon)$. If r is the remainder of P_i, we denote by $\varepsilon(r)$ the minimum radius such that $B(P_i,\varepsilon(r)) \cup B(P_{i+1},\varepsilon(r))$ covers the segment

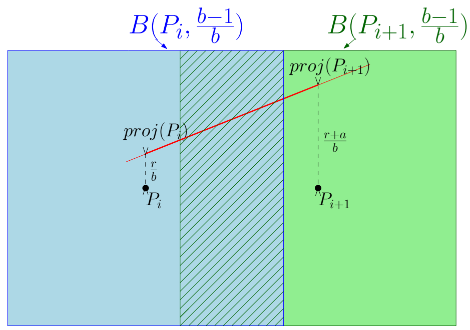


Fig. 4. When P_i and P_{i+1} are 4-connected, $B(P_i, \frac{b-1}{b}) \cup B(P_{i+1}, \frac{b-1}{b})$ covers the segment $[proj(P_i), proj(P_{i+1})]$.

$[proj(P_i), proj(P_{i+1})]$. We can write $\varepsilon(r)$ as the sum $\frac{r}{b} + \rho(r)$, with $\rho(r) \geq 0$ (see Figure 5). Using Thalès theorem, we have

$$\frac{\rho(r)}{\frac{a(b-r)}{b^2}} = \frac{\frac{b-r}{b} - \rho(r)}{\frac{b-r}{b}}$$

We obtain

$$\rho(r) = \frac{a(b-r)}{b(a+b)}$$

Then $\varepsilon(r) = \rho(r) + \frac{r}{b} = \frac{a+r}{a+b}$.

$\varepsilon(r)$ is increasing and is maximum for $r = b-1$, corresponding to $\varepsilon = \frac{a+b-1}{a+b}$.

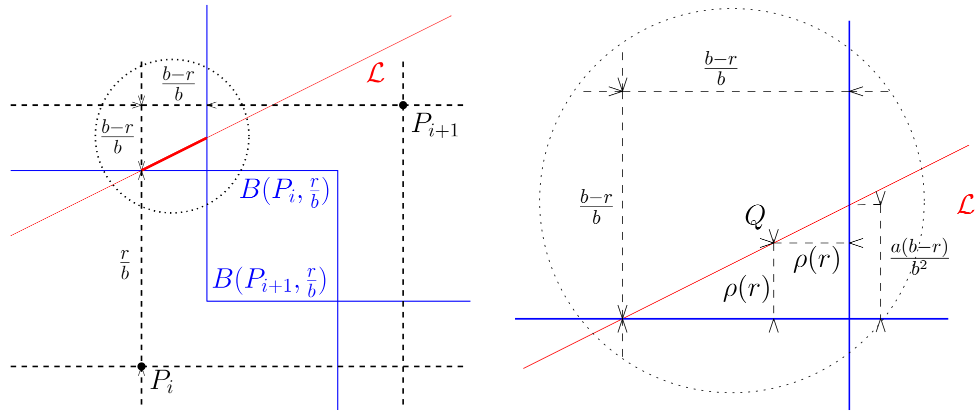
This proof was developed in the case $0 \leq a \leq b$. It can be easily extended in the general case to obtain $\varepsilon = \max(\frac{1}{2}, \frac{|a|+|b|-1}{|a|+|b|})$. Since $\varepsilon < 1$, P_i is the only digital point in the ball $B(P_i, \varepsilon)$ for all i, which ends the proof.

4 Discrete Convexity

4.1 Definitions

An important literature has been developed for digital convexity [12, 7, 5, 6, 1]. A large number of these definitions are considered to be equivalent in case of simply connected sets [10, 4].The following definition is issued from the transcription of convexity definition in Euclidean space to discrete space. In Euclidean geometry, a region $\mathcal{R}$ is convex if and only if for every pair of points p, q belonging to $\mathcal{R}$, the straight line segment $[p, q]$ is included in $\mathcal{R}$. The following definition of discrete convexity replaces inclusion by covering with balls [1].

Definition 1 (ε-convexity). *A connected component S is ε-convex, with ε belonging to interval $[\frac{1}{2}, 1[$ if:*



(a) (b)

Fig. 5. Illustration of the proof when P_i and P_{i+1} are 8-connected. r is the remainder of P_i. (b) is a close up of the region circled in (a).

– *for every pair of pixels P and Q of S,*
– *for every real value α belonging to $]0, 1[$,*

there exists a pixel R belonging to S such that the point $(\alpha P + (1-\alpha)Q)$ of the straight line supported by the two points PQ belongs to the balls $B(R, \varepsilon)$, centered on R with radius ε.

Definition 2 (Discrete convexity). *A connected component S is discrete convex if there exists a real $\varepsilon \in [\frac{1}{2}, 1[$ such that S is ε-convex.*

4.2 Algorithmic approach

The computation of the convex hull $Conv(S)$ is done in two steps :

1. first, the $x-y$ convex shape issued from S is computed;
2. then the convex hull is computed from the $x-y$ convex shape.

The $x-y$ convex shape of a connected component is defined as the convex shape along horizontal and vertical directions: for any points P and Q of the $x-y$ convex shape, if P and Q are on the same line or column of the grid, then all the points between P and Q (on this line or column) also belong to the $x-y$ convex shape. It is evident that the $x-y$ convex shape is included in the convex hull.

Algorithm 1 describes how to compute the $x-y$ convex shape. Suppose that the shape S is included in a binary image of size $m \times n$. The initialization step consists in:

1. compute the indices i_f and i_l of the upper and lower lines containing points of S respectively;

2. for each line i between i_f and i_l, compute $min(i)$ and $max(i)$ as the minimum and maximmum indices of points of S on line i.

This initilization step is done in $\mathcal{O}(nm)$. Afterwards, three

---

Algorithm 1: $x - y$ convex shape computation of a set S of discrete points

---

1 **for** i *from* i_f *to* i_l **do**
  $min_\downarrow(i) = \min(min(i), min_\downarrow(i-1))$; `//` $min_\downarrow(i)$ `is initially set to` m
  $max_\downarrow(i) = \max(max(i), max_\downarrow(i-1))$; `//` $max_\downarrow(i)$ `is initially set to` 0
end
2 **for** i *from* i_l *to* i_f **do**
  $min_\uparrow(i) = \min(min(i), min_\uparrow(i+1))$; `//` $min_\uparrow(i)$ `is initially set to` m
  $max_\uparrow(i) = \max(max(i), max_\uparrow(i+1))$; `//` $max_\uparrow(i)$ `is initially set to` 0
end
3 **for** i *from* i_f *to* i_l **do**
  $min(i) = \max(min_\downarrow(i), min_\uparrow(i))$
  $max(i) = \min(max_\downarrow(i), max_\uparrow(i))$
  $j_f = \min(j_f, min(i))$; `//` j_f `is initially set to` n
  $j_l = \max(j_l, max(i))$; `//` j_l `is initially set to` 0
end

Figure 6(b) illustrates the first two loops of the algorithm (lines 1 and 2). Figure 6(c) illustrates the third loop (line 3).

At the end of Algorithm 1, the variables $min(i)$ and $max(i)$ contain the indices of the minimum and maximum points of the $x - y$ convex shape for the line i. This algorithm also computes the coordinates i_f, i_l, j_f, j_l of the bounding rectangle of S.

Finally, the convex hull of S is computed as follows. To do so, we define the extremal point of the bounding rectangle as the points of S belonging to this rectangle and such that at least one of its two neighbours on the rectangle does not belong to S. These eight points are depicted on Figure 6(d).

Finally, the convex hull is computed from these extremal points using a technique similar to the algorithm of Sklansky [13][8](Chap. 13) on the polygon defined by the variables $min(.)$ and $max(.)$. Note that this polygon is simple and completely visible from the outside, such that the algorithm works in this case. This algorithm works in $\mathcal{O}(\max(m, n))$ time, which leads to a global complexity of $\mathcal{O}(nm)$ to compute the convex hull of S. The result on the example of Figure 6 is given in Figure 7.

4.3 Optimal convering and discrete convexity

Let us denote by $\{P_i, i = 0..n\}$ the ordered set of vertices of the convex hull $Conv(S)$. To each edge $[P_i, P_i + 1]$ we associate the slope parameters (a_i, b_i) such that $\gcd(a_i, b_i) = 1$. From Property 1, this edge is covered by balls centered

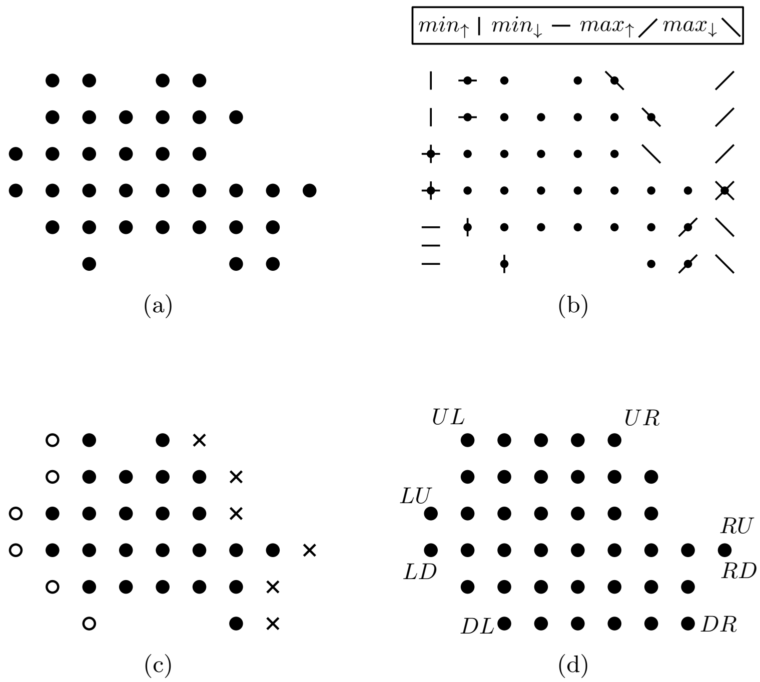


Fig. 6. (a) A discrete shape S. (b) Illustration of the variables $min_\downarrow$, $min_\uparrow$,$max_\downarrow$, $max_\uparrow$ used in Algorithm 1. (c) Illustration of the variables $min(.)$ and $max(.)$ at the end of Algorithm 1. (d) Extremal points of S.

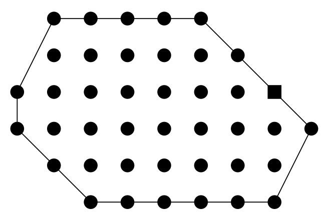
on the pixels of its OBQ digitization with radius $\max(\frac{1}{2}, \frac{|a_i|+|b_i|-1}{|a_i|+|b_i|})$.

It is easy to prove that the connected set $\mathrm{Conv}(S)$ of discrete points included in $Conv(S)$ is ε-convex with $\varepsilon = \max_{i \in 0 \dots n} \max(\frac{1}{2}, \frac{|a_i|+|b_i|-1}{|a_i|+|b_i|})$. S is discrete convex if and only if it is equal to $\mathrm{Conv}(S)$.

Moreover such definition is fully compatible with the continuous one as it is proved by the theorem [1]:

Theorem 2. *Let $\mathcal{S}$ be a connected component in space $\mathbb{R}^2$.*
If, for every sampling step, the discrete connected component S attached to $\mathcal{S}$ is discrete convex, then $\mathcal{S}$ is convex in space $\mathbb{R}^2$.

Using S and $\mathrm{Conv}(S)$ a lot of features can be extracted to estimate a measure of convexity of S (number of missing pixels, distribution of concavities)[11]. These features are used to measure the degree of convexity of a shape in presence of many concavities.



(a)

Fig. 7. Convex hull of the discrete shape S: the point marked by a square is added during the application of Sklansky's algorithm. The discrete points belonging to the polygonal line are the vertices of $Conv(S)$. Conv(S) is defined as the discrete points inside $Conv(S)$.

5 Conclusion

The transcription of continuous concept like convexity into digital space needs specific attention for its definition as well as for its properties. In case of convexity we use the definition of digital straight line and we replace the notion of inclusion by the notion of covering. The main result obtained in this paper is issued from parametric representation of digital straight line to characterize the optimal radius for covering balls centered on digital straight line pixels.

References

1. Chassery, J.M.: Discrete convexity: Definition, parametrization, and compatibility with continuous convexity. Computer Vision, Graphics, and Image Processing 21(3), 326 – 344 (1983)
2. Coeurjolly, D., Montanvert, A., Chassery, J.M.: Géométrie discrète et images numériques. Hermès (2007), traité IC2, série signal et image
3. Debled-Rennesson, I., Reveillès, J.P.: A linear algorithm for segmentation of digital curves. International Journal of Pattern Recognition and Artificial Intelligence 9(6), 635–662 (1995)
4. Eckhardt, U.: Digital lines and digital convexity. In: Digital and image geometry: advanced lectures, Lecture Notes in Computer Science, vol. 2243, pp. 209–228. Springer (2001)
5. Kim, C.E.: Digital convexity, straightness, and convex polygons 4(6), 618–626 (1982)
6. Kim, C.E., Rosenfeld, A.: Digital straight lines and convexity of digital regions 4(2), 149–153 (1982)
7. Kim, C.E., Sklansky, J.: Digital and cellular convexity. Pattern Recognition 15(5), 359–367 (1982)
8. Klette, R., Rosenfeld, A.: Digital Geometry: Geometric methods for digital picture analysis. Morgan Kaufmann (2004)

9. Reveilles, J.P.: Géométrie discrète, calcul en nombres entiers et algorithmique. Thèse d'etat, Université Louis Pasteur, Strasbourg, France (1991)
10. Ronse, C.: A bibliography on digital and computational convexity (1961-1988) 11(2), 181–190 (1989)
11. Roussillon, T.: Algorithmes d'extraction de modèles géométriques discrets pour la représentation robuste des formes. Ph.D. thesis, Université Lumière Lyon 2 (2009)
12. Sklansky, J.: Recognition of convex blobs. Pattern Recognition 2(1), 3–10 (1970)
13. Sklansky, J.: Measuring concavity on a rectangular mosaic. IEEE Transactions on Computers (12), 1355 – 1364 (1972)