

Construction of Digital Ellipse by Recursive Integer Intervals

Papia Mahato^(✉) and Partha Bhowmick

Department of Computer Science and Engineering, Indian Institute of Technology,
Kharagpur, India
papiamahatostar@gmail.com, bhowmick@gmail.com

Abstract. In this paper, we revisit the problem of ellipse construction in the integer plane. Our perspective is elementary number-theoretic analysis of a digital ellipse having an integer point as its center and two integer values specifying the lengths of its semi-major and semi-minor axes. We characterize a digital ellipse to derive certain recurrences on the integer intervals that contain the integer values of a specific square term corresponding to the integer points comprising the digital ellipse. This, in turn, helps in designing ellipse drawing algorithm on the integer plane. We propose two algorithms—one using floating-point-based distance computation, and another using purely integer operations. Some test results have also been presented to exhibit further research possibilities related to digital ellipse.

Keywords: Digital ellipse · Digital geometry · Integer intervals · Integer algorithm

1 Introduction

Ellipse is an important geometric primitive with a multitude of applications in different fields of science and engineering. Hence, like various other geometric primitives, its characterization and discretization is one of the necessary and interesting research topics in the subject of digital geometry. Its computational gamut in the discrete space brings in fundamental theoretical issues, which are related to several fields of computer science, and to name a few, these are computer graphics, computer vision, projective geometry, and image analysis.

Several methodologies have been proposed over the last fifty years for generation of digital ellipse, which may be seen in [3, 6, 8–13]. The adopted techniques are predominantly based on incremental raster approximation [8], double-step generation [11, 12], run slicing [13], and hybridization [5]. Most of these algorithms basically owe in concept to the one proposed in the very early stage of digitization [2].

Unlike other 2D geometric primitives like straight lines and circles, which have been continuously and extensively studied by the digital geometers, ellipse is possibly not yet studied up to its merit till date. This motivates us to focus on making out some novel characterization of ellipse in the discrete space—in particular, in \mathbb{Z}^2 —with the objective of designing efficient algorithms as per the

requirement of different applications. We report in this paper our first set of results, which centers around some of the elementary number-theoretic properties of digital ellipse when it has an integer specification. It culminates to certain interesting recurrences on integer intervals that correspond to the integer points comprising a digital ellipse.

The paper is organized as follows. In Sect. 2, we explain the preliminary concepts and the theoretical framework adopted in our work. In Sect. 3, we derive some elementary number-theoretic properties of digital ellipse. These properties are used in Sect. 4 for designing efficient algorithms on construction of digital ellipse. Here we first propose an algorithm that uses a few floating-point operations, which we improve to a second algorithm that uses strictly integer operations. In Sect. 5, we present some test results and point out further research issues related to digital ellipse.

2 Preliminaries

We start with some definitions and metrics in 2D space, which are used in the sequel. We mostly follow the conventions from [7].

An *integer point* means a point with integer coordinates. A *pixel* is equivalent to a *2-cell*, perceived as a unit square on the xy -plane, and hence uniquely identified by its center, as it is an integer point. Two pixels are said to be *0-adjacent* if they share a vertex (*0-cell*) and *1-adjacent* if they share an edge (*1-cell*). A *0-connected* (digital) curve is a sequence of pixels such that every two consecutive pixels are 0- or 1-adjacent (i.e., in 8-neighborhood). For a *1-connected* curve, every two consecutive pixels in the sequence have to be 1-adjacent (i.e., in 4-neighborhood). A curve is *open-ended* if it has two distinct endpoints, s and t . A curve is *closed* if it partitions the integer plane into an interior and an exterior. If removal of a pixel p from an open-ended curve does not break the connectedness between s and t , then p is said to be a *simple pixel*. For a closed curve, on the other hand, removal of a simple pixel does not give rise to connectedness between its interior and exterior. For further details on simple pixel, we refer to [7]. If a curve does not contain any *simple pixel*, then it is said to be *irreducible*. In an open-ended irreducible curve, s and t has one neighbor each and the rest have two neighbors each. For a closed irreducible curve, each constituent pixel has two neighbors each. Hence, for an open-ended irreducible k -connected curve, where $k \in \{0, 1\}$, there is always a unique k -path between any two of its pixels. And for a closed irreducible k -connected curve, there are exactly two k -paths between any two pixels.

Between two points $p(i, j)$ and $p'(i', j')$, we define x -distance as $d_x(p, p') = |i - i'|$, y -distance as $d_y(p, p') = |j - j'|$, and the isothetic distance (Minkowski norm [7]) as $d_\infty(p, p') = \max\{d_x(p, p'), d_y(p, p')\}$. Based on this, the isothetic distance of a point $p(i, j)$ from a 2D curve C (which is an ellipse in our work) is defined as $d_\perp(p, C) = \min\{d_x(p, C), d_y(p, C)\}$, where,

$$d_x(p, C) = \begin{cases} d_x(p, q) & \text{if there exists a point } q(x, j) \text{ on } C \\ \infty & \text{otherwise,} \end{cases}$$

$$d_y(p, C) = \begin{cases} d_y(p, q) & \text{if there exists a point } q(i, y) \text{ on } C \\ \infty & \text{otherwise.} \end{cases}$$

While measuring $d_x(p, q)$ and $d_y(p, q)$, if there lies more than one point q on C , then we consider the nearest. Figure 1 shows the basic idea.

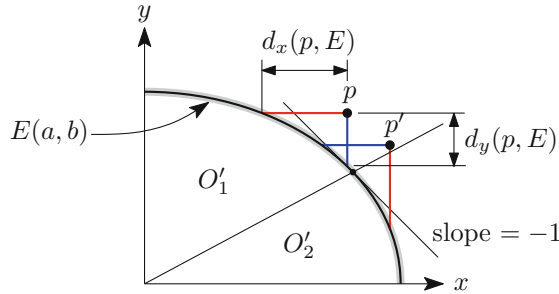


Fig. 1. Isothetic distance and its anomaly with pseudo-octants O'_1 and O'_2 . Both p and p' are in O'_1 , but their respective isothetic distances from $E(a, b)$ are $d_{\perp}(p, E) = d_y(p, E)$ and $d_{\perp}(p', E) = d_x(p', E)$, thus forming the anomaly. This anomaly gives rise to disconnectedness in the digital ellipse if p' is an integer point and $d_{\perp}(p', E) \leq \frac{1}{2}$.

We use $E(a, b)$ to denote a real ellipse, where a and b are the respective lengths of its semi-major and semi-minor axes. In our work, we consider a and b as positive integers. Further, an ellipse is considered in its *canonical form*, which means its center is $(0, 0)$ and its axes are parallel to the coordinate axes [13]. So, its equation is

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1 \tag{1}$$

and the corresponding digital ellipse is defined as follows.

Definition 1. A digital ellipse $E(a, b)$ is the 0-connected irreducible sequence of integer points obtained by discretization of the real ellipse $E(a, b)$ such that each point in $E(a, b)$ has an isothetic distance of at most $\frac{1}{2}$ from $E(a, b)$.

2.1 Elliptic Octants

An ellipse, whether the real or the digital, possesses four-way symmetry about its major and minor axes. As a result, we get four symmetric arcs of an ellipse, which lie in four different quadrants. For digitization purpose, the real ellipse is considered, and its arc in the 1st quadrant—and so for each other quadrant—is further divided into two arcs by a radial line that passes through the point on the ellipse at which its slope is -1 . The four quadrants are thus subdivided into eight *pseudo-octants*, in which the respective real arcs are digitized to generate

the full digital ellipse. This can be seen in the existing literature; see, for example, [4, 13], and the bibliographies therein.

The above idea works fine as far as a and b are comparable, but fails in case of a sufficiently high eccentricity—for example, when a is sufficiently large in value compared to b . A discussion on this can be seen in [13]. This happens due to the anomaly that for $a \gg b$, an integer point p may satisfy $d_y(p, E(a, b)) > d_x(p, E(a, b))$ although lying in the 1st pseudo-octant. An illustration is given in Fig. 1, and some examples of failure are shown in Fig. 2.

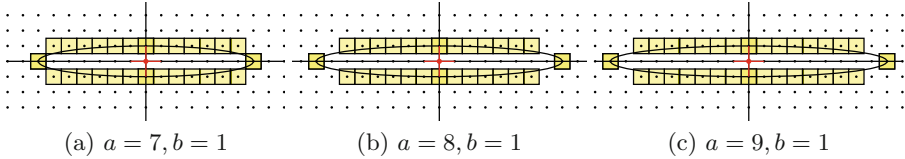


Fig. 2. Disconnectedness arising due to anomaly of pseudo-octants for $a \geq 8, b = 1$.

To circumvent the aforesaid problem, we introduce the concept of *elliptic octant* w.r.t. $E(a, b)$, which we henceforth refer simply as *e-octant* for brevity. We use \mathbb{O}_t to denote the t th ($1 \leq t \leq 8$) e-octant. We have the following proposition based on the notion of isothetic distance.

Proposition 1. *A point p lies in \mathbb{O}_t , where*

$$t(\bmod 4) \in \begin{cases} \{0, 1\} & \text{if and only if } d_y(p, E(a, b)) \leq d_x(p, E(a, b)) \\ \{2, 3\} & \text{if and only if } d_x(p, E(a, b)) \leq d_y(p, E(a, b)). \end{cases}$$

By Proposition 1, $\mathbb{O}_1, \mathbb{O}_4, \mathbb{O}_5$, and \mathbb{O}_8 become mutually symmetric with their isothetic distance as y -distance; and so also for $\mathbb{O}_2, \mathbb{O}_3, \mathbb{O}_6$, and \mathbb{O}_7 with their isothetic distance as x -distance. Owing to the above symmetry, it suffices to characterize the arc of a digital ellipse that lies in the first quadrant (i.e., \mathbb{O}_1 and \mathbb{O}_2). In \mathbb{O}_1 , the y -distance serves as the isothetic distance; whereas in \mathbb{O}_2 , it is the x -distance.

The above characterization by isothetic distance helps in generating a digital ellipse as a sequence of runs comprising its arcs in the eight e-octants. A *run* is defined as a maximum sequence of successive integer points with equal y - or equal x -coordinate. So, a digital ellipse in \mathbb{O}_1 consists of horizontal runs, and in \mathbb{O}_2 consists of vertical runs. This property is used during the construction of a digital ellipse. In \mathbb{O}_1 , we start from the point $(0, b)$ and go on generating the subsequent points along the horizontal runs, as long as the isothetic distance of an integer point from $E(a, b)$ is its y -distance. In \mathbb{O}_2 , we do the reverse—starting from the point $(a, 0)$, we generate the vertical runs, as long as the isothetic distance is x -distance. The other e-octants are easily constructed from the arcs obtained in \mathbb{O}_1 and \mathbb{O}_2 by using the 4-symmetry of $E(a, b)$.

3 Properties of Digital Ellipse

We denote by $E_t(a, b)$ the arc of the real ellipse $E(a, b)$ that lies in \mathbb{O}_t , and by $\mathbf{E}_t(a, b)$ that of the digital ellipse $\mathbf{E}(a, b)$ lying in \mathbb{O}_t . We have the following lemma on isothetic distance in an e-octant.

Lemma 1. *The isothetic distance of a point $p(i, j)$ from $E_t(a, b)$ is $d_{\perp}(p, E_t(a, b)) = d_y(p, E_t(a, b))$ if and only if $t \pmod 4 \in \{0, 1\}$, or equivalently, $|ab|j| - b^2\sqrt{a^2 - i^2}| \leq |ab|i| - a^2\sqrt{b^2 - j^2}|$.*

Proof. By definition of isothetic distance and by Eq. 1, $d_y(p, E_t(a, b)) = ||j| - (b/a)\sqrt{a^2 - i^2}|$ and $d_x(p, E_t(a, b)) = ||i| - (a/b)\sqrt{b^2 - j^2}|$. So, by Proposition 1, $d_y(p, E_t(a, b)) \leq d_x(p, E_t(a, b))$ if and only if

$$\begin{aligned} & t \pmod 4 \in \{0, 1\} \\ \iff & ||j| - (b/a)\sqrt{a^2 - i^2}| \leq ||i| - (a/b)\sqrt{b^2 - j^2}| \\ \iff & |ab|j| - b^2\sqrt{a^2 - i^2}| \leq |ab|i| - a^2\sqrt{b^2 - j^2}|. \end{aligned} \tag{2}$$

This completes the proof. □

We consider the 1st and the 2nd e-octants to obtain a characterization of digital ellipse. Owing to 4-symmetry, this will be applicable to other e-octants. Lemma 1 provides the way to decide whether a (real or integer) point p lies in \mathbb{O}_1 or in \mathbb{O}_2 . But it does not tell whether p belongs to $\mathbf{E}_1(a, b)$ or to $\mathbf{E}_2(a, b)$. The following theorem states the necessary and sufficient condition to decide whether an integer point belongs to $\mathbf{E}_1(a, b)$.

Theorem 1. *An integer point $p(i, j)$ with $j > 0$ belongs to $\mathbf{E}_1(a, b)$ if and only if*

$$4b^2a^2 - a^2(2j + 1)^2 \leq 4b^2i^2 \leq 4b^2a^2 - a^2(2j - 1)^2 - 1. \tag{3}$$

Proof. A point $p(i, j)$ belongs to $\mathbf{E}_1(a, b)$ if and only if $d_y(p, \mathbf{E}_1(a, b)) \leq \frac{1}{2}$ (by Proposition 1 and Definition 1), or equivalently, there exists a real point $q(i, j - \delta)$ on $E(a, b)$ such that $-\frac{1}{2} \leq \delta < \frac{1}{2}$. Note that $j > 0$ ensures that $q \in \mathbb{O}_1$. Further, to ensure that $\mathbf{E}_1(a, b)$ is irreducible (Definition 1), p is not considered to be in $\mathbf{E}_1(a, b)$ when $\delta = \frac{1}{2}$. Now, as mentioned in the proof of Lemma 1, $d_y(p, E_t(a, b)) = ||j| - (b/a)\sqrt{a^2 - i^2}|$, or, $d_y(p, \mathbf{E}_1(a, b)) = (j - (b/a)\sqrt{a^2 - i^2})$. So, the last condition is true if and only if

$$\begin{aligned} & -\frac{1}{2} \leq j - \frac{b}{a}\sqrt{a^2 - i^2} < \frac{1}{2} \\ \iff & j - \frac{1}{2} < \frac{b}{a}\sqrt{a^2 - i^2} \leq j + \frac{1}{2} \\ \iff & a(2j - 1) < 2b\sqrt{a^2 - i^2} \leq a(2j + 1) \\ \iff & -4b^2a^2 + a^2(2j - 1)^2 < -4b^2i^2 \leq -4b^2a^2 + a^2(2j + 1)^2 \\ \iff & 4b^2a^2 - a^2(2j + 1)^2 \leq 4b^2i^2 < 4b^2a^2 - a^2(2j - 1)^2, \end{aligned}$$

hence we get Eq. 3, as a, b, i, j are all integers. □

Theorem 1 provides all integer points of $E_1(a, b)$ with positive ordinate, i.e., $j > 0$ for a point $p(i, j) \in E_1(a, b)$. Now, Eq. 3 does not give a valid integer interval if $-a^2(2j + 1)^2 > -a^2(2j - 1)^2 - 1$, or, $-4a^2j > 4a^2j - 1$, or, $8a^2j < 1$, which implies $j = 0$, since a is a positive integer and j is a non-negative integer in \mathbb{O}_1 . Consequently, we have the following corollary.

Corollary 1. *An integer point $p(i, j)$ belongs to $E_1(a, b)$ but does not satisfy Eq. 3 only if $j = 0$.*

The equation for points constituting $E_2(a, b)$ is given in the following corollary, which is also obtained in a similar way.

Corollary 2. *An integer point (i, j) with $i > 0$ belongs to $E_2(a, b)$ if and only if*

$$4b^2a^2 - b^2(2i + 1)^2 \leq 4a^2j^2 \leq 4b^2a^2 - b^2(2i - 1)^2 - 1. \tag{4}$$

Moreover, if $p(i, j)$ belongs to $E_2(a, b)$ but does not satisfy Eq. 4, then $i = 0$.

Now, to efficiently compute the integer points comprising the digital ellipse in the first e-octant, we need the following theorem.

Theorem 2. *An integer point (i, j) with $j > 0$ belongs to $E_1(a, b)$ if and only if $4b^2i^2$ lies in the interval $I_k = [u_k, v_k := u_k + l_k - 1]$, where $j = b - k$, $k \geq 0$, and u_k and l_k are given as follows.*

$$\begin{aligned} u_k &= \begin{cases} -4a^2b - a^2 & \text{if } k = 0 \\ u_{k-1} + l_{k-1} & \text{otherwise} \end{cases} \\ l_k &= \begin{cases} 8a^2b & \text{if } k = 0 \\ l_{k-1} - 8a^2 & \text{otherwise} \end{cases} \end{aligned} \tag{5}$$

Proof. We get u_0 and l_0 corresponding to $k = 0$ by substituting $j = b$ in Eq. 3.

To get the recurrence of l_k for $k > 0$, observe that $l_k = 4b^2a^2 - a^2(2j - 1)^2 - 4b^2a^2 + a^2(2j + 1)^2 = 8a^2j = 8a^2(b - k)$, in accordance with Eq. 3. Hence, $l_k - l_{k-1} = 8a^2(b - k) - 8a^2(b - k + 1) = 8a^2$. To get the recurrence of u_k , we substitute $j = b - k$ in Eq. 3 to get $u_k = 4b^2a^2 - a^2(2(b - k) + 1)^2$, and substitute $j = b - k - 1$ to get $v_{k-1} = 4b^2a^2 - a^2(2(b - k - 1) - 1)^2 - 1 = u_k - 1$. Thus, $u_k - v_{k-1} = 1$, or, $u_k = v_{k-1} + 1 = u_{k-1} + l_{k-1}$. \square

Since I_0 contains only perfect squares of the form $4b^2i^2$, can we reset $u_0 = 0$. However, to make the recurrence of u_k work for $k \geq 1$, we adhere to a negative value of u_0 , and this does not affect the performance of the algorithms discussed in Sect. 4.

The recurrence relations for $E_2(a, b)$, obtained in a similar fashion, are put in the following corollary.

Corollary 3. *An integer point (i, j) with $i > 0$ belongs to $E_2(a, b)$ if and only if $4a^2j^2$ lies in the interval $I_k = [u_k, v_k := u_k + l_k - 1]$, where $i = a - k$, $k \geq 0$, and u_k and l_k are given as follows.*

$$\begin{aligned} u_k &= \begin{cases} -4b^2a - b^2 & \text{if } k = 0 \\ u_{k-1} + l_{k-1} & \text{otherwise} \end{cases} \\ l_k &= \begin{cases} 8b^2a & \text{if } k = 0 \\ l_{k-1} - 8b^2 & \text{otherwise} \end{cases} \end{aligned} \tag{6}$$

We conclude this section with an important observation. Notice in the proof of Theorem 2 that $u_k = v_{k-1} + 1$, which holds for both \mathbb{O}_1 and \mathbb{O}_2 . We put this in the following corollary.

Corollary 4. *For $k = 0, 1, 2, \dots$, the intervals I_k are disjoint and contiguous.*

4 Algorithms for Ellipse Construction

The basic idea of digital ellipse construction by the proposed technique is demonstrated in Fig. 3. Let $p(i, j)$ be a point of $E_1(a, b)$. Then, considering the clockwise traversal of $E_1(a, b)$, the next point would be either $(i + 1, j)$ or $(i + 1, j - 1)$, the former being when it is the same run and the latter being for a change of run. The recurrences in Theorem 2 are used for this, as demonstrated in Fig. 3 for $E_1(a, b)$ with $a = 6$ and $b = 4$. In the first e-octant, the first run starts at the point $(0, 4)$. We get $u_k = -612$ and $v_k = 539$ for the first run, and the points that satisfy the interval $[u_k, v_k]$ are $(0, 4)$, $(1, 4)$, and $(2, 4)$. After this, the value of j becomes 3 for the second run, and the values of u_k and v_k for $k = 1$ are computed recursively using Eq. 5. The process goes on till we are in \mathbb{O}_1 . For \mathbb{O}_2 , a similar process runs starting from $(6, 0)$, as shown in Fig. 3.

4.1 Algorithm DRAW-ELLIPSE-FLOAT

We have designed two algorithms for generation of a digital ellipse, which are based on the above-mentioned technique. We first discuss here Algorithm 1 that uses both

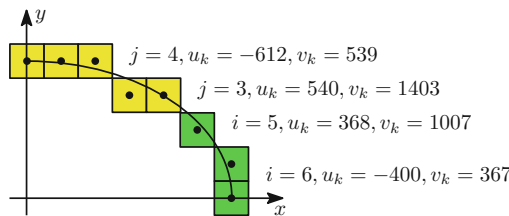


Fig. 3. An example showing generation of $E_1(6, 4)$ (yellow cells) and $E_2(6, 4)$ (green cells) by the proposed technique (Color figure online).

integer and floating-point operations, the latter type of operations being needed for determining the octant between \mathbb{O}_1 and \mathbb{O}_2 based on Lemma 1. Necessary variables are defined and initialized accordingly in Steps 1–2. The first **while** loop (Lines 4–5) and the second **while** loop (Lines 6–8) run in \mathbb{O}_1 and generate all integer points of $E_1(a, b)$. In each iteration of the first **while** loop, the procedure **drawRunFloat** generates a horizontal run (i.e., a maximum-length sequence of successive points with same y -coordinate) of $E_1(a, b)$. This procedure is again used later to generate the vertical runs of $E_2(a, b)$ in the **while** loop for \mathbb{O}_2 (Lines 11–12). The value of t is set to 1 and to 2 while invoking it in \mathbb{O}_1 (Line 5) and in \mathbb{O}_2 (Line 12), respectively. The ellipse parameters (a, b) and the point coordinates (i, j) are passed in appropriate order to generate the respective horizontal and vertical

Algorithm 1. DRAW-ELLIPSE-FLOAT

Input: int a, b
Output: Digital ellipse with semi-major axis a and semi-minor axis b

```

1 int  $i \leftarrow 0, j \leftarrow b, u \leftarrow -4a^2b - a^2, l \leftarrow 8a^2b, v \leftarrow 4a^2b - a^2 - 1, s \leftarrow 0$ 
2 float  $d_y \leftarrow 0, d_x \leftarrow 0$ 
3  $E \leftarrow \{\}$   $\triangleright$  output set of integer points
4 while  $(d_y \leq d_x) \wedge (u \leq v)$  do
5    $\lfloor$  drawRunFloat( $i, j, a, b, 1, s, d_y, d_x, u, l, v, E$ )  $\triangleright$  Lemma 1 & Theorem 1
6 while  $(u > v) \wedge (i < a)$  do
7   include4SymPoints( $i, j, E$ )  $\triangleright$  Corollary 1
8    $\lfloor$   $i \leftarrow i + 1, s \leftarrow 4b^2i^2$ 
9  $i \leftarrow a, j \leftarrow 0, u \leftarrow -4b^2a - b^2, l \leftarrow 8b^2a, v \leftarrow 4b^2a - b^2 - 1, s \leftarrow 0$ 
10  $d_y \leftarrow 0, d_x \leftarrow 0$ 
11 while  $(d_x \leq d_y) \wedge (u \leq v)$  do
12    $\lfloor$  drawRunFloat( $j, i, b, a, 2, s, d_y, d_x, u, l, v, E$ )  $\triangleright$  Lemma 1 & Corollary 2
13 while  $(u > v) \wedge (j < b)$  do
14   include4SymPoints( $i, j, E$ )  $\triangleright$  Corollary 2
15    $\lfloor$   $j \leftarrow j + 1, s \leftarrow 4a^2j^2$ 
16 return  $E$ 
```

Procedure drawRunFloat($i, j, a, b, t, s, d_y, d_x, u, l, v, E$)

```

1 repeat
2    $\lfloor$  if  $t = 1$  then include4SymPoints( $i, j, E$ )
3   else include4SymPoints( $j, i, E$ )  $\triangleright$  Proposition 1
4    $\lfloor$   $i \leftarrow i + 1, s \leftarrow 4b^2i^2$ 
5 until  $s > v$ ;
6  $j \leftarrow j - 1$ 
7 if  $t = 1$  then  $d_y \leftarrow |abj - b^2\sqrt{a^2 - i^2}|, d_x \leftarrow |abi - a^2\sqrt{b^2 - j^2}|$ 
8 else  $d_y \leftarrow |bai - a^2\sqrt{b^2 - j^2}|, d_x \leftarrow |baj - b^2\sqrt{a^2 - i^2}|$   $\triangleright$  Lemma 1
9  $u \leftarrow v + 1, l \leftarrow l - 8a^2, v \leftarrow u + l - 1$   $\triangleright$  Theorem 2, Corollary 3 & 4
```

Algorithm 2. DRAW-ELLIPSE-INT

Input: int a, b
Output: Digital ellipse with semi-major axis a and semi-minor axis b

```

1 int  $i \leftarrow 0, j \leftarrow b, u \leftarrow -4a^2b - a^2, l \leftarrow 8a^2b, v \leftarrow 4a^2b - a^2 - 1, s \leftarrow 0, i_1, j_1$ 
2  $E \leftarrow \{\}$ 
3 while  $(u \leq s) \wedge (s \leq v)$  do
4    $\lfloor$  drawRunInt( $i, j, a, b, 1, s, u, l, v, i_1, j_1, E$ )  $\triangleright$  Theorem 1
5    $i_1 \leftarrow i - 1, j_1 \leftarrow j + 1$ 
6 while  $(u > v) \wedge (i < a)$  do
7   include4SymPoints( $i, j, E$ )  $\triangleright$  Corollary 1
8    $\lfloor$   $i_1 \leftarrow i, j_1 \leftarrow j, i \leftarrow i + 1, s \leftarrow 4b^2i^2$ 
9    $i \leftarrow a, j \leftarrow 0, u \leftarrow -4b^2a - b^2, l \leftarrow 8b^2a, v \leftarrow 4b^2a - b^2 - 1, s \leftarrow 0$ 
10 while  $(u \leq s) \wedge (s \leq v) \wedge ((i \neq i_1) \vee (j \neq j_1))$  do
11    $\lfloor$  drawRunInt( $j, i, b, a, 2, s, u, l, v, i_1, j_1, E$ )  $\triangleright$  Corollary 2
12 while  $(u > v) \wedge (j < b)$  do
13   include4SymPoints( $i, j, E$ )  $\triangleright$  Corollary 2
14    $\lfloor$   $j \leftarrow j + 1, s \leftarrow 4a^2j^2$ 
15 return E
```

Procedure drawRunInt($i, j, a, b, t, s, u, l, v, i_1, j_1, E$)

```

1 repeat
2   if  $t = 1$  then include4SymPoints( $i, j, E$ )
3   else if  $(i = i_1) \wedge (j = j_1)$  then return
4   else include4SymPoints( $j, i, E$ )  $\triangleright$  Proposition 1
5    $i \leftarrow i + 1, s \leftarrow 4b^2i^2$ 
6 until  $s > v$ ;
7  $j \leftarrow j - 1$ 
8  $u \leftarrow v + 1, l \leftarrow l - 8a^2, v \leftarrow u + l - 1$   $\triangleright$  Theorem 2, Corollary 3 & 4
```

runs for $E_1(a, b)$ and $E_2(a, b)$. For \mathbb{O}_2 , the parameters are properly initialized in Lines 9–10.

Inside the procedure **drawRunFloat**, there is a **repeat-until** loop (Lines 1–5) that computes the points of the k (≥ 0)th horizontal (if $t = 1$) or vertical (if $t = 2$) run using the integer interval $[u, v]$. For every s generated in succession and lying in $[u, v]$, the procedure **include4SymPoints** includes the corresponding point $p(i, j)$ and its three symmetric points $\{(i, -j), (-i, -j), (-i, j)\}$ in E . The parameters (i, j) are passed in **include4SymPoints** depending on the value of t . After generation of all the points of the k th run, all the related parameters are updated in Lines 6–8 of **drawRunFloat** to generate the points of the next run.

As discussed in Sect. 3 (Corollary 1), integer points of the form $(i, j = 0)$ and belonging to $E_1(a, b)$ cannot be tracked by the integer intervals used in the first **while** loop (Lines 4–5). This happens particularly when a is significantly large compared to b . In order to track such points, the second **while** loop (Lines 6–8)

is used with necessary conditional checks. A similar **while** loop (Lines 13–15) is also used in \mathbb{O}_2 to ensure the completeness of $E_2(a, b)$.

4.2 Algorithm DRAW-ELLIPSE-INT

Algorithm 1 is not free from floating-point operations and hence susceptible to computational pitfalls. So, as an improvement, we design Algorithm 2 where we use only integer operations. This algorithm, contrary to the previous, does not resort to computation of isothetic distance. Instead, it uses two extra integer variables, i_1 and j_1 , meant to store the coordinates of the last point in $E_1(a, b)$.

It is worth mentioning here now a few details concerning x - and y -distances of a (real or integer) point $p(i, j)$ from $E(a, b)$. Let, w.l.o.g., p lie in \mathbb{O}_1 . Then, by Proposition 1, $d_y(p, E_1(a, b)) \leq d_x(p, E_1(a, b))$. Now, $p \in E_1(a, b)$ if and only if $d_y(p, E_1(a, b)) \leq \frac{1}{2}$, which, however, does not give any idea about the value of $d_x(p, E_1(a, b))$. Interestingly, it may also happen that $d_x(p, E_1(a, b)) \leq \frac{1}{2}$ for the same integer point p . In such situation, the point p would be included only in $E_1(a, b)$ by Algorithm 1. However, if neither of i and j is zero, then p would satisfy the integer intervals corresponding to both $E_1(a, b)$ and $E_2(a, b)$ (Eqs. 3 and 4).

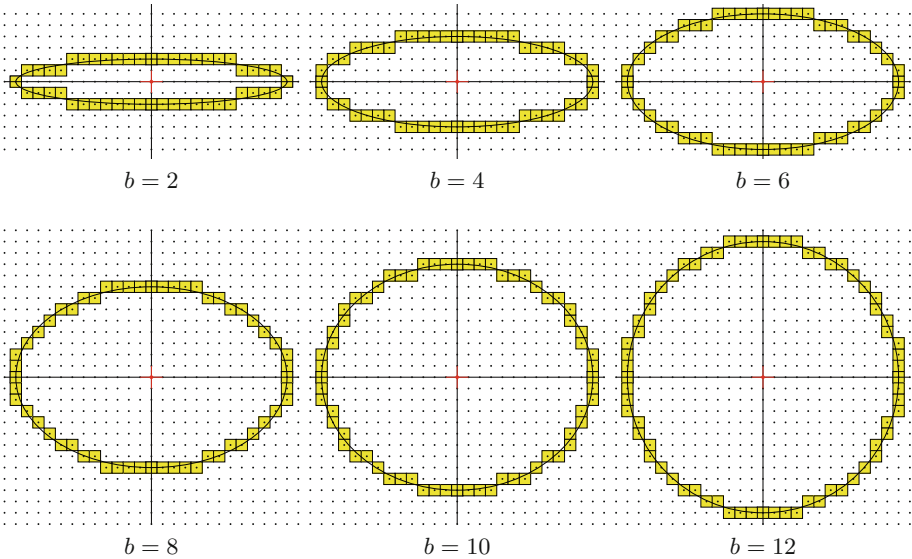


Fig. 4. Digital ellipses with $a = 12$ and increasing values of b .

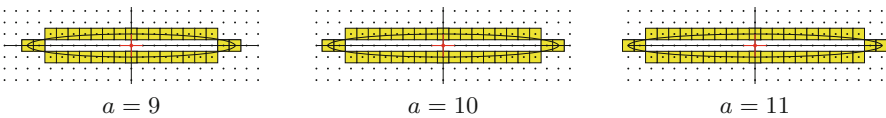


Fig. 5. Digital ellipses with $b = 1$ and increasing values of a .

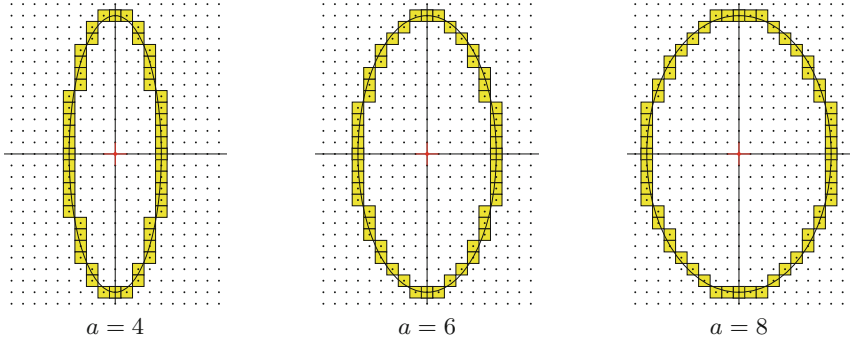


Fig. 6. Digital ellipses with $b = 12$ and increasing values of a .

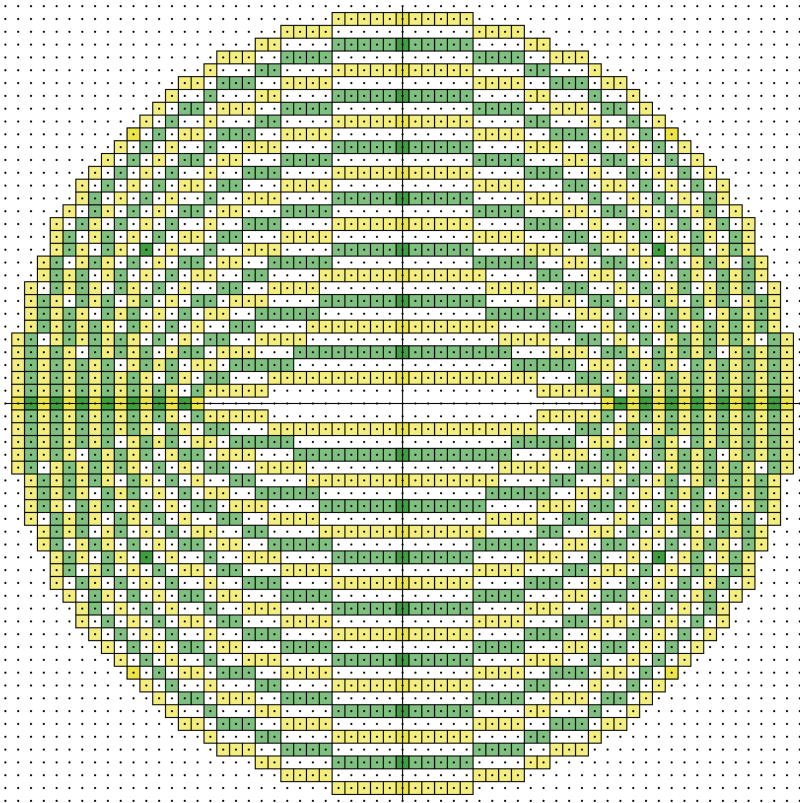


Fig. 7. A set of digital ellipses: $\{E(a, b) : (a, b) = (16, 2), (17, 4), (18, 6), \dots, (30, 30)\}$.

As a result, this point p would be generated twice—first in \mathbb{O}_1 and then in \mathbb{O}_2 . In fact, there may occur many such points in succession—around the junction of \mathbb{O}_1 and \mathbb{O}_2 —which would result to overlap between the two digital arcs $E_1(a, b)$

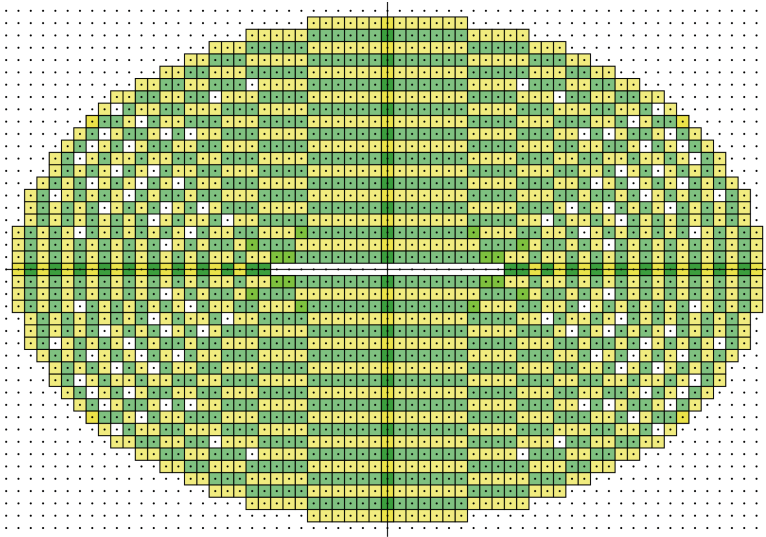


Fig. 8. Another set: $\{E(a, b) : (a, b) = (11, 1), (12, 2), (13, 3), \dots, (30, 20)\}$.

and $E_2(a, b)$. This overlap is prevented by using i_1 and j_1 in Algorithm 2, as mentioned above. The necessary conditional check is put in Line 10. Updating the values of i_1 and j_1 are done appropriately when the algorithm executes for \mathbb{O}_1 . Notice that, except for the above-mentioned points, the **while** loops of Algorithm 2 work in the same way as those of Algorithm 1.

5 Concluding Notes

Algorithms 1 and 2 both run with an optimal time complexity, which is linear in the number of integer points comprising a digital ellipse, i.e., $\Theta(a + b)$. This owes to the fact that the number of every type of operations (comparison, addition, multiplication, etc.) required to report each integer point of a digital ellipse is upper-bounded by a small constant. To derive tight asymptotic bounds on the number of operations in either of the two algorithms, a thorough analysis has to be done. Nature of distribution of runs with change in major- and minor-axis lengths can also be studied as a prospective future work.

We present here some instances of digital ellipse produced by the proposed technique. These are identical irrespective of the algorithm used.

Figure 4 displays some digital ellipses with a fixed semi-major axis ($a = 12$) and increasing values of b . The corresponding real ellipses are also shown; it helps us to understand the pattern of points forming their digitization. Figures 5 and 6 display two different sets of results with a fixed semi-minor axis—one for small $b(= 1)$ and another with usual $b(= 12)$ —and with increasing values of a . All these results show the 4-symmetry of digital ellipse along with the properties of their composition as 0-connected irreducible sequences.

Figure 7 contains a set of 15 digital ellipses whose semi-major axis increases from 16 to 30 units at unit step, and semi-minor axis from 2 to 30 at double step, finally reaching a digital circle of radius 30. Notice for this set that all the fifteen digital ellipses are disjoint, as there is no common pixel between any two. Some ‘gaps’ are left between two consecutive digital ellipses, which indicates an interesting similitude with concentric integer circles, since the latter class also exhibits the aforesaid disjointness property, as recently shown in [1]. Characterization of such an ellipse class would be an interesting research issue in the context of covering the integer plane by coaxial ellipses.

In Fig. 8, we have shown another set containing 20 digital ellipses, whose semi-major axis increases from 11 to 30, and semi-minor axis from 1 to 20—both at unit steps. The gaps formed here are less in number, since there is overlap between two consecutive ellipses. This set, together with the previous set, indicates that overlapping ellipses can reduce the gaps, although not completely. What follows as a natural question is to find a minimal or a minimum set of coaxial digital ellipses with major- and minor-axis lengths specified by two rational numbers, so as to cover the interior of an ellipse of given integer specification. Characterization of this set and designing efficient algorithms for digital ellipse construction with non-integer (rational, for definiteness) specification are two important follow-up research problems in the context of our work.

References

1. Bera, S., Bhowmick, P., Stelldinger, P., Bhattacharya, B.: On covering a digital disc with concentric circles in \mathbb{Z}^2 . *Theoret. Comput. Sci.* **506**, 1–16 (2013)
2. Bresenham, J.E.: Algorithm for computer control of a digital plotter. *IBM Syst. J.* **4**(1), 25–30 (1965)
3. Fellner, D.W., Helmbert, C.: Robust rendering of general ellipses and elliptical arcs. *ACM Trans. Graph.* **12**(3), 251–276 (1993)
4. Foley, J.D., Dam, A.V., Feiner, S.K., Hughes, J.F.: *Computer Graphics: Principles and Practice*. Addison-Wesley, Reading (1993)
5. Haiwen, F., Lianqiang, N.: A hybrid generating algorithm for fast ellipses drawing. In: *International Conference on Computer Science and Information Processing (CSIP)*, pp. 1022–1025 (2012)
6. Kappel, M.: An ellipse-drawing algorithm for raster displays. In: Earnshaw, R.A. (ed.) *Fundamental Algorithms for Computer Graphics*. NATO ASI Series, vol. 17, pp. 257–280. Springer, Heidelberg (1991)
7. Klette, R., Rosenfeld, A.: *Digital Geometry: Geometric Methods for Digital Picture Analysis*. Morgan Kaufmann, San Francisco (2004)
8. McIlroy, M.D.: Getting raster ellipses right. *ACM Trans. Graph.* **11**(3), 259–275 (1992)
9. Pitteway, M.L.V.: Algorithm for drawing ellipses or hyperbolae with a digital plotter. *Comput. J.* **10**, 282–289 (1967)
10. Van Aken, J., Novak, M.: Curve-drawing algorithms for raster displays. *ACM Trans. Graph.* **4**(2), 147–169 (1985)
11. Van Aken, J.: An efficient ellipse-drawing algorithm. *IEEE Comput. Graphics Appl.* **4**(9), 24–35 (1984)

12. Wu, X., Rokne, J.: Double-step generation of ellipses. *IEEE Comput. Graphics Appl.* **9**(3), 56–69 (1989)
13. Yao, C., Rokne, J.G.: Run-length slice algorithms for the scan-conversion of ellipses. *Comput. Graph.* **22**(4), 463–477 (1998)