A Linear-time Algorithm for the Generation of Random Digital Curves

Partha Bhowmick
Indian Institute of Technology, Kharagpur
India
bhowmick.gmail.com

Oishee Pal
Ushacomm India Pvt Ltd, Kolkata
India
Oishee.Pal@ushacomm.com

Reinhard Klette
The University of Auckland
New Zealand
r.klette@auckland.ac.nz

Abstract

We propose an algorithm to generate random digital curves of finite length, generating points of a digital path ρ on the fly. Path ρ never intersects or touches itself, and hence becomes simple and irreducible. This is ensured by detecting every possible trap formed by the previously generated part of ρ, which, if entered into, cannot be exited without touching or intersecting ρ. The algorithm is completely free of any backtracking and its time complexity is linear in the length of ρ. Implemented and tested exhaustively, it shows that it produces results as specified by the user.

1. Introduction

The generation of random digital curves of finite length is, for example, required when testing geometric algorithms in image analysis. Assume that algorithm A takes a digital curve as its input; see, for example, [2, 3, 9, 13]. Ideally, one would like to test the behavior and performance of A on various real-world digital curves being of practical relevance and having also *ground truth* (e.g, length or curvature of the curve) available for the studied curves. However, it is often very difficult to obtain a sufficiently large number of practically relevant input data sets. An other option is to run A on a reasonably large number of random curves (e.g., using high-resolution curves for generating ground truth, and lower resolution curves for the actual input), which necessitates a proven algorithm that can generate a sufficiently large set of random curves of finite length in the digital plane in feasible time.

Work on generating random curves can be traced back to studies on Brownian motion in $\mathbb{R}^2$, see [5, 6], or on random walks, see [8, 10, 11]. The generation of closed random polylines also received considerable attention in recent times, see [1, 7, 12, 14]. However, there is no published work so far on generating random digital curves. Polygon-generation algorithms work with input vertices of the polyline which are generated randomly but *a priori*. On the contrary, our algorithm generates new points *on the fly* (also called *online* in [9]) while creating a digital curve ρ, starting from a random point (or vertex) p_1, choosing the next point p_2 randomly, connecting points p_1 and p_2, choosing the next point p_3 randomly, connecting it with p_2, and so forth, eventually returning to the start point p_1.

The difficulty of the problem lies in generating such a digital curve ρ such that ρ *is one pixel wide everywhere and never intersects or touches itself*, hence it becomes irreducible and simple. This calls for detecting every possible "narrow-mouthed" *trap* formed by the previously generated part of ρ, which, if entered into, cannot be exited without touching or intersecting ρ. A trap may be multiply nested and needs to be detected 'fast' (i.e., without any backtracking through the mouth of the trap) in order to ensure linear computation time. Furthermore, it must be ensured that the (4- or 8-connected) path ρ, traced by the algorithm, finally reaches the start point again. The proof of correctness of our algorithm follows from the principle of mathematical induction, and its time complexity is linear in the length of ρ. Instances of a few variations of random digital curves produced by the algorithm are shown in Fig. 1.

Preliminaries. We use both the grid point and the cell model of digital geometry in the plane [9]. The *canvas* is a set $\mathbb{G}_n$ of grid cells, forming an $n \times n$ square in the plane. Vertices and centers of cells in $\mathbb{G}_n$ are assumed to be grid points in $\mathbb{Z}^2$, also simply called *points* for brevity in this paper. Let $n = 2^k$.

The canvas may be partitioned into a set $\mathbb{G}_m$ of $m \times m$ square cells, $m \leq n$ (see Fig. 2, top), each of size $s \times s$, where $s = 2^t$, with $t \geq 2$. The cell belonging to the ith row and the jth column of $\mathbb{G}_m$ is denoted by $c(i,j)$; it is a *border cell* if and only if $\{i,j\} \cap \{1,m\} \neq \emptyset$, and a *corner cell* if and only if $(i,j) \in \{1,m\} \times \{1,m\}$.

In the digital plane, 1- or edge-adjacency [0- or vertex-adjacency] of cells is equivalent to 4-adjacency [8-adjacency] of centers of cells. For a cell c, adjacency sets are denoted by $A_\alpha(c)$ and neighborhoods by $N_\alpha(c) = A_\alpha(c) \cup \{c\}$, for $\alpha \in \{0,1\}$. For a center point p, the sets are $A_\alpha(p)$ and $N_\alpha(p) = A_\alpha(p) \cup \{p\}$, for $\alpha \in \{4,8\}$.

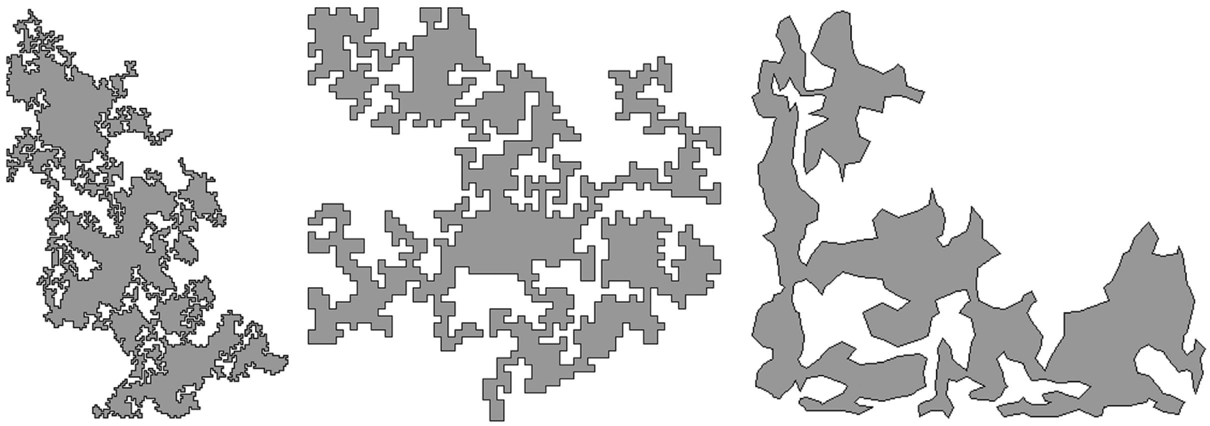


Figure 1. Instances of irreducible, 4- or 8-connected random simple curves generated by the proposed algorithm on a canvas with $n = 400$. Interiors are shown in gray for better visibility of the curves.

A cell $c(k,l)$, that is 0-adjacent to $c(i,j)$, is denoted by $c^{(t)}$ if $t = \tan^{-1}((k-i)/(l-j))/(\pi/4)$, considering that $\tan^{-1} x \in [0, 2\pi)$. We have that $t \in \{0,1,2,\ldots,7\}$ in case of 0-adjacency, and $t \in \{0,2,4,6\}$ in case of 1-adjacency (see Fig. 2, lower left).

In this paper, a *curve* ρ is a simple, irreducible, α-connected digital path $\langle p_1, p_2, \ldots, p_n \rangle$ of points in $\mathbb{Z}^2$ such that each of p_1 and p_2 has at least one (at most two), and each other point exactly two α-neighbors in ρ. If both p_1 and p_n have one α-neighbor in ρ only, then ρ is a *simple arc* with *end points* p_1 and p_n. If each point in ρ has exactly two α-neighbors in ρ, then ρ is a *simple curve*.

A *hole* in an α-connected set $S \subset \mathbb{Z}^2$ is a finite $\bar{\alpha}$-component of $\mathbb{Z}^2 \smallsetminus S$. If $\alpha = 8$ then $\bar{\alpha} = 4$, and vice versa. A simple arc defines no hole, whereas a simple curve always defines exactly one hole. A simple curve divides $\mathbb{Z}^2 \smallsetminus \rho$ into two regions, namely the *interior* (the hole) and the *exterior* (also known as the 'background').

We generate a random simple arc or curve ρ such that points $p_i \in \rho$ are in the canvas $\mathbb{G}_n$ only (see Fig. 1).

2. Proposed Algorithm

A cell c is said to be a *occupied* if and only if the generated part of curve ρ already passes through c; otherwise it is *free*. We use the following parameters for a cell c:

The *blocking factor* $\beta(c)$ is a 5-bit number given by the combinatorial arrangement of the occupied and the free cells in $N_1(c)$. The most significant bit of $\beta(c)$ corresponds to c itself, and the other four bits correspond to the four cells lying right, top, left, and below of c in that order. If a cell in $N_1(c)$ is occupied then the corresponding bit of $\beta(c)$ equals 1, otherwise 0. Thus, $\beta(c) = 0$ implies that ρ is not (yet) passing through any cell in $N_1(c)$. If $0 < \beta(c) < 16$ then c is free but one or more cells in $A_1(c)$ are occupied. If $\beta(c) \geq 16$ then c is occupied.

The *directional label* $\delta(c)$ is used if $0 < \beta(c) < 16$ which takes its value then from $\{\texttt{L}, \texttt{R}, \texttt{B}\}$, with the interpretation: $\texttt{L}$ = left, $\texttt{R}$ = right, $\texttt{B}$ = both left and right, depending on the position of c relative to the direction of traversal of ρ in the cell(s) of $A_0(c)$. We use $\texttt{X}$ for the initialized value. While the construction of ρ is in progress, blocking factors and directional labels have interim values, which are updated and become final values when ρ is finished.

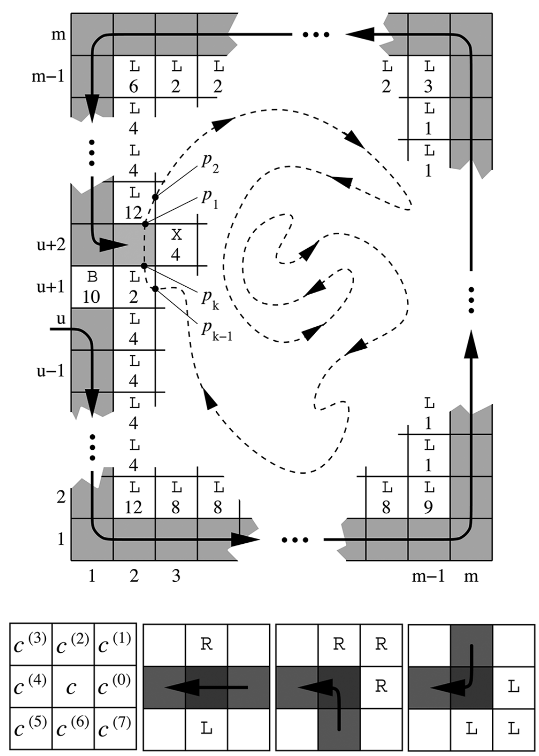


Figure 2. *Top:* The canvas and its initialization. Occupied cells are shown in gray. Cells not shown have $\beta = 0$ and $\delta = \texttt{X}$. The initialized part of the curve (passing through the border cells) is shown as a solid line, and the random part is shown as dotted. *Bottom, left to right:* Cells adjacent to a cell c and three types of turns, with only one (out of four) combinatorial cases shown (dark gray: current cell c_i, light gray: previous and next cells).

2.1. Initialization of the Canvas

The initialization of the canvas is illustrated in Fig. 2. A cell $c(u,1)$ is randomly chosen from $\{c(i,1) : 2 < i < m-2\}$. Curve ρ is assumed to enter $c(u,1)$ from its left edge, and then progresses (in straight or right-angle moves) through the border cells, finally reaching cell $c(u+2,1)$. From $c(u+2,1)$, ρ is now entering $c(u+2,2)$. In other words: by this initialization, $c(u+1,1)$ is free and has B as δ-value, whereas all other border cells are occupied. The free cells, adjacent to the border cells, have L as δ-value, as shown in Fig. 2, top. While generating the random curve, if some cell c is visited which is adjacent to some border cell, then the corresponding parameters of c are updated accordingly. These parameters help advancing the curve in a random and yet 'safe' direction.

Clearly, that *virtual* part of ρ lying in the border cells [except $c(u+1,1)$] of $\mathbb{G}_m$ is not random, and hence not considered as being a part of the random curve. The random curve starts and ends at the cell $c_1 := c(u+2,2)$.

In general, the *start cell* c_1 has three cells in $A_1(c_1)$ which are free. We enter randomly one of those via either the right, top, or bottom edge of c_1. On the selected edge, a random start point p_1 is selected (i.e., p_1 is not a grid point in general). After initialization of the canvas, all cells except those adjacent to already occupied cells have $\beta = 0$, as shown in Fig. 2, top. Blocking factors and directional labels of those cells are later updated while generating the random curve.

2.2. Updating the Cell Parameters

The *current cell*, which ρ has currently entered, is denoted by c_i ($i > 1$), unless mentioned otherwise. The cell c_i corresponds to the ith iteration of our algorithm. Parameters β and δ are updated in (appropriate cells of) $A_0(c_i)$, as shown in Fig. 2, bottom row. Each current cell c_i has a *previous cell*, c_{i-1}, from where ρ has entered c_i, and a *next cell*, c_{i+1}, where ρ will enter next. We do not label those cells of $A_0(c_i)$ that are common with $A_1(c_{i-1})$ or $A_1(c_{i+1})$, because cells in $A_0(c_i) \cap A_1(c_{i-1})$ have been already labeled when c_{i-1} was the current cell in the previous (i.e., $(i-1)$th) iteration, and those in $A_0(c_i) \cap A_1(c_{i+1})$ will be labeled when c_{i+1} is the new current cell in the next (i.e., $(i+1)$th) iteration.

Thus, only the cells belonging to the region $\widetilde{N}(c_i) := A_0(c_i) \smallsetminus (A_1(c_{i-1}) \cup A_1(c_{i+1}))$ are labeled in the ith iteration, as illustrated in Fig. 2. In the $(i+1)$th iteration, there are at most three possibilities of choosing the next cell, since there can be at most three free cells in $A_1(c_i)$, as given by $\beta(c_i)$. In each case, a cell of $A_0(c_i)$, which is not labeled in the ith iteration, is either labeled or chosen as the next cell (i.e., gets occupied) in the $(i+1)$th iteration. As evident from Fig. 2, bottom row, at least two and at most three cells

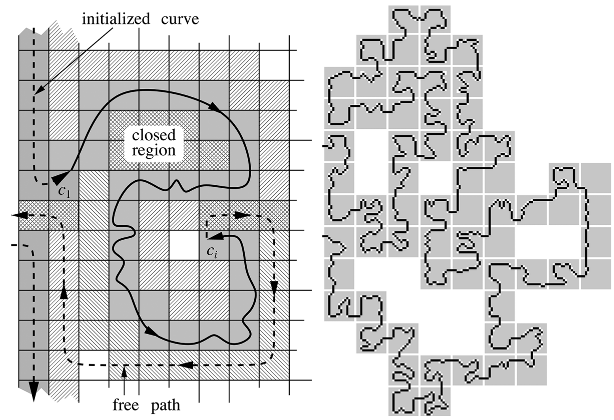


Figure 3. Open and blocked regions and holes formed during construction of ρ. *Left:* There exists at least one free path $\rho(c_i, c_1)$ from the current cell c_i to c_1, where ρ could also enter, e.g., the white cell when leaving c_i. On the contrary, there exists no free path from any cell of the blocked region to c_1. *Right:* Part of a random curve with four holes. Cells in holes are free but do not offer any path to start cell c_1 (not shown here).

will be labeled in each iteration. For the current cell c_i, the entry point of ρ can lie on one of its four edges, and the exit point on one of the remaining three edges. As a result, there are three *types of turns* at c_i, each having four sub-types, considering the previous cell c_{i-1} and the next cell c_{i+1}, which are dealt with as follows:

NO TURN: $\widetilde{N}(c_i)$ consists of two cells; the one lying left to the direction of the traversal of ρ is labeled by L, and the other, lying right to the direction of traversal, by R.

LEFT TURN: All three cells in $\widetilde{N}(c_i)$ get label R.

RIGHT TURN: All three cells in $\widetilde{N}(c_i)$ get label L.

The most significant bit of $\beta(c_i)$ is updated from 0 to 1 because cell c_i is now occupied. Blocking factors of the free cells [as given by $\beta(c_i)$] in $A_1(c_i)$ are updated accordingly. Cells in $\widetilde{N}(c_i)$ which are free ($\beta < 16$) are labeled with proper directional labels.

2.3. Choosing the Next Cell

From the current cell c_i, the next cell c_{i+1} is (randomly) chosen in such a way that there exists at least one *free path* from c_{i+1} to c_1. (A free path from a cell c_i to a cell c_{i+k}, $k > 1$, is given by a sequence of cells, $\rho(c_i, c_{i+k}) := \langle c_i, c_{i+1}, \ldots, c_{i+k} \rangle$, such that each cell in $\langle c_{i+1}, \ldots, c_{i+k-1} \rangle$ is free and distinct, and every two consecutive cells in $\rho(c_i, c_{i+k})$ are 1-adjacent.) A *safe edge* of c_i is a possible exit edge; the algorithm selects randomly one of the safe edges for exit.

For the current cell c_i we have the *free region* R_i of all free cells c of the canvas $\mathbb{G}_m$ such that there exists still at least one free path from c to the free border cell (i.e., to $c_1^{(5)}$). Similarly, a *blocked region* H is a maximal (connected) region of free cells such that there does not exist any free path from any cell of H to $c_1^{(5)}$. A cell in H is said to be *blocked*, and edges of a blocked cell are also *blocked*.

Theorem 1 *There exists a free path from the current cell c_i to the start cell c_1 if and only if $A_1(c_i) \cap R_i \neq \emptyset$.*

Proof: If $A_1(c_i) \cap R_i \neq \emptyset$, then there exists a free cell $c_i^{(t)} \in A_1(c_i)$ lying in R_i. Thus, by definition of a free region, there exists a free path from $c_i^{(t)}$ to $c_1^{(5)} \in A_0(c_1) \smallsetminus A_1(c_1)$. Since $c_1^{(6)}$ is the only free cell (of the canvas, after initialization) in $A_1(c_1^{(5)})$, a free path from $c_i^{(t)}$ to $c_1^{(5)}$ always contains $c_1^{(6)} \in A_1(c_1)$. Thus, there exists a free path from c_i to c_1. – Conversely, the existence of a free path from c_i to c_1 implies that at least one cell of $A_1(c_i)$ is in R_i, thus $A_1(c_i) \cap R_i \neq \emptyset$. □

Using Theorem 1 we can decide whether an edge e of the current cell c_i is safe or not, since e is also incident with another cell $c_i^{(t)}$ that lies in $A_1(c_i)$. Since c_i is occupied, it does not lie in a blocked region, wherefore $c_i^{(t)}$ lies in a blocked region if and only if the edge e belongs to (the border of) a blocked region. If e is selected as the exit edge of c_i, then $c_i^{(t)}$ becomes the next cell from where a free path to c_1 is not possible:

Corollary 1 *The edge between c_i and $c_i^{(t)}$ is safe if and only if $c_i^{(t)}$ belongs to R_i.*

2.3.1 Determining Safe Edges

To ensure that ρ is simple and irreducible, it is allowed to enter and exit a cell at most once. Hence, an exit edge of the current cell c_i cannot be an entry edge of the next cell if the latter is already occupied. This is determined using the blocking factor $\beta(c_i)$. Furthermore, a blocked edge cannot be an exit edge (Sec. 2.3). The crux of the problem is, therefore, to decide whether or not an edge of c_i is a blocked edge. Since a hole is surrounded by occupied cells, and the next cell is never an occupied cell, ρ enters a hole H if and only if c_i (which had been free until ρ entered c_i) gives rise to such a hole H. Thus, each event of forming a hole is detected based on (changes in the components of the cells in) $A_0(c_i)$. The advantage of detecting such a *hole event* is that, once ρ enters the next cell c_{i+1} from c_i by selecting a safe edge, it can never enter the hole H formed by c_i, since H gets surrounded by occupied cells after it is formed.

Thus, it has to be ensured that the exit edge of c_i is a safe edge, which is determined using the "interim label" of a cell.[1] However, only the interim label of a cell is not sufficient to decide about a hole event. Further characterizations of cells in the local neighborhood of c_i are required to distinguish whether there is a hole event or an event of an *ensuing hole*, as explained next. We use the notation $\delta(c, i)$ to indicate the label of cell c when the current cell is c_i.

[1] As mentioned in Sec. 2, the directional label $\delta(c)$ provides only an interim value, since $\delta(c)$ may be updated when ρ visits some other cell(s) in $A_0(c)$ later on.

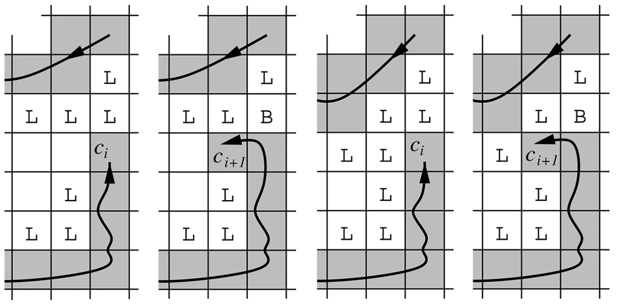
Definition 1 *$E_i \subset R_i$ is an ensuing hole corresponding to c_i if and only if*
(e1) *there exists $c \in \widetilde{N}(c_i)$ such that $\delta(c, i) = \mathtt{B}$,*
(e2) *for each $c' \in E_i$, we have that $\delta(c', i) \in \{\mathtt{L}, \mathtt{R}, \mathtt{X}\}$,*
(e3) *there exists a free path $\rho(c_{i+1}, c_1^{(5)})$, and for any such path, c is on $\rho(c_{i+1}, c_1^{(5)})$.*

The formation of an ensuing hole is illustrated in Fig. 4. It should be mentioned here that the initialization of the canvas (Fig. 2) is done in a way that gives rise to an ensuing hole, E_1, corresponding to c_1 as the current cell. The ensuing hole E_1 comprises of all the cells inside the canvas except the border cells and c_1, the reason being as follows: The cell $c_1^{(5)} := c(1, u+1)$ had its label L with $c(1, u), c(1, u-1), \ldots, c(1, u+2)$ as the current cells, which changed to B when the current cell becomes c_1. Thus, the condition (e1) of Definition 1 is true. It is easy to see that conditions (e2) and (e3) are also true. Such an ensuing hole E_1 is formed during the initialization to ensure that ρ finally returns to $c_1^{(5)}$ in order to become closed. In short, the algorithm finally converts the cell of each ensuing hole into a occupied cell or a hole-cell by the time ρ is closed; and more importantly, each cell having the interim label B is finally occupied. The following theorem captures this concept.

Theorem 2 *When the construction of ρ is over, each cell c can have either $\beta(c) \geq 16$ (c is occupied) or $\delta(c) \in \{\mathtt{L}, \mathtt{R}, \mathtt{X}\}$; that is, B can be the interim label but cannot be the final label of any free cell.*

Proof: Assume that cell c had the interim label $\delta(c) = \mathtt{R}$ when the current cell was c_{i-k}, and let it be updated to $\delta(c) = \mathtt{B}$ when the current cell becomes c_i. Thus, both c_i and c_{i-k} belong to $A_0(c)$. With c_i as the current cell, the part of ρ from c_{i-k} to c_i, therefore, corresponds to a sequence of occupied cells $\langle c_{i-k}, \ldots, c_i \rangle$ with two options:
Case 1: The (only possible) next cell is c with label B.
Case 2: The next cell can be c or some other cell in $A_1(c_i)$ having a label different from B.

If $\langle c_{i-k}, \ldots, c_i, c \rangle$ does not enclose any free cell, then the only possible next cell is c (Case 1), and c becomes occupied. If $\langle c_{i-k}, \ldots, c_i, c \rangle$ encloses a region R consisting of one or more free cells, then, in Case 1, there does not exist any free path from a cell in R to the cell $c_1^{(5)}$ once c becomes the current cell; thus, R becomes a hole. In Case 2, if the next cell is different from c, then ρ will visit c at a later point of time after visiting (and blocking) some cells of R, which implies that R would be dissociated from the free region in future. In Case 2, therefore, either all free cells of R finally get occupied or some get occupied and the rest gives rise to one or more blocked regions. Hence, the region R in Case 2 is an ensuing hole. From c_i on, ρ visits cells of the ensuing hole, and later on ρ must come out of the ensuing



(a) (b) (c) (d)

Figure 4. Distinguishing the formation of a hole (a,b) from an ensuing hole (b,c). *Ensuing hole:* (a) Before formation, all the concerned cells have label L. (b) After formation, label of $b_i := c_i^{(2)}$ gets modified to B, and a free path exists from each free cell in $E_i \cap A_1(c_{i+1}) := \{c_{i+1}^{(2)}, c_{i+1}^{(4)}, c_{i+1}^{(6)}\}$ to b_i. *Hole:* (c) Before formation, cells have label L. (d) After formation, label of $b_i := c_i^{(2)}$ becomes B, and a free path to b_i is not possible from $c_{i+1}^{(4)}$ and $c_{i+1}^{(6)}$, as $c_{i+1}^{(3)}$ is occupied.

hole, failing which ρ cannot reach $c_1^{(5)}$ in order to become a simple curve. The only cell through which ρ comes out of the ensuing hole is c (Definition 1), whereby c becomes finally occupied. Thus, if a cell c has the interim label B, then it would finally have $\beta(c) \geq 16$; otherwise, its final label is $\delta(c) \in \{\mathtt{L}, \mathtt{R}, \mathtt{X}\}$. □

From the proof of Theorem 2, it is clear that a change of label L or R (of a free cell) in $\widetilde{N}(c_i)$ indicates either a hole event or an ensuing hole event, as stated in the following corollary.

Corollary 2 *Either a hole or an ensuing hole is created if and only if at least one free cell in $\widetilde{N}(c_i)$ gets the label* B *as c_i becomes the current cell.*

It is, therefore, necessary to distinguish a hole event from an ensuing hole event if there occurs a label B in $\widetilde{N}(c_i)$ corresponding to the current cell c_i. The following theorem explicates the necessary and sufficient conditions to decide whether such a label B corresponds to an ensuing hole event. If not, then the only other possible event associated with the label B is a hole event, based on which the safe edges can be determined accordingly.

Theorem 3 *The current cell c_i gives rise to an ensuing hole E_i if and only if there exists a free cell $b_i \in \widetilde{N}(c_i)$ such that*
(E1) $\delta(b_i, i) = \mathtt{B}$;
(E2) *there exists $\rho(a_i, b_i) \subseteq A_0(c_{i+1})$ for each $a_i \in E_i \cap A_1(c_{i+1})$.*

Proof: Condition E1 follows from Corollary 2. W.l.o.g., let c_{i+1} lie left of c_i, and let the the cell $b_i := c_{i+1}^{(1)}$ get the label B when there is a left turn at the current cell c_i (Fig. 4). In order that the concerned region E_i is an ensuing hole, the cell next to c_{i+1} should be one of the cells constituting $A_1(c_{i+1}) \smallsetminus \{c_i\} = \{c_{i+1}^{(2)}, c_{i+1}^{(4)}, c_{i+1}^{(6)}\}$.

If $E_i \subset A_0(c_{i+1})$, and E_i be such that a free path from a cell a_i in $E_i \cap A_1(c_{i+1})$ cannot be made to b_i, then ρ may enter the cell a_i in the next iteration from where a free path to b_i is not possible. This prevents E_i to be an ensuing hole. For example, if $E_i = \{c_{i+1}^{(2)}, c_{i+1}^{(4)}\}$ and the cell next to c_{i+1} is $c_{i+1}^{(4)}$, then a free path from $c_{i+1}^{(4)}$ to b_i is not possible; the free path, however, exists if $E_i = \{c_{i+1}^{(2)}, c_{i+1}^{(3)}, c_{i+1}^{(4)}\}$. Hence, if $E_i \subset A_0(c_{i+1})$ be such that a free path from each cell $a_i \in E_i \cap A_1(c_{i+1})$ to b_i is possible, then E_i is an ensuing hole. Conversely, if there exists a free path from each $a_i \in E_i \cap A_1(c_{i+1})$ to b_i, then there is always a free path from $c_{i+2} \in E_i$ to b_i. Thus, $E_i \subset A_0(c_i)$ is an ensuing hole if and only if E1–E2 are true.

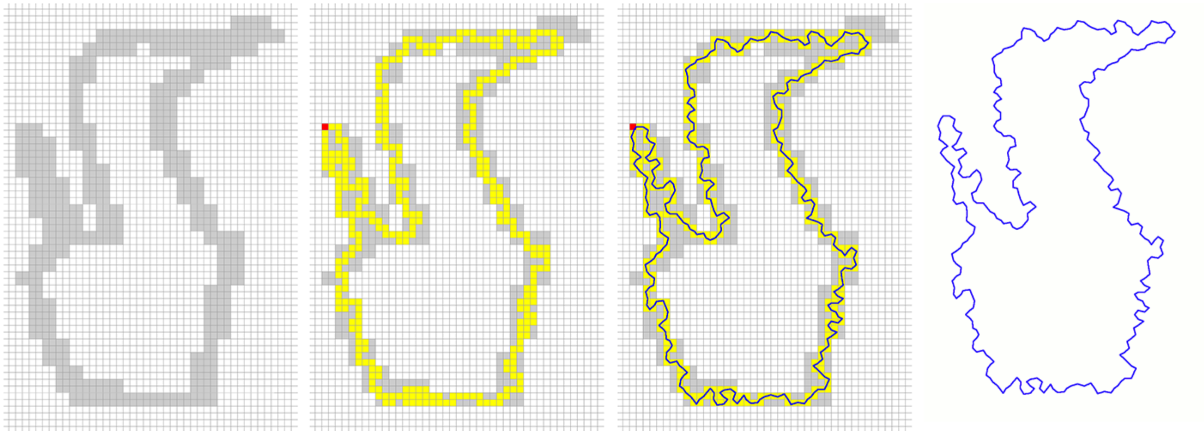
The general proof for E2, corresponding to the case $E_i \not\subset A_0(c_{i+1})$, follows from mathematical induction on the number n of (free) cells in E_i. Let there exist a free path $\rho(a_i, b_i)$ for each $a_i \in E_i \cap A_1(c_{i+1})$. The basis of induction is $n = 1$, which occurs only when $E_i = \{c_{i+1}^{(2)}\}$. The inductive hypothesis is, if E2 is true for less than n cells, then the corresponding E_i is an ensuing hole. In the inductive step, we show that, if E2 is true for n cells, then E_i is an ensuing hole.

The cell c_{i+1} is the current cell in the $(i+1)$th iteration. If the label of some cell b_{i+1} in $\widetilde{N}(c_{i+1})$ is B, then either a hole or an ensuing hole E_{i+1} is again formed in the $(i+1)$th iteration (Corollary 2). Considering the facts that $b_{i+1} \in E_i$, $b_{i+1} \notin E_{i+1}$ (since $\delta(b_{i+1}; c_{i+1}) = \mathtt{B}$), $c_{i+2} \in E_i$, and $c_{i+2} \notin E_{i+1}$, we get $E_{i+1} \subset E_i$, whence $|E_{i+1}| < |E_i| = n$. If the label of no cell in $\widetilde{N}(c_{i+1})$ is B, then also $|E_{i+1}| < n$, since $E_{i+1} = E_i \smallsetminus \{c_{i+2}\}$. We have the following three cases corresponding to E_{i+1}:
Case 1: E1 and E2 are true; E_{i+1} is an ensuing hole (inductive hypothesis).
Case 2: E1 is true, but E2 is not; E_{i+1} is a hole (Corollary 2).
Case 3: E1 is not true; E_{i+1} is neither a hole nor an ensuing hole (Corollary 2).[2]

In Case 1, there exists a free path $\rho(c_{i+2}, b_{i+1})$. Since $b_{i+1} \in \widetilde{N}(c_{i+1}) \subset A_0(c_{i+1})$ and $b_{i+1} \in E_i$, we have $b_{i+1} \in E_i \cap A_0(c_{i+1})$, which means either $b_{i+1} \in E_i \cap A_1(c_{i+1})$ or $b_{i+1} \in E_i \cap (A_0(c_{i+1}) \smallsetminus A_1(c_{i+1}))$. If $b_{i+1} \in E_i \cap A_1(c_{i+1})$, then we get a free path from b_{i+1} to b_i, and hence a free path from c_{i+2} to b_i (via b_{i+1}). If $b_{i+1} \in E_i \cap (A_0(c_{i+1}) \smallsetminus A_1(c_{i+1}))$, then b_{i+1} is adjacent to some cell in $E_i \cap A_1(c_{i+1})$, and since each cell in $E_i \cap A_1(c_{i+1})$ has a free path to b_i (E2), a free path exists from c_{i+2} to b_i.

In Case 2, since E_{i+1} is a hole, ρ does not enter E_{i+1}; i.e., $c_j \notin E_{i+1}$ for $j > i+1$. Since $c_{i+2} \in E_i \cap A_1(c_{i+1})$, a

[2] For example, in Fig. 4(b), for $c_{i+2} = c_{i+1}^{(2)}$, E_{i+1} is a hole; for $c_{i+2} = c_{i+1}^{(4)}$ or $c_{i+1}^{(6)}$, E_{i+1} is an ensuing hole as $\delta(c_{i+1}^{(2)})$ changes from L to B.



(a) Canvas (b) Visited cells (c) Traversed curve (d) Final output

Figure 5. Result of a random curve (blue) blocking some canvas cells (yellow) generated in a "duck"-shaped canvas (gray).

free path exists from c_{i+2} to b_i, wherefore E_i is an ensuing hole.

In Case 3, E_{i+1} has size $n-1$, and is neither a hole nor an ensuing hole. Hence, after entering $c_{i+2} \in E_i \cap A_1(c_{i+1})$, in case the next cell c_{i+3} leads to a hole, ρ would traverse the free path $\rho(c_{i+2}, b_i) \subseteq A_0(c_{i+1})$, whence E_i becomes an ensuing hole.

Conversely, if E_i is an ensuing hole, then E1 is true, and each path from $c_{i+1} \in E_i$ (via any cell in E_i) to $c_i^{(5)}$ contains b_i (Definition 1). Hence, there exists a free path from each cell in $E_i \cap A_1(c_i)$ to b_i, which implies E2 is true. $\square$

A hole event is evident from Corollary 2 and Theorem 3:

Corollary 3 *c_i gives rise to a hole H_i if and only if E1 is true and E2 is false.*

3. Conclusions

The strength of Theorem 3 and Corollary 3 is that, starting from c_1, each cell is randomly and safely selected from the available free cells in the neighborhood of the current cell using the cell components β and δ. Whether there arises any ensuing hole or hole is determined at the current cell by checking E1 and E2, which needs constant time. Since each cell, getting the interim label B, is finally occupied (Theorem 2), and the canvas is initialized in such a way that $\delta(c_1^{(5)}) = \mathtt{B}$ (Fig. 2), ρ reaches the cell $c_1^{(6)}$ prior to $c_1^{(5)}$ [as any free path from c_1 to $c_1^{(5)}$ contains $c_1^{(6)}$] after traversing randomly in the canvas. The algorithm is made to terminate when the next cell c_{i+1} is $c_1^{(6)}$ so that the point p_{k-1} (selected randomly) on the right edge of $c_1^{(6)}$, can be joined with the point p_k on the top edge of $c_1^{(6)}$, and then to p_1, as shown in Fig. 2, thereby producing a random curve. Leaving aside the initialization time of the canvas, the time complexity of generating a random curve that blocks k cells of the canvas is, therefore, given by $\Theta(k)$.

References

[1] T. Auer and M. Held. Heuristics for the generation of random polygons. In *Proc. CCCG*, pages 38–44, 1996. 1

[2] P. Bhowmick and B. B. Bhattacharya. Fast polygonal approximation of digital curves using relaxed straightness properties. *IEEE Trans. PAMI*, 29(9):1590–1602, 2007. 1

[3] I. Debled-Rennesson, R. Jean-Luc, and J. Rouyer-Degli. Segmentation of discrete curves into fuzzy segments. *Electronic Notes in Discrete Mathematics*, 12:372–383, 2003. 1

[4] I. D.-Rennesson and J. P. Reveilles. A linear algorithm for segmentation of digital curves. *IJPRAI*, 9:635–662, 1995.

[5] A. Einstein. Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Annalen der Physik*, 17:549–560, 1905. 1

[6] A. Einstein. *Investigations on the Theory of Brownian Movement*. Dover Publications, NY, 1926 (reprint: 1956). 1

[7] P. Epstein. Generating geometric objects at random. CS Dept., Master thesis, Carleton University, Canada, 1992. 1

[8] S. Finch. Pòlya's random walk constant. In § *5.9 in Mathematical Constants*, Cambridge University Press, 2003. 1

[9] R. Klette and A. Rosenfeld. *Digital Geometry: Geometric Methods for Digital Picture Analysis*. Morgan Kaufmann, San Francisco, 2004. 1

[10] G. Pòlya. Über eine Aufgabe der Wahrscheinlichkeitsrechnung betreffend die Irrfahrt im Strassennetz. *Mathematische Annalen*, 84:149–160, 1921. 1

[11] S. Rohde and O. Schramm. Basic properties of sle. *Annals Math.*, 161:879–920, 2005. 1

[12] J. Rourke and M. Virmani. Generating random polygons. TR:011, CS Dept., Smith College, Northampton, 1991. 1

[13] A. W. M. Smeulders and L. Dorst. Decomposition of discrete curves into piecewise straight segments in linear time. *Contemporary Math.*, 119:169–195, 1991. 1

[14] C. Zhu, G. Sundaram, J. Snoeyink, and J. S. B. Mitchell. Generating random polygons with given vertices. *CGTA*, 6:277–290, 1996. 1