The Annals of Applied Probability
2017, Vol. 27, No. 2, 651–685
DOI: 10.1214/16-AAP1183

RECONSTRUCTION OF A MULTIDIMENSIONAL SCENERY WITH A BRANCHING RANDOM WALK

BY HEINRICH MATZINGER, ANGELICA PACHON[1] AND SERGUEI POPOV

Georgia Institute of Technology, University of Turin and University of Campinas

We consider a d-dimensional scenery seen along a simple symmetric branching random walk, where at each time each particle gives the color record it observes. We show that up to equivalence the scenery can be reconstructed a.s. from the color record of all particles. To do so, we assume that the scenery has at least $2d + 1$ colors which are i.i.d. with uniform probability. This is an improvement in comparison to Popov and Pachon [*Stochastics* **83** (2011) 107–116], where at each time the particles needed to see a window around their current position, and in Löwe and Matzinger [*Ann. Appl. Probab.* **12** (2002) 1322–1347], where the reconstruction is done for $d = 2$ with a single particle instead of a branching random walk, but millions of colors are necessary.

1. Introduction. The classical *scenery reconstruction problem* considers a coloring $\xi : \mathbb{Z} \to \{1, 2, \ldots, \kappa\}$ of the integers $\mathbb{Z}$. To ξ, one refers as *the scenery* and to κ as the *number of colors of* ξ. Furthermore, a particle moves according to a recurrent random walk S on $\mathbb{Z}$ and at each instant of time t observes the color $\chi_t := \xi(S_t)$ at its current position S_t. The scenery reconstruction problem is formulated through the following question: From a color record $\chi = \chi_0\chi_1\chi_2\ldots$, can one reconstruct the scenery ξ? The complexity of the scenery reconstruction problem varies with the number of colors κ, it generally increases as κ decreases.

In [10], it was shown that there are some strange sceneries which cannot be reconstructed. However, it is possible to show that a.s. a "typical" scenery, drawn at random according to a given distribution, can be reconstructed (possibly up to shift and/or reflection).

In [13, 14], it is proved that almost every scenery can be reconstructed in the one-dimensional case. In this instance, combinatorial methods are used; see, for example, [9, 11, 15–17, 19] and [18]. However, all the techniques used in one dimension completely fail in two dimensions; see [8].

In [12], a reconstruction algorithm for the 2-dimensional case is provided. However, in [12] the number of colors in the scenery is very high (order of several

Received March 2014; revised June 2015.
[1]Supported in part by the project AMALFI (Universitá di Torino/ Compagnia di San Paolo).
MSC2010 subject classifications. 60J05, 60J80.
Key words and phrases. Random walk, branching random walk, reconstruction algorithm.

billions). The d-dimensional case is approached in [21] using a branching random walk and assuming that at each instant of time t, each particle is seeing the observations contained in a box centered at its current location.

In the present article, we show that we can reconstruct a d-dimensional scenery using again a branching random walk but only observing at each instant of time t, the color at the current positions of the particles. For this, we only need at least $2d + 1$ colors. Our method exploits the combinatorial nature of the d-dimensional problem in an entirely novel way.

The scenery reconstruction problem goes back to questions from Kolmogorov, Kesten, Keane, Benjamini, Perez, Den Hollander and others. A related well-known problem is to *distinguish sceneries*: Benjamini, den Hollander and Keane independently asked whether all nonequivalent sceneries could be distinguished. An outline of this problem is as follows: Let η_1 and η_2 be two given sceneries, and assume that either η_1 or η_2 is observed along a random walk path, but we do not know which one. Can we figure out which of the two sceneries was taken? Kesten and Benjamini in [1] proved that one can distinguish almost every pair of sceneries even in two dimensions and with only two colors. For this, they take η_1 and η_2 both i.i.d. and independent of each other. Prior to that, Howard had shown in [3, 4] and [5] that any two periodic one-dimensional nonequivalent sceneries are distinguishable, and that one can a.s. distinguish single defects in periodic sceneries. *Detecting a single defect in a scenery* refers to the problem of distinguishing two sceneries which differ only in one point. Kesten proved in [7] that one can a.s. recognize a single defect in a random scenery with at least five colors. As fewer colors lead to a reduced amount of information, it was conjectured that it might not be possible to detect a single defect in a 2-color random scenery. However, in [14] the first named author showed that the opposite is true—a single defect in a random 2-color scenery can be detected. He also proved that the whole scenery can be reconstructed without any prior knowledge.

One motivation to study scenery reconstruction and distinguishing problems was the T, T^{-1}*-problem* that origins in a famous conjecture in ergodic theory due to Kolmogorov. He demonstrated that every Bernoulli shift T has a trivial tail-field and conjectured that also the converse is true. Let $\mathcal{K}$ denote the class of all transformations having a trivial tail-field. Kolmogorov's conjecture was shown to be wrong by Ornstein in [20], who presented an example of a transformation which is in $\mathcal{K}$ but not Bernoulli. His transformation was particularly constructed to resolve Kolmogorov's conjecture. In 1971, Ornstein, Adler and Weiss came up with a very natural example which is $\mathcal{K}$ but appeared not to be Bernoulli; see [22]. This was the so-called T, T^{-1}*-transformation*, and the T, T^{-1}-problem was to verify that it was not Bernoulli. It was solved by Kalikow in [6], showing that the T, T^{-1}-transformation is not even loosely Bernoulli. A generalization of this result was recently proved by den Hollander and Steif [2].

2. Model and statement of results. We consider a random coloring of the integers in d-dimension. Let ξ_z be i.i.d. random variables, where the index z ranges over $\mathbb{Z}^d$. The variables ξ_z take values from the set $\{0, 1, 2, \ldots, \kappa - 1\}$ where $\kappa \geq 4$, and all the values from $\{0, 1, 2, \ldots, \kappa - 1\}$ have the same probability. A realization of $\xi = \{\xi_z\}_{z \in \mathbb{Z}^d}$ thus is a (random) coloring of $\mathbb{Z}^d$. We call this random field ξ a *d-dimensional scenery*.

Now assume that a branching random walk (BRW) on $\mathbb{Z}^d$ observes the d-dimensional scenery ξ, that is, each particle of the BRW observes the color at its current position.

Formally a branching random walk in $\mathbb{Z}^d$ is described as follows. The process starts with one particle at the origin, then at each step, any particle is substituted by two particles with probability b and is left intact with probability $1 - b$, for some fixed $b \in (0, 1)$. We denote by N_n the total number of particles at time n. Clearly, $(N_n, n = 0, 1, 2, \ldots)$ is a Galton–Watson process with the branching probabilities $\tilde{p}_1 = 1 - b$, $\tilde{p}_2 = b$. This process is supercritical, so it is clear that $N_n \to \infty$ a.s. Furthermore, each particle follows the path of a simple random walk, that is, each particle jumps to one of its nearest neighbors chosen with equal probabilities, independently of everything else.

To give the formal definition of the observed process, we first introduce some notation. Let $\eta_t(z)$ be the number of particles at site $z \in \mathbb{Z}^d$ at time $t \geq 0$, with $\eta_0(z) = \mathbf{1}\{z = 0\}$. We denote by $\eta_t = (\eta_t(z))_{z \in \mathbb{Z}^d}$ the configuration at time t of the branching random walk on $\mathbb{Z}^d$, starting at the origin with branching probability b.

Let G be the genealogical tree of the Galton–Watson process, where $G_t = \{v_1^t, \ldots, v_{N_t}^t\}$ are the particles of tth generation. Let $S(v_j^t)$ be the position of v_j^t in $\mathbb{Z}^d$, that is, $S(v_j^t) = z$ if the jth particle at time t is located on the site z. Recall that we do not know the position of the particles, only the color record made by the particles at every time, as well as the number of particles at each time.

According to our notation, $\{z \in \mathbb{Z}^d : \eta_t(z) \geq 1\} = \{\exists j; S(v_j^t) = z, j = 1, \ldots, N_t\}$. The *observations* χ, to which we also refer as *observed process*, is a coloring of the random tree G. Hence, the observations χ constitute a random map:

$$\chi : G_t \to \{0, 1, 2, \ldots, \kappa - 1\},$$
$$v_i^t \mapsto \chi(v_i^t) = \xi(S(v_i^t)), \qquad i = \{1, \ldots, N_t\}.$$

In other words, the coloring χ of the random tree G yields the color the particle i at time t sees in the scenery from where it is located at time t.

Denote by $\Omega_1 = \{(\eta_t)_{t \in \mathbb{N}}\}$ the space of all possible "evolutions" of the branching random walk, by $\Omega_2 = \{0, 1, 2, \ldots, \kappa - 1\}^{\mathbb{Z}^d}$ the space of all possible sceneries ξ and by $\Omega_3 = \{(\chi_t)_{t \in \mathbb{N}}\}$ the space of all possible realizations of the observed process. We assume that $(\eta_t)_{t \in \mathbb{N}}$ and ξ are independent and distributed with the laws P_1 and P_2, respectively.

Two sceneries ξ and ξ' are said to be *equivalent* (in this case we write $\xi \sim \xi'$), if there exists an isometry $\varphi : \mathbb{Z}^d \mapsto \mathbb{Z}^d$ such that $\xi(\varphi x) = \xi'(x)$ for all x.

Now we formulate the main result of this paper. The measure P designates the product measure $P_1 \otimes P_2$.

THEOREM 2.1. *For any $b \in (0,1)$ and $\kappa \geq 2d+1$, there exists a measurable function $\Lambda : \Omega_3 \to \Omega_2$ such that $P(\Lambda(\chi) \sim \xi) = 1$.*

The function Λ represents an "algorithm," the observations χ being its input and the reconstructed scenery ξ (up to equivalence) being its output. The main idea used to prove Theorem 2.1 is to show how to reconstruct a finite piece of the scenery (close to the origin).

2.1. *Main ideas.* We start by defining a reconstruction algorithm with parameter n denoted by Λ^n, which works with all the observations up to time n^2 to reconstruct a portion of the scenery ξ close to the origin. The portion reconstructed should be the restriction of ξ to a box with center close to the origin. This means closer than $\sqrt{n}$, which is a lesser order than the size of the piece reconstructed. We will show that the algorithm Λ^n works with *high probability* (w.h.p. for short, meaning with probability tending to one as $n \to \infty$).

Let us denote by $\mathcal{K}_x(s)$ the box of size s centered at x in $\mathbb{Z}^d$, that is, $x + [-s, s]^d$, and by $\mathcal{K}(s)$ the box of size s centered at the origin of $\mathbb{Z}^d$. For a subset A of $\mathbb{Z}^d$, we designate by ξ_A the restriction of ξ to A, so $\xi_{\mathcal{K}_x(s)}$ denotes the restriction of ξ to $\mathcal{K}_x(s)$.

In what follows, we will say that w is a word of size k of ξ_A, if it can be read in a straight manner in ξ_A. This means that w is a word of ξ_A if there exist an $x \in \mathbb{Z}^d$ and a canonical vector $\vec{e}$ (it defined to be one that has only one nonzero entry equal to $+1$ or -1), so that $x + i\vec{e} \in A$, for all $i = 0, 1, 2, \ldots, k-1$, and

$$w = \xi_x \xi_{x+\vec{e}} \xi_{x+2\vec{e}} \cdots \xi_{x+(k-1)\vec{e}}.$$

Let A and B be two subsets of $\mathbb{Z}^d$. Then we say that ξ_A and ξ_B are equivalent to each other and write $\xi_A \sim \xi_B$, if there exists an isometry $\varphi : A \mapsto B$ such that, $\xi_A \circ \varphi = \xi_B$.

2.2. *The algorithm Λ^n for reconstructing a finite piece of scenery close to the origin.* The four phases of this algorithm are described in the following way:

1. *First phase*: *Construction of short words of size* $(\ln n)^2$. The first phase aims at reconstructing words of size $(\ln n)^2$ of $\xi_{\mathcal{K}(n^2)}$. The set of words constructed in this phase is denoted by *SHORTWORDS*^{n}. It should hopefully contain all words of size $(\ln n)^2$ in $\xi_{\mathcal{K}(4n)}$, and be contained in the set of all words of size $(\ln n)^2$ in $\xi_{\mathcal{K}(n^2)}$. The accurate definition of *SHORTWORDS*^{n} is as follows: A word w_2 of size $(\ln n)^2$ is going to be selected to be in *SHORTWORDS*^{n} if there exist two strings w_1 and w_3 both of size $(\ln n)^2$ and such that:

(a) $w_1 w_2 w_3$ appears in the observations before time n^2, and

(b) the only word w of size $(\ln n)^2$ such that $w_1 w w_3$ appears in the observations up to time n^4 is w_2.

Formally, let $W(\xi_{\mathcal{K}(4n)})$ and $W(\xi_{\mathcal{K}(n^2)})$ be the sets of all words of size $(\ln n)^2$ in $\xi_{\mathcal{K}(4n)}$ and $\xi_{\mathcal{K}(n^2)}$, respectively, then

$$W(\xi_{\mathcal{K}(4n)}) \subseteq \mathit{SHORTWORDS}^n \subseteq W(\xi_{\mathcal{K}(n^2)}). \tag{2.1}$$

We prove that (2.1) holds w.h.p. in Section 3.1.

2. *Second phase*: *Construction of long words of size* $4n$. The second phase assembles the words of $\mathit{SHORTWORDS}^n$ into longer words to construct another set of words denoted by $\mathit{LONGWORDS}^n$. The rule is that the words of $\mathit{SHORTWORDS}^n$ to get assembled must coincide on $(\ln n)^2 - 1$ consecutive letters, and it is done until getting strings of total size exactly equal to $4n + 1$. In this phase, let $\mathcal{W}_{4n}(\xi_{\mathcal{K}(4n)})$ and $\mathcal{W}_{4n}(\xi_{\mathcal{K}(n^2)})$ be the set of all words of size $4n$ in $\xi_{\mathcal{K}(4n)}$ and $\xi_{\mathcal{K}(n^2)}$, respectively, then

$$\mathcal{W}_{4n}(\xi_{\mathcal{K}(4n)}) \subseteq \mathit{LONGWORDS}^n \subseteq \mathcal{W}_{4n}(\xi_{\mathcal{K}(n^2)}). \tag{2.2}$$

We achieve that (2.2) holds w.h.p. in Section 3.2.

3. *Third phase*: *Selecting a seed word close to the origin*. The third phase selects from the previous long words one which is close to the origin. For that, Λ^n applies the previous two phases, but with the parameter being equal to $n^{0.25}$ instead of n. In other words, Λ^n chooses one (any) word w_0 of $\mathit{LONGWORDS}^{n^{0.25}}$, and then chooses the word w_L in $\mathit{LONGWORDS}^n$ which contains w_0 in such a way that the relative position of w_0 inside w_L is centered. (The middle letters of w_0 and w_L must coincide.) Λ^n places the word w_L so that the middle letter of w_0 is at the origin. See Section 3.3.

4. *Fourth phase*: *Determining which long words are neighbors of each other*. The fourth phase place words from $\mathit{LONGWORDS}^n$ in the correct relative position to each other, thus assembling the scenery in a box near the origin. For this, Λ^n starts with the first long-word which was placed close to the origin in the previous phase. Then words from the set $\mathit{LONGWORDS}^n$ are placed parallel to each other until a piece of scenery on a box of size $4n$ is completed.

Let us briefly explain how Λ^n chooses which words of $\mathit{LONGWORDS}^n$ are *neighbors* of each other in the scenery $\xi_{\mathcal{K}(n^2)}$, (i.e., they are parallel and at distance 1). Λ^n estimates that the words v and w which appear in $\xi_{\mathcal{K}(n^2)}$ are neighbors of each other iff the three following conditions are all satisfied:

(a) First, there exist 4 words v_a, v_b, v_c and w_b having all size $(\ln n)^2$ except for v_b which has size $(\ln n)^2 - 2$, and such that the concatenation $v_a v_b v_c$ is contained in v, whilst up to time n^4 it is observed at least once $v_a w_b v_c$.
(b) Second, the word w_b is contained in w.
(c) Finally, the relative position of v_b in v should be the same as the relative position of w_b in w. By this, we mean that the middle letter of v_b has the same position in v as the middle letter of w_b in w has.

See the precise definition in Section 3.4.

Let $\mathcal{B}$ be the set of all finite binary trees, and let χ_t designates all the observations made by the branching random walk up to time t. That χ_t is the restriction of the coloring χ to the subtree $\bigcup_{i \leq t} G_i$.

The next result means that the algorithm Λ^n works w.h.p.

THEOREM 2.2. *Assume that the number of colors κ satisfies the condition that $\kappa \geq 2d+1$. Then, for every $n \in \mathbb{N}$ large enough, the map*

$$\Lambda^n : \{0, 1, \ldots, \kappa - 1\}^{\mathcal{B}} \to \{0, 1, \ldots, \kappa - 1\}^{\mathcal{K}(4n)},$$

defined above as our algorithm satisfies

$$P\big[\exists x \in \mathcal{K}(\sqrt{n}) \text{ so that } \Lambda^n(\chi_{n^4}) \sim \xi_{\mathcal{K}_x(4n)}\big] \geq 1 - \exp\big(-C(\ln n)^2\big), \tag{2.3}$$

where $C > 0$ is a constant independent of n.

In words, the algorithm Λ^n manages to reconstruct w.h.p. a piece of the scenery ξ restricted to a box of size $4n$ close to the origin. The center of the box has every coordinate not further than $\sqrt{n}$ from the origin. The reconstruction algorithm uses only observations up to time n^4.

We will give the exact proof of the above theorem in the next section, but, before we present the main ideas in a less formal way in the remainder of this section. We first want to note that the algorithm Λ^n reconstructs a piece of the scenery ξ in a box of size $4n$, but the exact position of that piece within ξ will in general not be known after the reconstruction. Instead, the above theorem insures that w.h.p. the center of the box is not further than an order $\sqrt{n}$ from the origin.

For what follows, we will need a few definitions: Let $[0, k-1]$ designate the sequence $\{0, 1, 2, 3 \ldots, k-1\}$ and R be a map $R : [0, k-1] \to \mathbb{Z}^d$ such that the distance between $R(i)$ and $R(i+1)$ is 1 for every $i = 0, 1, 2, \ldots, k-2$. We call such a map a *nearest neighbor path of length* $k-1$. Let x and y be two points in $\mathbb{Z}^d$. If $R(0) = x$ and $R(k-1) = y$, we say that R goes from x to y. We also say that R starts in x and ends in y. Let w be a string of colors of size k, such that

$$w = \xi\big(R(0)\big)\xi\big(R(1)\big)\cdots\xi\big(R(k-1)\big).$$

In that case, we say that R *generates* w *on the scenery* ξ.

2.2.1. *The DNA-sequencing trick.* We use the same trick as is used in modern methods for DNA-sequencing where instead of reconstructing the whole DNA-sequence at once, one tries to reconstruct smaller pieces simultaneously. Then to obtain the entire piece one puzzles the smaller pieces together. In this paper, the scenery is multidimensional unlike DNA-sequences. Nonetheless, we first present the reconstruction method for DNA-sequencing in this subsection, because it is easiest to understand. The trick goes as follows. Assume that you have a genetic

sequence $D = D_1 D_2 \cdots D_{n^\alpha}$, where $\alpha > 0$ is a constant independent of n, and D is written in an alphabet with $\kappa > 0$ equiprobable letters. For instance in DNA-sequences, this alphabet is $\{A, C, T, G\}$. To determine the physical order of these letters in a sequence of DNA, modern methods use the idea of do not go for the whole sequence at once, but first determine small pieces (subsequences) of order at least $C \ln(n)$ and then assemble them to obtain the whole sequence. (Here, $C > 0$ is a constant independent of n, but dependent of α.) Let us give an example.

EXAMPLE 2.1. Let the sequence be equal to

$$D = TATCAGT,$$

and suppose a biologist in his laboratory is able to find all subsequences of size 5. Typically, the direction in which these subsequences appear in D is not known. The set of all subsequences of size 5 which appear in D is

$$TATCA, ATCAG, TCAGT,$$

as well as their reverses

$$ACTAT, GACTA, TGACT.$$

Which one are to be read forward and which one backward is not known to the biologist. He is only given all these subsequences in one bag without extra information. If he was given all the subsequences of size 5 appearing in D, he could reconstruct D by assembling these subsequences using the assembly rule that they must coincide on a piece of size 4. Why? Consider the set of all subsequences of size 4:

$$TATC, ATCA, TCAG, CAGT$$

and their reverses

$$CTAT, ACTA, GACT, TGAC.$$

Note that each of these appears only once. Hence, four consecutive letters determine uniquely the relative position of the subsequences of size 5 with respect to each other, and given the bag of subsequences of size 5 is possible to assemble theme one after the other. How? Start by picking any subsequence. Then put down the next subsequence from the bag which coincides with the previous one on at least 4 letters. For example, take *GACTA* and put it down on the integers in any position, for instance,

G	A	C	T	A
0	1	2	3	4

.

Then take another subsequence from the bag which coincides with the previously chosen one, on at least 4 contiguous letters. For example, *TGACT* satisfies this

condition. Now superpose the new subsequence onto the previous one so that on 4 letters they coincide:

T	G	A	C	T		
	G	A	C	T	A	.
-1	0	1	2	3	4	

This leads to:

T	G	A	C	T	A	
-1	0	1	2	3	4	.

Next, observe that the subsequence *ACTAT* coincides on at least 4 consecutive letters, with what has been reconstructed so far. So, put *ACTAT* down in a matching position:

		A	C	T	A	T	
T	G	A	C	T	A		.
-1	0	1	2	3	4	5	

This leads to:

T	G	A	C	T	A	T	
-1	0	1	2	3	4	5	.

The final result is the sequence *TGACTAT* which is D read in reverse order.

But how do we know that this method works? In the example, we saw the sequence D before hand, and could hence verify that each subsequence of size 4 appears in at most one position in D. However, the biologist can not verify this condition. He only gets the bag of subsequences as only information. So, the idea is that if we know the stochastic model which generates the sequence, we can calculate that w.h.p. each subsequence of size $C\ln(n)$ appears only once in D, provided the constant $C > 0$ is taken large enough. Take for example the i.i.d. model with $\kappa > 1$ equiprobability letters. Then the probability that two subsequences located in different and nonintersecting parts of D be identical is given by

$$P(E_{i,j}^{n+}) = \left(\frac{1}{\kappa}\right)^{C\ln n} = n^{-C\ln(\kappa)}, \tag{2.4}$$

where $E_{i,j}^{n+}$ is the event that

$$D_{i+1}\cdots D_{i+C\ln n} = D_{j+1}\cdots D_{j+C\ln n}. \tag{2.5}$$

Similarly, let $E_{i,j}^{n-}$ be the event that

$$D_{i+1}\cdots D_{i+C\ln n} = D_{j-1}\cdots D_{j-C\ln n}. \tag{2.6}$$

Note that even if the subsequences given on both sides of (2.5) or (2.6) intersect, as long as they are not exactly in the same position, we get that (2.4) still holds. To see this, take for example the subsequence $w = D_{i+1}D_{i+2}\cdots D_{i+C(\ln n)}$ and the subsequence $v = D_{i+2}D_{i+3}\cdots D_{i+C(\ln n)+1}$. These two subsequences are not at all independent of each other since up to two letters they are identical to each other. However, we still have that $P(w = v)$ is equal to

$$P(D_{i+1}D_{i+2}\cdots D_{i+C(\ln n)} = D_{i+2}D_{i+3}\cdots D_{i+C(\ln n)+1}) = \left(\frac{1}{\kappa}\right)^{C\ln n}. \tag{2.7}$$

To see why this holds, simply note that D_{i+2} is independent of D_{i+1}, so w and v agree on the first letter with probability $1/\kappa$. Then D_{i+3} is independent of D_{i+2} and D_{i+1}, so the second letter of v has a probability of $1/\kappa$ to be identical to the second letter of w, and so on. Thus, to get that w.h.p. no identical subsequence appears in any two different positions in D, we need to get a suitable upper bound of the right-hand side of (2.4). If we take i and j both in n^α, we find that the probability to have at least one subsequence appearing in two different positions in D, can be bounded in the following way:

$$P\Big(\bigcup_{i\neq j} E_{i,j}^{n+}\Big) \le \sum_{i\neq j} P\big(E_{i,j}^{n+}\big) \le n^{2\alpha}\cdot n^{-C\ln(\kappa)}. \tag{2.8}$$

The same type of bound also holds for $P(E_{i,j}^{n-})$ using (2.6). We take C strictly larger than $2\alpha/\ln\kappa$ in order to get the right-hand side of (2.8) be negatively polynomially small in n.

For our algorithm Λ^n, we will take subsequences (the short words) to have size $(\ln n)^2$ instead of being linear in $\ln n$. Thus, we do not have to think about the constant in front of $\ln n$. Then the bound we get for the probability becomes even better than negative polynomial in n. It becomes of the type $n^{-\beta n}$ where $\beta > 0$ is a constant independent of n.

2.2.2. *Applying the DNA-sequencing method to a multidimensional scenery.* In our algorithm Λ^n, we do not have the task to reconstruct a sequence, instead we have to reconstruct a multidimensional scenery restricted to a box. We use the same idea as the DNA-sequencing method except that we will reconstruct several long words instead of reconstructing just one long word (one sequence). This corresponds to the second phase of Λ^n where with a bag of short words, it constructs a collection of long words. These words will be the different restrictions of the scenery ξ to straight lines-segments parallel to some direction of coordinates. These long words, of course, do not yet tell us exactly how the scenery looks; we will still need to position these long words correctly with respect to each other. (This is done in the fourth phase of Λ^n.) Let us explain that with another example.

EXAMPLE 2.2. Let us assume $d = 2$ and the scenery ξ restricted to the box $[0,4]\times[0,4]$ be given by

$$\xi_{[0,4]\times[0,4]} = \begin{array}{|c|c|c|c|c|}\hline 1&9&4&3&7\\\hline 5&0&7&6&1\\\hline 4&3&9&1&2\\\hline 6&1&4&0&4\\\hline 2&7&8&0&3\\\hline\end{array}. \tag{2.9}$$

Here, we assume that the alphabet has size $\kappa = 10$ and that we would be given a bag of all short words of size 4 appearing in (2.9). That is words which can be read

horizontally or vertically in either direction: from up to down, down to up, left to right, right to left and of size 4. This bag of short-words is

$$\begin{aligned}&\{1943, 5076, 4391, 6140, 27803, 9437, 0761, 3912, 1404, 7803,\\ &1546, 9031, 4794, 3610, 7124, 5462, 0317, 7948, 6100, 1243\},\end{aligned} \tag{2.10}$$

and their reverses. As assemble rule we use that words must coincide on at least 3 consecutive letters. We can, for example, assemble 1943 with 9437 to get 19437. Similarly, we assemble 1546 with 5462 and obtain 15462. So, we apply the DNA-puzzling trick, but instead of reconstruct only one long word, we will reconstruct several long words. In this example, we reconstruct the set of 10 long words and their reverses:

$$\begin{aligned}&\{19437, 50761, 43912, 61404, 27803,\\ &15462, 90317, 47948, 36100, 71243\},\end{aligned} \tag{2.11}$$

where again each of the above long words could be its own reverse.

Thus, the previous example shows how the second phase of the algorithm works: From a set of shorter words $SHORTWORDS^n$, we obtain a set of longer words $LONGWORDS^n$ in the same manner how we obtained the set (2.11) from the bag of words (2.10). Note that the long words are successfully reconstructed in this example, because in the restriction of the scenery in (2.9), any word of size 3 appears at most in one place.

The differences in the numeric example presented above and how the second phase of Λ^n works are the following: the short words in Λ^n have length $(\ln n)^2$ and the long words have length $4n$, instead of 4 and 5. Moreover, Λ^n will need to assemble in the second phase many short words to get one long word, despite in this example where we just used two short words to get each long word. On the other side, in the previous example is given to us a bag of all words of size 4 of the restriction of ξ to the box $[0, 4] \times [0, 4]$. However, the second phase of Λ^n has the bag of short words $SHORTWORDS^n$, which is not exactly equal to all words of size $(\ln n)^2$ of ξ restricted to a box. Instead, it is the bag of words that contains all words of size $(\ln n)^2$ of ξ, restricted to the box $\mathcal{K}(4n)$, but augmented by some other words of the bigger box $\mathcal{K}(n^2)$.

The bag of short words $SHORTWORDS^n$ is obtained in the first phase of Λ^n. The reason why the first phase is not able to identify which words are in the box $\mathcal{K}(4n)$ and which are in $\mathcal{K}(n^2)$ is as follows: the observations in the first phase of Λ^n are taken up to time n^2. Since we assume that the first particle starts at time 0 at the origin, by time n^2 all particles must be contained in the box $\mathcal{K}(n^2)$, and the probability for one given particle at time n^2 to be close to the border of $\mathcal{K}(n^2)$ is exponentially small. Since we have many particles, a few will be close to the border of $\mathcal{K}(n^2)$ by time n^2, and these will be enough particles to provide some

few words close to the border of $\mathcal{K}(n^2)$ by time n^2, and selected in the first phase of the algorithm.

There is an important consequence to this in the second phase of the algorithm, since some reconstructed long words in $LONGWORDS^n$ might also be "far out in the box $\mathcal{K}(n^2)$" and not in $\mathcal{K}(4n)$.

2.2.3. *The diamond trick to reconstruct all the words of* $\xi_{\mathcal{K}(4n)}$. In the previous subsection, we showed how to assemble shorter words to get longer ones, but we have not yet explained the main idea of how we manage to obtain short words in the first phase of Λ^n, being given only the observations. The basic idea is to use the *diamonds associated with a word appearing in the scenery*. Let us look at an example.

EXAMPLE 2.3. Take the following piece of a two-dimensional scenery which would be the restriction of ξ to the $[0,6]\times[0,4]$:

$$\xi_{[0,6]\times[0,4]} = \begin{array}{|c|c|c|c|c|c|c|}\hline 2&1&9&4&3&7&4\\\hline 7&5&0&7&6&1&1\\\hline 7&4&3&9&1&2&1\\\hline 8&6&4&4&0&4&3\\\hline 2&2&7&8&0&3&9\\\hline\end{array}. \tag{2.12}$$

Consider the word

$$w = 43912 = \xi_{(1,2)}\xi_{(2,2)}\xi_{(3,2)}\xi_{(4,2)}\xi_{5,2}$$

which appears in the above piece of scenery in green. That word "appears between the points $x = (1,2)$ and $y = (5,2)$." We only consider here words which are written in the scenery "along the direction of a coordinate." In blue, we highlighted the *diamond associated with the word* 43912. More precisely, the diamond consists of all positions which in the above figure are green or blue.

The formal definition is that if x and y are two points in $\mathbb{Z}^d$ so that $x\bar{y}$ is parallel to a canonical vector $\vec{e}$, then the diamond associated with the word $w = \xi_x\xi_{x+\vec{e}}\xi_{x+2\vec{e}}\cdots\xi_{y-\vec{e}}\xi_y$ consists of all points in $\mathbb{Z}^d$ which can be reached with at most $(|x-y|/2)-1$ steps from the point $(y-x)/2$. We assume here that the Euclidean distance $|x-y|$ is an odd number.

The useful thing will be that w.h.p. for a given nonrandom point z outside the diamond associated with w, there is no nearest neighbor walk path starting at z and generating as observations w. So, w.h.p. a word w in the scenery can only be generated as observations by a nearest neighbor walk path starting (and also ending) in the diamond associated with w. (At least if we restrict the scenery to a box of polynomial size in the length of w.) To see this, take for instance the following path R:

$$\big((0,0),(1,0),(1,1),(1,2),(2,2)\big).$$

In the previous example, a random walker which would follow this little path would observe the sequence of colors given by

$$\xi \circ R = (2, 2, 6, 4, 3). \tag{2.13}$$

Note that the path R and the straight path from x to y intersect. So, (2.13) and the green word $w = 43912$ are not independent of each other. However, because the path R starts outside the diamond, we get the probability of the event that (2.13) and the word w are identical, has the same probability as if they would be independent. That is assuming that R is a (nonrandom) nearest neighbor path starting (or ending) at a given point z outside the diamond associated with a word w,

$$P(\xi \circ R = \xi_x \xi_{x+\vec{e}} \xi_{x+2\vec{e}} \cdots \xi_{y-\vec{e}} \xi_y) = \left(\frac{1}{\kappa}\right)^m,$$

where m designates the size of the word. In fact looking at our example, R starts at $(0, 0)$ which is outside the diamond. So, the starting point of R is different from $x = (1, 2)$, and hence by independence

$$P\big(\xi\big(R(0)\big) = \xi_x\big) = \frac{1}{\kappa}.$$

Then the second letter in w, that is $\xi_{x+\vec{e}}$ independent of the first two letters of the observations along the path, $\xi(R_0)\xi(R_1)$, since both points $R(0) = (0, 0)$ and $R(1) = (1, 0)$ are different from $x + \vec{e} = (2, 2)$. Thus, we get the probability that the first two letters of w coincide with the first two observations made by R is equal to

$$P\big(\xi\big(R(0)\big)\xi\big(R(1)\big) = \xi_x \xi_{x+\vec{e}}\big) = \left(\frac{1}{\kappa}\right)^2. \tag{2.14}$$

The proof goes on by induction: the kth letter in the word $w_{x+k\vec{e}}$ is independent of the first k observations made by R. The reason is that the first k positions $R(0)R(1)\cdots R(k-1)$ visited by R do never contain the point $x + k\vec{e}$, since "the walker following the path R never catches up with the walker going straight from x to y." On the other hand, observe that in our example, we see a path starting inside the diamond and producing as observation the same green word w. Take the path

$$(2, 1), (2, 2), (3, 2), (4, 2), (5, 2).$$

Thus, this "counterexample" illustrates how "easy" it is for a path starting "inside" the diamond associated with a word w, to generate the same word. This a second path different to the straight path "going from x to y."

Now, using (2.14) we can calculate an upper bound for the probability that for a given nonrandom point z outside the diamond, there exists at least one nonrandom path R starting at z and producing as observation w. Observe that in d-dimensional scenery, for a given starting point $z \in \mathbb{Z}^d$, there are $(2d)^{k-1}$ nearest neighbor walk

paths of length k. So, we find that the probability that there exists a nearest neighbor path R starting at z, with z outside the diamond associated with a word w, and R generating w, has a probability bounded from above as follows:

$$P(\exists \text{ a nearest neighbor path } R \text{ starting at } z \text{ with } \xi \circ R = w) \leq \left(\frac{2d}{\kappa}\right)^{k-1}. \tag{2.15}$$

Note that the bound above is negatively exponentially small in k as soon as $2d < \kappa$. The inequality $2d < \kappa$ is precisely the inequality given in Theorem 2.2 which makes our reconstruction algorithm work.

Thus, in the next section we will define the event B_3^n as follows: Let w be a word of size $(\ln n)^2$ in $\xi_{\mathcal{K}(n^2)}$, and R a nearest neighbor path, $R : [0, k-1] \to \mathcal{K}(n^4)$, so that $\xi \circ R = w$, and R begins and ends in the diamond associated with w.

Note then that $P(B_3^{nc})$ is bounded from above by (2.15) times the number of points in the box $\xi_{\mathcal{K}(n^2)}$. But expression (2.15) is negatively exponentially small in the size of the word $(\ln n)^2$, and hence dominates the polynomial number of points in the box, that is, it goes to zero as n goes to infinity. Thus, B_3^n holds w.h.p. (To see the exact proof, go to Lemma 3.3.)

Now, we know that with high probability the words can only be generated by a nearest neighbor walk path starting (and ending) in the diamond associated with w (at least when we restrict ourselves to a box of polynomial size in the length of w). But, how can we use this to reconstruct words? The best is to consider an example.

EXAMPLE 2.4. For this, let the restriction of the scenery ξ to $[0, 16] \times [0, 4]$ be equal to

$$\xi_{[0,16]\times[0,4]} = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|}\hline 2&1&9&4&3&7&4&1&2&5&2&2&7&8&0&6&9\\\hline 7&5&0&7&6&1&1&8&2&5&8&6&7&4&0&4&2\\\hline 7&4&3&9&1&2&1&7&8&4&7&6&1&7&7&7&4\\\hline 8&6&4&4&0&4&3&5&3&6&7&5&1&9&9&9&1\\\hline 2&2&7&8&0&3&9&4&3&7&2&1&9&4&5&7&0\\\hline\end{array}. \tag{2.16}$$

Let the word which appears in green be denoted by w_1 so that

$$w_1 = 43912.$$

Let the word written in brown be denoted by w_3 so that

$$w_3 = 61777.$$

Finally, let the word which is written when we go straight from the green word to the brown word be denoted by w_2 so that

$$w_2 = 17847.$$

In the current example, the diamond D_1 associated with the green word w_1 is given in blue and the diamond D_3 associated with the brown word w_3 is highlighted in

red. Note that there is only one shortest nearest neighbor path to go from D_1 to D_3, walking straight from the point $(5, 2)$ to the point $(11, 2)$ in exactly six steps. There is not other way to go in six steps from D_1 to D_3. When doing so a walker will see the word w_2. Now assume that the size of our short words is 5 [i.e., the size which in the algorithm is given by the formula $(\ln n)^2$]. Assume also that the rectangle $[0, 16] \times [0, 4]$ is contained in $\mathcal{K}(n^2)$.

Remember that if B_3^n holds, we have that within the box $\mathcal{K}(n^2)$ a nearest neighbor walk can only generate a short word of $\xi_{\mathcal{K}(n^2)}$ if it starts and ends in the diamond associated with that word. Using this to the words w_1 and w_3, we see in the observations the following pattern:

$$w_1 * * * * * w_3,$$

where $*$ is a wild card which stands for exactly one letter, then w.h.p. the walker between w_1 and w_3 was walking in a straight manner from D_1 to D_3. Hence, the wild card sequence $* * * * *$ must then be the word w_2. Of course, we need at least one particle to follow that path in order to observe $w_1 w_2 w_3$ up to time n^2. This will be taken care in Λ^n by the event B_2^n which stipulates that any nearest neighbor path of length $3(\ln n)$ contained in $\mathcal{K}(4n)$, will be followed by at least one particle up before time n^2. In other words, we have proven that if B_2^n and B_3^n both hold, then w_2 gets selected by the first phase of the algorithm as a short word of $SHORTWORDS^n$. The argument of course works for any short word of $\xi_{\mathcal{K}(4n)}$, and hence we have that

$$B_2^n \cap B_3^n \implies W(\xi_{\mathcal{K}(4n)}) \subset SHORTWORDS^n.$$

Thus, the previous example shows how the first phase of the algorithm manages to reconstruct all short words in $\xi_{\mathcal{K}(4n)}$.

2.2.4. *How to eliminate junk observation-strings which are not words of* $\xi_{\mathcal{K}(n^2)}$. In the previous subsection, we have shown how to reconstruct enough words. But now the next question is "how do we manage to not reconstruct too many words?" By this we mean, how do we make sure that observations which do not correspond to words of $\xi_{\mathcal{K}(n^2)}$ do not get selected by the first phase of our algorithm? This means that we have to be able to eliminate observations which do not correspond to a word of $\xi_{\mathcal{K}(n^2)}$. The best is again to see an example.

EXAMPLE 2.5. Take for this

$$\xi_{[0,16]\times[0,4]} = \begin{array}{|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|c|}\hline 2&1&9&4&3&7&4&1&2&5&2&2&7&8&0&6&9\\\hline 7&5&0&7&6&1&1&8&2&5&8&6&7&4&0&4&2\\\hline 7&4&3&9&1&2&1&7&8&4&7&6&1&7&7&7&4\\\hline 8&6&4&4&0&4&3&5&3&6&7&5&1&9&9&9&1\\\hline 2&2&7&8&0&3&9&4&3&7&2&1&9&4&5&7&0\\\hline\end{array}. \tag{2.17}$$

Observe this is the same piece of scenery as was shown in (2.16), but the brown word was moved two units to the left. So, let again w_1 denote the green word

$$w_1 = 43912$$

and let this time w_3 be the "new" brown word:

$$w_3 = 47617.$$

Now a particle in between following the green word and then the brown word could for example do the following little dance step:

$$right, up, right, down, right, right,$$

and then produce the observation string w_2 given by

$$w_2 = 11878.$$

How can we make sure the observation string w_2 which is not a word (it does not follow a straight path) of our piece of scenery, does not get selected by the first phase of our algorithm as a short word? To see how w_2 gets eliminated in the first phase of our algorithm, consider the following dancing step:

$$right, down, right, up, right, right.$$

When doing this nearest neighbor path, a particle would produce the observations $\bar{w}_2$ where

$$\bar{w}_2 = 13578.$$

Let B_5^n be the event that up to time n^4 every nearest neighbor path of length $3(\ln n)^2$ in $\mathcal{K}(n^2)$ gets followed at least once. Then, assuming our piece of scenery (2.17) is contained in $\mathcal{K}(n^2)$, we would have that: Up to time n^4, we will observe both strings

$$w_1 w_2 w_3 \quad \text{and} \quad w_1 \bar{w}_2 w_3,$$

at least once. Since $w_2 \neq \bar{w}_2$, the second short-word-selection criteria of the first phase assures that the words w_2 and $\bar{w}_2$ do not get selected in the first phase of Λ^n. The crucial thing here was that from the point $(5, 2)$ to $(8, 2)$ there were two different 6 step nearest neighbor paths generating different observations, w_2 and $\bar{w}_2$. In Section 3.1, we will show that w.h.p. in the box $\mathcal{K}(n^2)$ for any pair of points x and y so that a nearest neighbor walk goes in $(\ln n)^2 - 1$ steps from x to y, either one of the two following things hold:

1. The segment $x\bar{y}$ is parallel to a direction of a coordinate and the distance $|x - y| = (\ln n)^2 - 1$, or
2. there exist two different nearest neighbor walk paths going from x to y in $(\ln n)^2 - 1$ steps and generating to different observation-strings.

So, this implies that w.h.p., the first phase of our algorithm can eliminate all the strings as it does in the example with w_2 and $\bar{w}_2$ which are not words of $\xi_{\mathcal{K}(n^2)}$.

2.2.5. *Why we need a seed.* The second phase produces a bag of long words which w.h.p. contains all long words of $\xi_{\mathcal{K}(4n)}$. Unfortunately, this bag is likely to contain also some long words of $\xi_{\mathcal{K}(n^2)}$ which are "not close to the origin."

Note that the size of the long words is $4n$, so if such a long word appears close to the border of $\mathcal{K}(n^2)$, then it would not serve to our reconstruction purpose. The reason is that the algorithm Λ^n aims to reconstruct the scenery in a box close to the origin. That is why in the third phase of Λ^n we apply the first two phases but with the parameter n being replaced by $n^{0.25}$. We then take any long word w_0 from the bag of words produced by the second phase of the algorithm $\Lambda^{n^{0.25}}$. That long word w_0 of size $4n^{0.25}$ is then w.h.p. contained in $\xi_{\mathcal{K}((n^{0.25})^2)} = \xi_{\mathcal{K}(n^{0.5})}$.

In other words, the long word w_0 chosen as seed in the third phase of the algorithm is w.h.p. not further away than $\sqrt{n}$ from the origin. Since it is likely that any word of that size appears only once in $\mathcal{K}(n^2)$, then we can use this to determine one long word w_L from the bag created by Λ^n which is not further from the origin than $\sqrt{n}$. We simply chose w_L to be any word from the bag of long words created by Λ^n which contains w_0. Finally, in the fourth phase of the algorithms we will then add neighboring long words to that first chosen long word. If the one long word which gets chosen in the third phase is close to the origin, we can determine which long words are neighbor on each other, then this will ensure that the other long words used for the reconstruction in the fourth phase are also close to the origin.

2.2.6. *Finding which long words are neighbors in the* 4*th phase of the algorithm* Λ^n. The fourth phase of the algorithm is then concerned with finding the relative position of the longer reconstructed words to each other. More precisely it tries to determine which long words are neighbors of each other. For the exact definition of neighboring long words, check out Section 3.4. Let us explain it through another example.

EXAMPLE 2.6. Consider the piece of scenery $\xi_{[0,16]\times[0,4]}$ given in (2.17) and let us designate by v_a the green word, by v_c the brown word and by v_b the word between v_a and v_c, that is,

$$v_a = \xi_{(1,2)}\xi_{(2,2)}\xi_{(3,2)}\xi_{(4,2)}\xi_{(5,2)} = 43912,$$
$$v_c = \xi_{(9,2)}\xi_{(10,2)}\xi_{(11,2)}\xi_{(12,2)}\xi_{(13,2)} = 47617,$$

and

$$v_b = \xi_{(6,2)}\xi_{(7,2)}\xi_{(8,2)} = 178.$$

Finally, let w_b designate the word "one line higher" than v_b, so

$$w_b := \xi_{(5,3)}\xi_{(6,3)}\xi_{(7,3)}\xi_{(8,3)}\xi_{(9,3)} = 11825.$$

Note that w_b has two digits more than v_b, and the middle letter of w_b has the same x-coordinate than the last letter of v_b in (2.17). Furthermore, in the piece of scenery (2.17), we designate the third line by v so that

$$v := \xi_{(0,2)}\xi_{(1,2)}\xi_{(2,2)}\cdots\xi_{(16,2)} = 74391217847617774,$$

and by w the long word written one line above, that is,

$$w := \xi_{(0,3)}\xi_{(1,3)}\xi_{(2,3)}\cdots\xi_{(16,3)} = 75076118258674042.$$

Assume that the two words v and w have already been reconstructed by the two first phases of our algorithm Λ^n. Consider next the straight path R_1:

$$(1,2),(2,2),(3,2),(4,2),(5,2),(6,2),(7,2),$$
$$(8,2),(9,2),(10,2),(11,2),(12,2),(13,2),$$

so a particle following this straight path generates the observations $\xi \circ R_1 = v_a v_b v_c$.

Consider now a second path R_2 that is similar to R_1 but at the end of the word v_a, it goes one step up to read w_b and then, at the end of w_b it goes one step down to read v_c. So, the path R_2 is defined as follows:

$$(1,2),(2,2),(3,2),(4,2),(5,2),(5,3),(6,3),(7,3),$$
$$(8,3),(9,3),(9,2),(10,2),(11,2),(12,2),(13,2),$$

and generates the observations $\xi \circ R_2 = v_a w_b v_c$.

If now up to time n^4 there is at least one particle following the path R_1 and another particle following R_2, then in the observations before time n^4, we will observe once $v_a v_b v_c$ and also $v_a w_b v_c$. So, v_a, v_b, v_c and w_b pass all the criteria in the fourth phase of our algorithm together with the long words v and w. So, v and w are detected (correctly) as being neighboring long words. Again, for this to happen we only need particles to go on the path R_1 and R_2.

So, what the previous example tell us is that to recognize correctly which words in $LONGWORDS^n$ are neighbors in the fourth phase of our algorithm, we need first to guarantee that all nearest neighbor paths of length $3(\ln n)^2$ within the box $\mathcal{K}(n^2)$, being followed by at least one particle up to time n^4. The event B_3^n guaranties this and we prove it holds w.h.p. in the first subsection of the next section.

On the other side, we still need the fourth phase of Λ^n not to classify pairs of long words as neighbors if they are not in the scenery. This problem of "false positives" is solved as soon as the previously defined diamond property holds, that is when the event B_3^n holds and also short words do not appear in different places in $\xi_{\mathcal{K}(n^2)}$. The proof of this is a little intricate and is given as proof of Lemma 3.10 in the next section.

2.2.7. *Placing neighboring long words next to each other in the* 4*th phase of our algorithm.* Here, we show how the long words are located next to each other in the fourth phase, in order to reconstruct a piece of the scenery ξ.

EXAMPLE 2.7. Let us assume that the scenery is 2-dimensional and that the seed word is $w_0 = 012$. To start, we place w_0 at the origin, and it would be the first part of the "reconstruction done by the algorithm Λ^n," thus w_0 is the restriction of $\Lambda^n(\chi_{n^4})$ to the set $\{(-1,0),(0,0),(1,0)\}$.

Say then that we find a long word w_L which is 60123. This long word contains the seed word 012 centered in its middle; so, if we superpose w_L over w_0 so that it matches we get again 60123, that corresponds to the restriction of $\Lambda^n(\chi_{n^4})$ to the set $\{(-2,0),(-1,0),(0,0),(1,0),(2,0)\}$.

Next, we find that the two long words 01111 and 02222 of $LONGWORDS^n$ are neighbors of w_L. In that case, we place these two long words in a neighboring position to get the following piece of scenery:

$$\begin{matrix} 01111 \\ 60123, \\ 02222 \end{matrix}$$

and assume that in our bag $LONGWORDS^n$ of long words we find the word 03333 to be a neighbor of 01111 and 04444 to be neighbor of 02222. Then our reconstruction yields the following piece of scenery:

$$\begin{matrix} 03333 \\ 01111 \\ 60123. \\ 02222 \\ 04444 \end{matrix} \tag{2.18}$$

This is then the "end product" produced by the algorithm Λ^n. So, this final output of the algorithm is a piece of scenery on a box of 5×5 given in (2.18). We assumed the size of the long words to be 5.

2.2.8. *Overview over the events which make the algorithm* Λ^n *work.* A handful of events ensure that the algorithm Λ^n works as already mentioned. These events will all be defined again in a more exhaustive way in the next section. But, here let us list them in a somewhat informal way. The *first phase* works through the following events:

- The event

$$B_2^n$$

guaranties that up to time n^2 every nearest neighbor path of length $3(\ln n)^2$ contained in $\mathcal{K}(4n)$ gets followed at least once by at least one particle.

- The event

$$B_3^n$$

asserts that up to time n^4 for every word w of $\xi_{\mathcal{K}(n^2)}$, any nearest neighbor walk path R in $\mathcal{K}(n^4)$ generating w must start and end in the diamond associated with w. This event is needed to guaranty that all the words of $\xi_{\mathcal{K}(4n)}$ of length $(\ln n)^2$ get selected in the first phase of the algorithm, and put into the bag of words $SHORTWORDS^n$.

- The event

$$B_4^n$$

states that for any two points x and y in $\mathcal{K}(n^2)$ so that there is a nearest neighbor path going in $(\ln n)^2 - 1$ steps from x to y, either one of the following holds:
  - The points x and y are at distance $(\ln n)^2 - 1$ from each other and the segment $x\bar{y}$ is parallel to a direction of coordinate. In that case, there can only be one nearest neighbor walk in $(\ln n)^2 - 1$ steps from x to y, and that nearest neighbor walk goes "straight."
  - There exist two different nearest neighbor walk paths going from x to y in exactly $(\ln n)^2 - 1$ steps and generating different observations.

- The event

$$B_5^n$$

guaranties that up to time n^4 every nearest neighbor path of length $3(\ln n)^2 - 1$ contained in $\mathcal{K}(n^2)$ gets followed at least once. Furthermore, together with the event B_4^n, these make sure that observations which are not words of $\xi_{\mathcal{K}(n^2)}$ get eliminated in the first phase of Λ^n.

In Lemma 3.6, the combinatorics of the *first phase* is proven. It is shown that when all the events $B_2^n, B_3^n, B_4^n, B_5^n$ hold, then the first phase works. In the first subsection of the next section it is also proven that all these events hold w.h.p., which implies that the first phase works w.h.p.

For the *second phase* of the algorithm, we only need that any word of $\xi_{\mathcal{K}(n^2)}$ of size $(\ln n)^2 - 1$ appears in only one place in $\mathcal{K}(n^2)$. That is given by the event C_1^n and needed to assemble short words into longer words.

The *third phase* of the algorithm is just using the first two phases of the algorithm but with a parameter different from n. Instead the parameter is $n^{0.25}$. So, we do not need any other special event to make this phase work. We only need what we have proven for the first two phases of the algorithm.

Finally, the *fourth phase* of the algorithm needs the diamond property to hold for the short words in $\xi_{\mathcal{K}(n^2)}$, that is, the event B_3^n to hold. Furthermore, the event C_1^n which guaranties that short words cannot appear in two different places of $\xi_{\mathcal{K}(n^2)}$ is also needed. These events are defined already for the first two phases of the algorithm.

The next section gives all the detailed definitions of these events, the proofs for their high probability and also the rigorous proofs that these events make the different phases of the algorithm work. Although most of the main ideas are already given in the present section, and it might seem a little redundant, we feel that presenting them once informally but with all the details in a rigorous manner, will be very useful to understand better the algorithm.

3. Proof of Theorem 2.2. In what follows, we will we say that the Branching random walk BRW visits $z \in \mathbb{Z}^d$ at time t if $\eta_t(z) \geq 1$.

3.1. *First phase.* Here, we will construct the set of *SHORTWORDS*^{n}. Recall that a string w_2 of size $(\ln n)^2$ is in *SHORTWORDS*^{n} if there exist two sequences w_1 and w_3 both of size $(\ln n)^2$ and such that:

1. $w_1 w_2 w_3$ appears in the observations before time n^2.
2. The only string w of size $(\ln n)^2$ such that $w_1 w w_3$ appears in the observations up to time n^4 is w_2.

Let $W(\xi_{\mathcal{K}(4n)})$ and $W(\xi_{\mathcal{K}(n^2)})$ be the sets of all words of size $(\ln n)^2$ in $\xi_{\mathcal{K}(4n)}$ and $\xi_{\mathcal{K}(n^2)}$ respectively, then we are going to show that with high probability the set *SHORTWORDS*^{n} satisfies that

$$W(\xi_{\mathcal{K}(4n)}) \subseteq \textit{SHORTWORDS}^n \subseteq W(\xi_{\mathcal{K}(n^2)}).$$

We need the following results.

LEMMA 3.1. *Let B_1^n be the event that up to time n^2 all the sites in $\mathcal{K}(4n)$ are visited by the BRW more than* $\exp(cn)$ *times, where c is a constant independent of n, then there exists $C > 0$ such that*

$$P(B_1^n) \geq 1 - e^{-Cn^2}.$$

PROOF. We only sketch the proof: it is elementary to obtain that by time $n^2/2$ the process will contain at least $e^{\delta n^2}$ particles (where δ is small enough), with probability at least $1 - e^{-Cn^2}$. Then consider any fixed $x \in \mathcal{K}$; for each of those particles, at least one offspring will visit x with probability at least $cn^{-(d-2)}$, and this implies the claim of Lemma 3.1. □

LEMMA 3.2. *Let B_2^n be the event that up to time n^2 for every nearest neighbor path of length $3(\ln n)^2 - 1$ contained in $\mathcal{K}(4n)$; there is at least one particle which follows that path. Then, for all n large enough we have*

$$P(B_2^n) \geq 1 - \exp[-c_1 n],$$

where $c_1 > 0$ is constant independent of n.

PROOF. Let R_j^z be the jth nearest neighbor path of length $3(\ln n)^2 - 1$ in $\mathcal{K}(4n)$ starting at $z \in \mathcal{K}(4n)$, $j = 1, \ldots, (2d)^{3(\ln n)^2-1}$. By Lemma 3.1, we know that with high probability up to time n^2 all the sites in $\mathcal{K}(4n)$ are visited by the BRW more than $\exp(cn)$ times, where c is a constant independent of n. Suppose we have been observing the BRW up to time n^2, then define for any $z \in \mathcal{K}(4n)$ the following variables:

$$Y_{ij} = \begin{cases} 1, & \text{if after the } i\text{th visit to } z, \\ & \text{the corresponding particle follows } R_j^z, \\ 0, & \text{otherwise,} \end{cases} \tag{3.1}$$

where $i = 1, \ldots, \exp(cn)$ and $j = 1, \ldots, (2d)^{3(\ln n)^2-1}$. Note that the variables Y_{ij}'s are independent because all the particles are moving independently between themselves. We are interested on the event

$$\left\{ \bigcap_{j=1}^{(2d)^{3(\ln n)^2-1}} \bigcup_{i=1}^{e^{cn}} \{Y_{ij} = 1\} \right\}, \tag{3.2}$$

that is, for every path of length $3(\ln n)^2 - 1$ starting at $z \in \mathcal{K}(4n)$, up to time n^2, there is at least one visit to z, such that the corresponding particle on z follows it.

Let $Z_j = \sum_i Y_{ij}$, thus Z_j counts the number of times R_j^z is followed, and it is binomially distributed with expectation and variance given by

$$E[Z_j] = \exp(cn)\left(\frac{1}{2d}\right)^{3(\ln n)^2-1} \quad \text{and}$$

$$V[Z_j] = \exp(cn)\left(\frac{1}{2d}\right)^{3(\ln n)^2-1}\left(1 - \left(\frac{1}{2d}\right)^{3(\ln n)^2-1}\right).$$

Observe that (3.2) is equivalent to $\{\bigcap_{j=1}^{(2d)^{3(\ln n)^2-1}} \{Z_j \geq 1\}\}$, then by Chebyshev's inequality we have

$$P(Z_j \leq 0) \leq \frac{V[Z_j]}{E^2[Z_j]} < \exp\big[\big(3(\ln n)^2 - 1\big)\ln 2d - cn\big],$$

and

$$P\left(\bigcup_{j=1}^{(2d)^{3(\ln n)^2-1}} \{Z_j \leq 0\}\right) < \exp\big[\big(6(\ln n)^2 - 2\big)\ln 2d - cn\big]. \tag{3.3}$$

Since the number of sites in $\mathcal{K}(4n)$ is $(8n+1)^d$, by (3.3) it follows that

$$P\big(B_2^{nc}\big) < (8n+1)^d \exp\big[\big(6(\ln n)^2 - 2\big)\ln 2d - cn\big]. \tag{3.4}$$

Now note that for any constant $0 < c_1 < c$, the right-hand side of (3.4) is less than $\exp(-c_1 n)$, for n large enough. Thus, $P(B_2^{nc}) \to 0$ as $n \to \infty$. $\square$

In what follows, we will denote by T_w the diamond associated with a word w appearing in a certain place in the scenery. For the definition of diamond associated with a word, see (2.2.3).

LEMMA 3.3. *Let B_3^n be the event that for any word w of size $(\ln n)^2$ contained in $\xi_{\mathcal{K}(n^2)}$, every nearest neighbor walk path R generating w on $\xi_{\mathcal{K}(n^4)}$ must start and end in the diamond associated to w, that is, $R(0) \in T_w$ and $R((\ln n)^2 - 1) \in T_w$. Then*

$$P(B_3^n) \geq 1 - 2^{2d+1} \exp\bigl(8d \ln(n+1) + (\ln n)^2 \ln(2d/\kappa)\bigr),$$

which goes to 1 as $n \to \infty$ because we have assumed $\kappa > 2d$.

PROOF. Take without loss of generality a sequence read in a straight way from left to right, that is,

$$w = \xi(x)\xi(x + \vec{e_1})\xi(x + 2\vec{e_1}) \cdots \xi\bigl(x + \bigl((\ln n)^2 - 1\bigr)\vec{e_1}\bigr),$$

and let R^z be a nearest neighbor walk (nonrandom) of length $(\ln n)^2 - 1$ starting at z with $z \in \mathcal{K}(n^4) \setminus T_w$. Since a nearest neighbor path at each unit of time only make steps of length one, then it follows that w_i is independent of $\xi(R(0)), \xi(R(1)), \ldots, \xi(R(i-1))$, as well as of $w_1, \ldots, w_{i-1}$. Hence,

$$P\bigl[w = \xi\bigl(R(0)\bigr)\xi\bigl(R(1)\bigr) \cdots \xi\bigl(R\bigl((\ln n)^2 - 1\bigr)\bigr)\bigr] = (1/\kappa)^{(\ln n)^2} \tag{3.5}$$

(recall that κ is the number of colors).

Let P_z^n be the set of all nearest neighbor paths of length $(\ln n)^2 - 1$ starting at z, and note that P_z^n contains no more than $(2d)^{(\ln n)^2 - 1}$ elements. For a fix $z \in \mathcal{K}(n^4) \setminus T_w$, let $B_{3,z}^n$ be the event that there is no nearest neighbor path R^z of length $(\ln n)^2 - 1$ generating w. By (3.5), it follows that

$$\begin{aligned} P(B_{3,z}^{nc}) &= P\bigl[\exists R^z; \xi\bigl(R^z\bigr) = w\bigr] \\ &\leq \sum_{R^z \in P_z^n} P\bigl(\xi\bigl(R^z\bigr) = w\bigr) \\ &\leq \frac{(2d)^{(\ln n)^2 - 1}}{\kappa^{(\ln n)^2}} \\ &\leq \left(\frac{2d}{\kappa}\right)^{(\ln n)^2}. \end{aligned}$$

Now for any $z \in \mathcal{K}(n^4) \setminus T_w$, let B_{3S}^n be the event that there is no nearest neighbor path R^z of length $(\ln n)^2 - 1$ generating w. Hence,

$$P(B_{3S}^{nc}) = P\Bigg(\bigcup_{z\in\mathcal{K}(n^4)\setminus T_w} B_{3,z}^{nc}\Bigg) \le \sum_{z\in\mathcal{K}(4n)\setminus T_w} P(B_{3,z}^{nc})$$

$$\le (2n^4+1)^{2d}\left(\frac{2d}{\kappa}\right)^{(\ln n)^2}.$$

Now by symmetry $P(B_3^{nc}) \le 2P(B_{3S}^{nc})$, so that

$$P(B_3^{nc}) \le 2(2n^4+1)^{2d}\left(\frac{2d}{\kappa}\right)^{(\ln n)^2}$$

$$\le 2\big(2(n+1)^4\big)^{2d}\left(\frac{2d}{\kappa}\right)^{(\ln n)^2}$$

$$= \big(8d\ln(n+1) + (\ln n)^2\ln(2d/\kappa)\big). \qquad \square$$

LEMMA 3.4. *Let B_4^n be the event that for every two points $x, y \in \mathcal{K}(n^2)$ and such that there is a nearest neighbor path R of length $(\ln n)^2 - 1$ going from x to y, one of the following alternatives holds*:

(a) *x and y are at distance $(\ln n)^2 - 1$ and on the same line, that means, along the same direction of a coordinate, or*

(b) *there exist two different nearest neighbor paths R_1 and R_2 of length $(\ln n)^2 - 1$ both going from x to y but generating different observations, that is, $\xi(R_1) \ne \xi(R_2)$.*

Then we have

$$P(B_n^4) \ge 1 - \exp\big[-0.5(\ln n)^2\ln\kappa\big].$$

PROOF. Let x and y be two points in $\mathcal{K}(n^2)$ and such that there is a nearest neighbor path R of length $(\ln n)^2 - 1$ going from x to y, so the distance between x and y, $d(x, y) \le (\ln n)^2 - 1$.

Suppose that $d(x, y) = (\ln n)^2 - 1$, which is defined as the length of the shortest path between x and y. If x and y are not on the same line, then there exist two paths R_1 and R_2 of length $(\ln n)^2 - 1$ both going from x to y and not intersecting anywhere, except in x and y. This means that $R_1(0) = R_2(0) = x$ and $R_1((\ln n)^2 - 1) = R_2((\ln n)^2 - 1) = y$, but for all j_1, j_2 strictly between 0 and $(\ln n)^2 - 1$, we find $R_1(j_1) \ne R_2(j_2)$. Since the scenery ξ is i.i.d., it thus follows that

$$P\big[\xi(R_1(0)) = \xi(R_2(0)), \ldots, \xi(R_1((\ln n)^2 - 1)) = \xi(R_2((\ln n)^2 - 1))\big]$$
$$= \left(\frac{1}{\kappa}\right)^{(\ln n)^2 - 2}. \tag{3.6}$$

Now suppose that $d(x, y) < (\ln n)^2 - 1$. Let R_1 be a path which makes a cycle from x to x and then going in shortest time from x to y, and R_2 be a path which follows first a shortest path between x and $y-1$, next makes a cycle from $y-1$ to $y-1$ and then go to y. If neither the cycle from x to x intersects the shortest path which makes part of R_2 nor the cycle from $y-1$ to $y-1$ intersects the shortest path which makes part of R_1, then we have that for $i = 0, \ldots, (\ln n)^2 - 1$, the positions taken by R_1 and R_2 are different, that is, $R_1(1) \neq R_2(1), \ldots, R_1((\ln n)^2 - 2) \neq R_2((\ln n)^2 - 2)$. Hence, $\xi(R_1(i))$ is independent of $\xi(R_2(i))$ for $i = 1, 2, \ldots, (\ln n)^2 - 2$ and (3.6) holds again.

Let B^n_{4xy} be the event that there exist two nearest neighbor paths S and T going from x to y with $d(x, y) \leq (\ln n)^2 - 1$, but generating different observations, and let $B^n_4 = \bigcap_{x,y} B^n_{4xy}$, where the intersection is taken over all $x, y \in \mathcal{K}(n^2)$ such that $d(x, y) \leq (\ln n)^2 - 1$. By (3.6), it follows that

$$P\big(B^{nc}_{4xy}\big) \leq \exp\big[-\big((\ln n)^2 - 2\big)\ln\kappa\big], \tag{3.7}$$

then

$$\begin{aligned} P\big(B^{nc}_4\big) &\leq \sum_{x,y} P\big(B^{nc}_{4xy}\big) \\ &< \big[2(\ln n)^2 - 3\big]^d \big(2n^2 + 1\big)^d \exp\big[-\big((\ln n)^2 - 2\big)\ln\kappa\big] \qquad (3.8) \\ &= \exp\big[d\ln\big(2(\ln n)^2 - 3\big) + d\ln\big(2n^2 + 1\big) - \big((\ln n)^2 - 2\big)\ln\kappa\big], \end{aligned}$$

and observe that the right-hand side of (3.8) is for all n large enough less than $\exp(-0.5(\ln n)^2 \ln\kappa)$. This completes this proof. □

LEMMA 3.5. *Let B^n_5 be the event that up to time n^4 every path of length $3(\ln n)^2 - 1$ in $\mathcal{K}(n^2)$ gets followed at least once by a particle. Then*

$$P\big(B^n_5\big) \geq 1 - \exp\big[-c_2 n^2\big],$$

where $c_2 > 0$ is a constant not depending on n.

PROOF. Note that the event B^m_2 from Lemma 3.2, where we take $m = n^2$ gives that up to time n^4, every path in $\mathcal{K}(4n^2)$ of length $12\ln n - 1$ gets followed by at least one particle. So, this implies that B^n_5 holds, and hence

$$B^{n^2}_2 \subset B^n_5.$$

The last inclusion above implies

$$P\big(B^n_5\big) \geq P\big(B^{n^2}_2\big). \tag{3.9}$$

We can now use the bound from Lemma 3.2 for bounding the probability of $P(B^{n^2}_2)$. Together with (3.9), this yields

$$P\big(B^n_5\big) \geq 1 - \exp\big[d\ln\big(8n^2 + 1\big) + \big(6\big(\ln n^2\big)^2 - 2\big)\ln 2d - cn^2\big]. \tag{3.10}$$

Now note that for any constant $c_2 > 0$ for which $c > c_2$, we have that: for all n large enough we have that the bound on the right-hand side of inequality (3.10) is less than $\exp(-c_2 n^2)$ which completes this proof. □

LEMMA 3.6 (The first phase works). *Let B^n designate the event that every word of size $(\ln n)^2$ in $\xi_{\mathcal{K}(4n)}$ is contained in the set SHORTWORDS^n, and also all the strings in SHORTWORDS^n belong to $W(\xi_{\mathcal{K}(n^2)})$, then*

$$B_1^n \cap B_2^n \cap B_3^n \cap B_4^n \cap B_5^n \subset B^n.$$

PROOF. We start by proving that every word in $W(\xi_{\mathcal{K}(4n)})$ of size $(\ln n)^2$, say w_2, is contained in *SHORTWORDS*^{n}. Then there exist two integer points $x, y \in \mathcal{K}(4n)$ on the same line, that is, along the same direction of a coordinate, and at a distance $3(\ln n)^2 - 1$ such that "w_2 appears in the middle of the segment $x\bar{y}$." By this we mean that there exists a canonical vector $\vec{e}$ (such a vector consists of only one nonzero entry which is 1 or -1), so that

$$w_1 w_2 w_3 = \xi_x \xi_{x+\vec{e}} \xi_{x+2\vec{e}} \cdots \xi_y,$$

where w_1 and w_2 are words of size $(\ln n)^2$ and $w_1 w_2 w_3$ designates the concatenation of w_1, w_2 and w_3. By the event B_2^n, up to time n^2 there is at least one particle which will go from x to y in exactly $3(\ln n)^2 - 1$ steps. That is, up to time n^2 there is a particle which follows the "straight path from x to y" given by

$$x, x + \vec{e}, x + 2\vec{e}, \ldots, y.$$

When doing so, we will see in the observations the string $w_1 w_2 w_3$. Thus, the triple (w_1, w_2, w_3) satisfies the first criteria of the first phase of Λ^n. It needs also to pass the second criteria to be selected.

To see that (w_1, w_2, w_3) satisfies second the criteria, let us assume that w is a word of size $(\ln n)^2$ so that the concatenation $w_1 w w_2$ appears in the observation before time n^4. Then there exists a nearest neighbor walk path R of length $3(\ln n)^2 - 1$ generating $w_1 w w_2$ on $\xi_{\mathcal{K}(n^4)}$. By this, we mean that $im R \subset \mathcal{K}(n^4)$ and $\xi \circ R = w_1 w w_2$. By the event B_3^n, we have that $R((\ln n)^2 - 1)$ is in the diamond T_{w_1} associated with w_1 and $R(2(\ln n)^2)$ is in the diamond T_{w_3} associated with w_3. So, when we take the restriction of R to the time interval $[(\ln n)^2 - 1, 2(\ln n)^2]$ we get a nearest neighbor walk going in $(\ln n)^2$ steps from T_{w_1} to T_{w_3}. The only way to do this is to go in a straight way from the point $x + ((\ln n)^2 - 1)\vec{e}$ to $x + (2(\ln n)^2)\vec{e}$. [The reason being that the distance between T_{w_1} and T_{w_3} is $(\ln n)^2$ and the only pairs of points (x', y') so that $x' \in T_{w_1}$ and $y' \in T_{w_3}$ and located at that distance $(\ln n)^2$ from each other, are $x' = x + ((\ln n)^2 - 1)\vec{e}$ and $y' = x + (2(\ln n)^2)\vec{e}$.] So, during the time interval $[(\ln n)^2 - 1, 2(\ln n)^2]$ we have that R is walking in a straight way on the middle part of the segment $x\bar{y}$, that is walking in a straight way from x' to y'. Hence, during that time R is generating in the observation the

word w_2. This proves that $w = w_2$. Hence, the triple (w_1, w_2, w_3) also passes the second criteria of the first phase of our algorithm, which implies that

$$w_2 \in SHORTWORDS^n,$$

and hence

$$W(\xi_{\mathcal{K}(4n)}) \subset SHORTWORDS^n.$$

It remains to show that if the triple (w_1, w_2, w_3) gets selected through the first phase of our algorithm (hence passes the two selection criteria given there), then indeed w_2 is a word of $\xi_{\mathcal{K}(n^2)}$. Now, to pass the first selection criteria, we have that the concatenation $w_1 w_2 w_3$ must appear before time n^2 in the observations. Since the first particle starts at time 0 in the origin, by time n^2 all the particles must be still contained in the box $\mathcal{K}(n^2)$. Hence, there must exist a nearest neighbor path R of length $3(\ln n)^2 - 1$ which generates $w_1 w_2 w_3$ on $\xi_{\mathcal{K}(n^2)}$. Hence, $im R \subset \mathcal{K}(n^2)$ and $w_1 w_2 w_3 = \xi \circ R$.

Assume that the restriction of R to the time interval $[(\ln n)^2 - 1, 2(\ln n)^2]$ would not be a "straight walk" on a line. Then, by the event B_4^n there would exist a modified nearest neighbor walk R' of length $3(\ln n)^2 - 1$ for which the following two conditions are satisfied:

1. Restricted to the time interval $[(\ln n)^2 - 1, 2(\ln n)^2]$, we have that R' generates a string w different from w_2 on ξ.
2. Outside that time interval, R' generates the same observations w_1 and w_3 as R.

Summarizing: the nearest neighbor walk R' generates $w_1 w w_3$ on $\xi_{\mathcal{K}(n^2)}$, where $w \neq w_2$. But by the event B_5^n, every nearest neighbor walk of length $3(\ln n)^3 - 1$ in $\mathcal{K}(n^2)$ gets followed at least once by a particle up to time n^4. Hence, at least one particle follows the path of R' before time n^4. Doing so it produces the string $w_1 w w_3$ with $w \neq w_2$ before time n^4 in the observations. This implies, however, that the triple (w_1, w_2, w_3) fails the second selection criteria for phase one of our algorithm. This is a contradiction, since we assumed that (w_1, w_2, w_3) gets selected through the first phase of Λ^n (and hence passes the two selection criteria given there). This proves by contradiction that R restricted to the time interval $[(\ln n)^2 - 1, 2(\ln n)^2]$ can only be a "straight walk." Hence, the sequence generated during that time, that is, w_2 can only be a word of the scenery ξ in $\mathcal{K}(n^2)$. This proves that w_2 is in $W(\xi_{\mathcal{K}(n^2)})$ and then

$$SHORTWORDS^n \subset W(\xi_{\mathcal{K}(n^2)}). \qquad \square$$

3.2. *Second phase.* In this phase, the words of $SHORTWORDS^n$ are assembled into longer words to construct the set of $LONGWORDS^n$ using the assembling rule: To puzzle two words together of $SHORTWORDS^n$, the words must coincide on at least $(\ln n)^2 - 1$ consecutive letters. To get a correct assembling, we will need that the short words could be placed together in a unique way.

LEMMA 3.7. *Let C_1^n be the event that, for all $x, y \in \mathcal{K}(n^2)$, the words $\xi_x \xi_{x+\vec{e}_i} \cdots \xi_{x+((\ln n)^2-1)\vec{e}_i}$ and $\xi_y \xi_{y+\vec{e}_j} \cdots \xi_{y+((\ln n)^2-1)\vec{e}_j}$ are identical only in the case $x = y$ and $\vec{e}_i = \vec{e}_j$. Then*

$$P(C_1^n) \geq 1 - \exp\bigl[2d \ln(2n^2+1) - ((\ln n)^2 - 1)\ln\kappa\bigr].$$

PROOF. Let x and y be two points in $\mathcal{K}(n^2)$ and define the event

$$C_{x,y} = \{\xi_x \xi_{x+\vec{e}_i} \xi_{x+2\vec{e}_i} \cdots \xi_{x+((\ln n)^2-1)\vec{e}_i} = \xi_y \xi_{y+\vec{e}_j} \xi_{y+2\vec{e}_j} \cdots \xi_{y+((\ln n)^2-1)\vec{e}_j}\},$$

with $\vec{e}_i$ and $\vec{e}_j$ two canonical vectors in $\mathbb{Z}^d$. Clearly,

$$C_1^n = \bigcap_{x,y} C_{x,y},$$

where the intersection above is taken over all $(x, y) \in \mathcal{K}(n^2)$, and it leads to

$$P(C_1^{nc}) \leq \sum_{x,y} P(C_{x,y}^c). \tag{3.11}$$

If $x = y$ and $\vec{e}_i \neq \vec{e}_j$, observe that the words intersect themselves only at the first position and since the scenery is i.i.d., then $P(C_{x,y}^c) = (\frac{1}{\kappa})^{(\ln n)^2-1}$. On the other hand, when $x \neq y$, it is somewhat similar to the previous case, that is, the words intersect themselves at most by only one position, thus $P(C_{x,y}^c) \leq (\frac{1}{\kappa})^{(\ln n)^2-1}$. Hence,

$$\begin{aligned} P(C_1^{nc}) &\leq (2n^2+1)^{2d} \left(\frac{1}{\kappa}\right)^{(\ln n)^2-1} \\ &< \exp\bigl[2d \ln(2n^2+1) - ((\ln n)^2 - 1)\ln\kappa\bigr]. \end{aligned} \tag{3.12}$$

□

LEMMA 3.8 (The second phase works). *Let C^n designate the event that every word of size $4n$ in $\xi_{\mathcal{K}(4n)}$ is contained in LONGWORDS^n, and all the words in LONGWORDS^n belong to $\mathcal{W}_{4n}(\xi_{\mathcal{K}(n^2)})$, that is,*

$$\mathcal{W}_{4n}(\xi_{\mathcal{K}(4n)}) \subseteq \mathit{LONGWORDS}^n \subseteq \mathcal{W}_{4n}(\xi_{\mathcal{K}(n^2)}),$$

where $\mathcal{W}_{4n}(\xi_{\mathcal{K}(4n)})$ and $\mathcal{W}_{4n}(\xi_{\mathcal{K}(n^2)})$ are the set of all words of size $4n$ in $\xi_{\mathcal{K}(4n)}$ and $\xi_{\mathcal{K}(n^2)}$, respectively, then

$$B^n \cap C_1^n \subset C^n.$$

PROOF. Let $W(\xi_{\mathcal{K}(4n)})$ and $W(\xi_{\mathcal{K}(n^2)})$ be the set of all words of size $(\ln n)2$ in $\xi_{\mathcal{K}(4n)}$ and $\xi_{\mathcal{K}(n^2)}$, respectively. Once the first phase has worked, that is, when B^n occurs we have

$$W(\xi_{\mathcal{K}(4n)}) \subseteq \mathit{SHORTWORDS}^n \subseteq W(\xi_{\mathcal{K}(n^2)}).$$

If the short words can be placed together in a unique way using some assembling rule, and it is done until getting strings of total exactly equal to $4n$, then every word of size $4n$ in $\xi_{\mathcal{K}(4n)}$ is contained in the set of assembled words, that is, in $LONGWORDS^n$. On the other hand, if the assembled process is made using all words in $SHORTWORDS^n$, then all the words in $LONGWORDS^n$ belong to $\mathcal{W}_{4n}(\xi_{\mathcal{K}(n^2)})$ because $SHORTWORDS^n \subseteq W(\xi_{\mathcal{K}(n^2)})$.

Under the assembling rule given in Lemma 3.7, we conclude that $B^n \cap C_1^n \subset C^n$. □

3.3. *Third phase*. In this phase, we use the previous two phases of Λ^n but with the parameter $n^{0.25}$ instead of n. The idea is to take one long word from $LONGWORDS^{n^{0.25}}$, say v, which will be of size $4n^{0.25}$ instead of $4n$, then, select any long word from $LONGWORDS^n$, say w, which contains v in its middle. In this manner, we should hopefully get a word which has its middle not further than $\sqrt{n}$ from the origin.

In the next lemma, we show that when the first two phases of our algorithm with parameter n as well as $n^{0.25}$ both work, then the third phase must work as well.

LEMMA 3.9 (The third phase works). *Let D^n be the event that the third phase of Λ^n works. This means that the long-word of $LONGWORD^n$ selected by the third phase has its center not further than $n^{0.5}$ from the origin. Thus,*

$$C_1^n \cap C^n \cap C^{n^{0.25}} \subset D^n.$$

PROOF. Consider any word from $LONGWORD^n$ that contains in its middle a word from $LONGWORDS^{n^{0.25}}$, that is, take

$$w = \xi_x \xi_{x+\vec{e}_i} \xi_{x+2\vec{e}_i} \cdots \xi_{x+4n-1\vec{e}_i}$$

in $LONGWORD^n$ such that

$$v = \xi_{x+2n\vec{e}_i}, \ldots, \xi_{x+(2n+4n^{0.25}-1)\vec{e}_i}$$

belongs to $LONGWORDS^{n^{0.25}}$.

By $C^{n^{0.25}}$, the first two phases of the algorithm with parameter $n^{0.25}$ work. This implies that v is a word (of size $4n^{0.25}$) contained in $\xi_{\mathcal{K}(n^{0.5})}$. It is not difficult to see that C_1^n implies that any word of that size which appears in $\xi_{\mathcal{K}(n^2)}$, appears in a "unique position" therein. By C^n, all the words of $LONGWORD^n$ are contained in $\xi_{\mathcal{K}(n^2)}$. Thus, when a word w of $LONGWORD^n$ contains a word v of $LONGWORD^{n^{0.25}}$, then the two words must lie [in $\mathcal{K}(n^2)$] on top of each other in a unique way, which implies that the middle of w is also the middle of v. By $C^{n^{0.25}}$, we have that the middle of v (the way v appears in $\xi_{\mathcal{K}(n^{0.5})}$) is not further from the origin than $n^{0.5}$. Hence, the middle of w (in where it appears in $\xi_{\mathcal{K}(n^2)}$) is also not further than $n^{0.5}$ from the origin. This completes our proof. □

3.4. *Fourth phase.* For the fourth and last phase to work correctly, we need to be able to identify which words contained in $\xi_{\mathcal{K}(n^2)}$ are neighbors of each other. Let us give the definition of neighboring words.

Let I be a box of $\mathbb{Z}^d$ and w and v be two words (of the same length) contained in ξ_I. We say that w and v are *neighbors of each other* if there exist $x \in I$ and $\vec{u}, \vec{s} \in \{\pm\vec{e}_i, i = 1, \dots, d\}$ such that

$$w = \xi_x \xi_{x+\vec{u}} \xi_{x+2\vec{u}} \cdots \xi_{k\vec{u}}$$

and

$$v = \xi_{x+\vec{s}} \xi_{x+\vec{u}+\vec{s}} \xi_{x+2\vec{u}+\vec{s}} \cdots \xi_{x+k\vec{u}+\vec{s}},$$

where $\vec{s}$ is orthogonal to $\vec{u}$ and all the points $x + \vec{s} + i\vec{u}$ and $x + i\vec{u}$ are in I for all $i = 0, 1, 2, \dots, k$.

In other words, two words w and v contained in the restriction ξ_I are called neighbors if we can read them in positions which are parallel to each other and at distance 1.

LEMMA 3.10 (The fourth phase works). *Let F^n denote the event that for the words of LONGWORDS^n the three conditions in the fourth phase of Λ^n allows to correctly identify and chose neighbors. Interestingly, the events B_2^n, B_3^n and C_1^n are enough for F^n to occur. That is,*

$$B_2^n \cap B_3^n \cap C_1^n \subset F^n.$$

PROOF. We need to show that when all the events B_2^n, B_3^n and C_1^n occur, then we identify correctly and chose the long words from *LONGWORDS*^{n} which are neighbors of each other.

If two words v and w belonging to *LONGWORDS*^{N} were selected by the fourth phase of our algorithm to be put on top of each other, this means that Λ^n "estimated" that v and w were neighbors, is because the following three conditions were satisfied:

1. There exist 4 words v_a, v_b, v_c and w_b having all size $(\ln n)^2$ except for v_b which has size $(\ln n)^2 - 2$ and such that, the concatenation $v_a v_b v_c$ is contained in v, whilst up to time n^4 we observe at least once $v_a w_b v_c$.
2. The word w_b is contained in w.
3. The position of the middle letter of v_b in v should be the same as the middle letter position of w_b in w.

Assume that the three previous conditions are satisfied, and let $\vec{u}$ be the direction of the word v, so, we have two points x and y in $\mathbb{Z}^d$ such that $y = x + ((\ln n)^2 - 1)\vec{u}$ and:

- to the left (with respect to the direction of $\vec{u}$) from x we read v_a:

$$v_a = \xi_{x-\vec{u}((\ln n)^2+1)} \xi_{x-\vec{u}((\ln n)^2+2)} \xi_{x-\vec{u}((\ln n)^2+3)} \cdots \xi_x,$$

- between x and y we read v_b:

$$v_b = \xi_{x+\vec{u}}\xi_{x+2\vec{u}} \cdots \xi_{y-\vec{u}}, \quad \text{and}$$

- to the right of y we read v_c:

$$v_c = \xi_y \xi_{y+\vec{u}}\xi_{y+2\vec{u}} \cdots \xi_{y+((\ln n)^2-1)\vec{u}}.$$

By the first condition, we know that up to time n^4 we observe at least once the concatenated word $v_a w_b v_c$. Hence, there exists a nearest neighbor walk R on the time interval $[1, 3(\ln n)^2]$ which generated $v_a w_b v_c$ on $\xi_{\mathcal{K}(n^4)}$, so that

$$\xi \circ R = v_a w_b v_c.$$

Note that after time $2(\ln n)^2$ R generates v_c, then by B_3^n we have that $R(2(\ln n)^2 + 1)$ must be in the diamond T_c associated with v_c. Similarly, we get that $R((\ln n)^2)$ must be in the diamond T_a associated with v_a.

Now observe that to go in $(\ln n)^2 + 1$ steps with a nearest neighbor walk from the diamond T_a to the diamond T_c, there are only 3 possibilities [remember that x and y are at distance $(\ln n)^2 - 1$ from each other]:

(I) Going from x to y always making steps with respect to $\vec{u}$ (in 2 dimensions, say to the right), and once making one step with respect to $-\vec{u}$ (one step to the left). No steps in the directions orthogonal to $\vec{u}$.

(II) Starting at x, make one step in some direction orthogonal to $\vec{u}$, say with respect to $\vec{s}$ (in 2 dimensions, say one step up) then all steps with respect to $\vec{u}$ (to the right), and once making one step with respect to $-\vec{s}$ (one step down) in order to reach y.

(III) Starting at $x + \vec{s} - \vec{u}$ instead of x and arriving in $y + \vec{s} + \vec{u}$ instead of y.

Since the nearest neighbor walk R between time $(\ln n)^2$ and $2(\ln n)^2 + 1$ must be walking from the diamond T_a to the diamond T_c in $(\ln n)^2 + 1$ steps, then it must satisfy during that time one of the three conditions above.

If condition II holds, then w_b is the word which is "written" in the scenery ξ between $x + \vec{s}$ and $y + \vec{s}$, that is,

$$w_b = \xi_{x+\vec{s}}\xi_{x+\vec{s}+\vec{u}}\xi_{x+\vec{s}+2\vec{u}} \cdots \xi_{y+\vec{s}}.$$

That shows that the line between the two points $x + \vec{s}$ and $y + \vec{s}$ is where the word w_b is written in the scenery. Now observe that w appears in $\xi_{\mathcal{K}(n^2)}$ because w is in $LONGWORDS^N$, and by the second condition w must contain the word w_b, so w must contain the points $x + \vec{s}$ and $y + \vec{s}$. By the event C_1^n, any word of size $(\ln n)^2$ appears only in one place in $\xi_{\mathcal{K}(n^2)}$, then as w_b has size $(\ln n)^2$, the place where the word w is written in $\xi_{\mathcal{K}(n^2)}$ is a line parallel to the line $x\bar{y}$ and at distance 1. This means that the word w is a neighbor of the word v in $\xi_{\mathcal{K}(n^2)}$.

When condition (III) holds, a very similar argument leads to the same conclusion that w and v are neighbors in $\xi_{\mathcal{K}(n^2)}$.

Finally, when condition (I) holds, then we would have that $v_a w_b v_c$ appears in $\xi_{\mathcal{K}(n^2)}$, but we have also that $v_a v_b v_c$ appears in $\xi_{\mathcal{K}(n^2)}$ (since we take the words w and v from the set *LONGWORDS*^{n} and since by the event C_1^n these are words from the scenery $\xi_{\mathcal{K}(n^2)}$). However, the word w_b is longer than v_b. This implies that either v_a or v_c (or both together) must appear in two different places in $\xi_{\mathcal{K}(n^2)}$. But this would contradict the event C_1^n. Hence, we can exclude this case.

Thus, we have proven that if the events B_3^n and C_1^n all hold, then the words selected to be put next to each other in phase 4 of the algorithm are really neighbors in the way they appear in the restricted scenery $\xi_{\mathcal{K}(n^2)}$. So, when we use the criteria of the 4th phase of the algorithm to determine the words which should be neighbors in the scenery we make no mistake, by identifying as neighbors whilst they are not.

The next question is whether for all the neighboring words v and w in $\xi_{\mathcal{K}(4n)}$ we recognize them as neighbors, when we apply the fourth phase of our algorithm. This is indeed true, due to the event B_2^n. Let us explain this in more detail. Let v and w be to neighboring words in $\xi_{\mathcal{K}(4n)}$. We also assume that both v and w have length $4n$, are written in the direction of $\vec{u}$, and w is parallel and at distance 1 (in the direction $\vec{s}$ perpendicular to $\vec{u}$) of v. Now, take anywhere approximately in the middle of v, three consecutive strings v_a, v_b and v_c. Take them so that v_a and v_c have size $(\ln n)^2$, but v_b has size $(\ln n)^2 - 2$. Hence, we assume that the concatenation $v_a v_b v_c$ appears somewhere in the middle of v. Let $x \in \mathbb{Z}^d$ designate the point where v_a ends and let y be the point where v_c starts. Hence,

$$v_c = \xi_y \xi_{y+\vec{u}} \xi_{y+2\vec{u}} \cdots \xi_{y+((\ln n)^2-1)\vec{u}}.$$

Also, the points x and y are at distance $(\ln n)^2 - 1$ from each other and on the line directed along $\vec{u}$. Finally,

$$v_a = \xi_{x-((\ln n)^2+1)\vec{u}} \cdots \xi_{x-2\vec{u}} \xi_{x-\vec{u}} \xi_x.$$

So, the word v is written on the line passing through x and y. Note that the word w is written on the line which passes through the points $x + \vec{s}$ and $y + \vec{s}$. Now, because of the event B_2^n, for every nearest neighbor walk path of length $3(\ln n)^2 - 1$ contained in the region $\mathcal{K}(4n)$ there is at least one particle following that path up to time n^2.

Take the nearest neighbor walk path R on the time interval $[1, 3(\ln n)^2]$ which starts in $x - ((\ln n)^2 - 1)\vec{u}$ and then goes in the direction of $\vec{u}$ $(\ln n)^2$ steps. Next, goes one step in the direction of $\vec{s}$ (and hence reaches the line where w is written), and then R takes $(\ln n)^2$ steps in the direction of $\vec{u}$ and reads part of the word w. That part we designate by w_b. From there one step in the direction $(-\vec{s})$, to reach the point y and then all remaining $(\ln n)^2$ steps are taken in the direction $\vec{u}$. During those remaining steps, the walk generates the color record v_c. Such a nearest neighbor walk generates thus the color record $v_a w_b v_c$:

$$\xi \circ R = v_a w_b v_c.$$

Hence, by B_2^n at least one particle up to time n^2 follows R and generates the color record $v_a w_b v_c$. Similarly, we can choose a neighbor path that generates $v_a v_b v_c$. Since w_b and v_b are contained in w and v, respectively, then w and v get selected as neighbors by the fourth step of our algorithm. □

LEMMA 3.11 (The algorithm Λ^n works). *Let A^n be the event that Λ^n works. This means that the outcome after the fourth phase of Λ^n is "correct." Thus,*

$$B^n \cap C^n \cap D^n \cap F^n \subset A^n.$$

PROOF. Recall we said our algorithm works as a whole correctly, if there exists a box I with size $4n$ with center closer than $n^{0.5}$ from the origin and such that the restriction ξ_I is equivalent to our reconstructed piece of scenery. If the last phase of the algorithm works correctly, then we would like to see that outcome. The event that the outcome after the fourth phase is "correct," is denoted by A^n (as already mentioned). Now, assume that we have the correct short and long words constructed in phase one and two, that is, assume that the events B^n and C^n occur. Assume also that the third phase of our algorithm works correctly and we get the one long word picked at the end of phase 3 to be close enough to the origin, hence the event D^n occurs. When these three events occur (phase 1, 2 and 3 work) for the final phase of the algorithm to work, we then only need to identify which words of $LONGWORDS^n$ are neighbors of each other (remember that the words in $LONGWORDS^n$ are words of $\xi_{\mathcal{K}(n^2)}$), so, what we really need after is finding a way to determine which words of $\xi_{\mathcal{K}(n^2)}$ are neighbors of each other, and that holds by F^n.

Already at this point we observe that, if we have a collection of objects to be placed in a box of $\mathbb{Z}^d$ (in a way that all the sites of this box become occupied by exactly one object), and we know which objects should be placed in neighboring sites, then there is a unique (up to translations/reflections) way to do it. This will assure that the reconstruction works correctly once we identified the neighboring words. □

PROOF OF THEOREM 2.2. The last lemma above, Lemma 3.11, tells us that for the algorithm Λ^n to work correctly, we just need the events

$$B_1^n, B_2^n, B_3^n, B_4^n, B_5^n, C_1^n, C^{n^{0.25}}$$

to all hold simultaneously. Thus, Theorem 2.2 follows from Lemmas 3.1, 3.2, 3.3, 3.4, 3.5, 3.7 and 3.8. □

4. The infinite scenery: Proof of Theorem 2.1. How do we now reconstruct the infinite scenery ξ? So far, we have seen only methods to reconstruct a piece of ξ on a box of size $4n$ close to the origin. The algorithm was denoted by Λ^n, and the event that it works correctly is designated by the event A^n. By working correctly,

we mean that the piece reconstructed in reality is centered not further than $\sqrt{n}$ from the origin. In general, it is not possible to be sure about the exact location. So, instead we are only able to figure out the location up to a translation of order $\sqrt{n}$. Also, the piece is reconstructed only up to equivalence, which means that we do not know in which direction it is rotated in the scenery or if it was reflected. Now, the probability that the reconstruction algorithm Λ^n at level n does not work is small in the sense that it is finitely summable over n:

$$\sum_n P(A^{nc}) < \infty.$$

So, by the Borel–Cantelli lemma, when we apply all the reconstruction algorithms $\Lambda^1, \Lambda^2, \ldots$, we are almost sure that only finite many of them will not work. We use this to assemble the sceneries and get the whole scenery a.s. Let us call ξ^n the piece reconstructed by Λ^n. Hence,

$$\xi^n := \Lambda^n(\chi_0\chi_1 \cdots \chi_{n^4}),$$

where ξ^n is a piece of scenery on a box $\mathcal{K}(4n)$. The next task is to put the reconstructed pieces ξ^n together so that their relative position to each other is the same in the scenery ξ. For this, we will use the following rule. We proceed in an inductive way in n:

1. Let $\bar{\xi}^n$ designate the piece ξ^n which has been moved so as to fit the previously placed pieces. Hence, $\bar{\xi}^n$ is equivalent to ξ^n.

2. We place ξ^n by making it coincide with the previously placed ξ^{n-1} on a box of side length at least $\sqrt{n}$. In other words, $\bar{\xi}^n$ is defined to be any piece of the scenery equivalent to ξ^n, and such that on a restriction to a box of size $\sqrt{n}$ it coincides with $\bar{\xi}^{n-1}$. If no such box of size $\sqrt{n}$ can be found in $\bar{\xi}^{n-1}$ which is equivalent to a piece of the same size in ξ^n, then we "forget" about the previously placed pieces and just put the piece ξ^n on the box $\mathcal{K}(4n)$, that is, we center it around the origin.

3. The end result after infinite time is our reconstructed scenery denoted by

$$\bar{\xi} : \mathbb{Z}^d \to \{0, 1, \ldots, \kappa\}.$$

We will prove that a.s. $\bar{\xi}$ is equivalent to ξ. So, $\bar{\xi}$ represents our reconstructed ξ (since reconstruction in general is only possible up to equivalence). For $\bar{\xi}$ we simply take the point-wise limit of the $\bar{\xi}^n$:

$$\bar{\xi}(\vec{z}) := \lim_{n\to\infty} \tilde{\xi}^n.$$

For the above definition to be meaningful, take $\tilde{\xi}^n$ to be the extension of $\bar{\xi}^n$ to all $\mathbb{Z}^d$ by adding 0's where $\bar{\xi}^n$ is not defined.

To conclude the proof of Theorem 2.1, it is enough to prove that the above algorithm defines a scenery $\bar{\xi}$ which is almost surely equivalent to ξ.

THEOREM 4.1. *We have that*

$$P(\xi \approx \bar{\xi}) = 1.$$

PROOF. Let G^n be the event that in the restriction of ξ to the box $\mathcal{K}(2n+2)$ any two restrictions to a box of size $\sqrt{n}$ are different of each other. By this we mean, that if V_1 and V_2 are two boxes of size $\sqrt{n}$ contained in $\mathcal{K}(2n+1)$, then if the restriction ξ_{V_1} is equivalent to ξ_{V_2} then $V_1 = V_2$. Also, for G^n to hold, we require that for any box $V_1 \subset \mathcal{K}(2n+2)$ of size $\sqrt{n}$ the only reflection and/or rotation which leaves ξ_{V_1} unchanged is the identity.

Now note that when the event G^n occurs, and the piece ξ^{n+1} and ξ^n are correctly reconstructed in the sense defined before, then our placing them together works properly. This means that in that case, the relative position of $\bar{\xi}^{n+1}$ and $\bar{\xi}^n$ is that same as the corresponding pieces in ξ. It is elementary to obtain that the event G^n has probability at least $1 - c_1 n^d e^{-c_2 n^{d/2}}$. This means that $\sum_n P(G^{nc}) < \infty$, and so, by the Borel–Cantelli lemma, all but a finite number of the events G^n occur. Also, we have seen that the algorithm at level n has high probability to do the reconstruction correctly, and $\sum_n P(A^{nc}) < \infty$.

Hence, again by the Borel–Cantelli lemma, all but a finite number of the reconstructed scenery ξ^n will be equivalent to a restriction of ξ to a box close to the origin. We also see that the close to the origin for the box of the nth reconstruction means not further than $\sqrt{n}$. Thereof, we have that all but a finite number of the pieces $\bar{\xi}^n$ are positioned correctly with respect to each other. Since we take a limit for getting ξ, a finite number of $\bar{\xi}^n$'s alone have no effect on the limit, and so the algorithm works. □

REFERENCES

[1] BENJAMINI, I. and KESTEN, H. (1996). Distinguishing sceneries by observing the scenery along a random walk path. *J. Anal. Math.* **69** 97–135. MR1428097

[2] DEN HOLLANDER, F. and STEIF, J. E. (1997). Mixing properties of the generalized T, T^{-1}-process. *J. Anal. Math.* **72** 165–202. MR1482994

[3] HOWARD, C. D. (1996). Detecting defects in periodic scenery by random walks on $\mathbf{Z}$. *Random Structures Algorithms* **8** 59–74. MR1368850

[4] HOWARD, C. D. (1996). Orthogonality of measures induced by random walks with scenery. *Combin. Probab. Comput.* **5** 247–256. MR1411085

[5] HOWARD, C. D. (1997). Distinguishing certain random sceneries on $\mathbf{Z}$ via random walks. *Statist. Probab. Lett.* **34** 123–132. MR1457504

[6] KALIKOW, S. A. (1982). T, T^{-1} transformation is not loosely Bernoulli. *Ann. of Math.* (2) **115** 393–409. MR0647812

[7] KESTEN, H. (1996). Detecting a single defect in a scenery by observing the scenery along a random walk path. In *Itô's Stochastic Calculus and Probability Theory* 171–183. Springer, Tokyo. MR1439524

[8] KESTEN, H. (1998). Distinguishing and reconstructing sceneries from observations along random walk paths. In *Microsurveys in Discrete Probability* (*Princeton, NJ*, 1997). *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.* **41** 75–83. Amer. Math. Soc., Providence, RI. MR1630410

[9] LEMBER, J. and MATZINGER, H. (2008). Information recovery from a randomly mixed up message-text. *Electron. J. Probab.* **13** 396–466. MR2386738

[10] LINDENSTRAUSS, E. (1999). Indistinguishable sceneries. *Random Structures Algorithms* **14** 71–86. MR1662199

[11] LÖWE, M., MATZINGER, H. and MERKL, F. (2004). Reconstructing a multicolor random scenery seen along a random walk path with bounded jumps. *Electron. J. Probab.* **9** 436–507 (electronic). MR2080606

[12] LÖWE, M. and MATZINGER, H. III (2002). Scenery reconstruction in two dimensions with many colors. *Ann. Appl. Probab.* **12** 1322–1347. MR1936595

[13] MATZINGER, H. (1999). Reconstructing a three-color scenery by observing it along a simple random walk path. *Random Structures Algorithms* **15** 196–207. MR1704344

[14] MATZINGER, H. (2005). Reconstructing a two-color scenery by observing it along a simple random walk path. *Ann. Appl. Probab.* **15** 778–819. MR2114990

[15] MATZINGER, H. and PACHON, A. (2011). DNA approach to scenery reconstruction. *Stochastic Process. Appl.* **121** 2455–2473. MR2832409

[16] MATZINGER, H. and POPOV, S. (2007). Detecting a local perturbation in a continuous scenery. *Electron. J. Probab.* **12** 637–660. MR2318405

[17] MATZINGER, H. and ROLLES, S. W. W. (2003). Reconstructing a random scenery observed with random errors along a random walk path. *Probab. Theory Related Fields* **125** 539–577. MR1974414

[18] MATZINGER, H. and ROLLES, S. W. W. (2003). Reconstructing a piece of scenery with polynomially many observations. *Stochastic Process. Appl.* **107** 289–300. MR1999792

[19] MATZINGER, H. and ROLLES, S. W. W. (2006). Finding blocks and other patterns in a random coloring of $\mathbb{Z}$. *Random Structures Algorithms* **28** 37–75. MR2187482

[20] ORNSTEIN, D. (1971). A Kolmogorov automorphism that is not a Bernoulli shift. *Matematika* **15** 131–150. (In Russian.)

[21] POPOV, S. and PACHON, A. (2011). Scenery reconstruction with branching random walk. *Stochastics* **83** 107–116. MR2800083

[22] WEISS, B. (1972). The isomorphism problem in ergodic theory. *Bull. Amer. Math. Soc.* **78** 668–684. MR0304616

H. MATZINGER
SCHOOL OF MATHEMATICS
GEORGIA INSTITUTE OF TECHNOLOGY
686 CHERRY STREET
ATLANTA, GEORGIA 30332-0160
USA
E-MAIL: matzi@math.gatech.edu

A. PACHON
DIPARTIMENTO DI MATEMATICA "GIUSEPPE PEANO"
UNIVERSITY OF TURIN
VIA CARLO ALBERTO 10
10123, TURIN
ITALY
E-MAIL: apachonp@unito.it
angelicayohana.pachonpinzon@unito.it

S. POPOV
DEPARTMENT OF STATISTICS
UNIVERSITY OF CAMPINAS
RUA SÉRGIO BUARQUE DE HOLANDA 651
CIDADE UNIVERSITÁRIA
CEP 13083-859, CAMPINAS SP
BRASIL
E-MAIL: popov@ime.unicamp.br