Compressibility measures for two-dimensional data*

Lorenzo Carfagna[1][0009−0005−9591−057X] and
Giovanni Manzini[1][0000−0002−5047−0196]

University of Pisa
lorenzo.carfagna@gmail.com giovanni.manzini@unipi.it

Abstract. In this paper we extend to two-dimensional data two recently introduced one-dimensional compressibility measures: the γ measure defined in terms of the smallest string attractor, and the δ measure defined in terms of the number of distinct substrings of the input string. Concretely, we introduce the two-dimensional measures γ_{2D} and δ_{2D} as natural generalizations of γ and δ and study some of their properties. Among other things, we prove that δ_{2D} is monotone and can be computed in linear time, and we show that although it is still true that $\delta_{2D} \leq \gamma_{2D}$ the gap between the two measures can be $\Omega(\sqrt{n})$ for families of $n \times n$ matrices and therefore asymptotically larger than the gap in one-dimension. Finally, we use the measures γ_{2D} and δ_{2D} to provide the first analysis of the space usage of the two-dimensional block tree introduced in [Brisaboa *et al.*, Two-dimensional block trees, *The computer Journal*, 2023].

Keywords: Data compression · Repetitivenes Measures · Block Tree.

1 Introduction

Since the recent introduction of the notion of string attractor [6] different measures of string repetitiveness have been proposed or revisited [8,11]. It has been shown that such measures are more appropriate than the classical statistical entropy for measuring the compressibility of highly repetitive strings. In addition, these measures have been used to devise efficient compressed indices for highly repetitive string collections [10] an important setting which is hard for traditional entropy-based compressed indices.

In this paper we generalize the notion of attractor to two dimensional data, i.e. (square) matrices of symbols, and we initiate the study of the properties of the measure $\gamma_{2D}(M)$ defined as the size of the smallest attractor for the matrix M (Definition 1). As in the one-dimensional case, we introduce also the measure $\delta_{2D}(M)$ defined in terms of the number of distinct square submatrices (Definition 2) and we study the relationship between γ_{2D} and δ_{2D}. We prove that some properties that hold for strings are still valid in the two-dimensional

* Postprint version. The final authenticated publication is available online at https://doi.org/10.1007/978-3-031-43980-3_9

case: for example computing γ_{2D} is NP-complete while δ_{2D} can be computed in linear time, and for every matrix M it is $\delta_{2D}(M) \leq \gamma_{2D}(M)$. However, the gap between the two measures is larger than in one-dimension case since there are families of $n \times n$ matrices with $\delta_{2D} = O(1)$ and $\gamma_{2D} = \Omega(\sqrt{n})$, whereas for strings it is always $\gamma = O(\delta \log \frac{n}{\delta})$.

The study of the measures γ_{2D} and δ_{2D} is motivated by the fact that for two-dimensional data there is no clear definition of "context" of a symbol and therefore there is no universally accepted notion of statistical entropy. Therefore, alternative compressibility measures based on combinatorial properties such as γ_{2D} and δ_{2D} are worthwhile investigating. Indeed, in Section 3 we use the measures γ_{2D} and δ_{2D} to provide the first analysis of the size of the two-dimensional block tree introduced in [2]. In particular we show that the space used by a two-dimensional block tree for an $n \times n$ matrix M with delta measure δ_{2D} is bounded by $O((\delta_{2D} + \sqrt{n\delta_{2D}}) \log \frac{n}{\delta_{2D}})$, and that this space is optimal within a multiplicative factor $O(\log n)$.

For the rest of the paper, the RAM model of computation is assumed, with word size $w = \Theta(\log n)$ bits. Space is measured in words so when $O(x)$ space is indicated, the actual space occupancy in bits is $O(x \log n)$.

2 Two-dimensional compressibility measures

We consider a square matrix $M \in \Sigma^{n \times n}$ of size $n \times n$ where each of the n^2 symbols $M[i][j]$ of M are drawn from the alphabet Σ with $|\Sigma| = \sigma$. Every symbol in Σ is assumed to appear in M otherwise Σ is properly restricted. A submatrix of M with topmost left cell $M[i, j]$ is said to start at position (i, j) of M. An $a \times b$ submatrix of M starting at position (i, j) is written as $M[i : i+a-1][j : j+b-1]$, meaning that it includes any cell with row index in the range $[i, i + a - 1]$ and column index in $[j, j + b - 1]$. In this section two new repetitiveness measures for square matrices called γ_{2D} and δ_{2D} are proposed, as the generalisations of the γ and δ measures for strings respectively introduced in [6] and [13,3].

Definition 1. *An attractor Γ_{2D} for a square matrix $M \in \Sigma^{n \times n}$ is a set of positions of M: $\Gamma_{2D} \subseteq \{1, ..., n\} \times \{1, ..., n\}$ such that any square submatrix has an occurrence crossing (including) a position $p = (i, j) \in \Gamma_{2D}$. The measure $\gamma_{2D}(M)$ is defined as the cardinality of a smallest attractor for M.*

We say that a position $p = (i, j) \in \Gamma_{2D}(M)$ *covers* a submatrix I of M if there exists an occurrence of I which crosses p, and that a set of positions *covers* I if it includes a position p which *covers* I; when clear from the context, the parameter M is omitted from $\Gamma_{2D}(M)$ expression.

As a first result we show that, not surprisingly, the problem of finding the size of a smallest attractor is *NP-complete* also in two dimensions. The NP-completeness proof is done considering the decision problem "is there an attractor of size k for the given input?".

Lemma 1. *Given a string $S \in \Sigma^n$, let $R^S \in \Sigma^{n \times n}$ be the square matrix where each row is equal to the string S. Then there exists an (1-dim) attractor for S of size k if and only if there exists a (2-dim) attractor of size k for R^S.*

Proof. Given S and the corresponding R^S, the following observations hold: 1) any submatrix of R^S has an occurrence starting at the same column but on the first row of R^S; 2) any two $\ell \times \ell$ submatrices of R^S are equal if and only if the two respective substrings of S composing their rows are equal, formally: $R^S[i : i + \ell - 1][j : j + \ell - 1] = R^S[i' : i' + \ell - 1][j' : j' + \ell - 1]$ if and only if $S[j, j + \ell - 1] = S[j', j' + \ell - 1]$. From 1) and 2) the lemma follows: given a string attractor $\Gamma(S)$ for S of size k, the set $\Gamma_{2D} = \{(1, j) : j \in \Gamma(S)\}$ of size k is a two dimensional attractor for R^S and, vice versa, a string attractor Γ for S could be obtained from a matrix attractor $\Gamma_{2D}(R^S)$ for R^S projecting each couple by column index, formally, $\Gamma = \{j : (i, j) \in \Gamma_{2D}(R^S)\}$ is a one dimensional attractor for S. Note that if $\Gamma_{2D}(R^S)$ is a smallest attractor it does not include two positions on the same column, because, any distinct submatrix crossing one position has an occurrence (starting in the same column but at a different row) which crosses the other, hence in this case the projection does not generate any column index collision and $|\Gamma| = |\Gamma_{2D}(R^S)| = k$, otherwise, in case of collision, Γ could be completed with any $k - |\Gamma|$ positions not in Γ to reach size $k = |\Gamma_{2D}(R^S)|$. □

As an immediate consequence of the above lemma we have the following result.

Theorem 1. *Computing γ_{2D} is NP complete.* □

It is easy to see that $\gamma_{2D} \geq \sigma$ and γ_{2D} is insensitive to transpositions but, as for strings [9], γ_{2D} is not monotone. We show this by providing a family of matrices, built using the counterexample in [9] to disprove the monotonicity of γ, containing a submatrix with smaller γ_{2D}.

Lemma 2. *γ_{2D} is not monotone.*

Proof. Let w be the string $a\underline{b}\underline{b}\underline{b}a^n\underline{a}b$ with $n > 0$, having $\gamma(w) = 3$ minimal for the subset of positions $\Gamma(w) = \{2, 4, n + 5\}$ underlined in w. The string $w \cdot b = abb\underline{b}a^n\underline{a}bb$ obtained concatenating the letter b to w has a smaller compressibility measure $\gamma(w \cdot b) = 2$ corresponding to $\Gamma(w \cdot b) = \{4, n + 5\}$ [9], as the prefix $w[1, 3] = abb$ occurring as a suffix of $w \cdot b$ is already covered by position $n + 5$ in $\Gamma(w \cdot b)$. Consider $R^{w \cdot b}$ of size $(n + 7) \times (n + 7)$, from Lemma 1 follows that $\gamma_{2D}(R^{w \cdot b}) = \gamma(w \cdot b) = 2$, but the submatrix $R^{w \cdot b}[1 : n + 6][1 : n + 6]$ equal to R^w has a greater $\gamma_{2D}(R^w) = \gamma(w) = 3$. □

2.1 The measure δ_{2D}

The measure $\delta(S)$ for a string S, formally defined in [3] and previously introduced in [13] to approximate the output size of the Lempel–Ziv parsing, is the maximum

over $k \in [1, |S|]$ of the expression $d_k(S)/k$ where $d_k(S)$ is the number of distinct substrings of length k in S. We now show how to generalize this measure to two dimensions, by introducing the measure δ_{2D} which is defined in a similar way, considering $k \times k$ submatrices instead of length-k substrings.

Definition 2. *Given $M \in \Sigma^{n \times n}$, let $d_{k\times k}(M)$ be the number of distinct $k \times k$ submatrices of M, then*

$$\delta_{2D}(M) = \max\{d_{k\times k}(M)/k^2 : k \in [1,\ n]\}. \tag{1}$$

The measure δ_{2D} preserves some good properties of δ: δ_{2D} is invariant through transpositions and decreases or grows by at most 1 after a single cell edit since any $d_{k\times k}$ of the updated matrix could differ at most by k^2 from the initial one. δ_{2D} is monotone: given a submatrix M' of M having size $\ell \times \ell$ with $\ell \leq n$ any submatrix of M' appears somewhere in M then $d_{k\times k}(M') \leq d_{k\times k}(M)$ for any $k \in [1, \ell] \subseteq [1, n]$.

The next lemma shows that, as in the one-dimensional setting, δ_{2D} is upper bounded by γ_{2D}.

Lemma 3. *$\delta_{2D}(M) \leq \gamma_{2D}(M)$ for any matrix $M \in \Sigma^{n\times n}$.*

Proof. Let Γ_{2D} be a least size attractor for M i.e. $|\Gamma_{2D}| = \gamma_{2D}$. For any $k \in [1, n]$ an attractor position $p \in \Gamma_{2D}$ is included in at most k^2 distinct $k\times k$ submatrices, then we need at least $d_{k\times k}(M)/k^2$ distinct positions in Γ_{2D} to cover all $k \times k$ submatrices of M, formally, $|\Gamma_{2D}| \geq d_{k\times k}(M)/k^2$ holds for any $k \in [1, n]$ in particular for $k^* \in [1, n]$ such that $\delta_{2D} = d_{k^*\times k^*}(M)/(k^*)^2$. □

One of the main reasons for introducing δ was that it can be computed efficiently: [3] describes a linear algorithm to compute $\delta(S)$ with a single visit of the Suffix tree of S. We now show that an efficient algorithm for computing δ_{2D} can be derived in a similar way using the *Isuffix tree* introduced in [7] which can be built in $O(n^2)$ time, which is linear in the size of the input. A somewhat simpler algorithm can be obtained using the *Lsuffix tree* [4,5] but its construction takes $O(n^2 \log n)$ time.

The *Isuffix tree* $IST(A)$ of a matrix $A \in \Sigma^{n\times m}$ generalises the Suffix Tree to matrices: $IST(A)$ is a compacted trie representing all square submatrices of A. The Isuffix trees adopts a linear representation of a square matrix $C \in \Sigma^{q\times q}$: let $I\Sigma = \bigcup_{i=1}^{\infty} \Sigma^i$, each string in $I\Sigma$ is considered as an atomic *Icharacter*, the unique *Istring* associated to matrix C is $I_C \in I\Sigma^{2q-1}$ where $I_C[2i+1]$ with $i \in [0, q)$ is the $(i+1)^{th}$ *column-type Icharacter* $C[1 : i+1][i+1]$ and $I_C[2i]$ with $i \in [1, q)$ is the i^{th} *row-type Icharacter* $C[i+1][1 : i]$. See Figure 1 for an example. The k^{th} *Iprefix* of C is defined as the concatenation of the first k Icharacters $I_C[1] \cdot I_C[2] \cdot \ldots \cdot I_C[k] = I_C[1, k]$ of I_C. Note that an Iprefix ending in an odd position k is the Istring of the $\ell \times \ell$ square submatrix with $\ell = \lceil k/2 \rceil$ starting at C's top-left corner, that is, $C[1 : \ell][1 : \ell]$. For the example in Figure 1, the 3^{rd} Iprefix of C is the Istring "a a ba" which corresponds to the submatrix $C[1 : 2][1 : 2]$.

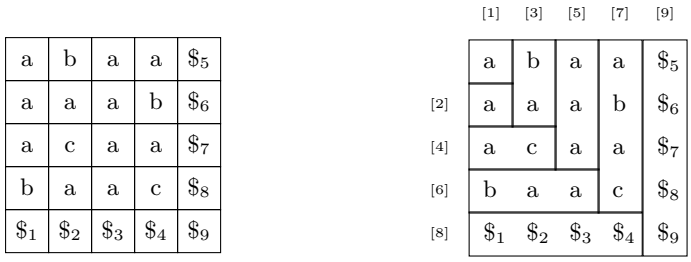

Fig. 1. A square matrix C on the left, and its Istring I_C on the right (last two Icharacters are omitted)

Given $A \in \Sigma^{n\times n}$, for $1 \leq i, j \leq n$, the *Isuffix* $I_{A_{ij}}$ of A is defined as the Istring of the largest square submatrix A_{ij} of A with upper left corner at position (i, j). From the above definitions it is clear that the Istring of any square submatrix of A, is an Iprefix (ending in a odd position) of some Isuffix $I_{A_{ij}}$. To ensure that no Isuffix $I_{A_{ij}}$ is Iprefixed by another Isuffix, A is completed with an additional bottom row and right column containing $2n+1$ distinct new symbols $\$_1, \ldots \$_{2n+1}$. For simplicity in the following we refer as A the input matrix already enlarged with $\$_i$ symbols. See Figure 2 for an example.

The Isuffix tree $IST(A)$ is a compacted trie over the alphabet $I\Sigma$ representing all the n^2 distinct Isuffixes $I_{A_{ij}}$ of A with, among others, the following properties [7]: 1) each edge is labeled with a non empty Isubstring $I_{A_{ij}}[\ell_1, \ell_2]$ of an Isuffix $I_{A_{ij}}$, that label is represented in constant space as the quadruple $\langle i, j, \ell_1, \ell_2\rangle$, the Isubstrings on any two sibling edges start with different Icharacters; 2) each internal node has at least two children and there are exactly n^2 leaves representing all the Isuffixes of A: let $L(u)$ be the Istring obtained concatenating the Isubstrings on the path from the root to a node u, for any leaf l_{ij}, the Istring $L(l_{ij})$ is equal to the linear representation $I_{A_{ij}}$ of the unique suffix A_{ij}; 3) The Isuffix tree satisfies the *common prefix constraint:* square submatrices of A with a common Iprefix share the same initial path in the tree; 4) The Isuffix tree satisfies the *completeness constraint* since all square submatrices of A are represented in $IST(A)$ as an Iprefix of some Isuffix of A.

Theorem 2. *δ_{2D} can be calculated in optimal time and space $O(n^2)$.*

Proof. Our algorithm is a generalization of the ideas used in [3] to compute the measure δ in linear time using a suffix tree. Given $A \in \Sigma^{n\times n}$, we build the array $d[1:n]$ which stores at position k the number of distinct $k \times k$ submatrices of A then we obtain δ_{2D} as $\max_k d[k]/k^2$. Initially the Isuffix Tree $IST(A)$ of A is built in time $O(n^2)$ [7], then $IST(A)$ is visited in depth first order. Let u be a node such that the path from the root to u contains $|L(u)|$ Icharacters. Let e be an edge outgoing from u labeled with $q_e = \langle i, j, \ell_1, \ell_2\rangle$ where $\ell_1 = |L(u)| + 1$. The Istring of a distinct square submatrix is obtained whenever appending a

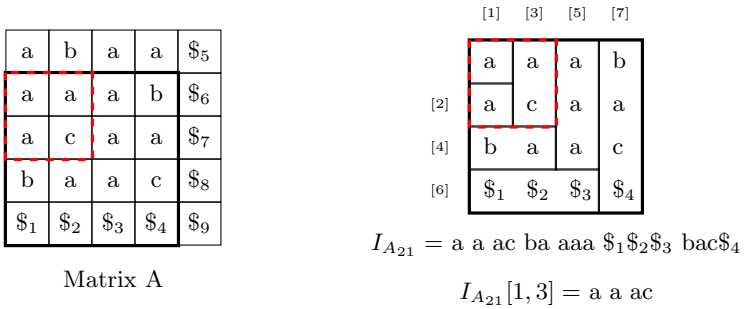

Fig. 2. The submatrix $A[2:5][1:4] = A_{21}$ with solid black border on the left and its Istring $I_{A_{21}}$ on the right. The Istring of the submatrix $A[2:3][1:2]$ (in red) is the third Iprefix of $I_{A_{21}}$.

prefix of the Isubstring $I_{A_{ij}}[\ell_1, \ell_2]$ labelling the edge e to $L(u)$ yields an Istring of odd length. Because the traversing of e may yield new square submatrices, $d[\cdot]$ must be updated accordingly. Let $s = \lceil \frac{\ell_1 - 1}{2} \rceil + 1$ and $t = \lceil \frac{\ell_2}{2} \rceil$. Every $d[k]$ with $k \in [s, t]$ should be increased by one: to do this in constant time we set $d[s] = d[s] + 1$ and $d[t+1] = d[t+1] - 1$ and we assume that each value stored in an entry $d[i]$ is *implicitly* propagated to positions $i+1, i+2, \ldots n$: so the $+1$ is propagated from s up to t and the propagation is canceled by the -1 added at the position $t+1$. At the end of the Isuffix tree visit, for each $k \in [1, n-1]$ we set $d[k+1] = d[k+1] + d[k]$ so that $d[k]$ contains the number of distinct $k \times k$ matrices encountered during the visit and we can compute δ_{2D} as $\max_k d[k]/k^2$.

Note that when leaf l_{ij} is reached via the edge e with label $q_e = \langle i, j, \ell_1, \ell_2 \rangle$, all the Iprefixes of $I_{A_{ij}}[\ell_1, \ell_2]$ that have an Icharacter which includes some $\$_x$ symbol should not be counted. The range of well formed Iprefixes can be determined in constant time since it suffices to access one symbol in each of the last two trailing Icharacters of $I_{A_{ij}}[\ell_1, \ell_2]$ to check whether these two contains any $\$_x$. Since the Isuffix Tree can be constructed and visited in $O(n^2)$ time the overall time and space complexity of the above algorithm is $O(n^2)$. □

We now study how large can be the gap between the two measures γ_{2D} and δ_{2D}, recalling that by Lemma 3 it is $\delta_{2D} \leq \gamma_{2D}$. In [8] Kociumaka et al. establish a separation result between measures δ and γ by showing a family of strings with $\delta = O(1)$ and $\gamma = \Omega(\log n)$. This bound is tight since they also prove that $\gamma = O(\delta \log \frac{n}{\delta})$. The next theorem proves that the gap between the two measures in two dimensions is much bigger: δ_{2D} can be (asymptotically) smaller than γ_{2D} up to a $\sqrt{n}$ factor.

Lemma 4. *There exists a family of $n \times n$ matrices with $\delta_{2D} = O(1)$ and $\gamma_{2D} = \Omega(\sqrt{n})$.*

Proof. Consider the matrix M of size $n \times n$ where the first row is the string S composed by $\sqrt{n}/2$ consecutive blocks of size $2\sqrt{n}$ each. The i^{th} block S_i

with $i = 1, \ldots, \sqrt{n}/2$ is the string $1^i 0^{(2\sqrt{n}-i)}$, so S_i contains (from left to right) i initial ones and the remaining positions are zeros. The remaining rows of the matrix are all equals to $\#^n$. Note that for any size k all distinct submatrices start in the first row or are equal to $\#^{(k\times k)}$. Let δ_k be $d_{k\times k}/k^2$, so that δ_{2D} can be rewritten as $\max\{\delta_k \mid k \in [1,\ n]\}$. We compute δ_k for each possible k. For $k = 1$, we have $\delta_1 = |\Sigma| = 3$. For $k \geq \sqrt{n}$ it is $\delta_k = O(1)$ since $k^2 \geq n$ and there at most $(n-k+1)+1 \leq n$ distinct $k \times k$ matrices. Now consider δ_k with $k \in [2, \ldots, \sqrt{n})$. All distinct $k \times k$ submatrices (excluded the $\#^{(k\times k)}$ one) are those having as first row a distinct substring of length k of S. All those substrings are included in the language $0^a 1^b 0^c$ with $a \in [0, \ldots, k], b \in [0, \ldots, k-a], c \in [0, \ldots, k-a-b]$ such that $a + b + c = k$, to see this note that no substring of length $k < \sqrt{n}$ can contain any two non adjacent (and non empty) groups of ones since there is a group of at least $\sqrt{n} > k$ consecutive zeros between each of them in S. Fixed k, to count the strings in $0^a 1^b 0^c$ is enough to count the possible choices for the starting/ending positions of the middle 1^b block: which are $O(k^2)$, then for $k \in [2, \ldots, \sqrt{n})$, $\delta_k = \frac{O(k^2)}{k^2} = O(1)$. This proves that $\delta_{2D} = O(1)$.

To estimate γ_{2D} consider the i^{th} block on the first row: $S_i = 1^i 0^{(2\sqrt{n}-i)}$. Each S_i with $i = 1, \ldots, \sqrt{n}/2$ is a unique occurrence since the sequence 1^i occurs only inside blocks S_j with $j \geq i$ which begins with at least i ones, but inside S_j the sequence 1^i is followed by $2\sqrt{n}-j < 2\sqrt{n}-i$ zeros, so the copy of S_i will intersect the $(j+1)^{th}$ block where no leading zeros are present. As a consequence each submatrix M_i of size $2\sqrt{n}\times 2\sqrt{n}$ having S_i as first row is a unique occurrence too. As each M_i does not overlap any other M_j with $j \neq i$ at least $\sqrt{n}/2$ positions are needed in Γ_{2D} to cover them. This proves that $\gamma_{2D} = \Omega(\sqrt{n})$. □

Given a set S, the worst-case entropy [8] of S defined as $\lceil \log_2 |S| \rceil$ is the minimum number of bits needed to encode all the elements in S. In the following Lemma, we extend the construction of Lemma 4 to define a family $\mathcal{F}$ of matrices with constant δ_{2D} and worst-case entropy $\Omega(\sqrt{n}\log n)$.

Lemma 5. *There exists a family of square matrices on a constant size alphabet Σ with common measure $\delta_{2D} = O(1)$ and worst-case entropy $\Omega(\sqrt{n}\log n)$.*

Proof. Consider again the matrix M of Lemma 4. Each of the $(\sqrt{n}/2)!$ matrices obtained permuting the $\sqrt{n}/2$ blocks S_i on the first row of M has still $\delta_{2D} = O(1)$. On the other hand, every encoding algorithm to distinguish among these matrices needs at least $\log((\sqrt{n}/2)!) = \Theta(\sqrt{n}\log n)$ bits. □

3 Space Bounds for Two-Dimensional Block Trees

Brisaboa et. al. [2] generalized the Block Tree concept [1] to two dimensional data providing a compressed representation for discrete repetitive matrices that offers direct access to any compressed submatrix in logarithmic time. Given a matrix $M \in \Sigma^{n\times n}$ and an integer parameter $k > 1$, assuming for simplicity that n is a power of k, i.e. $n = k^\alpha$, M is split into k^2 non overlapping submatrices, called blocks, each of size $(n/k) \times (n/k) = k^{\alpha-1} \times k^{\alpha-1}$. Each of these blocks

corresponds to a node at level $\ell = 1$ in the *2D-BT* and the root of the tree at level $\ell = 0$ represents the whole matrix M. A tree is obtained by splitting (some of) the blocks at level ℓ, which have size $(n/k^\ell) \times (n/k^\ell)$, into k^2 non overlapping blocks of size $(n/k^{\ell+1}) \times (n/k^{\ell+1})$. At any level ℓ, nodes whose corresponding submatrix intersects the first occurrence, in row major order, of a $(n/k^\ell) \times (n/k^\ell)$ submatrix (including themselves) are internal nodes, referred in the following as *marked ones*; all others nodes are the level-ℓ leaves of the *2D-BT*, and referred in the following as *unmarked nodes*. Only marked nodes are recursively split and expanded at level $\ell + 1$; instead an unmarked node corresponding to a block X points to the marked nodes in the same level corresponding to the level-ℓ blocks overlapping the first (in row major order) occurrence O of X, and stores the relative offset $\langle O_x, O_y \rangle$ of O inside the top left of such blocks (see Figure 3). The splitting process ends when explicitly storing blocks is more convenient than storing pointers to marked blocks. The resulting tree-shaped data structure has height $h = O(\log_k n)$. In the following the block related to node u in the tree is named B_u, and a block B_u is said to be marked (unmarked) if the corresponding node u is marked (unmarked). Note that if X is unmarked, then the (up to) four blocks intersecting the first occurrence O of X are marked by construction. If we call D the $(2n/k^\ell) \times (2n/k^\ell)$ submatrix formed by these four blocks, we observe that this is also a first occurrence (otherwise we would have another occurrence of X preceding O) and therefore the up to four blocks at level $\ell - 1$ containing D will be marked. Repeating this argument shows that an unmarked node points to marked nodes which always exist in the same level since none of their ancestors has been pruned in a previous level. Note that our marking scheme is slightly different than the one in [2] in which if a submatrix is pruned at some level its content is seen as all 0s in the subsequent levels. This approach removes the issue of possibly pointing to pruned nodes, but makes it difficult to estimate the number of marked nodes in terms of the matrix content, which is our next objective.

It has already been proved [8] that one-dimensional Block Trees are worst case optimal in terms of δ in the following sense: a BT on a string $S \in \Sigma^n$ uses $O(\delta \log \frac{n \log \sigma}{\delta \log n})$ space and there exist string families requiring that amount of space to be stored. No space analysis of the *2D-BT* was given in [2]. In the following we show that a *2D-BT* built on a matrix $M \in \Sigma^{n \times n}$ occupies $O((\delta_{2D} + \sqrt{n\delta_{2D}}) \log \frac{n}{\delta_{2D}})$ space.

Lemma 6. *The number of marked nodes in any level of a 2D block tree is* $O(\delta_{2D} + \sqrt{n\delta_{2D}})$.

Proof. Consider a generic tree level, and assume the blocks in this level have size $k^\ell \times k^\ell$. In the following the term *block* denotes a $k^\ell \times k^\ell$ submatrix of M whose upper left corner is an entry of the form $M[1 + \lambda k^\ell, 1 + \mu k^\ell]$ with λ, μ integers. For any distinct $k^\ell \times k^\ell$ submatrix in M, let O be the first occurrence of that submatrix in row major order. O intersects $m \in \{1, 2, 4\}$ blocks $B_1, \ldots, B_m$ that are therefore marked. Let D be a $2k^\ell \times 2k^\ell$ submatrix built including all the m blocks $B_1, \ldots, B_m$, and therefore containing O. We call D a *superblock*.

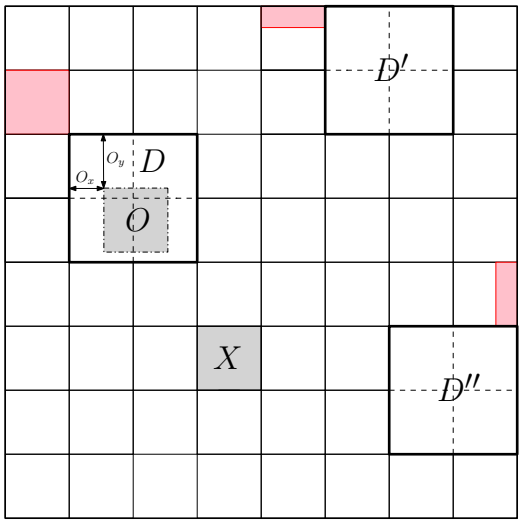

Fig. 3. The four blocks inside region D are marked since they overlap the first occurrence O, in row major order, of X. For the argument in the proof of Lemma 6 D is a type 3 superblock: by considering every entry in the red block above it as an upper left corner we obtain $k^{2\ell}$ distinct $3k^\ell \times 3k^\ell$ matrices containing D. The type 2 superblock D' borders the upper edge; by considering the k^ℓ entries in the first row marked in red we obtain k^ℓ distinct $3k^\ell \times 3k^\ell$ matrices containing D'. We also show a type 2 superblock D'' bordering the right edge; we obtain k^ℓ distinct $3k^\ell \times 3k^\ell$ matrices containing D'' by considering the $3k^\ell \times 3k^\ell$ matrices with the upper *right* corner in the portion of the last column marked in red.

If $m = 4$, D is unique, otherwise $4 - m$ more blocks are chosen arbitrarily to reach the desired size. Let u be the number of superblocks constructed in this way; the number of marked blocks at this level is at most $4u$, hence we proceed bounding u. We partition the superblocks into 3 types: 1) those on a corner of M, i.e. including one of the entries $M[1,1], M[1,n], M[n,1]$ or $M[n,n]$, these are at most four; 2) those not on a corner but including an entry in the first/last row/column; 3) those not including any entry in the first/last row/column. Let u_i be the number of superblocks of type i: clearly $u = u_1 + u_2 + u_3 = O(u_2 + u_3)$.

Given a superblock D of the third type, we observe that D is included into $k^{2\ell}$ distinct $3k^\ell \times 3k^\ell$ submatrices starting at any position in the $k^\ell \times k^\ell$ block touching the top left corner of D (see Figure 3). Summing over all type 3 superblocks, we have a total of $u_3 k^{2\ell}$ submatrices of size $3k^\ell \times 3k^\ell$ starting in distinct positions inside M. These submatrices are distinct: each submatrix contains, by construction, the first occurrence of some block; if two of those matrices were equal we would have two first occurrences of the same block starting in different positions which is impossible. Since by definition the number of distinct submatrices of size $3k^\ell \times 3k^\ell$ is at most $(3k^\ell)^2 \delta_{2D}$ we have

$$u_3 k^{2\ell} \leq 9k^{2\ell}\delta_{2D} \quad \Longrightarrow \quad u_3 \leq 9\delta_{2D}$$

Consider now a superblock D' of the second type bordering the upper edge of M (the other 3 cases are treated similarly, see superblock D'' in Figure 3). Any $3k^\ell \times 3k^\ell$ matrix which starts in the same row of D', but, in any of the k^ℓ columns preceding D' is distinct by the same argument presented before (see again Figure 3). Reasoning as above we find $u_2 k^\ell$ distinct $3k^\ell \times 3k^\ell$ matrices which implies $u_2 \leq 9k^\ell \delta_{2D}$. Since it is also $u_2 \leq n/k^\ell$, we have $u_2 \leq \min(9k^\ell \delta_{2D}, n/k^\ell) = O(\sqrt{n\delta_{2D}})$. We conclude that the number of marked blocks at any level is $O(u) = O(u_1 + u_2 + u_3) = O(\delta_{2D} + \sqrt{n\delta_{2D}})$. □

Theorem 3. *The 2D-BT takes $O((\delta_{2D} + \sqrt{n\delta_{2D}}) \log \frac{n}{\delta_{2D}})$ space. This space is optimal within a multiplicative factor $O(\log n)$.*

Proof. The *2D-BT* as described at the beginning of the section has height $\log_k n$. Such height can be reduced choosing a different size for the blocks at level $\ell = 1$. Assuming for simplicity $n = \sqrt{\delta_{2D}} k^\alpha$, M is initially divided into δ_{2D} blocks of size $k^\alpha \times k^\alpha$. In this way the height of the tree became $O(\log_k \frac{n}{\delta_{2D}})$, using the bound of the Lemma 6, we get an overall number of marked nodes $O((\delta_{2D} + \sqrt{n\delta_{2D}}) \log_k \frac{n}{\delta_{2D}})$. Each marked node produces at most k^2 unmarked nodes on the next level, hence the tree has at most $O(k^2(\delta_{2D} + \sqrt{n\delta_{2D}}) \log_k \frac{n}{\delta_{2D}})$ nodes which is $O((\delta_{2D} + \sqrt{n\delta_{2D}}) \log \frac{n}{\delta_{2D}})$ for $k = O(1)$. To prove the worst-case quasi-optimality: let $\mathcal{F}$ be the set of matrices having $\delta_{2D} = O(1)$ from Lemma 5, for any coder $C : \mathcal{F} \to \{0,1\}^*$ representing all the matrices in $\mathcal{F}$, there exist a matrix W such that $|C(W)| = \Omega(\sqrt{n} \log n)$ bits while the *2D-BT* takes $O(\sqrt{n} \log^2 n)$ bits of space for any matrix in $\mathcal{F}$. □

The following result shows that the bound in Lemma 6 cannot be substantially improved at least when $\delta_{2D} = O(1)$. Since the proof of Lemma 6 shows that the number of marked blocks *at the interior* of the matrix is bounded by $O(\delta_{2D})$, we consider a family of matrices that have a hard to compress first row.

Lemma 7. *There exists an infinite family of matrices $M \in \Sigma^{n \times n}$ with $\delta_{2D} = O(1)$, such that 2D-BT for M has $\Omega(\sqrt{n})$ marked nodes on a single level.*

Proof. Let $M \in \Sigma^{n \times n}$ be the matrix of Lemma 4 with $n = k^{2\alpha}$ so that n is both a power of k and a perfect square. We have already proven that $\delta_{2D}(M) = O(1)$. Consider the *2D-BT* built on M: note that for block size larger than $4\sqrt{n} \times 4\sqrt{n}$ each block on the upper edge of M includes entirely in its first row at least one of the strings S_i of the form $1^i 0^{(2\sqrt{n}-i)}$ composing S. Since each S_i is unique, any of those blocks is a first occurrence and hence marked. In particular, at level $\ell = \alpha - \lceil \log_k 4 \rceil$, counting levels from the root ($\ell = 0$) to the leaves, all $\Theta(\sqrt{n})$ blocks on M upper side are marked and the lemma follows. □

In [12] the authors introduced a variant of the one-dimensional block tree, called Γ-tree, in which, given a not necessarily minimum string attractor Γ, the marked nodes at each level are those close to an attractor position. The Γ-tree is then enriched with additional information that makes it a compressed full text index using $O(\gamma \log(n/\gamma))$ space where $\gamma = |\Gamma|$ is the size of the string attractor.

Following the ideas from [12], we now show how to modify the construction of the *2D-BT* assuming we have available a, not necessarily minimum, *2D-attractor* Γ_{2D}.

To simplify the explanation, again we assume that $n = k^\alpha$ for some $\alpha > 0$, given a matrix $M \in \Sigma^{n \times n}$ and an attractor $\Gamma_{2D} = \{(i,j)_1, \ldots, (i,j)_\gamma\}$ for M, the splitting process is unchanged but at level ℓ we mark any node u corresponding to a $n/k^\ell \times n/k^\ell$ block B_u which includes a position $p \in \Gamma_{2D}$ and all (the at most 8) nodes of the blocks adjacent to B_u. Remaining nodes are unmarked and store a pointer ptr_B to the marked block B on the same level ℓ where an occurrence O of their corresponding submatrix that spans a position $p \in \Gamma_{2D}$ begins, as well as the relative offset of O within B. The claimed occurrence O crossing $p \in \Gamma_{2D}$ exists otherwise Γ_{2D} would not be an attractor for M, and all the (at most) 4 blocks intersecting O are ensured to be marked as they contain p or are adjacent to a block containing p. We also point out that overlapping between an unmarked block B' and the pointed occurrence O containing $p \in \Gamma_{2D}$ is impossible as if this happens B' would be adjacent to a block B with $p \in B$, hence B' would be marked as well.

If n is not a power of k, blocks on right and bottom edges of M won't be squared, but no special treatment is needed as all the previous essential properties are still valid: consider a rectangular block B of size $a \times b$ on the edge, if B is marked, B is recursively split into smaller blocks (some of those with rectangular shape), if B is unmarked, the squared matrix B' of size $c \times c$ with $c = max(a,b)$ including B is squared and will occur somewhere else crossing an attractor position while spanning at most four marked blocks, then B would occur as well. Note that B, contrary to B', may not cross any attractor position but will certainly point to marked blocks only, avoiding unmarked blocks point to other unmarked ones.

Theorem 4. *Given $M \in \Sigma^{n \times n}$ and an attractor $\Gamma_{2D}(M) = \{(i,j)_1, \ldots, (i,j)_\gamma\}$ of size γ, the 2D-BT built using Γ_{2D} takes $O(\gamma \log \frac{n}{\gamma})$ space.*

Proof. Each position $p \in \Gamma$ could mark at most 9 distinct blocks per level: the block B including p and the (up to) eight blocks adjacent to B. Hence the number of marked blocks per level is $\leq 9\gamma$. Assuming again $n = \sqrt{\gamma}k^\alpha$, dividing initially M into blocks of size $k^\alpha \times k^\alpha$ we get a shallower tree of height $O(\log_k \frac{n}{\gamma})$ with $O(\gamma)$ nodes on the second level ($\ell = 1$) and an overall number of marked nodes $O(\gamma \log_k \frac{n}{\gamma})$. Since any marked node produces at most k^2 unmarked nodes on the next level, for any $k = O(1)$ the *2D-BT* built using any attractor Γ_{2D} of size γ takes $O(\gamma \log \frac{n}{\gamma})$ space. □

Access to a single symbol $M[i][j]$ is quite as in the one dimensional Γ-Tree: assume the node u at level ℓ is reached with a local offset $\langle O_x, O_y \rangle$, if u is a marked node, the child c of u where the searched cell falls is determined, the coordinates $\langle O_x, O_y \rangle$ are translated to local coordinates on c where the search routine proceeds in the next level. If instead node u is unmarked, the marked node v on the same level is reached via the pointer ptr_v stored in u, the actual

offset inside the block B_v is determined using the offset $\langle O'_x, O'_y \rangle$ stored in u and the access procedure continues on marked node v. The descending process halts when a marked block on the deepest level is reached and the corresponding explicit symbol is retrieved. Access procedure costs $O(\log n)$ as we visit at most 2 nodes on the same level before descend to next one.

Funding: This research was partially supported by MIUR-PRIN project "Multicriteria Data Structures and Algorithms: from compressed to learned indexes, and beyond" grant n. 2017WR7SHH, and by the PNRR ECS00000017 Tuscany Health Ecosystem, Spoke 6 "Precision medicine & personalized healthcare", CUP I53C22000780001, funded by the European Commission under the NextGeneration EU programme.

References

1. Djamal Belazzougui, Manuel Cáceres, Travis Gagie, Pawel Gawrychowski, Juha Kärkkäinen, Gonzalo Navarro, Alberto Ordóñez Pereira, Simon J. Puglisi, and Yasuo Tabei. Block trees. *J. Comput. Syst. Sci.*, 117:1–22, 2021.
2. N. Brisaboa, T. Gagie, A. Gómez-Brandón, and G. Navarro. Two-dimensional block trees. *The Computer Journal*, 2023. To appear.
3. Anders Roy Christiansen, Mikko Berggren Ettienne, Tomasz Kociumaka, Gonzalo Navarro, and Nicola Prezza. Optimal-time dictionary-compressed indexes. *ACM Trans. Algorithms*, 17(1):8:1–8:39, 2021.
4. Raffaele Giancarlo. A generalization of the suffix tree to square matrices, with applications. *SIAM J. Comput.*, 24(3):520–562, 1995.
5. Raffaele Giancarlo and Roberto Grossi. On the construction of classes of suffix trees for square matrices: Algorithms and applications. *Inf. Comput.*, 130(2):151–182, 1996.
6. Dominik Kempa and Nicola Prezza. At the roots of dictionary compression: string attractors. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 827–840. ACM, 2018.
7. Dong Kyue Kim, Joong Chae Na, Jeong Seop Sim, and Kunsoo Park. Linear-time construction of two-dimensional suffix trees. *Algorithmica*, 59(2):269–297, 2011.
8. Tomasz Kociumaka, Gonzalo Navarro, and Nicola Prezza. Toward a definitive compressibility measure for repetitive sequences. *IEEE Trans. Inf. Theory*, 69(4):2074–2092, 2023.
9. Sabrina Mantaci, Antonio Restivo, Giuseppe Romana, Giovanna Rosone, and Marinella Sciortino. A combinatorial view on string attractors. *Theor. Comput. Sci.*, 850:236–248, 2021.
10. G. Navarro. Indexing highly repetitive string collections, part I: Repetitiveness measures. *ACM Computing Surveys*, 54(2):article 29, 2021.
11. G. Navarro. Indexing highly repetitive string collections, part II: Compressed indexes. *ACM Computing Surveys*, 54(2):article 26, 2021.
12. Gonzalo Navarro and Nicola Prezza. Universal compressed text indexing. *Theoretical Computer Science*, 762:41–50, 2019.
13. Sofya Raskhodnikova, Dana Ron, Ronitt Rubinfeld, and Adam D. Smith. Sublinear algorithms for approximating string compressibility. *Algorithmica*, 65(3):685–709, 2013.