I/O Complexity for Range Queries on Region Data Stored Using an R-tree

Guido Proietti *
Dip. di Matematica Pura ed Applicata
University of L'Aquila
Via Vetoio, I-67010, Italy
proietti@univaq.it

Christos Faloutsos †
Dept. of Computer Science
Carnegie Mellon University
5000 Forbes Ave., Pittsburgh, 15213 PA
christos@cs.cmu.edu

Abstract

In this paper we study the node distribution of an R-tree storing region data, like for instance islands, lakes or human-inhabited areas. We will show that real region datasets are packed in minimum bounding rectangles (MBRs) whose area distribution follows the same power law, named REGAL *(REGion Area Law) [12], as that for the regions themselves. Moreover, these MBRs are packed in their turn into MBRs following the same law, and so on iteratively, up to the root of the R-tree. Based on this observation, we are able to accurately estimate the search effort for range queries, the most prominent spatial operation, using a small number of easy-to-retrieve parameters.*

Experiments on a variety of real datasets (islands, lakes, human-inhabited areas) show that our estimation is accurate, enjoying a maximum geometric average relative error within 30%.

1. Introduction

The area distribution of real region datasets has been discovered obeying to an hyperbolic power law, named *REGAL* (REGion Area Law) [12]. Korčak was the first to observe such a law, for the Aegean Islands [8]. Recent measurements on 2-d region datasets from diverse applications suggest that usually a similar power law holds [6]. In [12] this fact has been used to estimate the selectivity of range queries using only a few, easy-to-retrieve parameters, thus outperforming previous time-consuming approaches requiring the knowledge of the extent of all the regions in the dataset.

However, the above law has not yet been used to analyze the performances of any spatial access method used to store region data. In particular, an extensively studied problem is that of providing a realistic statistical model for the node distribution of the R-tree family [1, 4, 5, 13], which are among the most popular data structures in the spatial database community. In the past, most of the analysis efforts in this direction have focused on point and line data [3, 16], using the optimistic assumption that the leaf nodes (and iteratively, their parents) were square-like rectangles, roughly of the same size. Whereas this assumption can be accepted when working on point and line data, for region data it is too restrictive.

In this paper we deviate from this assumption, showing that real region datasets are packed in minimum bounding rectangles (MBRs) having a quite uniform aspect ratio and whose area distribution follows the REGAL law, as that for the regions themselves, independently, to some extent, of the capacity of the buckets. We also show that the law propagates towards further aggregation levels (i.e., buckets of MBRs follow the REGAL law). Based on this observation, we are able to accurately estimate the search effort for range queries posed on region data stored by means of R-trees. Since range queries are the most prominent spatial operation and R-trees are the most popular spatial access method, we provide a significant contribution to the theoretical analysis of query optimization for GIS and spatial DBMS.

Thus, the problem we focus on is the following: *We are given a real set of region data stored using an R-tree: what is the expected I/O complexity for range queries?* We answer this question in the rest of the paper, developing a realistic statistical model for R-trees storing region data, and showing how to use it to compute the I/O complexity of range queries. Its maximum geometric average relative error is within 30%.

*The research of this author was partially supported by the Italian National Research Council (CNR) under the fellowship N.215.29 and by the EU TMR Grant CHOROCHRONOS.

†The research of this author was partially supported by the NSF under Grant IRI-9625428, by the CMU's InforMedia, by the NSF, ARPA and NASA under NSF Cooperative Agreement No. IRI-9411299.

The paper is organized as follows: Section 2 gives a brief description of previous work on the topic. In Section 3 we develop our model and we show how it can be used to estimate the I/O complexity of window queries on regions datasets. Section 4 provides a large collection of experimental results on real region data (collection of islands, lakes, urban areas, etc.) and suggests some directions to a practitioner for an effective application of our model. Finally, Section 5 contains concluding remarks and future work.

2. Survey

The R-tree of Guttman [5] is a hierarchical spatial data structure that is derived from the B-tree. Each node in the tree corresponds to the smallest d-dimensional rectangle that encloses its child nodes. A leaf node contains pointers to the actual geometric object in the database, instead of child. All leaf nodes appear at the same level and parent nodes are allowed to overlap. This will guarantee at least 50% of space utilization and will maintain a balanced structure.

Due to its good space and time performances, the R-tree has been subject of further analysis and developments: among the most successful variations, we recall the R^+-tree [15] and the R^*-tree [1]. In particular, the latter outperforms previous approaches, deferring the splits by 'force-reinserting' some of the entries of the overflowing nodes.

As far as the analysis of R-trees is concerned, in [3] a formula that estimates the number of disk accesses for range queries posed on point datasets has been developed. Remember that range (or window) queries, are the most popular spatial access operation [11, 9]. In [16], a model for the prediction of R-tree performances is given, using the concept of *density* of data. However, the optimistic assumption that the leaf nodes (and iteratively, their parents) were square-like rectangles, roughly of the same size, is done. Whereas this is reasonable for the class of data analyzed in [16], for region data this is not the case.

3. Proposed method

In this section, we first give the problem definition and then we give the proposed solution. Table 1 gives a list of main symbols used throughout this section.

3.1. Problem definition

Let us rigorously state the problem we are concerned with. For the sake of clarity, we focus on the 2-dimensional space, but all the results can be extended to the d-dimensional space.

Symbol	Definition
$\mathcal{R}$	Dataset of rectangles
n	Count of rectangles of $\mathcal{R}$
N	Count of nodes of the R-tree
B	Patchiness exponent
D_H	Fractal dimension
A, W, H	Total area, width and height of $\mathcal{R}$
A_j, W_j, H_j	Area, width, height of the level-j nodes
σ	Ratio width/height of rectangles
$\alpha_j, \omega_j, \eta_j$	Area, width, height of largest level-j node
N_j	Count of level-j nodes
m	Count of levels in the R-tree
M	Maximum number of rectangles per node
$\overline{M}$	Average number of rectangles per node
$C(a)$	Count of regions having area at least a
$\vec{q} = (q_x, q_y)$	Query window of sides q_x, q_y
$DA(\mathcal{R}, \vec{q})$	Number of disk accesses for query $\vec{q}$
$U = [0,1]^2$	Image space

Table 1. Symbol table

PROBLEM: disk accesses for range queries
Given:

- A set of *similar* rectangles (i.e., having a fixed given aspect ratio σ between width and height) $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$ embedded in $U = [0,1] \times [0,1]$ and stored using an R-tree.
- A $q_x \times q_y$ *window query* $\vec{q}$.

Find the number of disk accesses $DA(\mathcal{R}, \vec{q})$ in $\mathcal{R}$ of the window query $\vec{q}$, that is, the number of R-tree nodes intersecting $\vec{q}$.

The formula in [7, 11] provides a solution to the above problem when we know the width w_i and the height h_i of every node in the R-tree. Let N be the total number of nodes in the R-tree. We have

$$DA(\mathcal{R}, \vec{q}) = \sum_{i=1}^{N} (q_x + w_i) \cdot (q_y + h_i) \qquad (1)$$

which can also be written

$$DA(\mathcal{R}, \vec{q}) = A_{tot} + q_x \cdot H_{tot} + q_y \cdot W_{tot} + q_x \cdot q_y \cdot N \quad (2)$$

where A_{tot}, W_{tot} and H_{tot} are the total area, width and height extent of the R-tree nodes. The question is to estimate the selectivity with much less information.

3.2. Proposed solution using the REGAL law

Real region datasets do not obey the uniformity assumption. Rather, it turns out that the complementary cumula-

tive distribution function[1] (CCDF) of the areas of the regions obeys the following hyperbolic power law (REGAL law [12]):

$$C(a) = k \cdot a^{-B} \quad k, B > 0, \;\; a \geq 0. \tag{3}$$

where k, B are constant.

Korčak was the first to observe such a law, for the Aegean Islands (he suggested $B \approx 0.5$) [8]. The exponent B is also called the *patchiness exponent*. Recent measurements on 2-d region datasets from diverse applications suggest that usually a similar power law holds [6], with B in the range $[0.5, 0.9]$. In [12] such a law has been tested on several real region datasets, it has been used to estimate the selectivity of range queries for region datasets and it has been shown its relationship with fractals.

We now show that under the realistic assumption that buckets of rectangles in $\mathcal{R}$ obey (3), we can compute accurate estimates on the number of disk accesses, once we are given the patchiness exponent B.

3.3. Theoretical result on a synthetic dataset

Power laws go hand-in-hand with self-similarity and fractals [14], and (3) is no exception. The concept of fractal dimension of a set of spatial data (e.g., points, lines, regions, etc.) is a well-established approach to better describe the inherent structure of the data themselves and to give an order in the complexity of spatial data. Fractals are either exactly or statistically self similar. Exact fractals are generated recursively, by applying a *generator* to an *initiator*. Let g be the number of pieces the generator is decomposed into and let r be the scaling factor; the fractal dimension D_H for a strictly self-similar fractal is defined as

$$D_H = \frac{\log g}{\log(1/r)}. \tag{4}$$

For a straight line, we have $D_H = 1$; for the Koch snowflake (see Figure 1) we have $D_H = \log 4 / \log 3$, slightly higher than 1, that is, it is more rugged than a straight line.

However, fractals like the Koch snowflake consist of a single region. When the generator is decomposed into disconnected pieces, multiple regions are created. These are the so-called $\sigma-$*fractals*. Figure 2 gives a possible $\sigma-$fractal after 2 iterations on the sides of the square with side $s_{\max}$.

It turns out that the following theorem holds:

[1]Remember that the cumulative distribution function of $f(x) : \Re \to \Re$ is defined as $D(x) = \int_{-\infty}^{x} f(t)dt$, while the complementary cumulative distribution function is defined as $C(x) = \int_{x}^{+\infty} f(t)dt$.

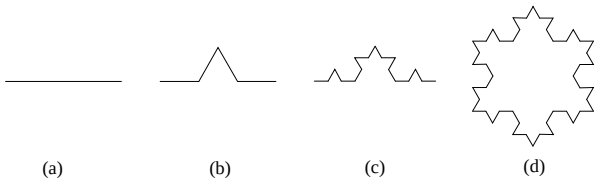


Figure 1. Koch snowflake: initiator (a), generator (b), second iteration (c) and relative Koch snowflake (d)

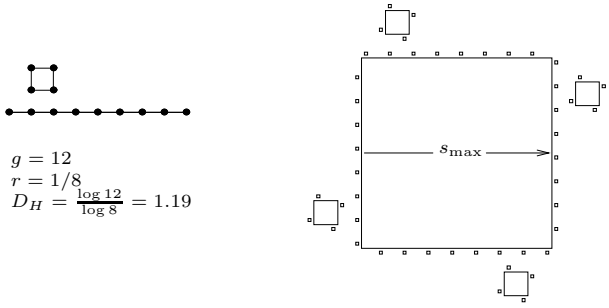


Figure 2. The region generator (left) and the synthetic dataset after two steps of generation (right)

Theorem 1 (Mandelbrot) *For a σ-fractal in a d-dimensional space, we have*

$$B = \frac{D_H}{d}$$

where B is the patchiness exponent of the regions (d-dimensional volumes) and D_H is the fractal dimension of their boundaries.

Proof. See [10]. □

Figure 2 provides a good arithmetic example of the above theorem. In fact, at the beginning, we have a square of side $s_{\max}$ and area $a_{\max} = s_{\max}^2$, to which the generator is recursively applied. Then, at the first stage of generation, we create 4 regions of side $r \cdot s_{\max}$. At the second stage we create $4 \cdot g$ regions of side $r^2 \cdot s_{\max}$. In general, at the k-th stage, we create $4 \cdot g^{k-1}$ regions of side $r^k \cdot s_{\max}$. Hence, as the side s is multiplied by r, the CCDF $C(s)$ of the number of regions of side at least s is multiplied by g. From Eq. (4), we have then

$$C(s) = k \cdot s^{-D_H} \tag{5}$$

in which the crucial exponent is D_H, and being $s^2 = a$

$$C(a) = k' \cdot a^{-B} \tag{6}$$

where $B = \frac{1}{2} D_H$. Interestingly, this result is independent of g and of the region contour roughness; moreover, it extends to the case when the generator involves two or more regions. Given the inherent self-similarity in real datasets, the above relationship, to some extent, holds for real datasets too. Our experiments on region datasets scattered around the world and previous studies [6] confirm that the law strongly holds for lakes, archipelagoes, vegetative ecosystems, urban areas and many others.

Let us now see how the synthetic dataset can be used to predict the performances of an R-tree used to store it. We start by proving that the number of neighbors of any island in the synthetic dataset follows a power law with exponent $D_H - 1$. Without loss of generality, let be $s_{\max} = 1$ and assume that the distance between the two pieces the generator is decomposed into is r. With these assumptions, we have that after $q \geq 1$ iterations, within a distance

$$\rho_q = 2 \cdot (r + \ldots + r^q) = 2 \cdot \frac{r - r^q}{1 - r} \tag{7}$$

from the biggest island, exactly

$$T_0(\rho_q) = 4 + 4g + \ldots 4g^{q-1} = 4 \cdot \frac{g^q - 1}{g - 1} \approx 4 \cdot \frac{g^q}{g - 1} \tag{8}$$

islands are included. Now, if we shrink the distance by a factor of $1/r$, we have that within a distance $\rho'_q = r \cdot \rho_q$ from the biggest island, exactly

$$T_0(\rho'_q) = T_0(\rho_q) - 4 \cdot (1 + 4 + 4g + \ldots 4g^{q-2}) \approx$$

$$\approx 4 \cdot \frac{g^{q-1} \cdot (g - 4)}{g - 1} = \frac{T_0(\rho_q)}{g \cdot r} \tag{9}$$

islands are enclosed. This means, as the distance ρ from the biggest island is multiplied by a factor of $1/r$, the number of neighbors $T_0(\rho)$ is multiplied by a factor of $g \cdot r$. In other words

$$T_0(\rho) = k_0 \cdot \rho^{\frac{\log g \cdot r}{\log 1/r}} = k_0 \cdot \rho^{D_H - 1} \tag{10}$$

where k_0 is a constant. Since after q steps, $4 \cdot r^{1-q}$ islands surrounding the biggest island at a distance $(2r)^q$ are created, we have from (10) that

$$4 \cdot r^{1-q} = k_0 \cdot (2r)^{q \cdot (D_H - 1)} \tag{11}$$

and therefore, expliciting k_0, it follows that (10) can be rewritten as

$$T_0(\rho) = \frac{4 \cdot r^{1-q}}{(2r)^{q \cdot (D_H - 1)}} \cdot \rho^{D_H - 1}. \tag{12}$$

Let now $T_i(\rho)$ denote the number of neighbors of a level-i island (i.e., an island generated after the i-th iteration). Using the same arguments as above, we have that

$$T_i(\rho) = k_i \cdot \rho^{D_H - 1} \quad 0 \leq \rho \leq \rho_i \tag{13}$$

where

$$\rho_i = 2 \cdot (r + \ldots + r^{q-i}). \tag{14}$$

Since it must be

$$T_i((2r)^q) \equiv 4 \cdot r^{i+1-q} = k_i \cdot (2r)^{q \cdot (D_H - 1)} \tag{15}$$

it follows from (11) that

$$k_i = r^i \cdot k_0 = \sqrt{a_i} \cdot k_0 \tag{16}$$

where a_i denotes the area of the level-i island. Hence, from (15,16), to enclose exactly M neighbors, a level-i island must be inflated of a radius

$$\Delta_i = \left(\frac{M}{\sqrt{a_i} \cdot k_0} \right)^{\frac{1}{D_H - 1}}. \tag{17}$$

This means that the area of the inflated island is

$$\tilde{a_i} = (\sqrt{a_i} + 2\Delta_i)^2 = \left(r^i + 2 \cdot \left(\frac{M}{r^i \cdot k_0} \right)^{\frac{1}{D_H - 1}} \right)^2. \tag{18}$$

To understand how the CCDF of the inflated areas will look like, let's plot it in a log-log diagram against the CCDF of the areas of the synthetic archipelago, setting $M = 64$ and stopping the generation process after $q = 8$ steps. Note that the inflation of the level-8 islands is not considered, since we assume these islands are absorbed by the largest ones. Results are shown in Figure 3.

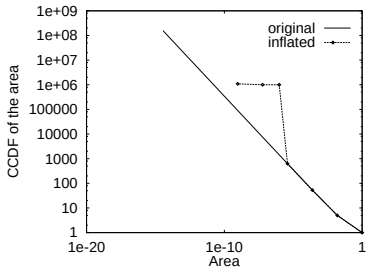


Figure 3. The CCDF of the area for the synthetic archipelago (solid line) against the inflated archipelago (dotted line).

Few words need to be spent about the plots. The first observation is that the two functions almost coincide. This is essentially due to the fact that the inflation of the largest islands is small. The only meaningful difference arises for the smallest islands, where the CCDF for the inflated archipelago has a tilting. However, this deviation, given the log-scale, only interests a limited portion of the function. Thus, from each practical point of view, we can conclude that the inflated archipelago follows almost perfectly the REGAL law, with the same patchiness exponent as for the original archipelago. The only thing we really need to pay attention when concerned with R-trees storing real datasets is the fact that the inflation of the largest objects will not be infinitesimal, partially because of the fact that packing in the R-tree is not optimal and partially because a real dataset, even if obeying to the REGAL law, generally has a complicated structure (namely, the biggest object is not surrounded by plenty of very small objects, as for the synthetic dataset). Therefore, if we want to give good estimations on the selectivity, we need to check the inflation of the biggest object, level by level. Once this is done, we can make use of the REGAL law to describe the CCDF of the areas of all the nodes of the R-tree on that level.

Summarizing, since the CCDF of the inflated areas must follow (3), we have that

$$\tilde{C}(\tilde{a}) = \tilde{k} \cdot \tilde{a}^{-B}. \tag{19}$$

To determine the constant $\tilde{k}$, we make use of the boundary condition $1 \equiv \tilde{C}(\tilde{a}_{\max}) = \tilde{k} \cdot \tilde{a}_{\max}^{-B}$ where $\tilde{a}_{\max}$ is the area of the largest inflated island. Therefore, (19) can be finally written as

$$\tilde{C}(\tilde{a}) = \tilde{a}_{\max}^{B} \cdot \tilde{a}^{-B}. \tag{20}$$

3.4. The main result

In this section, we make use of (20) to predict the I/O complexity of range queries posed on region data stored using an R-tree. Let M denote the maximum number of rectangles per node, and let $\overline{M}$ be the effective node utilization (typically, $\overline{M} = 0.7 \cdot M$). The number of leaf nodes is $N_0 = n/\overline{M}$, while the number of nodes at level $0 < j \leq m$, where leaves are at level 0 and the root is at level m, is $N_j = n/\overline{M}^{j+1}$. Let A_j be the area of all level-j nodes in the R-tree. We are now ready to prove the following:

Theorem 2 *Given a set $\mathcal{R} = \{r_1, r_2, \ldots, r_n\}$ of rectangles embedded in $U = [0,1]^2$ whose areas obey to the REGAL law, having a fixed given aspect ratio σ between width and height and a patchiness exponent B, the number of disk accesses for a rectangular window query $\vec{q} = (q_x, q_y)$ is*

$$DA(\mathcal{R}, \vec{q}) = \sum_{j=0}^{m} \alpha_j \cdot \frac{\left(\frac{n}{\overline{M}^{j+1}}\right)^{\left(1-\frac{1}{B}\right)} - 1}{1 - \frac{1}{B}} + \Big(q_x \cdot \sqrt{\frac{1}{\sigma}} +$$

$$+ q_y \cdot \sqrt{\sigma}\Big) \cdot \sum_{j=0}^{m} \alpha_j \cdot \frac{\left(\frac{n}{\overline{M}^{j+1}}\right)^{\left(1-\frac{1}{2B}\right)} - 1}{1 - \frac{1}{2B}} + q_x \cdot q_y \cdot N \tag{21}$$

where α_j is the area of the largest rectangle of the j-th level of the R-tree, $0 \leq j \leq m$.

Proof. We start with (2). We need to estimate A_{tot}, W_{tot} and H_{tot}. Let $C_j(a)$ be the CCDF for the level-j nodes of the R-tree. By assumption, $C_j(a)$ obeys almost perfectly to the REGAL law (20). As usual, we have that $1 \equiv C_j(\alpha_j) = k \cdot {\alpha_j}^{-B}$, where α_j is the area of the largest level-j rectangle. Therefore, it follows that

$$C_j(a) = {\alpha_j}^{B} \cdot a^{-B}.$$

From the inverse relation, we have

$$a(C_j) = \left(\frac{1}{{\alpha_j}^{B}} \cdot C\right)^{-\frac{1}{B}}.$$

Therefore, if a_{j_i} denotes the area of the i-th rectangle of the j-th R-tree level, it follows

$$A_j = \sum_{i=1}^{N_j} a_{j_i} \approx \alpha_j \int_1^{N_j} C_j^{-\frac{1}{B}}\, dC_j =$$

$$= \alpha_j \cdot \frac{\left(\frac{n}{\overline{M}^{j+1}}\right)^{\left(1-\frac{1}{B}\right)} - 1}{1 - \frac{1}{B}}. \tag{22}$$

Let now $CW_j(w)$ denote the number of level-j nodes having width at least w. Since the rectangles in $\mathcal{R}$ are similar, we assume that such a similarity propagates towards the upper levels of the R-tree. Therefore, for any R-tree rectangle of area a and width w, we have $a = \frac{1}{\sigma} w^2$. Then, from (3), $CW_j(w)$ obeys the following power law

$$CW_j(w) = (\sigma \cdot \alpha_j)^{B} \cdot w^{-2B}. \tag{23}$$

Denoting with

$$\omega_j = \sqrt{\sigma \cdot \alpha_j} \tag{24}$$

the width of the largest level-j node, (23) can be written

$$CW_j(w) = {\omega_j}^{2B} \cdot w^{-2B}.$$

Hence, from the inverse relation we have

$$w(CW_j) = \left(\frac{1}{{\omega_j}^{2B}} \cdot CW_j \right)^{-\frac{1}{2B}} = \omega_j \cdot {CW_j}^{-\frac{1}{2B}}.$$

Therefore, if $w_{j\,i}$ denotes the width of the i-th rectangle of the j-th R-tree level, it follows

$$W_j = \sum_{i=1}^{N_j} w_{j\,i} \approx \omega_j \int_1^{N_j} {CW_j}^{-\frac{1}{2B}}\, dCW_j =$$

$$= \omega_j \cdot \frac{\left(\frac{n}{M^{j+1}} \right)^{\left(1-\frac{1}{2B}\right)} - 1}{1 - \frac{1}{2B}}. \qquad (25)$$

Using similar arguments

$$H_j = \eta_j \cdot \frac{\left(\frac{n}{M^{j+1}} \right)^{\left(1-\frac{1}{2B}\right)} - 1}{1 - \frac{1}{2B}} \qquad (26)$$

where

$$\eta_j = \sqrt{\frac{\alpha_j}{\sigma}} \qquad (27)$$

is the height of the largest level-j node. Therefore, from (2, 22, 24, 25, 26 27) the thesis follows. □

As we show next, the above theorem will provide a good estimation for window selectivity on real region datasets, as soon as we are able to provide the area of the largest level-j rectangle.

4. Experiments on real datasets

To assess experimentally the accuracy of our analysis, we have used three different region datasets, that is:

- The Scandinavian Lakes (LAKES), available at `http://mapweb.parc.xerox.com/map/nogrid` (Xerox PARC Map Viewer) and consisting of 810 lakes.
- The Indonesia Archipelago (ISLANDS), available at `http://mapweb.parc.xerox.com/map/nogrid`, and consisting of 470 islands.
- A population density map of Europe (REGIONS). This map has been created starting from a population density map from a World Atlas. Each grid cell is turned to black if it has density above a threshold, namely 30 inhabitants/Km^2. It consists of 757 regions.

We also used three additional datasets: the Queen Elizabeth Islands (77 islands) and the Japan Archipelago (186 islands), both available at `http://mapweb.parc.xerox.com/map/nogrid`, and a map of Italy agricultural plains (228 regions), created starting from a geographic map from a World Atlas and turning to black a grid cell whenever it is at most 50 meters above the sea level. We do not give details about these datasets since the results were similar.

In the following subsections we present results for: (a) verifying that the R-tree nodes obey to the REGAL law (3), with patchiness exponent approximately equal to that of the regions themselves; (b) verifying the accuracy of our formula (21), once α_j is provided.

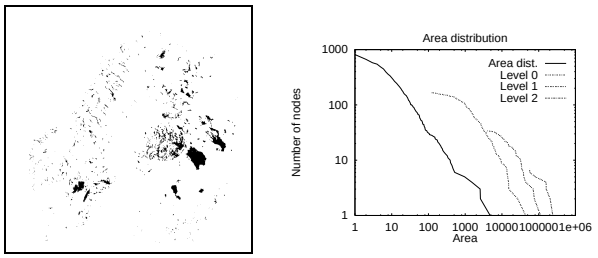
4.1. Verifying the REGAL law for the R-tree nodes

All the datasets were stored using 1024×1024 bitmaps, as shown in Figure 4a-c. Preliminary, we have identified all the regions and their MBRs in each dataset. Then, we have computed all the relevant features needed for checking our results. Note that to estimate B, we have computed the CCDF of the MBRs area for each dataset and we have interpolated the plotted points with a straight line using the classic least-square method. Note also that σ has been computed by averaging over all the MBRs' aspect ratios. After, we have stored the datasets using the *deferred-split R-tree* (DR-tree) [1], setting $M = 8$. All the resulting R-trees consist of three levels. Obtained data are summarized in Table 2.

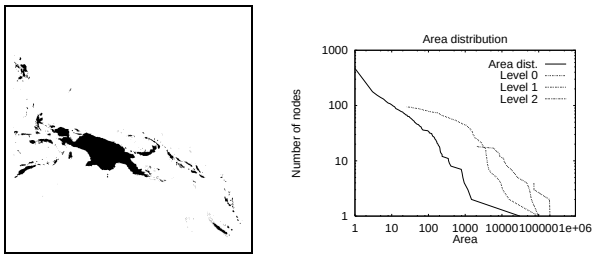
Feature	LAKES	ISLANDS	REGIONS
n	810	470	757
B	0.85	0.60	0.70
N	208	118	182
σ	1.13	1.98	0.53
α_0	43,925	99,008	49,364
α_1	112,158	99,008	100,098
α_2	241,776	202,620	195,714
A_{tot}	2,264,047	1,229,336	1,771,251
W_{tot}	18,118	6,359	10,341
H_{tot}	14,758	10,516	18,030

Table 2. Datasets features.

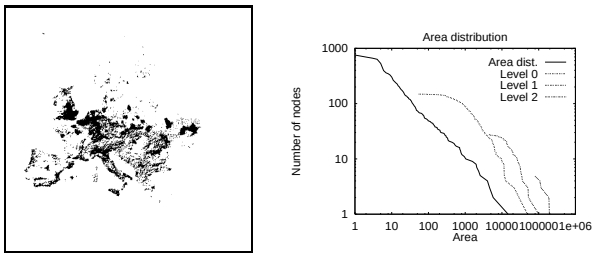
Figure 4a-c shows in a log-log diagram the resulting CCDFs. It is impressive that for all three datasets, independently of the R-tree level, the CCDF of the areas obey almost perfectly to (3). As anticipated, there is a shifting of the plots towards right as the level increases, since the largest island tends to have a not negligible inflation.



(a) LAKES



(b) ISLANDS



(c) REGIONS

Figure 4. Used datasets: (a) LAKES; (b) ISLANDS; (c) REGIONS, together with their patchiness plots log(count) *vs* log(area) for the R-tree nodes.

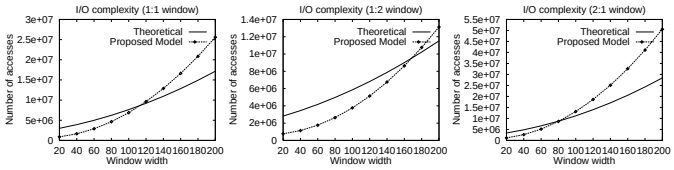
4.2. Accuracy of our I/O complexity estimation

To ascertain the accuracy of our formula (21), for each dataset we have initially computed the theoretical I/O complexity[2] using (1). After, we applied (21), for query windows of width $20 \cdot i, i = 1, \ldots 10$ and having three different aspect ratios: 1:1 (square), 1:2 and 2:1, so that the most usual window sizes and shapes are considered. Figure 5 shows the number of accesses estimated using our approach for the LAKES, ISLANDS and REGIONS dataset, respectively, as compared to the theoretical value (1). Finally, following the recommendations from statistics, we have also computed the *geometric average* of relative errors, for each dataset and for each different window aspect ratio, summarized in Table 3. Note that, for each dataset, our approach is usually around 25% far from the reality.

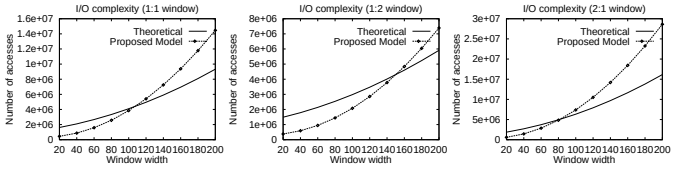
[2]Of course, all the computations have been normalized to the 1024×1024 image space.

Geometric average relative error (%)			
Dataset \ Ratio	1:1	1:2	2:1
LAKES	26.95	23.40	30.47
ISLANDS	29.07	24.46	31.85
REGIONS	28.21	23.62	33.33

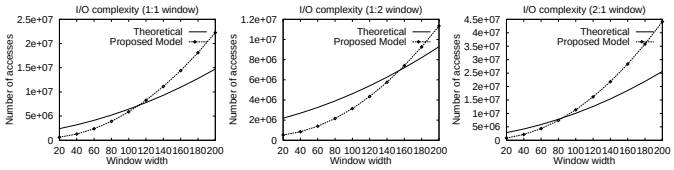
Table 3. Geometric average relative error (%) in estimating $DA(\mathcal{R}, \vec{q})$ of the proposed method (REGAL), for each dataset and for each aspect ratio of the query window.



(a) LAKES



(b) ISLANDS



(c) REGIONS

Figure 5. Number of accesses *vs* query window width, for square, 1:2 and 2:1 window queries (from left to right), for (a) LAKES, (b) ISLANDS and (c) REGIONS datasets.

4.3. Directions for a practitioner: fast estimation of B and α_j

The final question is: how can a practitioner benefit of our analysis? We have solid answers to this question. Up to now, no methods were provided to estimate R-tree performances in answering to range queries. Thus, to make use of developed formulas [7, 11], one was required to compute the total width and length extent of all the nodes in the R-tree. This can be done by scanning the entire index storing the data, that is, it is very time consuming.

On the contrary, the patchiness exponent B, the ratio σ and the maximum area α_j of the level-j nodes can be computed quickly. Concerning B, we suggest two possible fast ways to compute it, both of them based on sampling. The first makes use of the representing bitmap, while the second works on the database storing the data:

1. Focus on a subwindow of size $t \times t$ of the bitmap, extract the boundaries of the objects contained in it and apply the $O(t \log t)$ time algorithm [2] to compute their fractal dimension D_H. Assuming that regions are self-similar (and then subwindows of the bitmap are similar to the whole) and applying Theorem 1, we can conclude that $B = D_H/2$ is a good approximation for the real B of all the map.

2. Focus on a subwindow of the bitmap, retrieve from the database all the objects contained in it and compute the CCDF of their areas. Then, plot the obtained points in a log-log diagram and interpolate them with a straight line using the classic least-square method. The negated slope of such a line corresponds to the patchiness exponent of the subset of objects. Once again, assuming that regions are self-similar, we can be confident that such exponent is representative for the whole dataset.

Concerning σ, a robust solution is to average the aspect ratios over a small number of regions. Finally, to compute α_j for every R-tree level, it suffices to analyze the path from the root to the leaf node containing the largest object in the R-tree: in this way, we are not fully guaranteed to retrieve the largest node on each level, but most likely this will happen (for our experiments, it always happened).

Therefore we conclude that our analysis is suitable in practice and contributes to the solution to the problem of I/O complexity evaluation for range queries posed on R-trees storing region data.

5. Conclusions

The main contribution of this paper is the first tentative to estimate the I/O complexity for answering to range queries posed on region data stored in an R-tree. This has been done by studying how the area distribution of the regions in the underlying dataset propagates towards the upper levels of the R-tree.

We showed that very few measures are needed (essentially the average aspect ratio, the patchiness exponent and the largest node on each R-tree level), to achieve accurate results. Our experiments on diverse, real datasets, such as archipelagoes, city regions, plain maps, hydro-graphic systems and many others showed that our approach achieves estimates 30% close to the reality.

Promising future directions include the use of σ-fractals to study I/O complexity of additional query types (nearest neighbor, spatial join, etc.).

References

[1] N. Beckmann, H. Kriegel, R. Schneider, and B.Seeger. The R*-tree: an efficient and robust access method for points and rectangles. In *Proc. of the 9th ACM-SIGMOD Symposium on Principles of Database Systems*, pages 322–331, Nashville, TN, 1990.

[2] A. Belussi and C. Faloutsos. Estimating the selectivity of spatial queries using the 'correlation' fractal dimension. In *Proc. of the 21st VLDB Conference*, pages 299–310, Zurich, Switzerland, 1995.

[3] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *Proc. ACM SIGMOD*, pages 419–429, Minneapolis, MN, May 1994.

[4] C. Faloutsos, T. Sellis, and N. Roussopoulos. Analysis of object oriented spatial access methods. In *Proc. of the 6th ACM-SIGMOD Symposium on Principles of Database Systems*, pages 426–439, San Francisco, CA, 1987.

[5] A. Guttman. R-trees: a dynamic index structure for spatial searching. In *Proc. of the 3th ACM-SIGMOD Conference*, pages 47–57, Boston, MA, 1984.

[6] H. Hastings and G. Sugihara. *Fractals*. Oxford Science Publications, 1993.

[7] I. Kamel and C. Faloutsos. On packing R-trees. In *Proc. of the 2nd ACM Intl. Conf. on Information and Knowledge Management*, pages 490–499, Washington, DC, 1993.

[8] J. Korcak. Deux types fondamentaux de distribution statistique. *Bull. de l'Institute International de statistique*, 3:295–299, 1938.

[9] W. Lu and J. Han. Information associated join index for spatial range search. *Int. Journal of Geographical Information Systems*, 9(3):221–249, 1995.

[10] B. Mandelbrot. *The fractal geometry of nature*. W.H. Freeman and Company, 1982.

[11] B. Pagel, H. Six, H. Toben, and P. Widmayer. Towards an analysis of range query performances. In *Proc. of the ACM-SIGMOD Symposium on Principles of Database Systems*, pages 214–221, Washington, DC, 1993.

[12] G. Proietti and C. Faloutsos. Accurate modeling of region data. Technical Report CMU-TR-98-126, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1998.

[13] B. Salzberg. Access methods. In *The Computer Science and Engineering Handbook*, pages 1012–1037, 1997.

[14] M. Schroeder. *Fractals, Chaos, Power Laws: Minutes From an Infinite Paradise*. W.H. Freeman and Company, New York, 1991.

[15] T. Sellis, N. Roussopoulos, and C. Faloutsos. The R^+-tree: a dynamic index for multidimensional objects. In *Proc. of the 13rd VLDB Conference*, pages 507–518, England, 1987.

[16] Y. Theodoridis and T. Sellis. A model for the prediction of r-tree performance. In *Proc. of the 15th ACM Symposium on Principles of Database Systems (PODS'96)*, pages 161–171, Montreal, Canada, 1996.