A Systematic Review of Spatial Approximations in Spatial Database Systems

Pedro Gabriel Kohl Bertella[1], Yuri Kaszubowski Lopes[2], Rafael Alves Paes de Oliveira[1], Anderson Chaves Carniel[3]

[1] Federal University of Technology - Paraná, Brazil
pedrobertella@alunos.utfpr.edu.br, raoliveira@utfpr.edu.br
[2] Santa Catarina State University, Brazil
yuri.lopes@udesc.br
[3] Federal University of São Carlos, Brazil
accarniel@ufscar.br

Abstract. Many applications rely on spatial information retrieval, which involves costly computational geometric algorithms to process spatial queries. Spatial approximations simplify the geometric shape of complex spatial objects, allowing faster spatial queries at the expense of result accuracy. In this sense, spatial approximations have been employed to efficiently reduce the number of objects under consideration, followed by a refinement step to restore accuracy. For instance, spatial index structures employ spatial approximations to organize spatial objects in hierarchical structures (e.g., the R-tree). It leads to the interest in studying how spatial approximations can be efficiently employed to improve spatial query processing. This article presents a systematic review on this topic. We gather relevant studies by performing a search string on several digital libraries. We further expand the studies under consideration by employing a single iteration of the snowballing approach, where we track the reference list of selected papers. As a result, we provide an overview and comparison of existing approaches that propose, evaluate, or make use of spatial approximations to optimize the performance of spatial queries. The spatial approximations mentioned by the approaches are also summarized. Further, we characterize the approaches and discuss some future trends.

Categories and Subject Descriptors: A.1 [**General Literature**]: Introductory and Survey; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.2.8 [**Database Management**]: Database Applications—*Spatial databases and GIS*

Keywords: GIS, spatial approximation, spatial database systems, spatial information retrieval, spatial query processing

1. INTRODUCTION

Modern and advanced applications have increasingly required the representation, storage, and management of *spatial information*. These applications enrich data handling and analysis by modeling geographical and spatial phenomena as instances of *spatial data types* (e.g., points, lines, and regions) typically provided by spatial database systems and Geographical Information Systems (GIS) [Güting 1994]. Examples of applications handling spatial objects include agricultural applications for monitoring land use and land cover changes [Esquerdo et al. 2020], location-based services for delivering information based on the location of users [Jensen et al. 2004; Huang et al. 2018], and spatial data science applications for revealing useful information from spatial data [Anselin 2020; Carniel and Schneider 2021].

Spatial information retrieval is a common task of these applications and consists of capturing spatial objects through *spatial queries*. The conditions of a spatial query are often expressed as spatial relationships, such as *topological relationships* (e.g., overlaps, contains) [Egenhofer and Herring 1994;

Schneider and Behr 2006; Shen et al. 2018]. Hence, several types of spatial queries have been defined in the literature, such as the window query, containment query, and point query (see [Carniel 2020] for a survey).

Spatial approximations play an important role in the challenging task of reducing the elapsed time required to process a spatial query. Due to the high complexity of the geometric format of spatial objects (e.g., the number of points of a region or line object), spatial objects are approximated to a simpler geometric shape to reduce the complexity of the computation of spatial relationships. For instance, the Minimum Bounding Rectangle (MBR) [Papadias et al. 1995] is a very popular approximation employed in the literature.

On the other hand, the use of spatial approximations introduces at least another needed step to process spatial queries. This is because the result of a spatial query answered by spatial approximations is not fully accurate. In this case, the needed step called *refinement* has to evaluate the spatial relationship by accessing the original spatial object. This ensures the correct result of the spatial query; but it consists of computing costly geometric algorithms. A common goal is to reduce the number of objects to be checked in the refinement step by using *spatial index structures* built on a specific approximation. A typical index is the R-tree [Guttman 1984] and its variants like the R*-tree [Beckmann et al. 1990], which are based on MBRs.

Due to the importance of spatial information retrieval, several studies in the literature have been focused on optimizing the indexing of spatial objects (e.g., see [Gaede and Günther 1998] for a survey). However, for many types of queries (e.g., window queries), the step processed by the spatial index requires a small elapsed time if compared to the total elapsed time to deliver the final result of the spatial query (as pointed out in Section 3.2). This motivates the development of a multi-step spatial query processing that combines different types of approximations to reduce the cost of executing the refinement step. However, there is a lack of recent studies in the literature that compare, understand, and show the applicability of spatial approximations.

This article fulfills this gap by presenting a comprehensive study on the existing spatial approximations in the literature. Our contributions include the following topics:

—A systematic review of the literature by employing a reproducible methodology that can serve as a foundation for future trends on spatial approximations.

—An overview and comparison of existing approaches. We classify them as approaches that propose, evaluate, or make use of spatial approximations. Further, we summarize the main characteristic of the spatial approximations employed by the approaches.

—A characterization of existing approaches that allows us to understand how they use a spatial approximation to optimize spatial queries. Further, we present future trends in the area of spatial query processing.

This article is organized as follows. Section 2 discusses related work. Section 3 summarizes needed basic concepts. Section 4 introduces our systematic review and describes how existing approaches employ spatial approximations to spatial query processing. Section 5 analyzes existing implementations of spatial approximations and discusses open challenges and future trends. Section 6 concludes the article and presents future work.

2. RELATED WORK

In this article, we are interested in studying the role of spatial approximations in processing spatial queries. Another important related strategy to improve spatial query processing is the use of spatial index structures. Hence, in this section, we present an overview of such structures and discuss related surveys.

A spatial index, also called *spatial access method*, provides a specialized data structure and related algorithms to improve the retrieval of spatial objects according to the constraints imposed by spatial queries. Commonly, the underlying design of a spatial index is a hierarchy based on data partitioning or space partitioning. The first partitioning strategy organizes tree structures by using the geometric format of the spatial objects. The classical R-tree [Guttman 1984] and its variants like the R*-tree [Beckmann et al. 1990] and the revised R*-tree [Beckmann and Seeger 2009] are examples of indices that employ this strategy. The second strategy divides the space to accommodate the spatial objects. Examples include Quadtrees and related structures [Samet 1984]. There are also spatial index structures that combine both strategies, such as the Hilbert R-tree [Kamel and Faloutsos 1994]. There is a strong relationship between several types of spatial index structures and spatial approximations. Indexing spatial objects containing a different number of points (e.g., lines and regions) imposes the challenge of manipulating index pages (i.e., nodes in a tree) with variable sizes. On the other hand, spatial approximations allow the representation of complex objects by using a fixed-size data structure. Hence, several index structures (e.g., the R-tree) actually handle approximations and keep pointers that provide direct access to the original object.

Several surveys compare and summarize different types of spatial index structures. For instance, the authors in [Gaede and Günther 1998] discuss a variety of these structures designed for main memory and secondary memory. They also highlight the role of spatial index structures in a multi-step query processing strategy to answer different types of queries (as discussed in Section 3.2). Another related survey that compares other spatial index structures is presented in [Oosterom 2005]. Due to the importance of spatial joins in spatial database systems and GIS, other surveys [Jacox and Samet 2007; Bouros and Mamoulis 2019] concentrate on describing specific algorithms that possibly use spatial index structures to answer this type of spatial query. Other surveys are dedicated to describing spatial index structures applied to modern storage devices [Fevgas et al. 2020], spatial analytics systems based on Hadoop and Spark [Castro et al. 2020], and existing ecosystems for spatial data handling [Alam et al. 2021].

This article differs from the aforementioned studies since we present a systematic review of spatial approximations. The goal is to discuss how they have been used in the literature to improve spatial query processing. To the best of our knowledge, this article is the first work in the literature with this goal. This article extends our previous work [Bertella et al. 2021] as follows:

—We present additional descriptions of underlying concepts of the article, such as related studies, types of spatial queries, and an overview of multi-step query processing.

—We conduct an extended version of our previous systematic review by using a *snowballing approach* [Wohlin 2014]. For this, we have updated the previous systematic review and, after identifying the selected studies, we have used the reference list of these studies as input for another iteration of the systematic review.

—We identify more relevant studies in the context of spatial approximations that are compared and discussed in the article. Further, we provide an overview of the approximations employed by studies identified in our systematic review.

—We extend the discussion of the challenges and future trends.

3. NEEDED CONCEPTS FROM SPATIAL DATABASE SYSTEMS AND GIS

In this section, we sketch definitions related to spatial queries (Section 3.1) and discuss how they are processed by using two steps (Section 3.2).

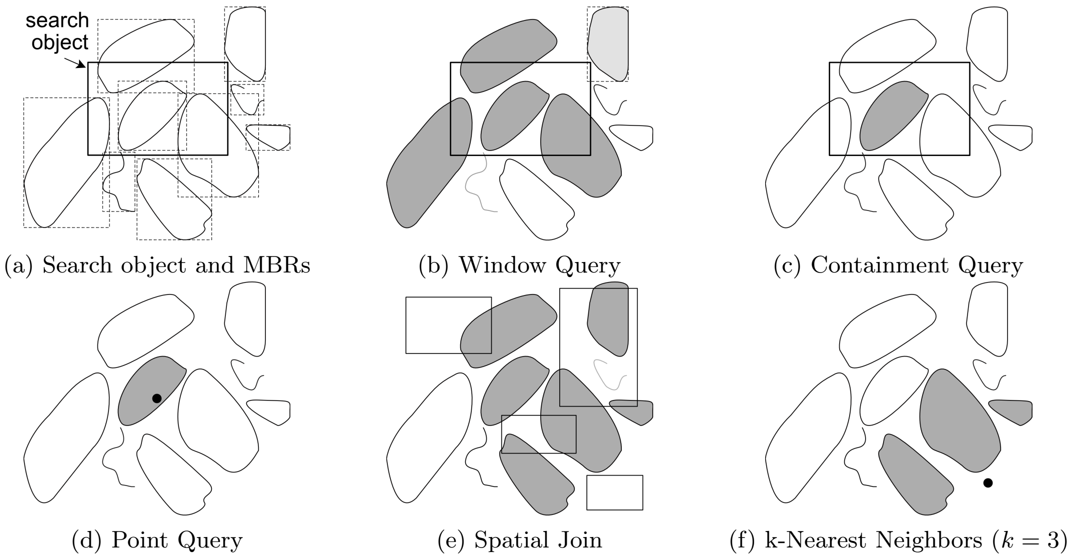


Fig. 1. The role of MBRs and examples of spatial queries. The final answer of each spatial query is shown in dark gray. In (b), the false candidate is depicted in light gray. In (e), the result indicates the intersecting pairs between the set of spatial objects and the set of rectangles. In (d) and (f), the point object represents the search object.

3.1 Spatial Data Types and Spatial Query Processing

Spatial database systems and GIS store, query, and handle spatial or geographic information represented by homogeneous geometries in the Euclidean space. For modeling them, formal definitions of *spatial data types* for *points*, *lines*, and *regions* have been provided in the literature [Schneider and Behr 2006]. Instances of these data types can be simple or complex where complex objects may contain finitely many components.

Commonly, spatial objects are handled by spatial operations, such as spatial relationships (e.g., topological relationships), geometric set operations (e.g., union, intersection), and metric operations (e.g., distance, area). Topological relationships (e.g., intersect, overlap, inside) are widely studied in the literature and determine how two or more spatial objects are related or connected in space. The evaluation of topological relationships can be performed by employing the 9-intersection model and variants [Egenhofer and Herring 1994; Schneider and Behr 2006]. They leverage point sets and point set topology to create a matrix from the nine possible intersections of the boundary, interior, and exterior of two spatial objects, considering the invariant criteria of emptiness and non-emptiness. However, this processing is costly since it involves the computation of complex geometric algorithms, especially for line and region objects that may vary in the number of points and format.

In this sense, spatial approximations play an important role. A spatial approximation represents a spatial object by using a simpler geometric format, such as rectangles or circles (more approximations are introduced in Section 4.2). For instance, a complex region containing several points can be approximated to a rectangle that minimally encompasses its boundary (i.e., an MBR) by storing only four numbers in the two-dimensional case. The computation of topological relationships on the simple geometric format provided by spatial approximations is a quick operation [Papadias et al. 1995]. Hence, spatial approximations are widely employed as a first step when processing spatial queries, as shown in Section 3.2.

As a basic feature, spatial database systems and GIS allow users to pose different types of spatial queries [Gaede and Günther 1998; Carniel 2020]. Examples are depicted in Figure 1. Given a search object, a spatial query has the goal of retrieving all objects that satisfy a set of conditions, usually

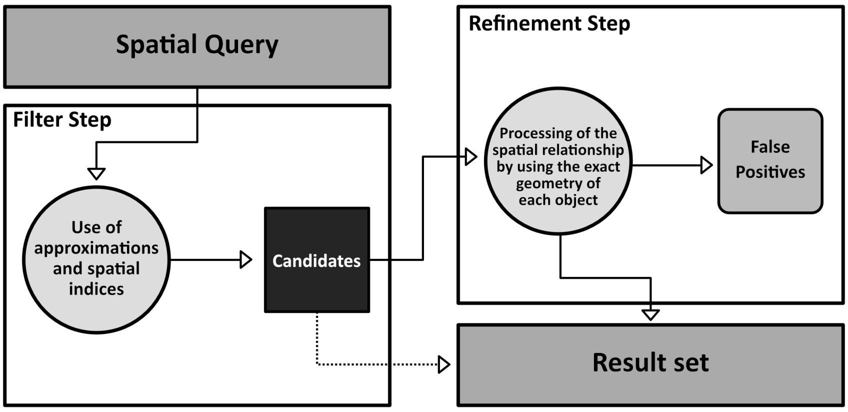


Fig. 2. The steps for processing spatial queries.

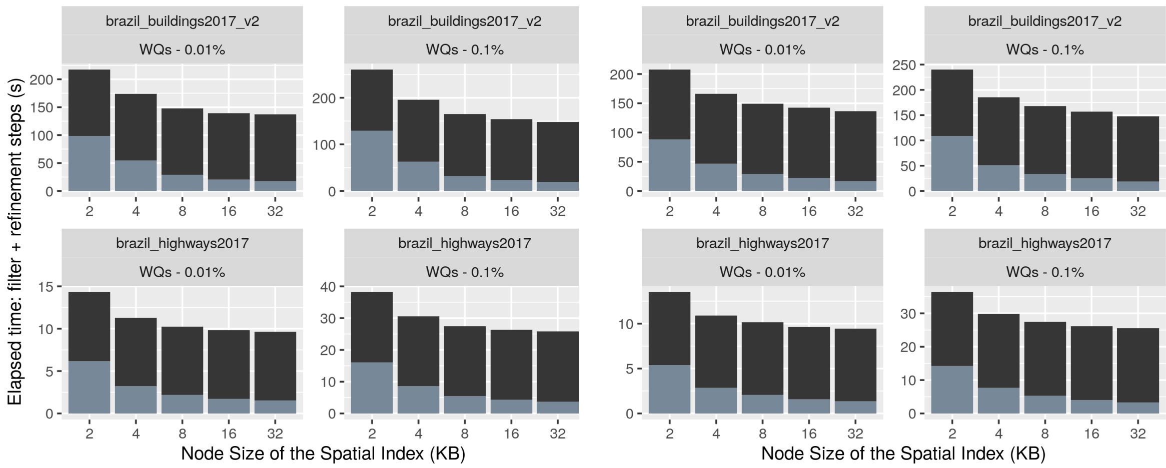
expressed by using topological relationships or distance functions. Figure 1a shows a rectangular-shaped search object (also known as *window*) and a set of spatial objects with their corresponding MBRs. By using the topological relationship *intersect*, we define a *window query* (WQ), which finds all spatial objects intersecting the search object (Figure 1b). A variant of this query is the *containment query* (CQ), which finds all objects that are *inside* the window (Figure 1c). If we consider that the search object is a point object, we design a *point query*. It returns all objects that have this particular simple point object (Figure 1d). In addition, two spatial datasets can be associated according to a particular topological relationship, leading to *spatial joins* (Figure 1e). Other typical spatial queries use the distance notion. For instance, the *k-nearest neighbor query* (kNN) finds a set of spatial objects that are the k-nearest neighbors of the search object (Figure 1f). Other types of spatial queries can be derived, as discussed in [Gaede and Günther 1998; Carniel 2020].

Some types of spatial queries can be directly answered by using spatial approximations only, such as the CQ (Figure 1c). However, this is not the case for other types of spatial queries, such as the WQ (Figure 1b). This is due to the introduction of an area that does not belong to the original spatial object, called *dead space*. When checking the intersection on approximations only, the result may contain spatial objects that do not belong to the final answer, as shown in Figure 1b. On the other hand, approximations avoid the computation of those objects that certainly do not belong to the result. Due to this and the fixed size to store an approximation, *spatial index structures* [Gaede and Günther 1998] often employ approximations to reduce the search space of spatial queries.

3.2 Filter and Refinement Steps

Figure 2 shows the two steps often employed to process a spatial query [Gaede and Günther 1998]. The first step called *filter* uses approximations (commonly in a spatial index structure) to get a set of candidates of spatial objects that answer the query. The next step called *refinement* gets these candidates as input and accesses their original geometric format to evaluate the spatial relationship (e.g., topological relationship) of the query and return the final answer of the query. This step usually requires more processing time than the filter step. Hence, there is an interest in reducing the number of objects that are evaluated by the refinement step.

Several approaches focus on improving the elapsed time of spatial index structures (i.e., filter step) (e.g., [Gaede and Günther 1998; Carniel et al. 2020]). However, the use of a spatial index is not sufficient to answer many types of spatial queries. In such cases, the refinement step requires more processing time than the filter step because of the number of objects evaluated by costly computational geometric algorithms. This is also supported by our experiments, as follows.



(a) R-trees

(b) R*-trees

Fig. 3. Results of our motivating experiments using two different spatial index structures. Since these structures are not sufficient to accurately answer WQs, the refinement step is required. The stacked bars show the elapsed times of the filter (in light gray) and refinement (in dark gray) steps. Note the different scales in the plots.

We used two real spatial datasets described in [Carniel et al. 2017][1]: (i) *brazil_buildings2017_v2*, containing 1,485,866 region objects, and (ii) *brazil_highways2017*, containing 2,644,432 line objects. We evaluated the execution of 200 WQs. Two different sets of 100 windows were used in the WQs. The first set had 0.01% of the area of the total extent of Brazil. The second set had 0.1%, increasing the number of objects returned in the WQs. In the filter step, we used two distinct spatial indices that make use of MBRs: (i) the *R-tree* with the linear split algorithm, and (ii) the *R*-tree* with 30% of reinsertion by using the close reinsert policy. They were built under different page sizes, from 2KB to 32KB, employing an LRU buffer of 512KB. The refinement step used the GEOS library, an open-source geometry engine implemented in C/C++ that is employed by PostGIS. We employed a local computer equipped with an Intel Core i3-4170 with a frequency of 3.70GHz, 8GB of main memory, and a Lexar NS100 128GB SSD. We used the FESTIval [Carniel et al. 2020] installed in the Ubuntu Desktop 18.04.5 64 bits. FESTIval is a PostgreSQL/PostGIS extension to conduct experimental evaluations of spatial indices. It provides several types of spatial queries and allows users to define workloads to be executed with spatial indices. We executed sequentially the WQs 5 times, flushing the system cache after the execution of each time. Then, we calculated the average elapsed time to execute the WQs with the two different sets of windows. We did not report the results of index construction (used by the filter step) because this kind of analysis goes beyond the scope of this article.

Figure 3 depicts the obtained results. The refinement step was the most costly part of the queries, independently of the employed filter step. The refinement step corresponded from 50.24% to 87% of the total elapsed time. Further, the experiments show that the cost of the refinement step is worst when manipulating spatial datasets containing region objects only. This motivates us to understand how to improve spatial query processing. However, the focus should be beyond the spatial indexing, which corresponds to less than 50% of the total elapsed time according to our experiments.

In this sense, our focus is to provide a comprehensive study on how the spatial approximations have been adopted in the literature to mitigate the effects of the refinement step. We also discuss how spatial approximations are imposing challenges and future trends when processing spatial queries, as discussed in Section 5.2.

[1]They are extracted from the OpenStreetMaps (`http://www.openstreetmap.org/` - accessed on November 11, 2021).

4. THE APPLICATION OF SPATIAL APPROXIMATIONS TO SPATIAL QUERY PROCESSING

Approximations play a fundamental role in the optimization of spatial queries by reducing the number of objects to be evaluated by the refinement step. In this section, we present a *systematic review* that aims to select available and relevant literature on spatial approximation (Section 4.1). We employ strict relevance criteria that strive for novelty (i.e., it goes beyond the incremental improvement in well-known approximations) (Section 4.2). With this, we are able to gather a small and preeminent group of studies that significantly advanced the state of the art in spatial approximations (Section 4.3).

4.1 Methodology

The research questions that guide the development of our systematic review are:

—Which spatial approximations are the most widely used and empirically evaluated?
—How have spatial approximations been employed in spatial query processing?
—Which spatial approximations provide the largest time reductions in spatial query processing?

To gather relevant studies on approximation-based spatial querying and thus answer these questions, we developed the following search string:

("spatial index" OR "spatial indexing" OR "spatial access method" OR "spatial information retrieval") AND ("spatial approximation" OR "spatial approximations" OR "bounding box" OR "approximation-based spatial query") AND ("performance" OR "efficiency" OR "experiment" OR "performance evaluation" OR "empirical evaluation" OR "empirical study")

On November 11, 2021, we employed the search string in the following digital libraries: IEEE[2], Science Direct[3], Springer[4], ACM DL[5], and Google Scholar[6], resulting in 1,497 studies. The inclusion criteria included studies that provide implementations and comparisons of spatial approximations, propose novel approximations, or use different spatial approximations when executing spatial queries. It leads to the exclusion of studies that apply well-known approximations only (i.e., MBRs) or approaches that focus on spatial indexing without the use of spatial approximations. Another inclusion criterion is the incorporation of studies written in English only. Further, we did not restrict the impact factor of the publication venues of the studies since our criteria have returned a concise number of studies, making it feasible to be analyzed in a full reading.

Aiming to expand our research reach, we have conducted one iteration of the *backward snowballing* [Wohlin 2014; Mourao et al. 2017] as an additional stage. Snowballing is a valuable strategy to search for relevant literature in systematic literature reviews. This strategy uses the reference list of a set of papers as input for another iteration of the systematic review. The combination of the traditional search in digital libraries with a snowballing approach leads to a (*hybrid*) search strategy that is considered the most complete in terms of efficiency in finding studies on a given topic. Additionally, snowballing has considerable value since it presents a temporal advance in a research field.

Figure 4 depicts the results of our search strategy, which is based on two stages: (i) search in digital libraries, and (ii) backward snowballing. Each stage has multiple steps where further steps are more detailed but over a reduced number of studies. The steps, in order, consider the publications (i) title and abstract, (ii) introduction and conclusion, and (iii) full text. Figure 4 also reveals the number

[2]https://ieeexplore.ieee.org/Xplore/home.jsp (accessed on November 11, 2021).
[3]https://www.sciencedirect.com (accessed on November 11, 2021).
[4]https://link.springer.com (accessed on November 11, 2021).
[5]https://dl.acm.org (accessed on November 11, 2021).
[6]https://scholar.google.com (accessed on November 11, 2021).

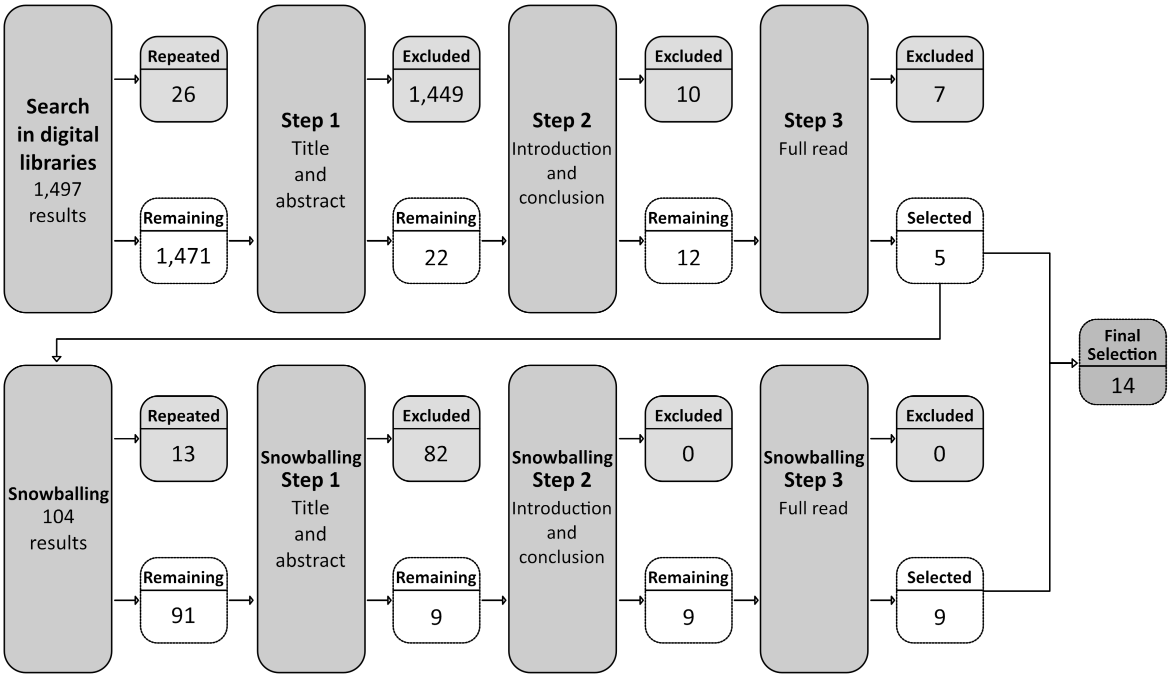


Fig. 4. Steps employed in our systematic review by using one iteration of the snowballing approach.

of studies removed and kept in each step. The five selected studies in Step 3 of the first stage are the initial set of papers for the backward snowballing. The backward snowballing returned 91 unique results (104 in total where 13 were repeated instances). These studies were submitted to the same multi-step inclusion criteria of the previous stage. From those 91 studies, some of them may have been already considered in the first stage of our search. Hence, we have removed such studies based on their title in Snowballing Step 1.

As a result, we have selected 14 studies and classified them according to the categories: (C_1) *novel spatial approximations or implementations*, (C_2) *performance evaluations of spatial approximations*, and (C_3) *spatial approximations for collections of spatial objects* (useful for representing nodes in spatial index structures). It is important to emphasize that a study can belong to more than one category.

4.2 General Characteristics of Existing Approaches

Table I presents the selected studies, a summary of their main features, and their categorization. In addition to the category and reference of the study (first two columns), we provide the source from that the study was retrieved by our selection (third column). Further, we list the approximations that are conceptually discussed and empirically compared by a study (fourth and fifth columns). The conceptual discussion is important to understand the intrinsic characteristics of a spatial approximation, whereas the empirical study highlights the performance behavior of the approximation. We also check whether the study employed some spatial index structure in its experiments (sixth column). Moreover, we list the types of spatial queries evaluated in the studies (last column).

According to the appearance order of the spatial approximations in Table I, the studies discuss and/or empirically compare the following approximations: (i) *Convex Hull* (CH), (ii) *Minimum Bounding n-corner* (n-corner), (iii) *Minimum Bounding Rectangle* (MBR), (iv) *Minimum Bounding Polybox* (MBP), (v) *Minimum Bounding Circle* (MBC), (vi) *Minimum Bounding Ellipse* (MBE),

Table I. Characterization of the studies selected by our systematic review. The studies are sorted by their category and publication year.

Category	Reference	Source	Conceptually Discussed Approx.	Empirically Compared Approx.	Spatial Index Structures	Spatial Queries
C_1	[Graham 1972]	Snow-balling approach	CH	-	-	-
C_1	[Aggarwal et al. 1985]	Snow-balling approach	n-corner	-	-	-
C_1	[Jagadish 1990]	Snow-balling approach	MBR, MBP	MBR, MBP	-	-
C_1	[Welzl 1991]	Snow-balling approach	MBC, MBE	MBC, MBE	-	-
C_2	[Brinkhoff et al. 1993b]	Springer	CH, MBE, MBR, n-corner	MBR, n-corner	R*-Tree	WQ
C_2	[Bandi et al. 2007]	Science Direct	CH, IA, MBR, MER, n-corner	IA, MBR	R-tree, Quadtrees	WQ, Spatial Join
C_1 and C_2	[Brodsky et al. 1995]	Snow-balling approach	MBP, MBR	MBP, MBR	-	-
C_1 and C_2	[Esperança and Samet 1997]	Springer	MBR, OP	MBR, OP	Quadtrees	WQ
C_1 and C_2	[Zimbrao and Souza 1998]	Snow-balling approach	4CRS, ELG, MBR, MER, n-corner	4CRS, ELG, MBR, MER, n-corner	R*-tree	Spatial Join
C_1 and C_2	[Kothuri and Ravada 2001]	Snow-balling approach	CH, IA, MBR	IA, MBR	R-Tree	WQ, CQ, kNN
C_1 and C_2	[Su et al. 2017]	IEEE	n-sided RP	n-sided RP	-	-
C_2 and C_3	[Brinkhoff et al. 1993a]	Snow-balling approach	CH, MBC, MBE, MBR, n-corner, RMBR	CH, MBC, MBE, MBR, n-corner, RMBR	R*-Tree	Point Query
C_2 and C_3	[Brinkhoff et al. 1994]	Snow-balling approach	CH, MBC, MBE, MBR, MEC, MER, n-corner, RMBR	CH, MBC, MBE, MBR, MEC, MER, n-corner, RMBR	R*-Tree	Spatial Join
C_1, C_2, and C_3	[Sidlauskas et al. 2018]	IEEE	CBB, CH, MBC, MBR, n-corner, RMBR	CBB, CH, MBC, MBR, n-corner, RMBR	R-Tree, Hilbert R-Tree, R*-Tree, *revised* R*-Tree	WQ, Spatial Join

(vii) *Interior Approximation* (IA), (ix) *Maximum Enclosed Rectangle* (MER), (x) *Orthogonal Polygons* (OP), (xi) *Four-colors Raster Signature* (4CRS), (xii) *Enclosed Line Segments* (ELG), (xiii) *n-sided Regular Polygon* (n-sided RP), (xiv) *Rotated Minimum Bounding Rectangle* (RMBR), (xv) *Maximum Enclosed Circle* (MEC), and (xvi) *Clipped Bounding Box* (CBB). Table II provides a short description of these approximations.

The MBR is widely employed for testing or validation in all studies, indicating that this particular spatial approximation is a standard benchmark. This is due to its popularity and the low cost of

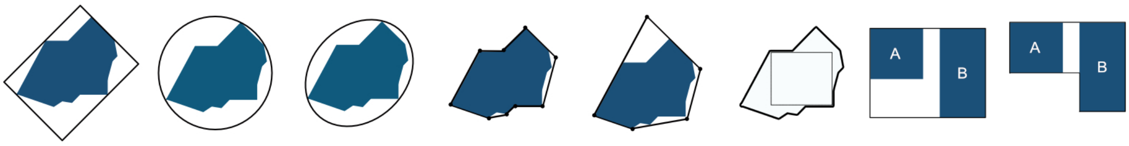
Table II. An overview of the spatial approximations identified in our systematic review of the literature.

Approximation	**Short Description**
Convex Hull (CH)	The smallest convex region object that contains a given spatial object.
Minimum Bounding n-corner (n-corner)	The smallest region object with less than or equal to n corners that encloses a given spatial object.
Minimum Bounding Rectangle (MBR)	The smallest axis-aligned rectangle that encloses a given spatial object. That is, the edges of the rectangle are parallel to the coordinate axes.
Minimum Bounding Polybox (MBP)	The smallest convex region object that encloses a given spatial object and whose edges are perpendicular to preselected axes (not necessarily the standard coordinate axes).
Minimum Bounding Circle (MBC)	The smallest circle that encloses a given spatial object.
Minimum Bounding Ellipse (MBE)	The smallest ellipse that encloses a given spatial object.
Interior Approximation (IA)	Based on the MER. It can also build a regular grid of rectangles (called quadrants) covering the MBR of a given spatial object to identify interior tiles (i.e., quadrants that belong to the interior of the spatial object).
Maximum Enclosed Rectangle (MER)	The largest axis-aligned rectangle inside a given spatial object.
Orthogonal Polygon (OP)	An orthogonal (i.e., rectilinear) region object that encompasses a given spatial object. That is, the sides of this region is parallel to the coordinate axes.
Four-colors Raster Signature (4CRS)	A raster approximation containing $n \times m$ cells, where each cell stores two bits of information to indicate the degree to which a cell belongs to a given spatial object (e.g., empty, weak intersection, strong intersection, and full).
Enclosed Line Segments (ELG)	The longest axis-aligned line segments inside a given spatial object.
n-sided Regular Polygon (n-sided RP)	An equilateral region object of n sides. If it minimally encompasses a given spatial object, then it is an outer-type RP. If it is the largest equilateral polygon inside a given spatial object, then it is an inner-type RP.
Rotated Minimum Bounding Rectangle (RMBR)	The smallest rectangle that encloses a given spatial object. It may not be an axis-aligned rectangle since it can be arbitrarily rotated.
Maximum Enclosed Circle (MEC)	The largest circle inside a given spatial object.
Clipped Bounding Box (CBB)	Given a collection of MBRs, the CBB is a non-convex region object built after rectangularly clipping off the corners of the MBRs.

computing topological relationships. In terms of validation usage, besides MBR, four other spatial approximations are typically used in experimental evaluations: RMBR, MBC, CH, and n-corner (with $n \in \{4, 5\}$). They are compared in terms of elapsed time to process topological relationships or the area of its dead space. Figure 5 depicts examples of some spatial approximations employed by the existing approaches.

Recall that spatial approximations are widely employed when indexing spatial data. The majority of the studies employ an R-tree variant in their performance evaluation. This is strongly related to the use of MBR as a baseline and popular spatial approximation. Further, we highlight that some other spatial index structures based on Quadtrees are also employed in experiments. This is due to the characteristic of the data being manipulated (e.g., point datasets) or to compare the spatial query processing by using other techniques.

With respect to spatial queries employed in the experiments, WQ is a popular type of query used by the studies. The main reason is that this type of query requires the refinement step. Hence, the spatial approximation is fundamental to reducing the overhead of such a step. We also note that spatial join is treated by approaches since it is a costly operation in spatial database systems.



(a) RMBR (b) MBC (c) MBE (d) CH (e) 5-corner (f) MER (g) MBR (h) CBB

Fig. 5. Examples of spatial approximations. In (g-h), the MBR and CBB are applied over a set of objects (i.e., a node of an R-tree), while the other methods (a-e) are applied to a single spatial object.

4.3 Describing Key Ideas of Existing Approaches

In this section, we describe the main characteristics of existing approaches. They are sorted by their publication year. The studies in [Graham 1972; Aggarwal et al. 1985; Welzl 1991] relate to geometric computational algorithms employed by other approaches. In [Graham 1972], the well-known algorithm to compute the convex hull on a set of points is given. This algorithm is widely employed by current spatial libraries, such as those discussed in Section 5.1. While the authors in [Aggarwal et al. 1985] provide an algorithm to compute the n-corner, the work in [Welzl 1991] supplies algorithms to obtain the MBC and MBE of a region object.

The approaches in [Jagadish 1990; Brodsky et al. 1995] make use of MBPs to improve the performance of spatial query processing. They discuss that the edges of an MBR are parallel to the standard axes (i.e., x and y in the two-dimensional case), leading to large areas of dead space. Hence, the underlying idea is to define additional axes in space and then identify their parallel lines that allow us to create a minimum convex polygon containing a spatial object being approximated. These additional axes can have different orientations. The optimal selection of the additional axes is discussed in [Brodsky et al. 1995], which relates to the storage overhead and runtime of spatial queries.

The authors in [Brinkhoff et al. 1993a] provide a performance evaluation that shows the importance of spatial approximations and introduce the term *approximation-based query processing*, which consists of the filter and refinement steps. In fact, many spatial query processing techniques are based on these steps, including the experiments conducted in Section 3.2. Three classes of approximations are identified: (i) conservative approximations, (ii) progressive approximations, and (iii) generalizing approximations. A conservative approximation preserves all points of the original spatial object; however, it may include points that do not belong to the spatial object. Examples include MBR, RMBR, MBC, MBE, CH, and n-corner (Figures 5a-e). Possible implementations and storage requirements of these approximations are discussed by the authors. A progressive approximation is inside of the spatial object; thus, it may not include all the points of the spatial object. This type of approximation can be used to identify objects that actually belong to the final answer of the query. A generalizing approximation is a simplification of the contour of the object that does not belong to the aforementioned approximations. The authors focus on empirically evaluating conservative approximations, which are analyzed by considering the approximation quality (i.e., a measure based on the dead space area). Further, the approximations are used on single elements stored in the leaf nodes of R*-trees. Finally, the authors conclude that the 5-corner delivered good performance results.

The authors in [Brinkhoff et al. 1993b] detail a *geo-architecture* that comprises concepts and techniques for efficiently processing spatial queries. The authors claim that an efficient architecture consists of combining spatial index structures, spatial approximations, decomposition techniques, and secondary indices for thematic attributes. Concerning spatial approximations, the authors show their importance as *geometric pretest* and conceptually describe some of them. The experiments conducted by the authors were focused on validating the proposed architecture and used previous results to justify the use of the 5-corner in the architecture.

The aforementioned studies are extended in [Brinkhoff et al. 1994] to deal with spatial joins. The authors consider that the filter step is processed by R*-trees. Then, according to empirical evaluations, the 5-corner provides good performance when employing it as an additional step. After executing performance evaluations, the authors propose to use a combination of progressive approximations to identify positive hits of the query without accessing the original geometry of the object. The proposed combination consists of the MER (Figure 5f) and the MEC.

The authors in [Esperança and Samet 1997] deal with orthogonal (i.e., axis-aligned) bounding structures for spatial objects, proposing a novel strategy of approximating spatial objects while achieving a balance between complexity and accuracy. The approach is based on an OP with a varying number of vertices, in contrast with the fixed number of four vertices of MBRs. To obtain the approximation, a coarsening algorithm is given by using a combination of two distinct approaches. The authors present the results of experiments based on the filter and refinement steps that highlight the benefits of using OPs in some types of queries, such as large WQs.

In [Zimbrao and Souza 1998], the authors propose the 4CRS, which is based on a raster "signature" of the spatial objects. It combines both conservative and progressive traits in a single approximation. It is composed of cells dividing the region object and small parts of its surrounding area, where each cell stores one of four "colors". The colors indicate the degree to which the cells intersect the approximated region object. It can be *empty*, *weak*, *strong*, or *full* to respectively represent the following situations: (i) the cell does not intersect the region object, (ii) the cell shares at most 50% of its area with the region object, (iii) the cell shares at least 50% of its area with the region object, and (iv) the cell is completely inside the region object. Given its raster nature, the approximation can also be compressed, grouping amounts of repeating data to reduce storage size. Experiments conducted by the authors demonstrated a significant decrease of up to 60% in the number of intersection tests, the number of disk access, and total query execution time when compared to a combination of n-corner, MER, and ELG approximations in spatial joins.

In [Kothuri and Ravada 2001], the authors also describe an intermediate step between the filter and refinement steps. The input of this intermediate step is the set of candidates returned by the filter step. The intermediate step is able to identify the objects that either (i) need to be computed by the refinement step, or (ii) can be included in the final result of the query. This is possible due to the use of progressive approximations called IA, which are based on the MER (Figure 5f). The authors extend the area of the MER by dividing the spatial object into several quadrants. Such quadrants are built on the MBR of the spatial object by using the decomposition technique of Quadtrees. Then, their algorithm selects the cells that are inside the original object. Experiments conducted by the authors show that subdividing the space four times delivers good performance results when computing the approximation and maximizing the IA area.

In [Bandi et al. 2007], the authors introduce the use of GPUs to process topological relations in spatial queries. They propose an intermediate step, between the filter and refinement steps, where additional filters are employed. For intersection queries an interior filter is used, while for within-distance joins the intermediate step uses 0-Object and 1-Object filters. The former consists of IAs and the latter are filters that calculate an upper bound of the distance between a pair of objects. This strategy reduces the number of objects sent to the refinement step. The GPU is accessed through OpenGL's graphical API. The authors integrated their implementation with Oracle Spatial, based on an architecture known as dual-thread. This architecture utilizes a primary and a *graphics thread*. The primary thread runs the initial MBR filter step and a Point in the Polygon test, whilst the graphics thread performs the hardware filter (i.e., by using IAs) by using the GPU. They report a reduction of 5.9 times in the query processing time due to the dual thread system and synchronization.

In [Su et al. 2017], the authors evaluate approximations that consist of n-sided RPs. The idea is to build outer-type and inner-type n-sided RPs. It is important to emphasize that a regular polygon is an equilateral polygon. Hence, an outer-type (inner-type) 4-sided RP is not necessarily equivalent to the MBR (MER) of an object. The authors discuss that the search precision can be improved by increasing the number of sides of the regular polygon. In this sense, the authors compare the search precision of outer-type and inner-type 4-sided regular polygons. The number of sides in the polygon can be adjusted, trading search precision and storage usage.

The authors in [Sidlauskas et al. 2018] propose the CBB to reduce the dead space of groups of spatial objects. Given an MBR that minimally encompasses a group of spatial objects (or MBRs), a CBB consists of this MBR and a set of *clip points*. A clip point is formed by a point in the d-dimensional space and an array of d bits. Each position of this array refers to a dimension in the space. The bit 0 represents the minimum extent of the object, whereas the bit 1 indicates the maximum extent. For instance, in the 2-dimensional case, bitmask 00 refers to the bottom-left corner of an MBR. Hence, the bitmask supplies the direction in which the clipping should be performed. The CBB is useful for indexing techniques that group MBRs of spatial objects into nodes, which in turn are organized hierarchically (e.g., the R-tree and its variants). In addition to the tree structure, the CBB requires a mapping between the nodes and their clip points, which is employed when traversing the tree in index operations (e.g., insertions and queries). Hence, the CBB is not directly stored in the tree but is calculated as needed. The authors experimentally show that the CBB reduces the dead space of nodes if compared to other spatial approximations. The CBB was also applied to the R-tree and other three variants, as shown in Table I. The main benefit of using the CBB is that it reduces the number of paths to be traversed in index operations.

5. DISCUSSIONS

In this section, we analyze existing implementations of spatial approximations (Section 5.1) and discuss open challenges and future trends (Section 5.2).

5.1 Existing Implementations

Spatial libraries implement *spatial type systems* that include data structures for spatial data types, geometric operations, and spatial index structures. They serve as an underlying basis of the operations executed by spatial database systems and GIS. An example is the PostGIS[7], a PostgreSQL extension that employs the spatial libraries GEOS and GDAL. Few studies compare existing spatial libraries. For instance, the work in [Pandey et al. 2021] conducts experiments to analyze the performance of spatial libraries when executing spatial queries by using spatial index structures. However, a study that compares the support for spatial approximations is still missing.

By using the conducted systematic review and considering eight popular open-source spatial libraries, Table III compares the support for spatial approximations provided by these libraries. A popular programming language of these libraries is the C++. In addition, common approximations provided by them are MBR, MBC, and CH. JTS, GEOS, and CGAL provide additional support for the RMBR as a more complex and higher-quality alternative to MBR. CGAL has more variety of spatial approximations. Compared to other libraries, CGAL is the only one to enable the use of MBE and Rotated Minimum Bounding Parallelogram (RMBP). Note that the RMBP is not considered or evaluated by the approaches identified in our systematic review.

5.2 Challenges and Future Trends

As a result of our systematic review of the literature, we identify some topics that can serve as a foundation for advances in spatial information retrieval by using spatial approximations. These topics are: (i) the incorporation of distinct spatial approximations within spatial index structures, (ii) the

[7]`https://postgis.net/` (accessed on November 11, 2021).

Table III. Spatial libraries and their support for spatial approximations.

Name	**Homepage**[8]	**Language**	**Approximations**
JTS	`https://locationtech.github.io/jts/`	Java	CH, MBC, MBR, RMBR
GEOS	`https://trac.osgeo.org/geos`	C/C++	CH, MBC, MBR, RMBR
CGAL	`https://www.cgal.org/`	C++	CH, MBC, MBE, MBR, RMBP, RMBR
Geometric Tools	`https://www.geometrictools.com/`	C++	CH, MBC, MBR
Boost.geometry	`https://www.boost.org`	C++	CH, MBR
Wykobi	`https://www.wykobi.com/`	C++	CH, MBC, MBR
ESRI Geom. API	`https://github.com/Esri`	Java	CH
JSI	`https://github.com/aled/jsi`	Java	MBR

use of spatial approximations in modern hardware environments, (iii) the conduction of empirical studies by using multi-step query processing with spatial approximations.

The first topic relates to the use of novel spatial approximations in the nodes of spatial index structures aiming to reduce the number of paths traversed by index operations. In the era of big data, trends in this topic include the use of distributed and parallel frameworks, such as Hadoop and Spark, to efficiently deal with a large volume of spatial data (see [Pandey et al. 2018; Castro et al. 2020] for a survey). In this sense, many spatial index structures are adapted to such environments. This has led to the need of rethinking how spatial indexing should be performed. Unfortunately, the power of spatial approximations is not well exploited and needs more comprehension, as discussed in [Sidlauskas et al. 2018]. Following the underlying idea of the CBB, a future research topic is the proposal and identification of spatial approximations applied to sets of objects.

The second topic corresponds to understanding the performance behavior of different spatial approximations in modern hardware environments. In addition to the parallel and distributed environment, the use of GPUs and modern storage devices may change how spatial approximations are employed in spatial query processing. The approach discussed in [Bandi et al. 2007] shows the importance of GPUs. However, more efforts are needed when considering modern storage devices like flash-based Solid State Drives (SSDs). Many spatial index structures have been redefined or improved to deal with the intrinsic characteristics of SSDs [Fevgas et al. 2020; Carniel et al. 2022], such as asymmetric read and write costs. Hence, spatial approximations can play an important role in a multi-step processing strategy that employs different modern technologies.

Finally, the last topic refers to the application of approximations as intermediate steps. This trend is identified by many studies discussed in this article. This is a potential strategy that can be applied to different contexts, such as SSDs and parallel and distributed systems. By combining the aforementioned topics, an open research task is how the multi-step processing can be fitted into the development of novel hardware environments. For instance, spatial approximations can be stored in SSDs, while refinement steps can access spatial objects stored in conventional Hard Disk Drives. Both steps can also process calculations by using GPUs.

6. CONCLUSIONS AND FUTURE WORK

In this article, we have provided a systematic review of the literature on the use of spatial approximation in spatial query processing. By using a reproducible methodology based on a snowballing approach, we have compared and described 14 approaches that employ different types of spatial approximations. We have identified that the spatial approximations are mainly used either as intermediate steps of the filter and refinement steps or within spatial index structures. We have also

[8]The homepages were accessed on November 11, 2021.

analyzed which spatial approximations are available in eight popular spatial libraries. This aspect helps us to understand the existing support for spatial approximations in core components of spatial database systems and GIS. Further, we discovered that there is at least one spatial approximation (i.e., the RMBP) that was not considered by the studies discussed in this article. Moreover, we have identified challenging topics and future trends with respect to the application of spatial approximations to spatial query processing. We highlight that the most challenging tasks include the use of approximations within spatial index structures and the strategy of employing approximations in multi-step query processing.

By considering the future trends introduced in this article, our main future work topic is the conduction of an empirical analysis of spatial approximations as intermediate steps. Given n intermediate steps, we plan to discover the most efficient order of execution of the spatial approximations in different types of spatial queries in distinct hardware environments.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001. Yuri K. Lopes is thankful for the partial financial support from Fundação de Amparo a Pesquisa e Inovação do Estado de SC, Brazil FAPESC 2021TR930 and 2021TR2074. Anderson C. Carniel has been supported by Google as a recipient of the 2022 Google Research Scholar Program.

REFERENCES

Aggarwal, A., Chang, J. S., and Yap, C. K. Minimum area circumscribing polygons. *The Visual Computer* vol. 1, pp. 112–117, 1985.

Alam, M. M., Torgo, L., and Bifet, A. A survey on spatio-temporal data analytics systems. *ACM Computing Surveys*, 2021.

Anselin, L. Spatial data science. In *International Encyclopedia of Geography*. John Wiley & Sons, Ltd, pp. 1–6, 2020.

Bandi, N., Sun, C., Agrawal, D., and El Abbadi, A. Fast computation of spatial selections and joins using graphics hardware. *Information Systems* 32 (8): 1073–1100, 2007.

Beckmann, N., Kriegel, H.-P., Schneider, R., and Seeger, B. The R*-tree: An efficient and robust access method for points and rectangles. In *ACM SIGMOD International Conference on Management of Data*. pp. 322–331, 1990.

Beckmann, N. and Seeger, B. A revised R*-tree in comparison with related index structures. In *ACM SIGMOD International Conference on Management of Data*. pp. 799–812, 2009.

Bertella, P. K., Lopes, Y. K., Oliveira, R. A. P., and Carniel, A. C. The application of spatial approximations to spatial query processing: A systematic review of literature. In *Brazilian Symposium on Databases*. pp. 229–240, 2021.

Bouros, P. and Mamoulis, N. Spatial joins: What's next? *SIGSPATIAL Special* 11 (1): 13–21, 2019.

Brinkhoff, T., Horn, H., Kriegel, H.-P., and Schneider, R. A storage and access architecture for efficient query processing in spatial database systems. In *International Symposium on Spatial Databases*. pp. 357–376, 1993b.

Brinkhoff, T., Kriegel, H.-P., and Schneider, R. Comparison of approximations of complex objects used for approximation-based query processing in spatial database systems. In *IEEE International Conference on Data Engineering*. pp. 40–49, 1993a.

Brinkhoff, T., Kriegel, H.-P., Schneider, R., and Seeger, B. Multi-step processing of spatial joins. In *ACM SIGMOD International Conference on Management of Data*. pp. 197–208, 1994.

Brodsky, A., Lassez, C., Lassez, J.-L., and Maher, M. J. Separability of polyhedra for optimal filtering of spatial and constraint data. In *ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. pp. 54–65, 1995.

Carniel, A. C. Spatial information retrieval in digital ecosystems: A comprehensive survey. In *International Conference on Management of Digital Ecosystems*. pp. 10–17, 2020.

Carniel, A. C., Ciferri, R. R., and Ciferri, C. D. A. Spatial datasets for conducting experimental evaluations of spatial indices. In *Satellite Events of the Brazilian Symposium on Databases - Dataset Showcase Workshop*. pp. 286–295, 2017.

Carniel, A. C., Ciferri, R. R., and Ciferri, C. D. A. FESTIval: A versatile framework for conducting experimental evaluations of spatial indices. *MethodsX* vol. 7, pp. 1–19, 2020.

Carniel, A. C., Roumelis, G., Ciferri, R. R., Vassilakopoulos, M., Corral, A., and Aguiar, C. D. Porting disk-based spatial index structures to flash-based solid state drives. *GeoInformatica* vol. 26, pp. 253–298, 2022.

Carniel, A. C. and Schneider, M. A survey of fuzzy approaches in spatial data science. In *IEEE International Conference on Fuzzy Systems*. pp. 1–6, 2021.

Castro, J. P. C., Carniel, A. C., and Ciferri, C. D. A. Analyzing spatial analytics systems based on Hadoop and Spark: A user perspective. *Software: Practice and Experience* 50 (12): 2121–2144, 2020.

Egenhofer, M. J. and Herring, J. R. Categorizing binary topological relations between regions, lines and points in geographic databases. In *The 9-Intersection: Formalism and Its Use for Natural-Language Spatial Predicates*, 1994.

Esperança, C. and Samet, H. Orthogonal polygons as bounding structures in filter-refine query processing strategies. In *International Symposium on Spatial Databases*. pp. 197–220, 1997.

Esquerdo, J. C. D. M., Antunes, J. F. G., Coutinho, A. C., Speranza, E. A., Kondo, A. A., and dos Santos, J. L. SATVeg: A web-based tool for visualization of MODIS vegetation indices in South America. *Computers and Electronics in Agriculture* vol. 175, pp. 105516, 2020.

Fevgas, A., Akritidis, L., Bozanis, P., and Manolopoulos, Y. Indexing in flash storage devices: a survey on challenges, current approaches, and future trends. *The VLDB Journal* vol. 29, pp. 273–311, 2020.

Gaede, V. and Günther, O. Multidimensional access methods. *ACM Computing Surveys* 30 (2): 170–231, 1998.

Graham, R. L. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters* vol. 1, pp. 132–133, 1972.

Güting, R. H. An introduction to spatial database systems. *The VLDB Journal* vol. 3, pp. 357–399, 1994.

Guttman, A. R-trees: A dynamic index structure for spatial searching. In *ACM SIGMOD International Conference on Management of Data*. pp. 47–57, 1984.

Huang, H., Gartner, G., Krisp, J. M., Raubal, M., and de Weghe, N. V. Location based services: ongoing evolution and research agenda. *Journal of Location Based Services* 12 (2): 63–93, 2018.

Jacox, E. H. and Samet, H. Spatial join techniques. *ACM Transactions on Database Systems* 32 (1): 1–44, 2007.

Jagadish, H. Spatial search with polyhedra. In *IEEE International Conference on Data Engineering*. pp. 311–319, 1990.

Jensen, C. S., Kligys, A., Pedersen, T. B., and Timko, I. Multidimensional data modeling for location-based services. *The VLDB Journal* vol. 13, pp. 1–21, 2004.

Kamel, I. and Faloutsos, C. Hilbert R-tree: An improved R-tree using fractals. In *International Conference on Very Large Data Bases*. pp. 500–509, 1994.

Kothuri, R. K. and Ravada, S. Efficient processing of large spatial queries using interior approximations. In *International Symposium on Spatial and Temporal Databases*. pp. 404–421, 2001.

Mourao, E., Kalinowski, M., Murta, L., Mendes, E., and Wohlin, C. Investigating the use of a hybrid search strategy for systematic reviews. In *ACM/IEEE International Symposium on Empirical Software Engineering and Measurement*. pp. 193–198, 2017.

Oosterom, P. V. a. N. Spatial access methods. In *Geographical Information Systems: Principles, Techniques, Management and Applications*, 2nd Edition ed., P. A. Longley, M. F. Goodchild, D. J. Maguire, and D. W. Rhind (Eds.). pp. 385–400, 2005.

Pandey, V., Kipf, A., Neumann, T., and Kemper, A. How good are modern spatial analytics systems? *VLDB Endowment* 11 (11): 1661–1673, 2018.

Pandey, V., van Rene, A., Kipf, A., and Kemper, A. How good are modern spatial libraries? *Data Science and Engineering* vol. 6, pp. 192–208, 2021.

Papadias, D., Sellis, T., Theodoridis, Y., and Egenhofer, M. J. Topological relations in the world of minimum bounding rectangles: A study with R-trees. In *ACM SIGMOD International Conference on Management of Data*. pp. 92–103, 1995.

Samet, H. The quadtree and related hierarchical data structures. *ACM Computing Surveys* 16 (2): 187–260, 1984.

Schneider, M. and Behr, T. Topological relationships between complex spatial objects. *ACM Transactions on Database Systems* 31 (1): 39–81, 2006.

Shen, J., Chen, M., and Liu, X. Classification of topological relations between spatial objects in two-dimensional space within the dimensionally extended 9-intersection model. *Transactions in GIS* 22 (2): 514–541, 2018.

Sidlauskas, D., Chester, S., Zacharatou, E. T., and Ailamaki, A. Improving spatial data processing by clipping minimum bounding boxes. In *IEEE International Conference on Data Engineering*. pp. 425–436, 2018.

Su, W.-T., Wei, H.-Y., Yeh, J.-H., and Chen, W.-C. An efficient approach based on polygon approximation to query spatial data on digital archiving system. In *International Conference on Applied System Innovation*. pp. 389–392, 2017.

Welzl, E. Smallest enclosing disks (balls and ellipsoids). In *New Results and New Trends in Computer Science*. pp. 359–370, 1991.

Wohlin, C. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *International Conference on Evaluation and Assessment in Software Engineering*. pp. 1–10, 2014.

Zimbrao, G. and Souza, J. M. A raster approximation for processing of spatial joins. In *International Conference on Very Large Data Bases*. pp. 558–569, 1998.