

The Application of Spatial Approximations to Spatial Query Processing: A Systematic Review of Literature

Pedro G. K. Bertella¹, Yuri K. Lopes², Rafael A. P. de Oliveira¹, Anderson C. Carniel³

¹Federal University of Technology - Paraná
Dois Vizinhos – PR – Brazil

pedrobertella@alunos.utfpr.edu.br, raoliveira@utfpr.edu.br

²Department of Computer Science – Santa Catarina State University
Joinville – SC – Brazil

yuri.lopes@udesc.br

³Department of Computer Science – Federal University of São Carlos
São Carlos – SP – Brazil

accarniel@ufscar.br

Abstract. *Spatial approximations simplify the geometric shape of complex spatial objects. Hence, they have been employed to alleviate the evaluation of costly computational geometric algorithms when processing spatial queries. For instance, spatial index structures employ them to organize spatial objects in tree structures (e.g., the R-tree). We report experiments considering two real datasets composed of ~ 1.5 million regions and ~ 2.7 million lines. The experiments confirm the performance benefits of spatial approximations and spatial index structures. However, we also identify that a second processing step is needed to deliver the final answer and often requires higher processing time than the step that uses index structures only. It leads to the interest in studying how spatial approximations can be efficiently used to improve both steps. This paper presents a systematic review on this topic. 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. Further, we characterize them and discuss some future trends.*

1. Introduction

Spatial information retrieval is a common task required by advanced applications [Carniel 2020], such as location-based services, smart cities, and mobile recommendation systems. The goal of such applications is to enrich the data analysis by taking into the importance of “where”. For this, geographical and spatial phenomena are modeled as instances of *spatial data types* (e.g., points, lines, and regions). Spatial database systems and Geographical Information Systems (GIS) are powerful tools for storing, managing, and retrieving *spatial objects*.

To retrieve spatial objects, applications perform spatial queries containing a set of conditions that the retrieved spatial objects must satisfy. Spatial relationships, such as *topological relationships* (e.g., overlaps, contains) [Schneider and Behr 2006], are often

employed as conditions in spatial queries. Thus, several types of spatial queries have been defined in the literature (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., 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. 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 Query), 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 2.2). 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 paper 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.
- 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 paper is organized as follows. Section 2 summarizes needed basic concepts. Section 3 introduces our systematic review and the overview of existing approaches. Section 4 discusses how existing approaches employ spatial approximations. Section 5 concludes the paper and presents future work.

2. Needed Concepts from Spatial Database Systems and GIS

In this section, we sketch definitions related to spatial queries (Section 2.1) and discuss how they are processed by using two steps (Section 2.2).

2.1. 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.

As a basic feature, spatial database systems and GIS allow users to pose different types of spatial queries [Gaede and Günther 1998, Carniel 2020]. Topological relationships are widely used to define conditions in spatial queries. For instance, the Window Query (WQ) finds all spatial objects that *intersect* an axis-parallel search object (i.e., a rectangle also known as *window*). A variant is the Containment Query (CQ), which finds all objects that are *inside* the window. Another example is the Point Query that returns all objects that have a particular simple point object. In addition, two spatial datasets can be associated according to a particular topological relationship, leading to Spatial Joins. Other 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.

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.

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 3.2). For instance, a complex region containing several points can be approximated to a rectangle that minimally encompasses its boundary (called MBR) by storing only four numbers in the two-dimensional case (Figure 1a). Spatial approximations allow us to improve the performance of evaluating topological relationships [Papadias et al. 1995].

Some types of spatial queries can be directly answered by using spatial approximations only, such as the CQ (Figure 1b). However, this is not the case for several other types of spatial queries, such as the WQ (Figure 1c). This is due to the introduction of an area that does not belong to the original spatial object, called *dead space*. When processing the relationship *intersects* on approximations only, the result may contain spatial objects that do not belong to the final answer (Figure 1c). 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.

2.2. Filter and Refinement Steps

Two steps are 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 of the query and return the final answer of the query. This step usually

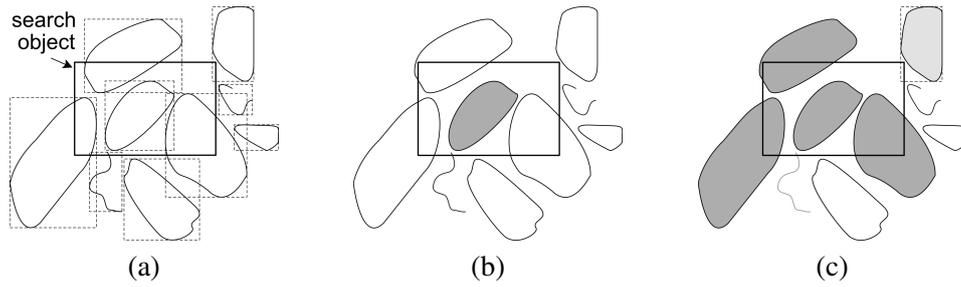


Figure 1. The use of MBRs (dashed lines) for a given search object (a), a CQ (b), and a WQ (c). The final answer of (b) and (c) consists of the spatial objects in dark gray, whereas the false candidate is shown in light gray.

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.

There are several approaches that focus on improving the elapsed time of spatial index structures (i.e., filter step) (as described in [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.

We used two real spatial datasets described in [Carniel et al. 2020]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 indexes 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 indexes. It provides several types of spatial queries and allows users to define workloads to be executed with spatial indexes. 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 paper.

Figure 2 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

¹They are extracted from the OpenStreetMaps (<http://www.openstreetmap.org/>)

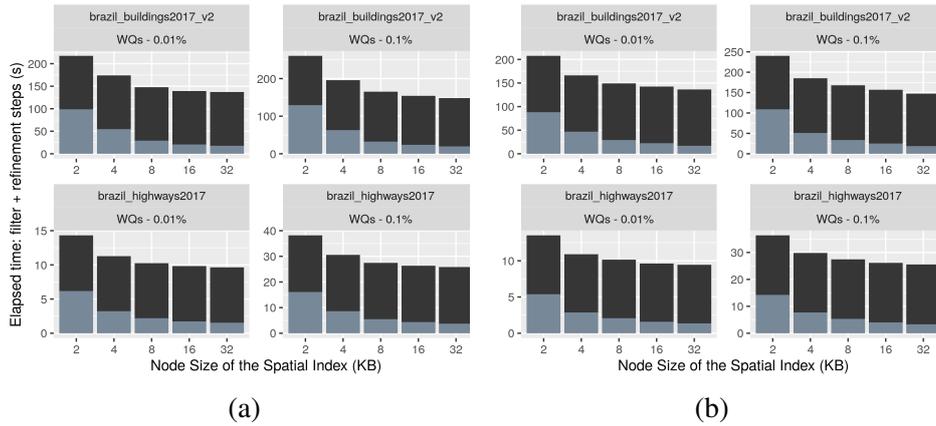


Figure 2. Results of our motivating experiments for the R-trees (a) and R*-trees (b). Since the spatial index structures are not sufficient to answer the WQs, the refinement step was required. The stacked bars show the elapsed times of the filter (in light gray) and refinement (in dark gray) steps. Note the different scales in the plots.

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 4.2.

3. 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 3.1). We employ strict relevance criteria that strive for novelty (i.e., it goes beyond the incremental improvement in well-known approximations) (Section 3.2). With this, we are able to gather a small and preminent group of studies that significantly advanced the state-of-the-art in spatial approximations.

3.1. Methodology

To gather relevant studies on approximation-based spatial querying, 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”)

We used the search string in the following search engines²: IEEE³, Science Di-

²The searching period was from April 27, 2020, to May 04, 2020.

³<https://ieeexplore.ieee.org/Xplore/home.jsp>

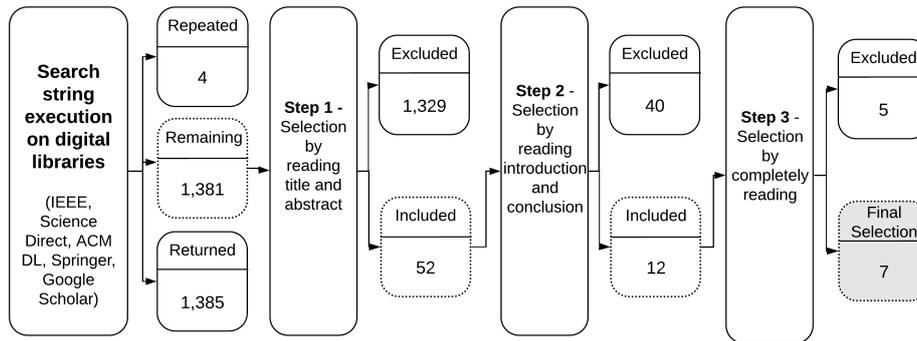


Figure 3. The employed methodology of our systematic review of the literature.

rect⁴, Springer⁵, ACM DL⁶, and Google Scholar⁷, resulting in 1,385 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 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.

For the sake of feasibility, the inclusion criteria were considered in 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. This process is detailed in Figure 3, which reveals the number of studies removed and kept in each step. In the last step, we classified the studies by considering the categories: (C_1) *novel spatial approximations*, (C_2) *performance evaluations of spatial approximations*, and (C_3) *spatial approximations within spatial data index structures*. It is important to emphasize that a study can belong to more than one category. We selected seven studies that are summarized in the following section.

3.2. Overview of Existing Approaches

Table 1 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 search engine 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). They are: (i) *Minimum Bounding Rectangle* (MBR), (ii) *Minimum Bounding Circle* (MBC), (iii) *Minimum Bounding Ellipse* (MBE), (iv) *Convex Hull* (CH), (v) *Minimum Bounding n-corner* (n-corner), (vi) *Rotated Minimum Bounding Rectangle* (RMBR), (vii) *Clipped Bounding Box* (CBB), (viii) *Maximum Inscribed Rectangle* (MIR), (ix) *Enclosed Line Segments* (ELG), (x) *Interior Approximation* (IA), and (xi) *n-sided Regular Polygon*. The conceptual discussion is important to understand the intrinsic characteristics of a spatial approximation, whereas the

⁴<https://www.sciencedirect.com>

⁵<https://link.springer.com>

⁶<https://dl.acm.org>

⁷<https://scholar.google.com>

Table 1. Characterization of the studies selected by our systematic review.

Category	Study	Search Engine	Conceptually Discussed Approx.	Empirically Compared Approx.	Spatial Index Structures	Spatial Queries
C ₂	ref1	Springer	MBR, MBE, CH, n -corner	MBR, n -corner	R*-Tree	WQ
	ref2	Science Direct	MBR, CH, n -corner, MIR, IA	IA, MBR	R-tree, Quadtrees	WQ, Spatial Join
C ₁ and C ₂	ref3	Springer	MBR, CH, IA	MBR, IA	R-Tree	WQ, kNN
	ref4	IEEE	n -sided RP	n -sided RP	-	Unknown
C ₂ and C ₃	ref5	IEEE	MBR, RMBR, MBC, MBE, CH, n -corner	MBR, RMBR, MBC, MBE, CH, n -corner	R*-Tree	Point Query
	ref6	Springer	MBR, RMBR, MBC, MBE, CH, n -corner, MIR, ELG	MBR, RMBR, MBC, MBE, CH, n -corner, MIR, ELG	R*-Tree	Spatial Join
C ₁ , C ₂ , and C ₃	ref7	IEEE	MBR, RMBR, MBC, CH, n -corner, CBB	MBR, RMBR, MBC, CH, n -corner, CBB	quadratic R-Tree, Hilbert R-Tree, R*-Tree, revised R*-Tree	WQ, Spatial Join

References: ref1: [Brinkhoff et al. 1993];, ref2:[Bandi et al. 2007], ref3: [Kothuri and Ravada 2001], ref4: [Su et al. 2017], ref5: [Brinkhoff et al. 1993], ref6: [Brinkhoff and Kriegel 1994, Brinkhoff et al. 1994], ref7: [Sidlauskas et al. 2018]

empirical study highlights the performance behavior of the approximation. 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 computing topological relationships. In terms of validation usage, besides MBR, four other spatial approximations are typically used for validation: 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 4 depicts examples of spatial approximations employed by the existing approaches, which are discussed in Section 4.

We also check whether the study employed some spatial index structure in its experiments (sixth column). Recall that spatial approximations are widely employed when indexing spatial data. All these 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. In terms of types of spatial queries evaluated in the studies (last column), WQ is a popular type of query used by four studies. The main reason is that this type of query requires the refinement step. Hence, the spatial approximation is fundamental to reduce 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.

4. Strategies for using Spatial Approximations to Optimize Spatial Queries

In this section, we shortly describe the seven selected approaches (Section 4.1). It provides the foundation to further discuss related topics (Section 4.2).

4.1. Existing Approaches

In this section, we describe the key ideas of existing approaches. They are sorted by their publication year. The authors in [Brinkhoff et al. 1993] provide a performance

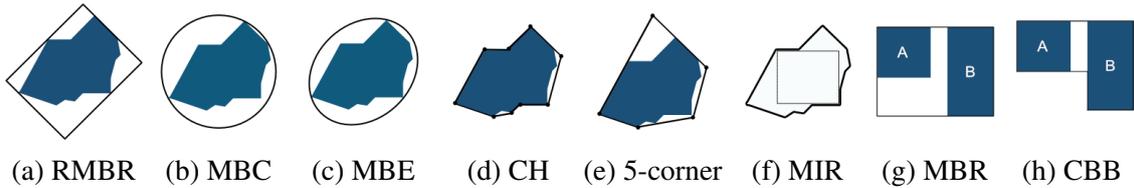


Figure 4. Examples of spatial approximations. In (g-h), the MBR and CBB are applied over a set of objects (i.e., a node of an R-tree), while the other methods (a-e) are applied over a single spatial object. Observe how CBB reduces the dead space of the MBR.

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 2.2. 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 (Figure 4a-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 (Table 1), 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-corners delivered good performance results.

The authors in [Brinkhoff et al. 1993] 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 (Table 1). 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 and Kriegel 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 MIR (Figure 4f) and the ELG.

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 *Interior Approximations* (IA), which are based on the MIR (Figure 4f). The authors extend the area of the MIR by dividing the spatial object into several partitions. Such partitions are built on the MBR of the spatial object by using Quadrees decomposition. Then, their algorithm selects the cells that are inside of 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's 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 regular polygons (n -sided RP). 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 (MIR) 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 *Clipped Bounding Box* (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 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

Table 2. Spatial libraries and their support for spatial approximations.

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

nodes if compared to other spatial approximations. The CBB was also applied to the R-tree and other three variants, as shown in Table 1. The main benefit of using the CBB is that it reduces the number of paths to be traversed in index operations.

4.2. Discussions

In this section, we analyze existing implementations of spatial approximations (Section 4.2.1) and discuss open challenges and future trends (Section 4.2.2).

4.2.1. 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 PostGIS⁸, a PostgreSQL extension that employs the spatial libraries GEOS and GDAL. There are few studies that compare existing spatial libraries, such as the work in [Pandey et al. 2020] that 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 2 compares the support for spatial approximations provided by the 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).

4.2.2. 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 use of spatial approximations in modern hardware

⁸<https://postgis.net/>

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] for a survey). In this sense, many spatial index structures are adapted to such environments. This has led to rethinking how spatial indexing should be performed in these environments. Unfortunately, the power of spatial approximations is not well exploited and needs more comprehension (e.g., the application of spatial approximations 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. As for the latter, efforts are needed when considering modern storage devices like flash-based Solid State Drives (SSDs). Many spatial index structures have been modified to deal with the intrinsic characteristics of SSDs (see [Fevgas et al. 2019] for a survey).

Finally, the last topic is the application of approximations as intermediate steps. This trend is identified by many studies discussed in this paper and leads to a potential strategy that can be applied to different contexts, such as SSDs and distributed systems.

5. Conclusions and Future Work

In this paper, we have provided a systematic review of literature on the use of spatial approximation in spatial query processing. By using a reproducible methodology, we have compared and described seven approaches that employ different types of spatial approximations. We identified that the spatial approximations are mainly used either as intermediate steps of the filter and refinement steps or within spatial index structures. The combination of both usages is still a challenging task.

By considering the future trends introduced in this paper, our future work include the 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.

References

- Bandi, N., Sun, C., Agrawal, D., and El Abbadi, A. (2007). Fast computation of spatial selections and joins using graphics hardware. *Information Systems*, 32(8):1073 – 1100.
- Beckmann, N., Kriegel, H.-P., Schneider, R., and Seeger, B. (1990). The R*-tree: An efficient and robust access method for points and rectangles. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 322–331.
- Brinkhoff, T., Horn, H., Kriegel, H.-P., and Schneider, R. (1993). A storage and access architecture for efficient query processing in spatial database systems. In *Int. Symp. on Spatial Databases*, pages 357–376.

- Brinkhoff, T., Kriegel, H. ., and Schneider, R. (1993). Comparison of approximations of complex objects used for approximation-based query processing in spatial database systems. In *Int. Conf. on Data Engineering*, pages 40–49.
- Brinkhoff, T. and Kriegel, H.-P. (1994). Approximations for a multi-step processing of spatial joins. In *IGIS '94: Geographic Information Systems*, pages 25–34.
- Brinkhoff, T., Kriegel, H.-P., Schneider, R., and Seeger, B. (1994). Multi-step processing of spatial joins. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 197–208.
- Carniel, A. C. (2020). Spatial information retrieval in digital ecosystems: A comprehensive survey. In *Int. Conf. on Management of Digital EcoSystems*, pages 10–17.
- Carniel, A. C., Ciferri, R. R., and Ciferri, C. D. A. (2020). FESTIVAL: A versatile framework for conducting experimental evaluations of spatial indices. *MethodsX*, 7:1–19.
- Egenhofer, M. J. and Herring, J. R. (1994). 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*.
- Fevgas, A., Akritidis, L., Bozanis, P., and Manolopoulos, Y. (2019). Indexing in flash storage devices: a survey on challenges, current approaches, and future trends. *The VLDB Journal*, 29:273–311.
- Gaede, V. and Günther, O. (1998). Multidimensional access methods. *ACM Computing Surveys*, 30(2):170–231.
- Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 47–57.
- Kothuri, R. K. and Ravada, S. (2001). Efficient processing of large spatial queries using interior approximations. In *Int. Conf. on Advances in Spatial and Temporal Databases*, pages 404–421.
- Pandey, V., Kipf, A., Neumann, T., and Kemper, A. (2018). How good are modern spatial analytics systems? *VLDB Endowment*, 11(11):1661–1673.
- Pandey, V., van Rene, A., Kipf, A., and Kemper, A. (2020). How good are modern spatial libraries? *Data Science and Engineering*, 6:192–208.
- Papadias, D., Sellis, T., Theodoridis, Y., and Egenhofer, M. J. (1995). Topological relations in the world of minimum bounding rectangles: A study with R-trees. In *ACM SIGMOD Int. Conf. on Management of Data*, pages 92–103.
- Schneider, M. and Behr, T. (2006). Topological relationships between complex spatial objects. *ACM Transactions on Database Systems*, 31(1):39–81.
- Sidlauskas, D., Chester, S., Tzirita Zacharitou, E., and Ailamaki, A. (2018). Improving spatial data processing by clipping minimum bounding boxes. In *Int. Conf. on Data Engineering*, pages 425–436.
- Su, W., Wei, H., Yeh, J., and Chen, W. (2017). An efficient approach based on polygon approximation to query spatial data on digital archiving system. In *Int. Conf. on Applied System Innovation*, pages 389–392.