Semantically Augmented Range Queries over Heterogeneous Geospatial Data

Goce Trajcevski
Iowa State University
USA
gocet25@iastate.edu

Booma Sowkarthiga Balasubramani
University of Illinois at Chicago, USA
bbalas3@uic.edu

Isabel F. Cruz
University of Illinois at Chicago
USA
ifcruz@uic.edu

Roberto Tamassia
Brown University
USA
rt@cs.brown.edu

Xu Teng
Iowa State University
USA
xuteng@iastate.edu

ABSTRACT

Geospatial data integration combines two or more data layers to facilitate advanced querying, analysis, reasoning, and visualization. In general, different layers (e.g., ZIP codes, census blocks, school districts, and land use parcels) have different spatial partitions and different types of associated semantic descriptors. In addition, geospatial data may contain errors (e.g., due to imprecision in the measurements or to representation constraints) causing uncertainty that needs to be incorporated and quantified in the query answers. In this paper, we leverage semantic descriptors in heterogeneous information layers to build a data structure that enables efficient processing of geospatial range queries by returning an estimate of the answer together with an error bound. We present the processing algorithms and evaluate our approach by means of experiments that encompass large datasets, demonstrating the benefits of our approach.

CCS CONCEPTS

• **Information systems → Geographic information systems**;

KEYWORDS

Geospatial Data integration, Semantics, Uncertainty, Range Queries

ACM Reference Format:
Goce Trajcevski, Booma Sowkarthiga Balasubramani, Isabel F. Cruz, Roberto Tamassia, and Xu Teng. 2020. Semantically Augmented Range Queries over Heterogeneous Geospatial Data. In *28th International Conference on Advances in Geographic Information Systems (SIGSPATIAL '20), November 3–6, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397536.3422271

1 INTRODUCTION

Geospatial data management is essential for applications in urban planning [38], smart cities [1, 3], geology [16], agriculture [5, 28], disaster remediation [12], infrastructure management [11], epidemiology [33], and many others. Geospatial vector data consists of geometric shapes, such as a line representing a river or road, a polygon representing the boundary of a farm, or a point representing a school building.

Geospatial data integration combines two or more *data layers* to facilitate advanced querying, analysis, reasoning, and visualization [2]. In general, each layer has a different spatial partition into regions, for example, ZIP codes, census blocks, school districts, and land use parcels. Each region may be annotated with semantic descriptors, which, for a land parcel, include Lake, Park, Commercial, and Residential. Descriptor Residential can be further subdivided into Single- and Multi-family residential. The hierarchy of land use codes can be modeled using an ontology to facilitate reasoning.

Our main objective is to develop effective methodologies for integrating different data layers, along with efficient querying algorithms, which will enable acquisition of information in targeted geospatial regions, as defined by *range queries*.

From the heterogeneity viewpoint, for each layer there may be a different granularity, for example, municipality vs. county, a different ontology, and a different format. In this paper we do not focus on bridging across ontologies, a process called *alignment* [13–15, 39], nor do we consider conversion between formats, which a library like GDAL [21] can perform. We do, however, take into account errors, and associated uncertainty, which are omnipresent in spatial data due to measurement imprecision [23], representation constraints [46], imprecise boundaries [34], and uncertainty of moving objects [49]. In our work, we aim to reduce errors and their propagation by considering more than one layer at a time. That is, we want to increase the quantity of information given by one layer—and therefore reduce uncertainty—by adding information about another layer, or layers. At the same time, we want to quantify the decrease in uncertainty.

Figure 1 shows two maps for the Lincoln Park neighborhood in Chicago: the Census block layer of Figure 1(a) and the high rise building layer of Figure 1(b). In each census block, most of the population is located in high rise buildings, rather than in single family homes. Therefore, by using both maps, one can more finely define the regions in each block that are more densely populated.

In this paper, for concreteness, we consider the problem of estimating the population within the region defined by a range query

SIGSPATIAL '20, November 3–6, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-8019-5/20/11.
https://doi.org/10.1145/3397536.3422271

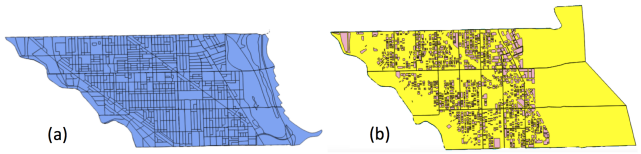

Figure 1: Maps of the Lincoln Park neighborhood in Chicago: (a) Census blocks [10]; (b) high rise buildings [9].

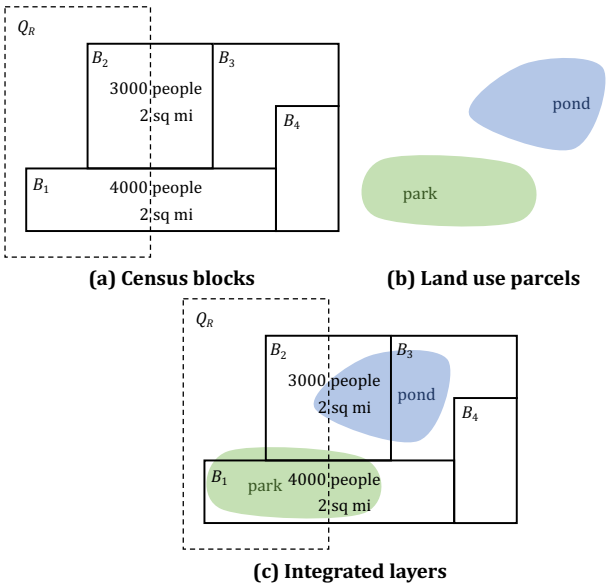

Figure 2: Querying two integrated layers.

using the census population data layer together with the land use data layer.

For example, in Figure 2, we have the following layers:

(a) A layer of census blocks, where block B_1 and B_2 have area 2 square miles and population 4000 and 3000, respectively. No measurement imprecision is considered in this example.
(b) A layer of land use parcels with two polygons corresponding to a pond and a park.

Rectangle Q_R is the region for which we want the total population count. For simplicity, we assume that exactly half of B_1 and half of B_2 overlap with Q_R. Further, we assume a uniform population density in these blocks. Hence, if we consider only the census blocks layer, we estimate the total population within Q_R to be $4000 \cdot 0.5 + 3000 \cdot 0.5 = 3500$. However, if we consider also the land use parcels layer, we can derive a more accurate estimate of the query result by incorporating the fact that the pond and park intersect portions of the blocks and thus reduce the populated regions of the blocks. In particular, note that a portion of B_1 is covered by the park and cannot be populated. Similarly, portions of B_2 are covered by the park and pond and cannot be populated. Thus, assume that Q_R overlaps with 20% of the populated region of B_1 and 60% of the populated region of B_2. We now estimate the population within Q_R to be $4000 \cdot 0.2 + 3000 \cdot 0.6 = 2600$, a more accurate answer given our assumptions.

To solve this problem and related ones, we introduce a modified R-tree, named *Semantically Augmented R-tree*, or *SeA-RT*, and a new range query, named *Semantically Augmented Range Query*, or *SeA-RQ*. Together with a few primitive spatial operations (such as the intersection of any two shapes) and the ontologies of the descriptors, we show how processing a SeA-RQ can be done more efficiently (in terms of running time) in comparison to using a regular R-tree to index only spatial shapes. In other words, the semantically augmented nodes may enable earlier pruning while processing the SeA-RQ to compute the expected value of the population in a query range and a bound on the error. We evaluate our approach by means of experiments that encompass large datasets and compare our results with those of applicable baselines.

The main contributions of this paper are as follows:

- A novel data structure SeA-RT (Semantically Augmented R-Tree) to index spatial and semantic attributes, including errors, for heterogeneous geospatial datasets (Section 2).
- An efficient algorithm for the evaluation of a SeA-RQ (Semantically Augmented Range Query), which incorporates semantic descriptors with explicit error bounds (Section 3).
- An experimental evaluation based on real and synthetic large datasets, demonstrating the tradeoffs between imprecision and efficiency (Section 4).

In the rest of the paper, we review related work in Section. 5 and present concluding remarks and discuss future work in Section 6.

2 DATA STRUCTURE

In this section, we provide basic background and notation and introduce the proposed data structure, the Semantically Augmented R-Tree (SeA-RT). A geospatial dataset, $\mathcal{D}$, is a collection of regions. Each region is described by a triple (oid, B, S), where:

- *oid* denotes the unique ID of the region;
- B denotes the polygonal boundary of the region (i.e., spatial extent), represented as a sequence of points in counter-clockwise order;
- S denotes the sequence of semantic descriptors of the region, $S = [S_1, \ldots, S_m]$ $(m \in \mathbb{N})$.

2.1 Semantic Descriptors

A *semantic descriptor* is a pair (a, x), where a is an attribute and x is the value of a. For brevity, we use the notation $oid.a$ to denote the value of attribute a of the region with ID *oid*. An attribute can be *numerical*, for example, $(population, 850)$ or *categorical*, for example, $(land\_use, \text{commercial})$. We assume that *area*, which denotes the numeric area of the region, is included among the semantic descriptors. If no such attribute is explicitly provided, it can be computed from the polygonal boundary.

Error. Associated with a numerical attribute, such as *resident population count*, there may be an *error* that characterizes the amount of uncertainty or measurement imprecision when collecting the data. To enable standard statistical error propagation analysis [41], we assume that the error is the *standard deviation* of the underlying attribute value.

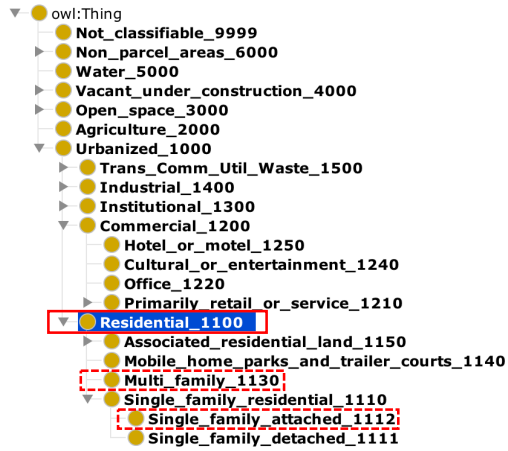

Figure 3: Example of land use ontology [9].

Ontology. For a categorical attribute, we assume its possible values are organized in an *ontology*. For example, Figure 3 shows an ontology for the values of attribute land use [9]. In practice, numerical codes may be used as a shorthand for categorical attributes. For example, in the land use ontology of Figure 3, the attribute value Residential is associated with numerical code 1100.

Attribute constraint and compatibility. In various real-world scenarios, the values of certain attributes may determine or constrain the values of other attributes. For example, if the *land_use* value is water, we expect the population count attribute to have value 0. Also, *land_use* attribute value single-family is normally associated with a lower population count than Multi-family.

In building the SeA-RT data structure, we leverage ontologies over attribute values and constraints between attributes to selectively merge the categorical attributes of the child regions into those of a parent region. This merge process is dictated by the notion of *compatibility* of attribute values.

Definition 2.1. Given an ontology for the values of a categorical attribute, let R be a subset of unrelated values, that is, for any two values in R, neither is descendant of the other in the ontology. We call the values in R, *reference values*. We say that two attribute values, v_1 and v_2 are compatible with respect to R, or, simply, *compatible*, if there exists $v \in R$ such that v_1 and v_2 are descendants of v.

For example, in the land use ontology of Figure 3, let the subset of reference values be $R = \{\text{Commercial, Residential}\}$. We have that attribute values Hotel-or-motel and Office are compatible since they are in the subtree of Commercial. Also, attribute values Single-family-attached and Multi-family are compatible since they are in the subtree of Residential. However, Single-family-attached and Office are not compatible.

2.2 Preprocessing: Overlay

The SeA-RT data structure is built from an input collection of heterogeneous geospatial datasets. Integrating heterogeneous spatial datasets is a problem with a rich history [6, 25]. In a preprocessing step, we leverage a standard technique called *polygon overlay*. The overlay operation [24] involves super-imposing two or more thematic maps (or data layers), to produce a composite map derived from the intersection of polygons in the individual data layers. The attributes and respective values for each *derived polygon* in the output layer are obtained by combining the corresponding attributes and values from the input layers. Figure 4.c illustrates the effect of overlaying the two input datasets from Figures 4.a and 4.b.

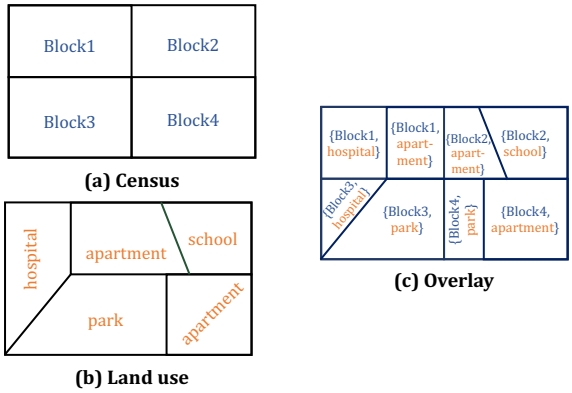

Figure 4: Overlay of Census and land use datasets.

Broadly, overlays can be carried out using different functions [7], depending on the geometry and the number of data layers (e.g., *intersection, union, symmetric difference*). Many GIS products, such as QGIS [35] and ArcGIS [18], support overlay of multiple datasets.

The preprocessing step consists of an overlay computation that involves additional work due to the presence of semantic descriptors. The output of the preprocessing step is a new geospatial dataset, denoted $\mathcal{D}^{ov}$, whose regions are the derived regions:

$$\mathcal{D}^{ov} = \{(oid_1, B_1^{ov}, S_1^{ov}), \cdots, (oid_n, B_n^{ov}, S_n^{ov})\} \quad (1)$$

In $\mathcal{D}^{ov}$, the object ID of a region is the list of the object IDs of its generating regions, that is, the regions from each input dataset whose intersection is the derived region.

We now explain how semantic descriptors are combined in the overlay process. For simplicity, we refer to the case of an overlay involving two datasets, $\mathcal{D}_1$ and $\mathcal{D}_2$ (see, for example, Figure 4).

- The list of attributes for the regions of $\mathcal{D}^{ov}$ is the union of the lists of attributes of $\mathcal{D}_1$ and $\mathcal{D}_2$. For simplicity, we assume that no value conflicts arise if some attributes are present in both $\mathcal{D}_1$ and $\mathcal{D}_2$.
- Following the standard principle of areal interpolation, the value of a numerical attribute of an input region is distributed to its derived regions proportionally to their areas (each dataset includes attribute *area*). However, the distribution method can be different when there are constraints between attributes. For example, in the overlay of Figure 2, Census block B_1 with population 4000 has two derived regions. Given the constraint that a park has no population, we set *population* = 4000 for the derived region with *land_use* Residential and *population* = 0 for the derived region with *land_use* Park, instead of distributing the population proportionally to the areas of the derived regions.

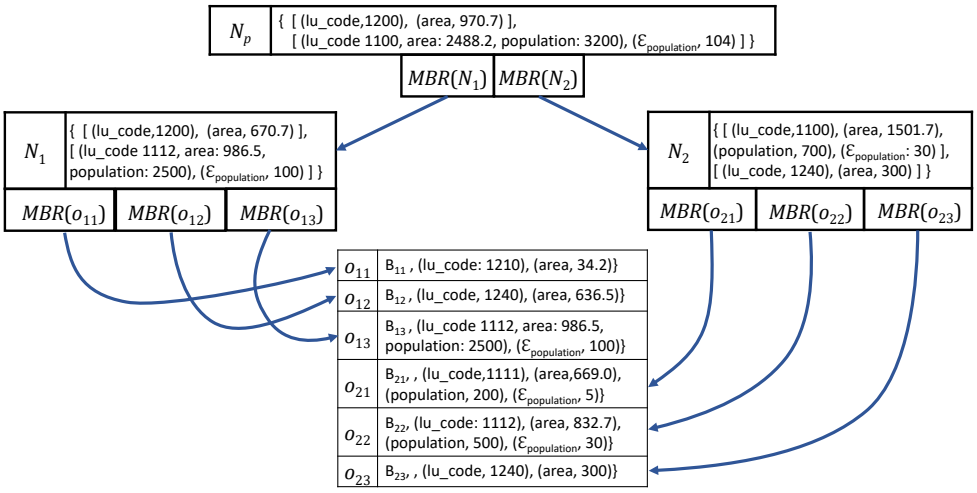

Figure 5: Example of a SeA-RT for the integration of a Census dataset with a land use dataset.

2.3 Semantically Augmented R-Tree (SeA-RT)

The R-tree [26] and its variants [20] are among the most popular structures used for indexing spatial data. An R-tree of order (m, M) has a root node with at least two entries (unless it is a leaf), and the internal and leaf nodes can store between $m \leq M/2$ and M entries. Internal nodes consist of a sequence of pointer ptr to a child node and the corresponding *Minimum Bounding Rectangle (MBR)* enclosing all the entries contained in the sub-tree rooted in that child. Each leaf node consists of pointers to actual data objects and corresponding *MBR*s.

A *semantically augmented R-tree (SeA-RT)*, illustrated in Figure 5, is a height-balanced tree that extends the R-tree by incorporating semantic information in its nodes. The tree indexes the overlay of two or more heterogeneous datasets with semantic descriptors, $\mathcal{D}^{ov}$ (Equation 1).

The nodes of the SeA-RT are structured as follows. A leaf node contains the polygonal boundary and list of semantic descriptors of a region of $\mathcal{D}^{ov}$. The innovative aspect of the SeA-RT is how an internal node *aggregates* the semantic descriptors of its children by using categorical attributes to estimate the numerical attributes.

An internal node, v, is associated with the region that is the union of the regions of the leaves in its subtree. Node v stores a list of semantic descriptors for its region, which is obtained by aggregating those of the regions of its children. In addition, node v stores the *MBR* of each child.

Aggregation of attribute values. In the following, for simplicity, we describe the process of aggregating semantic descriptors of the children of a node v for the case where there is a single categorical attribute, a, and multiple numerical attributes, $b_1, \cdots, b_k$. The method can be easily generalized to multiple categorical attributes.

Recall that there is an ontology for the values of the categorical attribute (see, for example, the ontology of Figure 3 for land use). Also, we assume that we have selected a set of reference values in the ontology so that the compatibility relation between values is well defined (Definition 2.1). The aggregation method is as follows:

- Consider the values assumed by categorical attribute a at the children of v. We partition these values into maximal subsets of compatible values. For each such compatibility subset, X, we store at node v the semantic descriptor $(a, LCA(X))$, where LCA denotes the *least common ancestor* of the values in subset X in the ontology. Thus, the list of semantic descriptors of node v may have multiple values for the same attribute.
- Each numerical attribute is aggregated by summing the values at the children over each compatibility subset of the categorical attribute.

Overall, the aggregation process results in node v storing a list of descriptors comprising sublists $((a, x_i), (b_1, y_{i1}), \cdots, (b_k, y_{ik}))$, one for each value x_i of attribute a that results from aggregating a compatibility subset.

Error aggregation. The aggregation of numerical attributes implies a corresponding aggregation of their errors. In particular, we rely on the theory of statistical error propagation in uncertain measurements [41]. Let $x_1, x_2, \ldots, x_k$ be values of a numerical attribute, where x_i has associated error (i.e., standard deviation) ϵ_i. We have that the error ϵ associated with $x = x_1 + x_2 + \ldots + x_k$ is given by

$$\epsilon = \sqrt{\epsilon_1^2 + \ldots + \epsilon_k^2}. \quad (2)$$

Note that the error of the sum is always greater than or equal to the error of each of its terms. Thus, as we aggregate numerical attributes from children to parent in the SeA-RT, we increase the associated error.

Example. An instance of SeA-RT is shown in Figure 5. Its regions, o_{11} through o_{23}, are the result of the overlay of two datasets: (1) a land use dataset with attributes *area* (numerical) and *lu_code* (categorical) with values from the ontology of Figure 3, where $R = \{\text{Commercial, Residential}\}$ is the subset of reference values (see Definition 2.1); and (2) a Census dataset with numerical attributes *area*, *population*, and $\epsilon_{population}$, which denotes the error associated with the population. The interior of the polygonal boundaries B_{11} through B_{23} are disjoint.

Node N_1 is the parent of o_{11}, o_{12}, and o_{13}. The land use values of o_{11} and o_{12}, Primary-retail-or-service (1210) and Cultural-or-entertainment (1240) are compatible. Thus, they are aggregated

into their *LCA*, Commercial (1200), in the land use ontology, and the corresponding area values, 636.5 and 34.2, are aggregated into their sum, 670.7. In contrast, o_{13} has land use value Single-family-attached (1112), which is incompatible with those of o_{11} and o_{12}. Thus, its land use value and the corresponding area, population and error values are carried over to N_1.

Next, consider node N_2. Children (o_{21} and o_{22}) have compatible land use values, Single-family-detached (1111) and Single-family-attached (1112), which are aggregated into their *LCA*, Single-family-residential (1110). Regarding numerical attributes *area* and *population*, their values are aggregated by summing them. Also, the errors for attribute population are aggregated, yielding error $\sqrt{5^2 + 30^2} = 30.4$ (rounded to 30). The third child o_{23} has an incompatible land use value, which is carried over to N_2.

Finally, node N_p with children N_1 and N_2 aggregates separately the attribute values compatible with Commercial (1200) and Residential (1100), respectively, yielding a total population count of 3200 and an error of $\sqrt{100^2 + 30^2} = 104.4$ (rounded to 104).

Complexity. We close this section by analyzing the space complexity of the SeA-RT. Let $n = |\mathcal{D}^{ov}|$ be the size of the integrated dataset indexed by the SeA-RT and let τ denote the number of maximal subsets of compatible attribute values in the ontology. We have that the space complexity of the SeA-RT is $O(\tau \cdot n)$.

3 QUERY PROCESSING

We now describe the syntax and semantics of a SeA-RQ and discuss its efficient processing using the SeA-RT data structure presented in Section 2.

3.1 Syntax and Error Calculation

A query $q \in$ *SeA-RQ* is a conjunctive query over spatial data augmented with semantic attributes, corresponding to $\mathcal{D}^{ov}$ (Section 2). For example:

SeA-RQ$_1$: *Retrieve the total park area and total population of the zones intersecting a given region* Q_R.

However, we recall that some of the numerical attributes in $\mathcal{D}^{ov}$ are associated with error due to imprecise measurement, implying uncertainty which is propagated along the hierarchy of a SeA-RT. As such, it needs to be explicitly captured in the syntax of the query, as well as in the processing algorithms [43]. In our case, this amounts to expressing the bounds on the error on certain numerical attributes that the user is willing to tolerate.

The following query uses two categorical attributes (land use and dominant political party) and one numerical attribute (population) in a country with multiple political parties that are arranged in an ontology with superclasses conservative and liberal:

SeA-RQ$_2$: *Retrieve the total population within a region* Q_R*, with error at most* 200*, living in single-family homes and in electoral precincts where the dominant party is of type liberal.*

Given the information in a SeA-RT, the bounds on the uncertainty can enable earlier pruning of some nodes (and corresponding sub-trees) when processing a SeA-RQ, which, as we will show in Section 4, can yield more efficient processing.

A SeA-RQ on geospatial dataset $\mathcal{D}^{ov}$ (Equation 1) can be specified as follows:

$$(Q_R, [S_i, \alpha_i, f_i]) \tag{3}$$

where

(1) Q_R is the spatial region of interest;
(2) S_i is the semantic attribute of interest;
(3) α_i is the error tolerance threshold on S_i which is applicable only to numerical attributes and has a null value for categorical attributes;
(4) f_i is the query function that extracts the portions of the geospatial objects within region Q_R and returns an estimate of the aggregate summary of semantic attribute S_i for the union of the portions, subject to constraints on the other semantic attributes and up to error tolerance α_i ;

For example, in *SeA-RQ*$_1$ above, we have three functions: f_1 over the categorical attribute land use (Park), f_2 over the boundary of the parks (area), and f_3 which is (sum) over the numerical attribute (Population), which is the only one associated with an error.

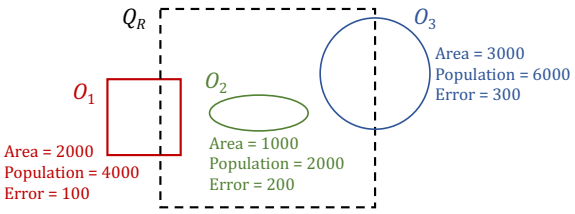

Figure 6: Areal interpolation.

To calculate the value to be reported in the answer along with the cumulative error from multiple regions with numeric attributes that are intersecting Q_R, we use areal interpolation. Specifically, we assume that the values of the numeric attributes and the errors are uniformly distributed across the regions and we weight each of them proportionally to the fraction of the area intersecting Q_R. For clarity, in the rest of this section we focus on queries of the form:

SeA-RQ$_{pc}$: *Retrieve the total population of the inhabited zones intersecting the region* Q_R*, with error at most* α.

Example. Consider the scenario shown in Figure 6, where objects O_1, O_2, and O_3 of $\mathcal{D}^{ov}$ have the same value for the categorical attribute (e.g., land use is residential (1100)) and intersect query region Q_R. Assume that O_1 has 25% overlap with Q_R, O_2 is fully contained in Q_R, and O_3 has 50% overlap with Q_R. Given the values for the area, population, and population error of each object shown in Figure 6, we obtain the following aggregate values for intersection of the objects with Q_R:

- $Area_{\cap Q_R} = 0.25{\cdot}2000 + 1000 + 0.5{\cdot}3000 = 3000$
- $Population_{\cap Q_R} = 0.25{\cdot}4000 + 2000 + 0.5{\cdot}6000 = 6000$
- $Error_{\cap Q_R} = \sqrt{(0.25 \cdot 100)^2 + 200^2 + (0.5 \cdot 300)^2} = 251.2$

3.2 Efficient SeA-RQ Processing

In a nutshell, processing a SeA-RQ involves descending the SeA-RT and determining the candidate nodes (i.e., the ones intersecting Q_R). However, it also further refines each candidate node by using the values of the semantic descriptors and associated errors, taking into account the given error-tolerance α.

Before we proceed with the details of the algorithm, we need to define a few terms that are used when executing the respective

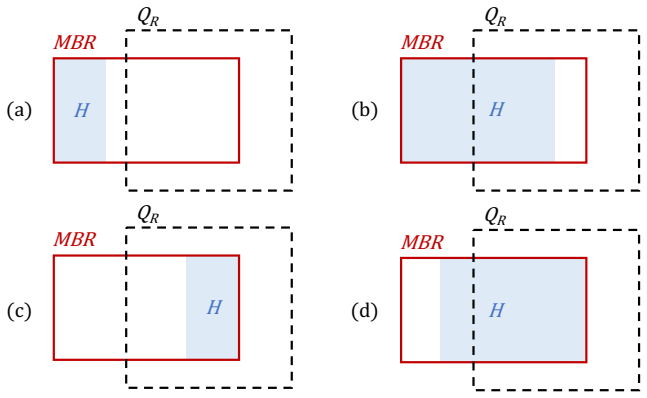

Figure 7: The four different cases of populated land as contained in the intersection of *MBR* and Q_R.

computations. Specifically, since the inner nodes (and the *MBR*s) contain aggregated values for the numerical attributes and the errors, we need to explain how those are obtained based on the intersection of a particular *MBR* with Q_R.

We now show how to estimate the output of the query that returns the population, P, in the intersection of the query range, Q_R with the minimum bounding rectangle, *MBR* of a node N. With reference to Figure 7, let H denote the inhabited portion of *MBR*, that is, the union of regions whose land use attribute value is a descendant of Residential. Also, let A_H denote the area of H. By selecting Residential as a reference attribute value in the land use ontology, we can keep track of A_H in the SeA-RT, but not of the exact boundary of H. If we were to have a full knowledge of the boundaries of the collection of all the nodes within the *MBR* of N (clearly, a huge overhead of replication), then we could compute the area of the inhabited portion in the intersection of *MBR* and Q_R, denoted A_{H_Q}, and we would estimate the query output as

$$A_{H_Q} \cdot \frac{P}{A_H} \tag{4}$$

Since the exact value of A_{H_Q} is not known from the aggregate information available at the current node, we estimate it by using the following modified areal interpolation method:

$$\widehat{A}_{H_Q} = \frac{A_{h_{\min}} + A_{h_{\max}}}{2} \tag{5}$$

$$A_{h_{\min}} = \max(0, A_H - Area(MBR - Q_R)) \tag{6}$$

$$A_{h_{\max}} = \min(A_H, Area(MBR \cap Q_R)) \tag{7}$$

Namely, Equation 5 estimates A_{H_Q} as the average of its minimum and maximum possible values. Equation 6, gives the minimum value, $h_{\min}$, for H_Q, as illustrated in Figure 7.a ($H_Q = 0$) and Figure 7.b ($H_Q = H - Area(MBR - Q_R)$). Equation 7, gives the maximum value, $A_{h_{\max}}$, for the area of H_Q, as illustrated in Figure 7.c ($H_Q = H$) and Figure 7.d ($A_{H_Q} = Area(MBR \cap Q_R)$).

Thus, for any numerical attribute a^{num}, its value in the the portion of N's *MBR* intersected with the query region Q_R, denoted $a^{num}_{Q_R}$, and the corresponding portion of the associated error, denoted $\epsilon^{a^{num}}_{Q_R}$, are obtained via areal interpolation as follows:

$$a^{num}_{Q_R} = \left(\frac{a^{num}}{A^{a^{num}}_H}\right) \cdot \widehat{A}^{a^{num}}_{H_Q} \tag{8}$$

$$\epsilon^{a^{num}}_{Q_R} = \epsilon^{a^{num}} \cdot \left(\frac{a^{num}_{Q_R}}{a^{num}}\right) = \epsilon^{a^{num}} \cdot \left(\frac{\widehat{A}^{a^{num}}_{H_Q}}{A^{a^{num}}_H}\right) \tag{9}$$

The next step is to refine the entries in the candidate nodes based on the function(s) f_j specified in the query (Equation 3). The entries are filtered based on the constraints on the semantic descriptors (e.g., land use category is Residential). The algorithm checks each entry in candidate nodes, and detects and prunes those entries that do not satisfy the constraints specified in the query. If the query involves retrieving the value of a numerical semantic descriptor in the query region Q_R, the algorithm returns the corresponding value $a^{num}_{Q_R}$ (Equation 8).

To describe query processing in a SeA-RT, consider a query that estimates the total population inside query region Q_R with error tolerance $\alpha = 5$. We illustrate in Figure 8 the processing of this query at an inner node, N_1, where land use codes are from Figure 3. Let $A_{MBR(N_1)} = 10000$. Since $MBR(N_1) \cap Q_R \neq \emptyset$, N_1 is a *candidate* node. Assume $A_{Q_R \cap MBR(N_1)} = 0.5 \cdot A_{MBR(N_1)} = 5000$. There are two sources of errors in the population count of N_1. One is aggregated from children C_{12} and C_{13} ($\sqrt{12^2 + 5^2} = 13$), as their land use attributes are compatible and merged into land use code 1200. The other is carried over from C_{14}, whose land use code (1130) is not compatible with those of C_{12} and C_{13}, and introduces a separate error (4). Note that C_{11} does not have the population attribute.

Let A^{1200}_H denote the area of the inhabited portion of $MBR(N_1)$ with land use code 1200 (i.e., due to C_{12} and C_{13}). As shown in Figure 8, $A^{1200}_H = 1225.6$. Thus, we have $A^{1200}_H < 5000 = A_{MBR(N_1) \cap Q_R} = A_{MBR(N_1) \setminus Q_R}$. Using Equation 5, we estimate the populated area with land use code 1200 inside $MBR(N_1) \cap Q_R$, as $\frac{1}{2} \cdot (0 + 1225.6) = \frac{1}{2} \cdot A^{1200}_H$. By Equation 9, the error associated with the estimate of the population for land use code 1200 in $MBR(N_1) \cap Q_R$ is $13 \cdot \frac{1}{2} = 6.5$. Similarly, the corresponding error for land use code 1130 is $4 \cdot \frac{1}{2} = 2$. Thus, the cumulative error for the estimated populated area inside

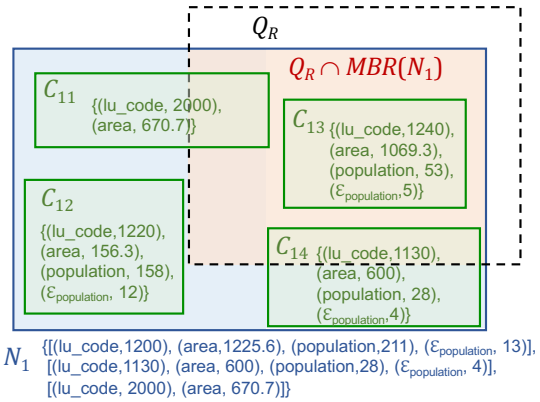

Figure 8: Processing an inner node of a SeA-RT.

$Q_R \cap MBR(N_1)$ from both land use codes is $\sqrt{6.5^2 + 2^2} \approx 6.8$, exceeding the tolerance error $\alpha = 5$. Thus, the algorithm proceeds with examining the children of N_1, whereby:

(1) The *MBR* of C_{11} intersects Q_R, but C_{11} is not a candidate as its land use, Agriculture (2000), is incompatible with population.
(2) C_{12} is not a candidate as its *MBR* does not intersect Q_R.
(3) Each of C_{13} and C_{14} is a candidate, but they have incompatible land use codes. Thus, we calculate their errors separately:
   (a) For C_{13}, $\epsilon_{population} = 5$ as its *MBR* is entirely contained within Q_R.
   (b) For C_{14}, assuming $A_{MBR(C_{14})} = 900$ and Q_R intersects one third of it, the residential area (600) can be either completely outside the intersection, or 50% inside of it. Thus, the associated error is $\frac{1}{2}(0 + 0.5) \cdot 4 = 1$ (Eq. 9)

   The combined error from C_{13}, and C_{14} is $\sqrt{5^2 + 1^2} \approx 5.09$, which still exceeds the user's tolerance. Thus, the algorithm continues by processing the children of C_{13} and C_{14}.

4 EXPERIMENTAL RESULTS

To quantitatively evaluate the benefits of SeA-RT on the efficiency of SeA-RQ processing, we conducted a comprehensive set of experiments using real census and land use datasets, and a synthetic dataset that we generated. We examined two basic variants: (1) queries involving attributes with numeric values, with and without error bound (e.g., retrieve the population count inside Q_R, with an error bound α); (2) queries involving attributes with categorical values (e.g., retrieve all the single family homes inside Q_R).

The experiments were run on macOS Sierra (version 10.12.1), with an Intel Core i7 CPU (3.1GHz with 16GB RAM) for the census and land use dataset, and on an AWS EC2 instance (t2.medium with 4GB RAM) for the synthetic dataset. The code implementing SeA-RT and the algorithms for processing the variants of SeA-RQ, as well as the datasets used, are publicly available on GitLab [4].

Datasets. We used two datasets of different sizes (Table 1):

- *DS1 (small).* We started with three datasets with attributes Census, Census Block Boundaries, and Land Use. Since spatial attributes and population data are not available in a single dataset, we integrated datasets Population by 2010 census block and Boundaries - census blocks - 2010), obtained from the City of Chicago's open data portal [10]. The Land use dataset is retrieved from CMAP's land use inventory [9], consisting of data for Northeast Illinois including the City of Chicago. The land use polygons are derived directly from parcel GIS files, which allows for greater accuracy compared to the "polygon-based" inventory. The SeA-RT was built on the overlay of the three data sources, resulting in an output layer which contains more polygons than the input layers combined [22, 32]. However, the space was far from the product of the cardinalities of the individual inputs. For example, overlaying 46311 polygons from Census Block Boundary and 506274 polygons from Land use, resulted in 552631 polygons.
- *DS2 (large).* We randomly generated 2 million records (rectangles, pentagons, hexagons and octagons) representing a spatial unit corresponding to a census block. For the semantic attributes associated with each polygon, we randomly assigned values for population, land use (from the classification scheme in CMAP's land use inventory [9]), and error on the population. The values for the areas were calculated during the generation of the polygons.

Dataset	# Polygons	Population	Error range	Land categories
DS1	555,632	2,024,373	[9, 110]	59
DS2	2,000,000	73,975,267	[2, 20]	59

Table 1: Properties of datasets *DS1* and *DS2*.

We note that the respective size of the index files were: 71MB for the SeA-RT and 38.9MB for the R-tree without any semantic descriptors for *DS1*; and 555.2MB for the SeA-RT and 311.9MB for the R-tree without any semantic descriptors for *DS2*.

Workload and Baseline. The experiments are based on executing 1000 queries with different parameters (Q_R, error threshold, land use), and the average of runs is reported. Specifically: (1) The query regions were rectangles, with varying size in terms of the overall spatial region covered by the datasets. (2) For selecting the tolerance threshold in SeA-RQ, we scanned the error values in each dataset to determine the interval of all the values. Then we randomly picked values within a certain percentile from the intervals.

The baseline that we used for comparison is the *(plain) R-tree* (i.e., indexing only the polygons/spatial data). When processing a SeA-RQ with an R-tree, we first determined the input polygons intersecting Q_R, and subsequently evaluated the other constraints (in terms of α and categorical attributes).

Figure 9 shows the comparison of execution times for SeA-RT vs. R-tree as a function of the size of the query region Q_R. The solid polylines pertain to *DS1* and the dashed ones to *DS2*, with a note that the time scales are indicated on the right side and left side of the Y-axis, respectively. The values for the error threshold were uniformly chosen from the respective ranges in each of *DS1* and *DS2*. We observe that SeA-RT executes 2-3 times faster.

We also evaluated the number of nodes accessed by each of SeA-RT and R-tree, using the same settings for the other parameters' selection. The results are shown in Figure 10. We observe that in Figure 10.b, where we also show Q_R of a size 80% of the total

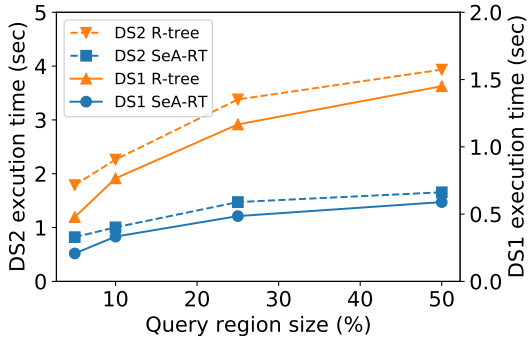

Figure 9: Average query time: R-tree vs. SeA-RT.

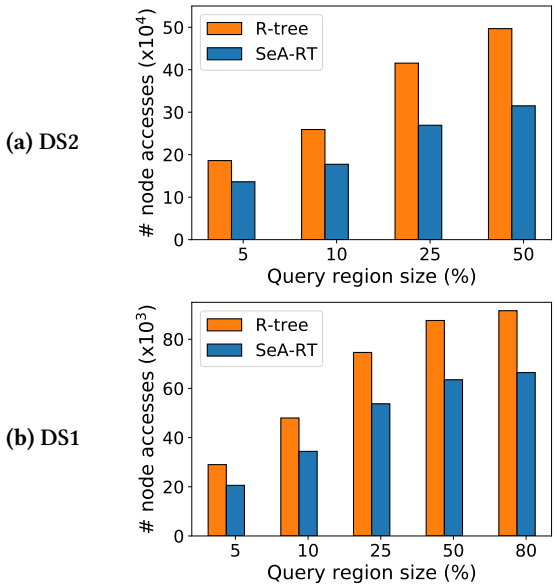

Figure 10: Average number of node accesses by query region size: R-tree vs. SeA-RT.

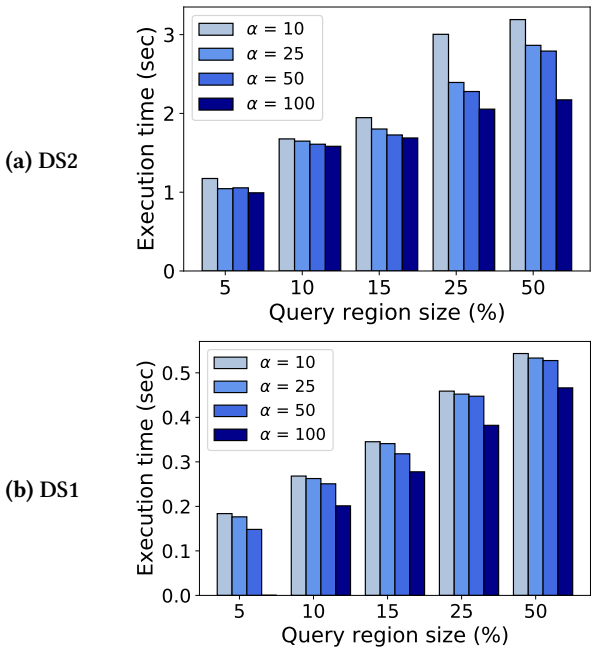

Figure 11: Impact of error tolerance on query time by query region size in a SeA-RT.

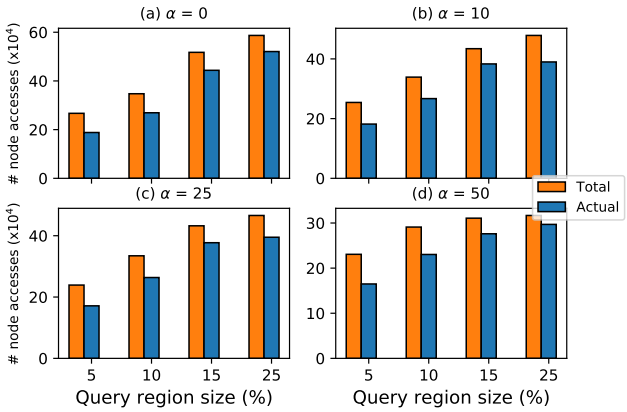

Figure 12: Impact of error tolerance on the number of node accesses by query region size in a SeA-RT (dataset *DS2*).

area, the ratio of nodes accessed by SeA-RT vs. R-tree is $> \frac{1}{2}$, not quite matching the ratio of the respective running times shown in Figure 9. However, there is an explanation for such behavior. Namely, the R-tree will only check for intersections with Q_R, as it does not have any descriptors embedded. Thus, in addition to accessing the inner nodes, answering SeA-RQs with R-trees has an additional overhead of checking the data objects for qualifying with respect to the attributes' values in the semantic descriptors.

Next, we report the impact of the error tolerance threshold on the query execution time on the SeA-RT. Figure 11 shows that, as expected, the execution time is proportional to the size of the query region. However, for smaller query regions, the impact of the error tolerance α is less significant. As the size of Q_R increases, α becomes more impactful in the sense that larger values imply faster processing time as the descent down the SeA-RT can stop sooner.

Experiments providing further insight into the impact of error tolerance α are presented in Figure 12, which shows the number of nodes accessed by SeA-RQs on dataset *DS2* as a function of the size of Q_R. Each of the four charts refers to a specific value of α, hence the range of values on the vertical axes varies (the larger α the fewer nodes are accessed). In these experiments, we have compared the number of nodes visited when processing queries using the SeA-RT, denoted *actual*, with the number of nodes that would be visited if one were to access all the nodes that individually satisfy both the spatial range *and* the semantic constraints, denoted *total*. The experiments show that the aggregation along the hierarchy in the SeA-RT enables visiting fewer nodes.

In particular, Figure 12.a reports on actual vs. total for $\alpha = 0$, that is, when exact query answers are required. In this case, the total number of accesses is the same as in the R-tree, hence the chart compares the SeA-RT with the R-tree on exact query processing. It is interesting that even in this case, the SeA-RT outperforms the R-tree. This is due to the benefit of performing aggregation on the categorical values in the ontology, a feature not available in the R-tree.

We also considered separately the impact of semantic descriptors with categorical values for the attributes on the efficiency. Specifically, we used SeA-RQs with different values for the land use, selected from the ontology of Figure 3. Figure 13 shows the execution times for different values of Q_R on the X-axis, and the comparative bars for the land use (LU) codes. As can be seen, if the

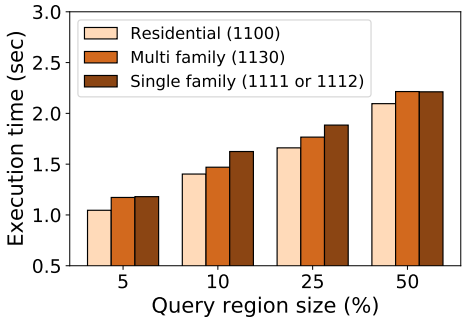

Figure 13: Impact of categorical attributes on query time (dataset *DS2*).

query goes to the level of detail 1100 (i.e., the Residential category), which is the root of the (sub)hierarchy for the other values, due to the nature of the augmentation in SeA-RT's hierarchy, the execution time is fastest because the descent can terminate earlier in the inner nodes.

We note that the difference between the land use codes of 1130 vs. (1111 of 1112) is negligible for the smallest and largest values of Q_R. The reason is the proximity in the hierarchy, which is reflected in the construction of SeA-RT. In smaller regions, there are fewer merges of 1111 and 1112 codes in the hierarchy, whereas for large regions, the processing will have to investigate many nodes.

5 RELATED WORK

Our work touches on many subjects related to spatial data structures, querying, data integration, and uncertainty. We group related work into two topics: *Augmented spatial indexes*, which includes efficient query processing and uncertainty, and *Combined geospatial data* for population estimation.

Augmented spatial indexes. Efficient processing of spatial and spatio-temporal queries (e.g., range, nearest-neighbor) over uncertain data has been investigated for over two decades [29, 42]. In particular, the R-tree, the R*-tree, and their extensions have been widely used for spatial data querying. Starting with the classical problem of map overlay, where efficient detection of intersecting segments is key [47], the R-tree has been used in such a way that more nodes are generated in areas with many edge segments, thus adapting to non-uniform distributions [44]. To efficiently process queries combining spatial and textual information for location-based web search (e.g., find web content related to a given place or region), the R* tree has been coupled with inverted files to enable pruning in both the textual and spatial dimensions [50]. Also, combining spatial and textual information for efficient query processing has been supported by extending the R-tree with signature files, the IR^2 tree [19], and with motion attributes (e.g., bus, walk) associated with a trajectory, the IRWI tree [27].

A variation of R–trees for probabilistic range queries over existentially uncertain objects was presented in [17]. Efficient processing of spatial range queries over multidimensional uncertain objects using a new spatial data structure, the U-tree, was presented in [40]. It enabled early pruning of subtrees that either do not intersect with the query region, or already satisfy a probabilistic threshold on the error. Our SeA-RT also extends the R-tree with additional semantic information. However, we allow for multiple types of semantic descriptors, ontologies for categorical attributes, and the notion of acceptable bound on the uncertainty due to the imprecision of the data in both the index structure and the query processing algorithm. Although less directly related to our work, other spatial indexes have been extended with semantic/textual information, such as combining inverted files with space-filling curves [8].

Combined geospatial data. The combination of disparate data has been a "persistent and perplexing" problem [25]. Facets of this problem include areal interpolation and polygonal overlay, which we use in our paper.

The problem to obtain spatially disaggregated population estimates has two sides: one where there is the absence of national population (e.g., in low income countries) and the other where housing census is unreliable. In the former case, the process involves a bottom up approach and, in the latter case, a top down approach. Both use covariates such as distance to major roads to improve the predictive accuracy [45]. Earlier work is based on areal interpolation and statistical modeling [48] and on a dasymetric mapping method, which transfers from a spatial unit system to another one using ancillary datasets [30, 31]. Examples of ancillary datasets include those captured by satellite remote sensing like land cover data. LIDAR remote sensing is used to derive building volumes as a population indicator thus using a third dimension, which is the height of the buildings [36, 37].

We have used the land use ancillary dataset in addition to areal interpolation to estimate population in a query region.

6 CONCLUSIONS AND FUTURE WORK

We have proposed semantically augmented range queries (SeA-RQ). which enable retrieving geospatial data enriched with semantic descriptors, organized in an ontology. Our approach integrates different kinds of semantic descriptors, both categorical and numerical, and incorporates the uncertainty of numerical values in real-world data sources. For efficient processing of a SeA-RQ, we have presented a novel indexing structure, the semantically augmented R-tree (SeA-RT), which aggregates spatial shapes together with semantic descriptor by leveraging ontologies and statistical error propagation for efficiency. An experimental evaluation of our approach has been conducted on real and synthetic datasets.

A first direction of future work is to extend our aggregation methods and areal interpolation techniques datasets with nonuniform spatial distributions. Another direction is to broaden the spectrum of aggregation functions, such as median. We also plan to study updates on a SeA-RT: insertions and deletions of objects originate at the leaves and propagate up the tree, requiring methods to efficiently update the semantic information at affected nodes. Finally, a major challenge is to seamlessly integrate the notion of semantic similarity in the construction of the SeA-RT and investigate its impact on efficient processing of SeA-RQs.

ACKNOWLEDGMENTS

This work was partially supported by NSF awards CNS-1646395, III-1618126, and CNS 1646107.

REFERENCES

[1] Booma Sowkarthiga Balasubramani, Omar Belingheri, Eric S. Boria, Isabel F. Cruz, Sybil Derrible, and Michael D. Siciliano. 2017. GUIDES: Geospatial Urban Infrastructure Data Engineering Solutions. In *ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*.

[2] Booma Sowkarthiga Balasubramani and Isabel F. Cruz. 2019. Spatial Data Integration. In *Encyclopedia of Big Data Technologies.*, Sherif Sakr and Albert Zomaya (Eds.). Springer. https://doi.org/10.1007/978-3-319-63962-8_218-1

[3] Booma Sowkarthiga Balasubramani, Vivek R. Shivaprabhu, Smitha Krishnamurthy, Isabel F. Cruz, and Tanu Malik. 2016. Ontology-based Urban Data Exploration. In *ACM SIGSPATIAL International Workshop on Smart Cities and Urban Analytics (UrbanGIS) (UrbanGIS)*. ACM, Article 10, 8 pages.

[4] Booma Sowkarthiga Balasubramani and Xu Teng. 2020. SUARTree Project. GitLab. https://gitlab.com/bbalas3/suartree

[5] Graeme Francis Bonham-Carter, Frederik Pieter Agterberg, and Donald F. Wright. 1988. Integration of Geological Datasets for Gold Exploration in Nova Scotia. *Photogrammetric Engineering and Remote Sensing* 54, 11 (1988), 1585–1592.

[6] Matthias Butenuth, Guido v. Gösseln, Michael Tiedge, Christian Heipke, Udo Lipeck, and Monika Sester. 2007. Integration of Heterogeneous Geospatial Data in a Federated Database. *ISPRS Journal of Photogrammetry and Remote Sensing* 62, 5 (2007).

[7] Jonathan E. Campbell and Michael Shin. 2011. *Essentials of Geographic Information Systems.* Saylor Foundation.

[8] Yen-Yu Chen, Torsten Suel, and Alexander Markowetz. [n. d.]. Efficient Query Processing in Geographic Web Search Engines. In *ACM SIGMOD International Conference on Management of Data*, Surajit Chaudhuri, Vagelis Hristidis, and Neoklis Polyzotis (Eds.).

[9] Chicago Metropolitan Agency for Planning (CMAP). (Accessed September 10, 2019). Land Use Inventory. https://www.cmap.illinois.gov/data/land-use/inventory.

[10] City of Chicago. (Accessed September 10, 2019). Chicago Data Portal. https://data.cityofchicago.org/.

[11] A. J. Clark, Patton Holliday, Robyn Chau, Harris Eisenberg, and Melinda Chau. 2010. Collaborative Geospatial Data as Applied to Disaster Relief: Haiti 2010. In *Security Technology, Disaster Recovery and Business Continuity*. Springer, 250–258.

[12] Thomas J. Cova. 1999. GIS in Emergency Management. In *Geographical Information Systems: Principles, Techniques, Applications, and Management*, P.A. Longley, M.F. Goodchild, D.J. Maguire, and D.W. Rhind (Eds.). John Wiley & Sons, Chapter 60, 845–858.

[13] Isabel F. Cruz, Flavio Palandri Antonelli, and Cosmin Stroe. 2009. AgreementMaker: Efficient Matching for Large Real-world Schemas and Ontologies. *PVLDB* 2, 2 (2009), 1586–1589.

[14] Isabel F. Cruz and Afsheen Rajendran. 2003. Semantic Data Integration in Hierarchical Domains. *IEEE Intelligent Systems* March-April (2003), 66–73.

[15] Isabel F. Cruz and Huiyong Xiao. 2008. Data Integration for Querying Geospatial Sources. In *Geospatial Services and Applications for the Internet*, John Sample, Kevin Shaw, Shengru Tu, and Mahdi Abdelguerfi (Eds.). Springer, 113–137.

[16] F. C. Dai, C. F. Lee, and X. H. Zhang. 2001. GIS-based Geo-environmental Evaluation for Urban Land-use Planning: A Case Study. *Engineering Geology* 61, 4 (2001), 257–271.

[17] Xiangyuan Dai, Man Lung Yiu, Nikos Mamoulis, Yufei Tao, and Michail Vaitis. 2005. Probabilistic Spatial Queries on Existentially Uncertain Data. In *International Symposium on Spatial and Temporal Databases (SSTD)*. Springer, 400–417.

[18] Environmental Systems Research Institute, Inc. (Esri). 2010. ArcGIS 9.2 Desktop Help.

[19] Ian De Felipe, Vagelis Hristidis, and Naphtali Rishe. 2008. Keyword Search on Spatial Databases. In *International Conference on Data Engineering (ICDE)*.

[20] Volker Gaede and Oliver Günther. 1998. Multidimensional Access Methods. *ACM Comput. Surv.* 30, 2 (1998), 170–231.

[21] GDAL/OGR contributors. 2020. *GDAL/OGR Geospatial Data Abstraction software Library.* Open Source Geospatial Foundation. https://gdal.org Accessed September 14, 2020.

[22] Manual Gimond. 2017. Introduction to GIS and Spatial Analysis. https://mgimond.github.io/Spatial/index.html

[23] Michael F. Goodchild. 1989. Modeling Error in Objects and Fields. In *The Accuracy of Spatial Databases*. Taylor & Francis, 107–113.

[24] Michael F. Goodchild and Sucharita Gopal. 1989. *The Accuracy of Spatial Databases.* CRC Press.

[25] Carol A. Gotway and Linda J. Young. 2002. Combining Incompatible Spatial Data. *J. Amer. Statist. Assoc.* 97, 458 (2002), 632–648.

[26] Antonin Guttman. 1984. R-trees: A Dynamic Index Structure for Spatial Searching. In *ACM SIGMOD International Conference on Management of Data*. ACM, 47–57.

[27] Hamza Issa and Maria Luisa Damiani. 2016. Efficient Access to Temporally Overlaying Spatial and Textual Trajectories. In *IEEE International Conference on Mobile Data Management, MDM*. 262–271.

[28] Charalambos Kontoes, G. G. Wilkinson, A. Burrill, S. Goffredo, and J. Megier. 1993. An Experimental System for the Integration of GIS Data in Knowledge-based Image Analysis for Remote Sensing of Agriculture. *International Journal of Geographical Information Systems* 7, 3 (1993), 247–262.

[29] Rui Li, Bir Bhanu, Chinya V. Ravishankar, Michael Kurth, and Jinfeng Ni. 2007. Uncertain spatial data handling: Modeling, indexing and query. *Comput. Geosci.* 33, 1 (2007), 42–61.

[30] Catherine Linard, Marius Gilbert, Robert W. Snow, Abdisalan M. Noor, and Andrew J. Tatem. 2012. Population Distribution, Settlement Patterns and Accessibility Across Africa in 2010. *PLOS ONE* 7, 2 (2012), 1–8.

[31] Catherine Linard, Marius Gilbert, and Andrew J. Tatem. 2011. Assessing the Use of Global Land Cover Data for Guiding Large Area Population Distribution Modelling. *GeoJournal* 76, 5 (2011), 525–538.

[32] David O'Sullivan and David Unwin. 2010. *Putting Maps Together – Map Overlay.* John Wiley & Sons, Chapter 11, 315–340.

[33] Dirk U. Pfeiffer, Timothy P. Robinson, Mark Stevenson, Kim B. Stevens, David J. Rogers, and Archie C. A. Clements. 2008. *Spatial Analysis in Epidemiology.* Oxford University Press.

[34] Dieter Pfoser, Nectaria Tryfona, and Christian S. Jensen. 2005. Indeterminacy and Spatiotemporal Data: Basic Definitions and Case Study. *GeoInformatica* 9, 3 (2005), 211–236.

[35] QGIS Development Team. 2009. QGIS Geographic Information System. http://qgis.osgeo.org Accessed September 14, 2020.

[36] Fang Qiu, Harini Sridharan, and Yongwan Chun. 2010. Spatial Autoregressive Model for Population Estimation at the Census Block Level Using LIDAR-derived Building Volume Information. *Cartography and Geographic Information Science* 37, 3 (2010), 239–257.

[37] Harini Sridharan and Fang Qiu. 2013. A Spatially Disaggregated Areal Interpolation Model Using Light Detection and Ranging-Derived Building Volumes. *Geographical Analysis* 45, 3 (2013), 238–258.

[38] D. Stevens, Suzana Dragicevic, and Kristina Rothley. 2007. iCity: A GIS-CA Modelling Tool for Urban Planning and Decision Making. *Environ. Model. Softw.* 22, 6 (2007), 761–773.

[39] William Sunna and Isabel F. Cruz. 2007. Structure-Based Methods to Enhance Geospatial Ontology Alignment. In *International Conference on GeoSpatial Semantics (GeoS) (Lecture Notes in Computer Science)*, Vol. 4853. Springer, 82–97.

[40] Yufei Tao, Xiaokui Xiao, and Reynold Cheng. 2007. Range Search on Multidimensional Uncertain Data. *ACM Transactions on Database Systems (TODS)* 32, 3 (2007), 15.

[41] John R. Taylor. 1997. *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements.* University Science Books.

[42] Goce Trajcevski, Roberto Tamassia, Isabel F. Cruz, Peter Scheuermann, David Hartglass, and Christopher Zamierowski. 2011. Ranking Continuous Nearest Neighbors for Uncertain Trajectories. *VLDB J.* 20, 5 (2011), 767–791.

[43] Goce Trajcevski, Ouri Wolfson, Klaus H. Hinrichs, and Sam Chamberlain. 2004. Managing Uncertainty in Moving Objects Databases. *ACM Trans. Database Syst. (TODS)* 29, 3 (2004).

[44] Peter van Oosterom. 1994. An R-tree Based Map-Overlay Algorithm. In *Proceedings EGIS*, Vol. 94. 318–327.

[45] N. A. Wardrop, W. C. Jochem, T. J. Bird, H. R. Chamberlain, D. Clarke, D. Kerr, L. Bengtsson, S. Juran, V. Seaman, and A. J. Tatem. 2018. Spatially Disaggregated Population Estimates in the Absence of National Population and Housing Census Data. *Proceedings of the National Academy of Sciences* 115, 14 (2018), 3529–3537.

[46] Stefan Werder. 2009. Formalization of Spatial Constraints. In *AGILE International Conference on Geographic Information Science*.

[47] Peter Y. F. Wu and W. Randolph Franklin. 1990. A Logic Programming Approach to Cartographic Map Overlay. *Computational Intelligence* 6, 2 (1990), 61–70.

[48] Shuo-sheng Wu, Xiaomin Qiu, and Le Wang. 2005. Population Estimation Methods in GIS and Remote Sensing: A Review. *GIScience & Remote Sensing* 42, 1 (2005), 80–96.

[49] Bing Zhang, Goce Trajcevski, and Liu Liu. 2016. Towards Fusing Uncertain Location Data from Heterogeneous Sources. *GeoInformatica* 20, 2 (2016), 179–212.

[50] Yinghua Zhou, Xing Xie, Chuang Wang, Yuchang Gong, and Wei-Ying Ma. 2005. Hybrid Index Structures for Location-based Web Search. In *ACM International Conference on Information and Knowledge Management (CIKM)*, Otthein Herzog, Hans-Jörg Schek, Norbert Fuhr, Abdur Chowdhury, and Wilfried Teiken (Eds.).