DEVELOPMENT OF OPTIMIZATION ALGORITHMS FOR GEOSPATIAL SEARCH IN LARGE VOLUMES OF DATA

Abstract. This paper deals with the issue of optimizing geospatial search algorithms in large volumes of geodata by integrating geospatial data into the structure of the Semantic Web. In this context, integration means transforming information from geodata sets into an RDF data model. This means that the information is structured by describing it in a vocabulary and correlating it with other information. A significant distinction is made between vector and raster data, as approaches to their integration are at different stages of development. Regarding the basic principles of vector data conversion, geometries are stored according to a specific syntax, usually in the form of a string. This string can, for example, contain information about the type of geometry (e.g., point, line, polygon) as well as the anchor points (as pairs of coordinates). Spatial relationships are defined before conversion to RDF and are explicitly stored as triples. There is no single standard for conversion, so the syntax may differ from organization to organization. To create geospatial queries, a modified version of SPARQL, GeoSPARQL, is used to work with geospatial data, which is both a query language and a data format. There are several approaches to transforming raster data, and, accordingly, methods should first be developed on how to implement such a transformation. This article presents a tool designed to enable users to seamlessly convert spatial data from common GIS formats to the RDF model, thereby greatly increasing the versatility and interoperability of spatial data representation. This tool not only facilitates the generation of geometries represented as GeoSPARQL WKT literals, but also offers structured geometries accessible exclusively through SPARQL queries, thus giving users full access to spatial data in a very adaptive and efficient way.

Keywords: geospatial data, geographic information systems (GIS), spatial querying, cartography, spatial indexing.

РОЗРОБКА АЛГОРИТМІВ ОПТИМІЗАЦІЇ ГЕОПРОСТОРОВОГО ПОШУКУ У ВЕЛИКИХ ОБСЯГАХ ДАНИХ

Анотація. Ця робота займається питанням оптимізації алгоритмів геопросторового пошуку у великих обсягах геоданих шляхом інтеграції геопросторових даних у структуру семантичної павутини. У цьому контексті інтеграція означає перетворення інформації з наборів геоданих у модель даних RDF. Це означає, що інформація структурована шляхом її опису на основі словника та співвіднесення з іншою інформацією. Істотно розрізняють векторні та растрові дані, оскільки підходи до їх інтеграції знаходяться на різних стадіях розвитку. Щодо основних принципів перетворення векторних даних, геометрії зберігаються відповідно до певного синтаксису, зазвичай у формі рядка. Цей рядок може, наприклад, містити інформацію про тип геометрії (наприклад, точка, лінія, багатокутник), а також опорні точки (як пари координат). Просторові відносини визначаються перед перетворенням у RDF і явно зберігаються як трійки. Немає єдиного стандарту для перетворення, тому синтаксис може відрізнятися від організації до організації. Задля створення геопросторових запитів використовується модифікована версія мови SPARQL, орієнтована на роботу з геопросторовими даними GeoSPARQL, яка одночасно є мовою запитів і форматом даних. Існує кілька підходів до перетворення растрових даних, і, відповідно, спочатку слід розробити методи, як можна реалізувати таке перетворення. У цій статті представлено інструмент, розроблений для того, щоб надати користувачам можливість плавно перетворювати просторові дані зі звичайних форматів ГІС у модель RDF, тим самим значно підвищуючи універсальність і сумісність представлення просторових даних. Цей інструмент не тільки полегшує генерацію геометрій, представлених у вигляді літералів GeoSPARQL WKT, але також пропонує структуровані геометрії, доступні виключно через запити SPARQL, таким чином надаючи користувачам повний доступ до просторових даних у дуже адаптивний та ефективний спосіб.

Ключові слова: геопросторові дані, геоінформаційні системи (ГІС), просторові запити, картографія, просторове індексування.

Introduction and Problem Statement. Cartography is of immense importance in various aspects of human society. A map, as its most important
product, serves as a tool for documenting political violence, planning human activities, and orienting and navigating people in space. Geospatial data, which are the main subject of study in cartographic materials, are collected, processed, and published in almost all areas of local governance. Typical geospatial data include streets, paths, squares, bus stops, development plans, and land use.

Geodata are recorded and stored in various systems, but are primarily managed and processed by GIS systems, which extend relational database management systems with geospatial connections. Although these systems are effective and optimized for the centralized processing of geospatial queries, they were not designed with the open and decentralized nature of web data in mind. On the other hand, Semantic Web technologies specifically promote the open, interoperable, and decentralized nature of web data.

There are various Semantic Web technologies that require technical work to achieve full integration of geospatial data. Federated query processing is one such technology. Federated queries allow users to execute queries that combine information from multiple data sources, hiding the details of data collection and the combination of partial query results from each source. Source selection involves matching each query operator with a subset of SPARQL endpoints that form the federation. GeoSPARQL is a standard for representing and querying geospatial linked data for the Semantic Web from the Open Geospatial Consortium (OGC). Defining a small ontology based on well-understood OGC standards is intended to provide a standardized basis for the exchange of RDF geospatial data, which can support both qualitative and quantitative spatial reasoning and queries using the SPARQL database query language. Source selection is usually based on characteristic properties and URI namespaces to eliminate sources that do not have relevant data and significantly enhance the efficiency of processing federated queries. This approach does not work when sets of geospatial data are distributed across a geographical scale.

Analysis of Recent Research and Publications. Scholars today have made significant contributions to developing methodologies for optimizing geospatial search algorithms in large volumes of data. A number of studies have been conducted to address this issue.

In work [1], a method of "Delimited Strokes" was developed for the mutual enrichment of two geometrically similar and semantically complementary datasets about roads in a single region. A Delimited Stroke (DS) is a linear structure created by connecting adjacent road segments that continue in a similar direction. Each "DS" is connected to other DSs through their end nodes. The matching process starts with finding the longest DS in the street network for its counterpart in another street network and extends the search to adjacent DSs. Thus, the correspondence of lines is extended to a graph match. The process is repeated until there are no unprocessed DSs left. Homologous road segments, identified through matching, are then linked.
to transfer semantic information available only in one road network to another road network. The process was effectively implemented using test data such as DLM-De TeleAtlas, Navteq, and OSM.

In work [2], a "Sparse Matching Algorithm" (SMA) was developed for matching between a vector database and a raster image with georeferencing. The vector database consists of structured road objects, semantically enriched through the DS process, while the raster image contains current line fragments often interrupted by shadows and other obstacles. SMA is used to merge fragments of sparse lines belonging to the road object and to obtain information about changes in the road network. A matching coefficient of 70% was found to be feasible for suburban and rural areas.

At the same time, in addition to the above documentation, it is worth mentioning the works of the following scholars: Pei Tao, Sun Ci, Guo Sihui, Shu Hua, Liu Yaxi, Du Yunyan, Ma, Ting, Zhou Chenghu [3], Car N., Homburg T. [4], Jovanovik M., Homburg T., Spasić M., and others [5], Bereta K., Caumont H., Daniels U. [6], Hagos D., Kakantousis T., Vlassov St [7], Alexander K., Cyganiak, R., Hausenblas M. [8], Montoya G. and others, H. Skaf-Molli, P. Molli [9], T. Hellmund, M. Schenk, P. Hertweck [10], O. Páez, L. Vilches-Blázquez [11], M. Masmoudi, S. Lamine, Zghal H. [12], Troumpoukis A., Konstantopoulos S., Prokopaki-Kostopoulou N. [13], Pei Tao, Huang Qian, Wang Xi, Chen Xiao, Liu Yaxi, Sun Ci, Chen Jie, Zhou Chenghu [14], Ferrari Elia, Striewski Friedrich, Tiefenbacher Fiona, Bereuter Pia, Oesch David, Donato Pasquale [16], Chen J., Jiménez-Ruiz E., Horrocks I., Antonyrajah D., Hadian A., Lee J. [17], and others.

However, considering mentioned scientific documentation, the issue related to methodologies for research and development of new algorithms and optimization methods for optimization of the search for geospatial information in large datasets, such as cartographic data or satellite images, remains insufficiently studied and requires further processing.

The research purpose. The objective of the work is the development of optimization algorithms for geospatial search in large volumes of data.

Main research material presentation. Geospatial Search in Large Datasets is typically achieved through a combination of data indexing, spatial querying methods, and algorithm optimization. An overview of the main stages of the process is outlined below:

− Data representation: geospatial data, such as points, lines, polygons, or raster grids, must be presented in a format suitable for efficient storage and retrieval.

− Indexing: to accelerate spatial queries, spatial indexing structures are often used. These indices organize geometric objects in the dataset to facilitate effective searching based on spatial relationships.

− Spatial queries: geospatial queries include operations such as point-in-polygon testing, nearest neighbor searches, range queries, and spatial joins.
Representation of Geospatial Data is typically conducted through data analytics. Data analytics for geospatial information entails searching for hidden patterns within datasets that have a spatial-temporal aspect. The hypothesis space consists of a feature space, which defines the identity and significance of individual geospatial objects and their relationships to one another, and a space-time dimension, which determines the locations, extents, and neighborhoods of these ge-objects.

From a methodological standpoint, two main approaches to geospatial data analytics are distinguished: supervised knowledge induction and unsupervised knowledge abduction. Supervised knowledge induction requires existing foundational knowledge, i.e., a hierarchy of concepts that includes the components of objects and their levels of abstraction. The sequence of processing, whether starting with spatial induction followed by feature-based induction or vice versa, does not impact the computational work.

In unsupervised knowledge abduction, there is no reliance on pre-existing hypotheses or foundational knowledge. Therefore, knowledge abduction involves deriving hypotheses from observations to best explain outcomes or to identify the most plausible causes of results. Hypotheses can be formed either through learning from examples or through the detection of clusters.

The proliferation of free web services such as OpenStreetMap (OSM) and Google MyMaps has significantly facilitated the voluntary exchange of geodata among internet users and has opened opportunities for data analytics on events (Event-Mining). From the perspective of geospatial data analysis, events represent spatial and/or semantic changes in geo-objects that have reached a certain level of significance. Each event can be characterized by five attributes: "what" (type of event), "where" (location), "when" (time of occurrence), "how" (type of change), and "who" (involved people or objects). An event may have various relationships (for example, correlation, cause-and-effect connection) with other events.

After the data analytics of geospatial data in the geospatial search process, as already mentioned, the next step typically involves the integration of geospatial queries themselves. These are specialized types of SQL queries. They differ from non-spatial SQL queries in the following ways:

− They use specialized types of geometric data: point, line, multiline, polygon, and multipolygon.
− They express relationships between types of geometric data, such as distance, equality, intersection, touch, overlap, non-intersection, and others.

With these queries, one can find the distance between two points, verify if one area (polygon) contains another, check if one line intersects another line or polygon, or touches them.
Typically, these queries are implemented using the SPARQL language. The fundamental principle of SPARQL is based on comparing graphs from the query with stored graphs. Most SPARQL queries consist of one or more triple patterns, which together are called basic graph patterns. A triple pattern is structured like a regular triple, except that the subject, predicate, or object can be replaced with variables. These patterns are matched against existing triples to retrieve the corresponding resources. Table 1 contains the source code with two triples that assign a name and a phone number to an employee with ID 657. The SPARQL query in Table 2, in the first triple pattern (line 6), queries for a resource that has the attribute vok:Name with the value "Roman". In the second triple pattern (line 7), it searches for the value of the vok:Tel attribute of this resource.

A SPARQL query typically consists of two parts: The SELECT section lists the variables that should appear in the output data. The WHERE section contains triple patterns for comparison with the RDF dataset. The given query requires two matches. Initially, it searches for graphs that match the first triple pattern (line 6). Then, based on the results, it looks for matches for the second triple pattern (line 7). The result of the query will be the variables ?ID with the value withID:452 and ?Phone with the value 098-789-57-36. If there were multiple people in the dataset with the same attributes, all of them would be displayed.

Table 1.

<table>
<thead>
<tr>
<th>Example of a triple for SPARQL queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 @prefix vok: &lt; <a href="http://example.org">http://example.org</a> &gt; .</td>
</tr>
<tr>
<td>2 @prefix mitID: &lt; <a href="http://example.org">http://example.org</a> / WorkerID / &gt; .</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4 mitID:452 vok:Name &quot; Roman &quot; .</td>
</tr>
<tr>
<td>5 mitID:452 vok:Tel &quot; 098 -789 -57 -36 &quot;</td>
</tr>
</tbody>
</table>

Table 2.

<table>
<thead>
<tr>
<th>SPARQL query</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 @prefix vok: &lt; <a href="http://example.org">http://example.org</a> &gt; .</td>
</tr>
<tr>
<td>2 @prefix mitID: &lt; <a href="http://example.org">http://example.org</a> / WorkerID / &gt; .</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4 SELECT ? ID ? Telephone</td>
</tr>
<tr>
<td>5 WHERE</td>
</tr>
<tr>
<td>6 ? ID vok:Name &quot; Roman &quot; .</td>
</tr>
<tr>
<td>7 ? ID vok:Tel ? Telephone .</td>
</tr>
<tr>
<td>8</td>
</tr>
</tbody>
</table>
On the other hand, the modified version of GeoSPARQL consists of three central components:

- A vocabulary that defines classes and properties for describing spatial data as well as for formulating simple SPARQL queries. The ontology, like some previous proposals, is based on the GML Simple Features model with some adaptations to RDF. The most important classes are geo:SpatialObject with subclasses geo:Feature and geo:Geometry, where the last two are connected by the geo:hasGeometry relationship.

- A filter function for GeoSPARQL queries is designed to enable spatial queries that include certain geoprocessing parameters, such as intersection or buffering.

- Query transformation rules allow GeoSPARQL to perform both attributive and geometric queries. Spatial analysis can be conducted based on topology, for example, based on RCC8, as well as through geometric calculations. For the first, spatial relationships are predefined based on geometries and stored as topological information in RDF, while for the latter, only geometries need to be available. In this case, the transformation rules allow extending an attribute to a geometric query while maintaining the same specification. For example, ogc:within includes both types of queries.

The Resource Description Framework (RDF) is a data model standardized by the W3C (World Wide Web Consortium) for describing structured information. Information in RDF is represented as directed graphs (a set of nodes connected by directed edges), where nodes and edges are uniquely identified and named by so-called Universal Resource Identifiers (URI). RDF is based on three fundamental concepts:

- Resources represent objects of interest. This means that resources can refer to any conceivable or actual object. Each resource is uniquely identified and named by a URI.

- Properties describe relationships between resources and thus represent a special type of resource that is also clearly identified by a URI.

- Statements describe resources in the form of triples. A triple always consists of a subject, predicate, and object, and thus has the structure of a simple sentence.

Table 3 with the source code demonstrates the definition of a polygon in the GeoSPARQL model using the N3 format.

Figure 1 illustrates that a filter, which only retains shapes within a distance \( d \) from point \( p \), can immediately (i.e., from the database index) discard all shapes in set \( s_1 \) if the distance between \( s_1 \) and \( p \) is greater than \( d \). This provides an optimization opportunity compared to calculating the distance between \( p \) and all the shapes in \( s_1 \), and then comparing them to \( d \).
Table 3.

Definition of a polygon in the GeoSPARQL model in N3 format

1 < http: // www . example . org / Points_of_Interest / Geometrie_Bern >
2 < http: // www . opengis . net / rdf # asWKT >
3 " < http: // www . opengis . net / def / crs / EPSG /0/4326 > Polygon ((5807 888 ,
4 [...] 5807 888 ) ) " ^ ^ < http: // www . opengis . net / rdf # WKTLiteral > .

Bringing this discussion to federated query processing, it becomes clear that sets of geospatial data are often published by state administrations or other organizations responsible for a specific geographic area. This motivates applying this optimization at the source selection level: if $s_1$ and $s_2$ were the bounding polygons of all resources served by two GeoSPARQL endpoints, then source selection could exclude $s_1$ from the execution plan.

![Fig. 1](image)

**Fig. 1** The boundaries of two sources, $s_1$ and $s_2$, with the polygon of interest $p$ located within $s_2$.

Typically, a GeoSPARQL query combines geospatial constraints with thematic triple patterns. In Figure 2, each of the three data sources contains triples using a specific list of codes or a vocabulary, which in spatial queries corresponds to a set of terms or keywords used to describe spatial relationships, attributes, and operations related to the query process. The situation illustrated in the figure may occur when different organizations publish data about various aspects of a geographic region, some of which are also published independently for each region. For a query that uses only the "green" vocabulary to retrieve interesting objects at a specified distance from $p$, it makes no sense to include $s_1$ in the execution plan. This separation is subjected to the usual source selection based on metadata about the vocabularies used in each data source. However, to have an optimal source choice, the federation mechanism should also exclude $s_3$ based on its geospatial scale and
pose the query only to s2. Such a source choice can be achieved by extending the usual federated source selection through a mechanism that combines metadata about the thematic content of the data source with metadata about its geospatial scale.

![Fig. 2 Boundaries of three sources and the polygon of interest. s1 uses the "blue" vocabulary; s2 and s3 use the "green" vocabulary.](image)

A source selection mechanism that recognizes the geospatial nature of sources can be useful in queries that involve geospatial joins. Figure 3 illustrates the combination of four geospatial sources. To find pairs of "red" and "blue" objects that are closer than distance $d$, it is necessary to exclude s4 from the query because this source contains only "green" entities. It is assumed that the distance between the boundaries of s1 and s3 is greater than $d$. Therefore, since all "red" figures are located in s1 and the distance between s1 and s3 is greater than $d$, it can be concluded that there are no "blue" figures within a distance $d$ of any red figures in s3. As a result, from a practical perspective, it makes sense to exclude s3 from the query evaluation as it contains irrelevant forms; and query only s1 and s2.

![Fig. 3 Four sources where s1 uses the "red" vocabulary to describe resources, s2 and s3 use the "blue" vocabulary, and s4 uses the "green" vocabulary.](image)
With a thorough understanding of the necessary theoretical components, there arises a need to develop an effective algorithm that allows manipulation of geospatial data aimed at enhancing the process of geospatial searching in large volumes of these data. This algorithm should include the following components:

− Input Parser – this component extracts all features and their descriptions from each file/database, such as schema, types, and CRS (Coordinate Reference Systems). The term "feature" refers to a "simple feature" as defined by the Open Geospatial Consortium (OGC) and the ISO 19125 standard of the International Organization for Standardization (ISO), which includes both spatial and non-spatial attributes. Spatial attributes have geometric values, and simple objects are based on two-dimensional geometry with linear interpolation between vertices (for example, multi-polygons, polygons, etc.).

− Function Parser – this component processes the objects extracted during the first step and classifies their properties based on their type, whether thematic or geometric. For each feature, the parser checks all properties and stores information such as their name, value, and type. If a property is geometric, additional information about this property (such as the number of geometries contained in a multi-polygon) will be stored.

− RDF Builder – generates a set of RDF triples based on the analyzed functions, expressed as subjects, predicates, and objects. Like in the first component, thematic and geometric properties are processed separately.

These components work together to enhance the efficiency and accuracy of geospatial data processing and querying, integrating sophisticated data handling and query mechanisms tailored to the needs of expansive geospatial datasets.

**Conclusion.** In this article, an algorithm for manipulating geospatial data was introduced, allowing the transformation of geospatial data into RDF. The input format supported by this algorithm, its main components, and its dependencies on external libraries were presented. Detailed descriptions were provided on how geometric information is structured according to an ontology that reuses and extends existing geographic vocabularies. Currently, the proposed conversion algorithm only accepts geospatial formats as input data. However, there is potential for developing a function to accept text file formats, such as CSV, which contain direct location information described by coordinates, and converting these textual coordinates into instances of the class geom:Point. Such location information can then be processed using query, linking, or visualization tools just like any other piece of geospatial information.

Future research perspectives include developing a more effective methodology for implementing more efficient algorithms for optimizing geospatial search in large data volumes.
References:
Література: