

# SEMANTIC FRAMEWORK FOR SPATIAL QUERY REFORMULATION FOR DISASTER MONITORING APPLICATIONS

Kuldeep R. Kurte<sup>1</sup>, Abhishek V. Potnis<sup>2</sup>, Surya S. Durbha<sup>2</sup>, Rajat C. Shinde<sup>2</sup>

<sup>1</sup>Oak Ridge National Lab, USA; <sup>2</sup>Centre of Studies in Resources Engineering, IIT Bombay, India  
<sup>1</sup>{kurtekr}@ornl.gov; <sup>2</sup>{abhishekvpotnis, sdurbha, rajatshinde}@iitb.ac.in

## Abstract

In disasters, since time is of the essence, quick decision making based on actionable insights is desired. In our earlier work, we have demonstrated that the spatial relationships-based queries can play a vital role in the disaster response phase. However, we found that the utilization of spatial relationships rules (i.e. encoded spatial knowledge) via rule reasoning process do not scale well with the increased number of image regions. Most of the available Resource Description Framework (RDF) triplestores do not support rule reasoning due to the computational complexity and undecidable nature of the rule reasoning process. In this paper, we propose an alternative approach for utilizing spatial knowledge encoded in the form of spatial relationship rules. The proposed approach reformulates the spatial query by expanding it with the configuration encoded in the corresponding spatial relationship rule. The preliminary results are promising and show the applicability of the proposed approach during the time critical events such as flood disaster.

**Index Terms**— Spatial relations, Query reformulation, SPARQL, RDF, SWRL, Linked data, Disaster response.

## 1. INTRODUCTION

Natural disaster—an event caused by natural processes, has deep impact on human life. Every year disaster events such as flood, earthquake, etc. are the major reasons for the loss of human lives. Such events impact economy and affect urban infrastructure. Although these events cannot be stopped, their effects can be mitigated through proper planning and disaster management activities. Disaster response is an important phase in the disaster management cycle in which various response majors are taken to reduce the disaster impacts and to save human lives. In the event of flood disaster, the affected areas may remain flooded for several days, roads remain submerged and places become inaccessible. In this scenario, identifying buildings that are strongly affected by flood is very important to deploy the rescue activities. Remote sensing (RS) data is a great source of information that provides a synoptic coverage of the

flood affected area and useful to extract the information such as flooded buildings, submerged roads, etc.

The spatial relation query such as, “Identify all building regions from the repository of a satellite X that are strongly affected by flood from west direction and also retrieve the information of the nearby hospitals”, is important during flood disaster. This query requires to model spatial relations between land use/cover regions, in this case the directional relation “building *strong\_east* flood water”. Figure 1 shows an example RS image that contains spatial configuration mentioned in the above query. The extent of a region, such as minimum bounding rectangles (MBRs) can be used to infer the spatial relation. Moreover, the above query also needs to retrieve the geospatially linked information such as nearby hospitals, news feeds, social media feeds.



**Figure 1.** An example RS image showing building regions strongly flooded from west direction. Regions are delineated with red color and their corresponding MBRs are shown with yellow border. (Image source: WV2, Srinagar, India, Flood event 2014).

For last few years, there has been an increased interest in representing geospatial data in a linked data format, also called as linkedgeodata [1]. This uses RDF data model to represent the information in the triplet form (i.e. “subject predicate object”), which can be stored in a special kind of database called triplestore. A triplestore can be exposed and queried over web, and also linked with other geospatial data represented in the similar format. RS image data can be processed to represent it in the linked data-based textual format. Recent research has highlighted the advantages of representing earth observation (EO) data as linked data. The corine land cover initiative of the copernicus earth observation program is actively being used for monitoring land cover changes<sup>1</sup>. Recently, a linked geodatabase-based

<sup>1</sup> <https://land.copernicus.eu/user-corner/land-use-cases>

framework was developed for wildfire monitoring using satellite images and ontology [2].

In our earlier work [3], we have proposed a semantics enabled framework for Spatial Image Information Mining (SIIM) for modeling spatial relationships among RS image regions and representing RS image information in the form of linked data, i.e. RDF representation of RS image. In this work, we have demonstrated the applicability of spatial relationship-based queries during flood disaster. The Web Ontology Language (OWL) [4] is used to define the semantics such as taxonomy of land use/cover classes, subsumption relation of spatial relation, e.g. *strong\_east* is a sub-property of *north*. In SIIM, the spatial knowledge is modeled in two ways, one, through the definitions of the subsumption relationships of spatial relations (i.e. object properties in OWL) and second, by defining the spatial relationships rules using semantic web rule language (SWRL) [5]. These rules can be used to infer the spatial relations among image regions (say, *strong\_east* in the above query) through the process of rule reasoning and the inferred information is then be used to answer the query.

Our past experience shows that the process of rule reasoning is time expensive and can quickly become undecidable with increased number of image regions. Moreover, the triplestores available today do not support rule reasoning. In this work, we propose an alternate approach to utilize the spatial knowledge encoded as spatial rules by reformulating user’s spatial query into expanded SPARQL query (RDF query language) which can then be posed on the triplestore storing the RS image’s RDF data. The work in [6] presents a similar approach for semantic query expansion using rules, however, the work presented in this paper specifically focuses on spatial queries and their expansion based on spatial relationships rules.

This paper is organized as follows, Section 2 describes the spatial relationships-based model approach in SIIM. Further, it presents proposed approach in formulating spatial queries using spatial semantic rules. Section 3 presents example spatial relationship queries that are useful in flood disaster and their equivalent and reformulated SPARQL queries. Finally, the section 4 give some concluding remarks and discusses the future work.

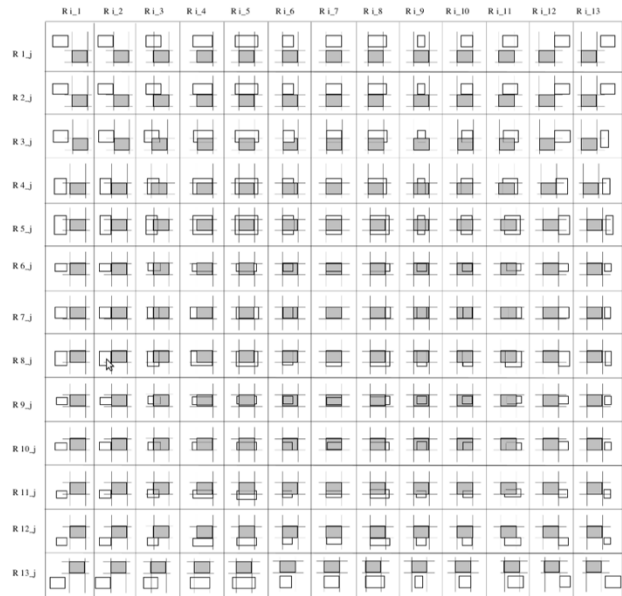
## 2. METHODOLOGY

As explained in the earlier section, spatial relationships-based queries can play important role in identifying flood affected buildings, submerged roads during flood disaster event. The retrieved information is useful to understand the flood inundation and receding patterns in the affected areas. This requires modeling spatial

relationships among image regions. In this section, we first describe the spatial relationships modeling in SIIM and later explain the proposed approach for spatial query reformulation.

### 2.1. Spatial relationships modeling in SIIM

In SIIM, image regions are represented by minimum bounding rectangles (MBRs) (see Figure 1). The SIIM ontology defines few datatype properties such as, *hasLeftLong*, *hasUpperLat*, to define the extent of MBRs. As shown by Papadias and Theodoridis, when two regions are represented by MBRs, a total of 169 spatial configurations are possible [7] (see Figure 2). This is an exhaustive set of possible spatial configurations between two MBRs. These spatial configurations encode both topological and directional relationships among MBRs.



**Figure 2. 169 spatial configurations among MBRs.  $R_{i,j}$  represents configuration at row ‘i’ and column ‘j’ [7].**

In SIIM, we have adopted these spatial configurations and have encoded this spatial knowledge in the form of ontology. This required to represent each spatial relation object property such as *strong\_east* as a conjunction of object and datatype properties. The current popular variants of OWL, i.e. OWL-DL and OWL2 do not provides constructs for such representation. Therefore,, we have used semantic web rule language (SWRL) to encode these spatial configurations and define the semantics of the spatial relations. These rules are intended to infer the spatial relationships among two MBRs from their spatial extent. Figure 3 shows the example SWRL rules for given spatial configurations.

At present, these rules are encoded in RDF and hence, can be exposed over web and any application that represents RS image regions using MBRs can consume these rules to infer the spatial relations among image regions. The fundamental requirement for other applications to use these spatial relationships rules is their compliance with the semantics of SIIM ontology.



**Figure 3. Example SWRL rules for topological relation Non-Tangential Proper Part (NTPP) and directional relationship *strong\_bounded\_east*. The corresponding spatial configuration is also shown. ‘^’ represents logical AND.**

## 2.2. Proposed approach for spatial query reformulation

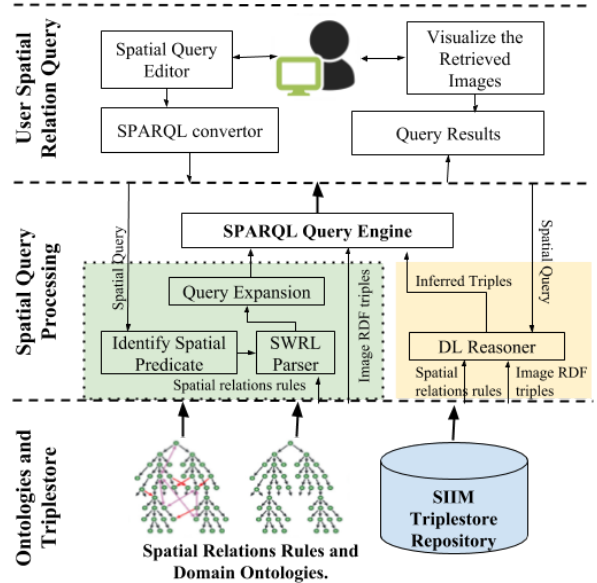
Figure 4 depicts two ways to use spatial knowledge that is encoded in the form of spatial relations rules. The part highlighted in yellow shows the earlier approach [3], which uses the appropriate spatial relations rule from spatial relation ontology to perform rule reasoning on the image triples to infer the spatial relations among region’s MBRs.

The part highlighted in green (in Figure 4) represents the proposed approach, which is the main contribution of this work. In this approach, instead of performing DL reasoning to produce inferred triples, which will be further queried to answer user’s spatial relation query, the spatial predicate in the query is identified and appropriate spatial relation rule is retrieved from the spatial relation ontology. The SWRL rule parser parses the rule and further, the query expansion module reformulate the SPARQL query by replacing the spatial relation predicate by corresponding pattern defined in the spatial rule. Next, the reformulated SPARQL query is posed on the triplestore to retrieve the information.

As shown in Figure 3, the SWRL rule consists of two parts: 1) the antecedent: the part before ‘→’, which consists of conjunction of series of terms (i.e. OWL datatype or object properties) that represent patterns, e.g. *hasLowerLat*(?B1, ?B1LowerLat) (refer Figure 3), and 2) the consequent: the part after ‘→’, which is a spatial relation predict (i.e OWL object property). Here, a spatial rule represents an object property in terms of conjunction of object and datatype properties. On the other hand, the WHERE clause in SPARQL query represents conjunction of

a series of triplet patterns.. For example, “?B1 *hasLowerLat* ?B1LowerLat” represents one such pattern.

The proposed approach identifies the spatial relation predicate in the query, which forms the consequent of the SWRL rule, and retrieves the appropriate rule from the spatial relation ontology. The SWRL rule parser is used to parse the rule, that is retrieved in the earlier step, to extract all the terms in its consequent part. The SPARQL query expansion module converts all the terms in the consequent part of the rule, which are in the form of “*F*(?a, ?b)”, to triplets, i.e. “?a *F* ?b ”. Along with these patterns, few consequent parts of SWRL rule contains comparison operators, e.g. “*lessThan*(?B1LowerLat, ?B2LowerLat)”. Such terms form FILTER clause in SPARQL query, e.g. “FILTER(?B1LowerLat > ?B2LowerLat)”. At the end, the query expansion module concatenates all these triplets (by appending period ‘.’ at the end of each triplet) and forms the pattern that replaces the spatial relation predicate in user’s query. The reformulated spatial query is then executed by SPARQL query engine to retrieve the information. The preliminary version of java-based tool to perform the proposed spatial query reformulation is provided here<sup>2</sup>.



**Figure 4. Two ways to utilize spatial relationship rules to answer the spatial relationships-based queries. The part highlighted in green shows the proposed approach and the part highlighted in yellow shows earlier approach.**

## 3. RESULTS AND DISCUSSION

In this section, we present two spatial relationships-based queries that are important during flood disaster event.

<sup>2</sup> <https://github.com/kuldeep3101988/SemQuery>

- Query1: Identify building regions from the repository of a satellite X that are surrounded by flood water.
- Query2: Identify building regions from the repository of a satellite X that are strongly bounded by flood water from east direction.

Query 1 contains topological relation *NTPP* (refer  $R_{9,9}$  in Figure 3) whereas query 2 contains a directional relation *strong\_bounded\_east* (refer  $R_{9,13}$  Figure 3). The SPARQL versions of both of these queries are shown in Figure 5.

Query 1	Query 2
<pre>select ?r1 where {   ?r1 rdf:type siim:LULCRegion .   ?r2 rdf:type siim:LULCRegion .   ?r1 siim:hasLULCClass "Building" .   ?r2 siim:hasLULCClass "FloodWater" .   ?r1 siim:hasBBox ?b1 .   ?r2 siim:hasBBox ?b2 .   ?b1 siim:NTPP ?b2 . }</pre>	<pre>select ?r1 where{   ?r1 rdf:type siim:LULCRegion .   ?r2 rdf:type siim:LULCRegion .   ?r1 siim:hasLULCClass "Building" .   ?r2 siim:hasLULCClass "FloodWater" .   ?r1 siim:hasBBox ?b1 .   ?r2 siim:hasBBox ?b2 .   ?b1 siim:strong_bounded_east ?b2 . }</pre>

Figure 5. SPARQL syntax for Query 1 and Query 2

<pre>select ?r1 where {   ?r1 rdf:type siim:LULCRegion .   ?r2 rdf:type siim:LULCRegion .   ?r1 siim:hasLULCClass "Building" .   ?r2 siim:hasLULCClass "FloodWater" .   ?r1 siim:hasBBox ?b1 .   ?r2 siim:hasBBox ?b2 .   ?b1 siim: hasLeftLong ?B1LeftLong .   ?b2 siim: hasLeftLong ?B2LeftLong .   ?b1 siim: hasLowerLat ?B1LowerLat .   ?b2 siim: hasLowerLat ?B2LowerLat .   ?b1 siim: hasRightLong ?B1RightLong .   ?b2 siim: hasRightLong ?B2RightLong .   ?b1 siim: hasUpperLat ?B1UpperLat .   ?b2 siim: hasUpperLat ?B2UpperLat .   FILTER(?B1LeftLong &gt; ?B2LeftLong &amp;&amp;   ?B1RightLong &lt; ?B2RightLong &amp;&amp;   ?B1LowerLat &gt; ?B2LowerLat &amp;&amp;   ?B1UpperLat &lt; ?B2UpperLat)</pre>	<pre>select ?r1 where {   ?r1 rdf:type siim:LULCRegion .   ?r2 rdf:type siim:LULCRegion .   ?r1 siim:hasLULCClass "Building" .   ?r2 siim:hasLULCClass "FloodWater" .   ?r1 siim:hasBBox ?b1 .   ?r2 siim:hasBBox ?b2 .   ?b1 siim: hasLeftLong ?B1LeftLong .   ?b1 siim: hasLowerLat ?B1LowerLat .   ?b2 siim: hasLowerLat ?B2LowerLat .   ?b2 siim: hasRightLong ?B2RightLong .   ?b1 siim: hasUpperLat ?B1UpperLat .   ?b2 siim: hasUpperLat ?B2UpperLat .   FILTER(?B1LeftLong &gt; ?B2RightLong &amp;&amp;   ?B1LowerLat &gt; ?B2LowerLat &amp;&amp;   ?B1UpperLat &lt; ?B2UpperLat &amp;&amp;)</pre>
--	--

Figure 6. The expanded version of the SPARQL queries shown in Figure 5. The highlighted section shows the expanded portion of respective spatial relation.

The reformulated versions of query 1 and query 2 are given in Figure 6. The highlighted portion in Figure 6 shows the expanded portion of the respective spatial relationship using the proposed approach. These reformulated queries can be executed in a triplestore to retrieve the information.

We have executed these reformulated queries in a triplestore and observed the execution timings for the spatial relations queries (refer Figure 7). The triplestore used for this experiment is Allegrograph<sup>3</sup> containing 457,885 RDF statements about image regions from RS imagery of a flood event happened in Srinagar, India in the year 2014. The triplestore server was setup on a system with intel i7 (3GHz) processor and 16GB memory. The LIMIT clause of SPARQL was used to limit the number of retrieved results. More comparisons in the FILTER clause of the reformulated query 1 is the potential reason for its greater

execution time compared to query 2. The exponential growth in execution timing shows the potential to further optimize the query performance.

Query 1 and Query 2 Execution Timings with Varying Number of Retrieved Results

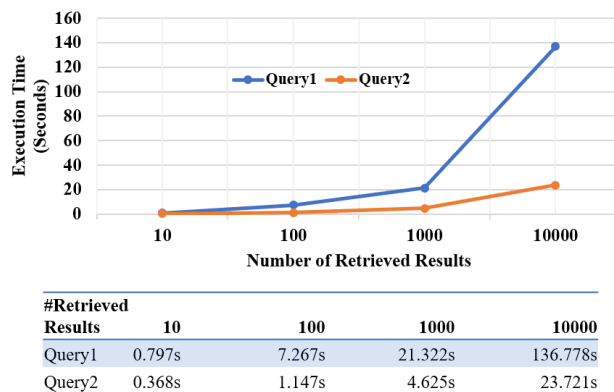


Figure 7. Execution times for queries shown in Figure 6.

#### 4. CONCLUSION

The spatial relationships-based queries can play important role in identifying the flood affected areas. However, timely retrieval of the information from the repository is desired. In this work, we have presented an alternate approach for utilizing the spatial knowledge encoded in the form of spatial relationships rules. This approach will overcome the scaling limitation of the rule reasoning process in our earlier work. Currently, the developed tool can handle up to one spatial predicate in the query. In future, we intend to extend the tool to support the queries containing multiple spatial predicates and perform time analysis on large triplestore.

#### 5. REFERENCES

- [1] C. Stadler et al., "Linkedgedata: A core for a web of spatial open data," *Semantic Web*, vol. 3, no. 4, pp. 333-354, Jan. 2012.
- [2] K. Kyzirakos et al., "Wildfire monitoring using satellite images, ontologies and linked geospatial data," *Web Semantics: Science, Services and Agents on the WWW*, vol. 24, pp. 18-26, Jan. 2014.
- [3] K. R. Kurte et al., "Semantics-enabled framework for spatial image information mining of linked earth observation data," *IEEE JSTARS*, vol. 10, no.1, pp. 29-44, Jan 2017.
- [4] D. L. McGuinness and F. V. Harmelen, "OWL web ontology language overview," *W3C Rec.*, vol. 10, no. 10, Feb. 2004.
- [5] I. Horrocks et al., "SWRL: A semantic web rule language combining OWL and RuleML," *W3C Member submission*, vol. 21, no. 79, May 2004.
- [6] S. M. Bowers et al., "Execution of semantic queries using rule expansion" *U.S. Patent No. 8,032,525*, 2011.
- [7] D. Papadias and Y. Theodoridis, "Spatial relations, minimum bounding rectangles, and spatial data structures," *Inter. Journal of Geographical Info. Sci.*, vol. 11, no. 2, pp. 111-138. Mar. 1997.

<sup>3</sup> <https://franx.com/agraph/allegrograph/>