Quality-Driven Approximate Methods for Integrating GIS Data

Ramaswamy Hariharan, Michal Shmueli-Scheuer, Chen Li and Sharad Mehrotra
School of Information and Computer Sciences
University of California, Irvine, CA 92697, USA
{rharihar, mshmueli, chenli, sharad}@ics.uci.edu

ABSTRACT

GIS data distributed in local, state, federal, and private data clearinghouses are being made accessible through the efforts of organizations such as Federal Geographic Data Committee (FGDC) and GeoData.gov. Many database applications, such as disaster management, transportation, and national infrastructure protection, need to access GIS information from such various data sources. In this paper we study how to answer keyword-based spatial queries approximately using information from heterogeneous GIS sources. An example query specifies the region of Orange County and keywords "`junior schools`," which asks for geospatial objects relevant to junior schools in Orange County. The answers to such a query provided by different sources differ widely in their content and quality. It is computationally expensive to access all the datasets to retrieve all the relevant objects. We develop approximate algorithms for answering such queries based on the local analysis of the query region using space-partitioning techniques. Our methods rank datasets in a partition based on parameters such as their spatial coverage and content matching the query keywords. The quality of the answers keeps improving progressively as we do deeper local analysis. We develop an efficient traversal strategy to maximize the quality refinement within a given time limit. We conducted experiments to evaluate the proposed techniques.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications – *spatial databases and GIS.*

General Terms

Algorithms, Management

Keywords

GIS data integration, approximate methods, heterogeneous data sources.

GIS'05, November 4, 2005, Bremen, Germany.

1. INTRODUCTION

Many applications such as emergency management, environmental analysis, and national infrastructure protection need to access GIS information for various kinds of analyses. For instance, during emergency situations [20], it is critical to provide the first responders quick access to GIS data to do damage assessment and make critical decisions, so that they can dispatch resources to save lives and properties. Often the GIS information comes from autonomous, heterogeneous sources, such as local, state, federal, and private agencies. These datasets are stored and maintained at different hierarchies of administrative jurisdictions. The commercial value of these datasets has also prompted various private agencies to produce datasets of their own. All these organizations provide access to datasets through their data-clearing houses. Therefore, it is becoming increasingly important to support seamless access to GIS information from these heterogeneous sources. A recent report on the World Trade Center attacks [19] has highlighted the need to integrate datasets across multiple jurisdictions. Recently, there has been multitude of efforts, such as the Federal Geographic Data Committee (FGDC) and GeoData.gov, to connect all the clearinghouses, so that users can have single access point to GIS data [8, 10].

One main challenge in integrating GIS data is the heterogeneity of different sources. Data integration problem has been studied from the perspective of how to resolve the schematic and semantic differences between heterogeneous data sources, mapping different schemas, and providing query interfaces for integrated access [4, 7, 9]. The adoption of GIS metadata standards solves the heterogeneity problem to some extent. The metadata provides information such as theme keywords, spatial bounding coordinates, and spatial references. We could utilize such metadata to deal with heterogeneity of the datasets as follows. When a user specifies a query (e.g., a few keywords and a bounding box) that asks for certain GIS information, we can rank these datasets based on their metadata, and return those that are ranked the highest. The user can then browse the data in these returned datasets to find the information. This metadata-based approach is adopted by organizations such as GeoData.gov. While this approach has the advantage of finding relevant datasets for a query, it also has limitations. First, without looking the real data in each dataset, a ranking of the datasets based on their metadata only might not be accurate. Secondly, the top ranked datasets could provide redundant information for the same region. For instance, there can be many data sets that provide information about hospitals in Orange County. Given a list of (top ranked) datasets, the user could be overwhelmed by the large amount of

information: she has to go through the dataset list, retrieve the data from each of them, and manually integrate the data. This process is very tedious and time consuming. Therefore, it is more desirable to provide one integrated result from these datasets.

In this paper we study how to integrate GIS information by supporting keyword queries on heterogeneous data sources. We consider the case where a user issues a query that consists of a geospatial region and a few keywords. An example query specifies the region of Orange County and keywords "`junior schools`." Intuitively, this query asks for geospatial objects that are in Orange County, and relevant to junior schools. Our goal is to retrieve such objects from the different datasets. One main advantage of supporting information retrieval (IR) style keyword search is that the user does not need to know the structure of the data at a source. One naïve approach to answering such a query is to retrieve all the objects from the datasets, and find those that are relevant to the query. This approach is impractical when it is expensive to go through these objects, especially when the datasets reside at remote sources.

In this paper we propose a novel technique to answer GIS keyword queries. The main idea is the following. Given a query, we rank the datasets from different data sources approximately based on their sample objects and relevance to the query. We choose the best data source, and retrieve its relevant objects as an approximate answer to the query. If the user is satisfied with the quality of these objects, there is no need to access more data sources. Otherwise, we need to further improve the quality of the current answer by partitioning the query region into subregions, and finding the best data source for each of them. We repeat this process until the user is satisfied with the quality of the final answer. In this way, our technique can compute an approximate answer to the query, and the quality of the answer improves progressively as we do deeper local analysis. We develop an efficient traversal algorithm to maximize the quality refinement within a given time limit. We have conducted experiments to evaluate the effectiveness of the proposed technique.

The remainder of the paper is organized as follows. In Section 2 we motivate and formally define the problem with an example. In Section 3, we present our space-partitioning algorithm and explain how to rank the datasets. In Section 4, we discuss our progressive refinement algorithms for efficient traversal of space partitions. We present the experimental results in Section 5. We discuss some related work in Section 6 and conclude in Section 7.

2. PROBLEM DEFINITION

In this section we use an example to motivate our work and formally define the studied problem.

EXAMPLE 2.1 Geospatial one-stops such as GeoData.gov or fgdc.gov provide access to GIS datasets through their metadata query interfaces. Their interface allows users to specify query parameters such as spatial region (where), theme keywords (what), and time (when). The query is matched against the metadata present in the underlying distributed catalog servers that index metadata, and retrieves maps, documents, downloadable datasets, etc. The downloadable datasets come as vector, raster, and digital line graphics (DLG) along with the metadata.

We are given a set of GIS datasets that reside at different sources. We consider queries specified in the following format:

$$(q, <w_1,...,w_p>),$$

in which q is a spatial region, and $w_1,...,w_p$ are keywords. An example is:

(Region of *Orange County,* '`junior schools`'),

which, intuitively, asks for geospatial objects in Orange County that are relevant to junior schools. For simplicity, we assume the spatial region is a bounding box. The keywords specify theme or other sub-categories of a theme that the user is interested in. Given this query, we want to compute one integrated result that combines the best matching portions from the relevant datasets in different spatial regions of the query. We focus on vector data that mainly stores geometric elements and their attributes in a relational table. The main challenge to answer the query is how to compute its approximate answer quickly and efficiently. We address this challenge by proposing techniques that provide quick approximate answers by doing local spatial analysis and progressively refining the answer as we do deeper analysis.

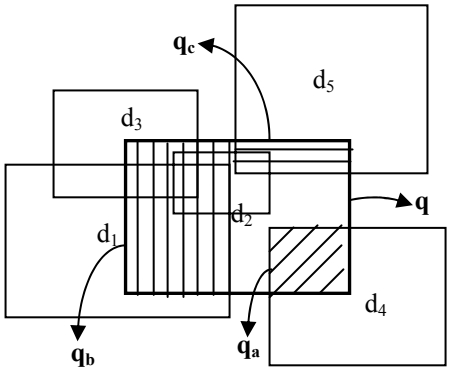

Figure 1. Example problem.

Hereafter when we refer to a data source, we mean its corresponding dataset relevant to queries. Let us assume that we have data sources $d_1,\ldots,d_5$ that can answer the query q, as illustrated in Figure 1. Let the subscripts denote the ranking of the data sources according to some metadata ranking mechanism, and d_1 ranks the highest. Traditional data integration techniques use the following two approaches to answering a query: (1) using all the sources or (2) using the best sources. There are limitations with both approaches. If the first approach is followed, there is a lot of overlap among the results from data sources d_1, d_2, and d_3 leading to redundant geospatial objects that represent the same real-world entity. It is difficult to identify such objects as they come from data sources that have different schemas, semantics, and data models. If the second approach is chosen, assuming that the system picks top three data sources, d_1, d_2, and d_3, then there are contributions from other sources d_4 and d_5 in sub-query regions q_a and q_c that are not considered. This leads to inadequacy in spatial coverage of the query.

Therefore, we need new techniques that can address these limitations. We address the redundancy problem by initially choosing only the data source that best matches the query. We address the spatial coverage problem by splitting the query region into sub-query regions, and further improving the quality of the answer, as requested by the user. In our example, initially d_1 is the

best data source. But for the top portion of the sub-region q_b, the contributions of d_2 and d_3 are better than that of d_1.

Now we formally define the GIS data integration problem studied in this work. Given a query (q, $<w_1,...,w_p>$) with matching set of data sources $D = (d_1, d_2, ..., d_n)$, integrate the results to produce a single dataset D_q, such that D_q contains the best contribution in terms of quality from all or subset of D, given as follows.

$$D_q = \{d_i\},\ 1 \le i \le n,$$

where $d_i = (d_i^1, d_i^2, ..., d_i^m)$ is the set of geospatial objects from source d_i matching the query. The result set can be considered as a set of records in a relational table format. Each record d_i^j in the result set has a spatial attribute (i.e., geometry) along with other attributes and m is the size of the result set. There are many parameters to measure the contributions of sources to the query, such as their spatial coverage, information content, accuracy, and currency. Some of these parameters are expressed in spatial metadata and used for ranking the datasets at the metadata level. In addition, we also consider the information content of geospatial objects present in each dataset that are relevant to the query. In this paper, we focus on spatial coverage and information content for evaluating a source's contribution to a query. We formally define these parameters as follows.

DEFINITION 2.1. Given a region q, the spatial coverage of a data source d_i is defined as the fraction of d_i's spatial region that falls within q, given as,

$$\frac{Area(d_i \cap q)}{Area(q)},$$

where the function *Area*(.) computes the area of a given region.

DEFINITION 2.2. Given a region q, the information content of d_i is defined as a score that measures how much information d_i contributes in q with respect to the keywords in the query. The score is estimated based on an extended version of *tf-idf* model used in information retrieval. We present the details of this model in Section 3.4. This information content is measured using a set of representative sample geospatial objects retrieved from d_i.

3. APPROXIMATE METHODS

In this section we discuss some naïve methods to solve the problem and present approximate methods using space partitioning techniques.

3.1 NAÏVE SOLUTIONS

An ideal condition of an integrated dataset is the one that has the maximum spatial coverage and maximum information content with the minimum redundancy. However, all of the criteria might not be satisfied simultaneously and hence, we need to weigh each criterion using user-defined weights. One naïve way of doing this is to take all the possible combinations of data sources. Then, we evaluate the information content and covered region for each combination and pick the best one using a weighted combination. This approach clearly is computationally expensive. Also, the combined datasets might overlap with each other and lead to a lot of redundancy in the output. Another way of doing this is to output all the relevant objects from the data sources satisfying the query. Again, the problem with this approach is that the output contains a lot of redundant objects. This information might be overwhelming for analysts to process, as in situations like crisis response, where time is a critical factor, this may not be acceptable. In addition, these methods cannot progressively refine the quality of integrated datasets.

3.2 SPACE PARTITIONING TECHNIQUES

If the user is not interested in an ideal output, and can tolerate certain error, we can use approximate methods to produce an integrated dataset. The idea is to trade off accurate and complete results for computational costs of performing the dataset integration and analysis overhead to identify the redundant objects. A number of approximate methods have been proposed in the literature based on multi-resolution data structures such as quadtrees, grids, and *R*-trees [12, 14, 16]. The main goal of these techniques is to provide approximate answers to a spatial range query with some quality guarantees. We use a similar idea of partitioning the query region into spaces. The main difference is that we employ this technique on the query region and not on the underlying data. The progressive partitioning of the query region refines the quality of the output dataset.

3.3 VIRTUAL QUADS

We propose an approximate technique that is based on the idea of partitioning the query region into virtual hierarchical quadrants. The reason for taking this space-partitioning technique is due to its simplicity and ease of maintenance. The idea of our approach is as follows: we start with the entire query region q as the root of a tree. We analyze a set of sampled geospatial objects from data sources that overlap with the root and pick the data source that best matches the query. The matching score of a data source is computed using scoring functions discussed in Section 3.4. The winning data source contributes to the overall approximate objects at the root compared to its counterparts, and the maximum spatial resolution is the root. On the other hand, as we go deeper into the tree, our analysis will pick contributions from other data sources as the spatial resolution increases, thus progressively improving the quality of the answer.

In the example of Figure 2(a), before splitting the query region q into four quadrants q_0, q_1, q_2, and q_3, the overall winning data source is d_2. After we split the quadrants shown in Figure 2(b), the overall winning data sources are d_2 in q_0, d_1 in q_1, d_3 in q_2, and d_4 in q_3. The example scores of the winning data sources before and after splitting the query region into quadrants are given in Table 1 and 2, respectively. The meaning of these scores and how we compute them are explained in Section 3.4. Now we show that after the split, the improvement in the quality of the resulting datasets does not decrease.

THEOREM 1. The total score of the winning data sources in a quadrant's children is always equal to or greater than the score of winning data source in the quadrant.

PROOF. Let d_a^q be the winning data source in a quadrant q and its score be s_a^q. Let q's children be q_0, q_1, q_2, and q_3, and datasets $d_l^{q_0}$, $d_m^{q_1}$, $d_n^{q_2}$, and $d_o^{q_3}$ be the winning datasets with scores $s_l^{q_0}$, $s_m^{q_1}$, $s_n^{q_2}$, and $s_o^{q_3}$, respectively. The winning data source in q's children could be the same winning data source as in q. It is known that d_a^q must overlap with at least one and at most four of

q's children. Let us assign scores for the overlapping portion of d_a^q with q's children as $s_a^{q_0}$, $s_a^{q_1}$, $s_a^{q_2}$, and $s_a^{q_3}$. Note that if d_a^q does not overlap with a q's child, its score in that quadrant is 0. We know that,

$$s_a^{q_0} + s_a^{q_1} + s_a^{q_2} + s_a^{q_3} = s_a^q$$

$$s_l^{q_0} \geq s_a^{q_0} \; ; \; s_m^{q_1} \geq s_a^{q_1} \; ; \; s_n^{q_2} \geq s_a^{q_2} \; ; \; s_o^{q_3} \geq s_a^{q_3}$$

Form above equations it follows that $s_l^{q_0} + s_m^{q_1} + s_n^{q_2} + s_o^{q_3} \geq s_a^q$.

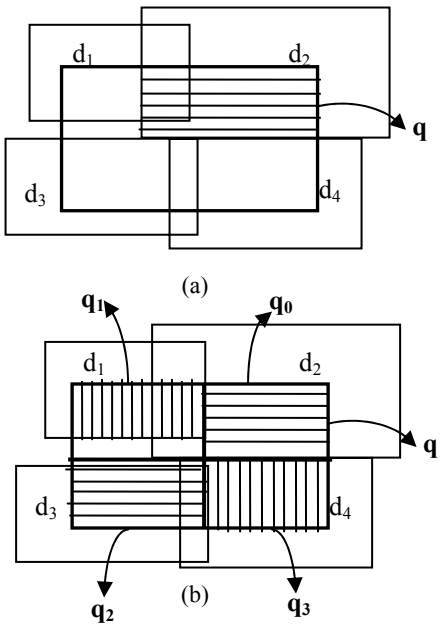

Figure 2. (a) Before split and (b) after split.

Table 1. Scores of datasets before split.

Dataset	Spatial Coverage	Info. Content	Total Score
d1	0.2	0.4	0.65
d2	**0.36**	**0.7**	**0.8**
d3	0.25	0.5	0.6
d4	0.28	0.5	0.7

Table 2. Scores of datasets after split.

Dataset	Spatial Coverage	Info. Content	Total Score
d1	0.01	0.1	0.1
d2	**1**	**0.2**	**0.5**

Quadrant 0

Dataset	Spatial Coverage	Info. Content	Total Score
d1	**0.75**	**0.3**	**0.55**
d2	0.4	0.3	0.3

Quadrant 1

Dataset	Spatial Coverage	Info. Content	Total Score
d3	**1**	**0.4**	**0.55**
d4	0.18	0.1	0.15

Quadrant 2

Dataset	Spatial Coverage	Info. Content	Total Score
d3	**1**	**0.1**	**0.55**
d4	0.02	0.4	0.1

Quadrant 3

3.4 SCORING FUNCTIONS

We rank each data source d_i in a quadrant based on scores of spatial coverage and information content of its sampled geospatial objects. While the score for spatial coverage $s_{d_i}^c$ determines how much of a query region is covered by d_i, the score for information content $s_{d_i}^{ic}$ captures how much of the query-specified keyword(s) are present in d_i. The importance of these two parameters is decided by the user and the scoring function is a weighted sum of the above scores. The following equation gives the scoring function for d_i.

$$s_{d_i} = w_c \cdot s_{d_i}^c + w_{ic} \cdot s_{d_i}^{ic} \; ,$$

where w_c and w_{ic} are user-defined weights. It is straightforward to compute $s_{d_i}^c$ (see Definition 2.1). To compute $s_{d_i}^{ic}$, we use IR techniques for relational tables.

The goal is to find a score for d_i that measures the information content for keyword(s) $<w_1,...,w_p>$ specified in the query. From the metadata information of d_i, we can extract the keywords and compute the score as s_{meta}. This technique can be further improved by considering the keywords in the records. Hence in order to get more accurate scoring, we will use a common *tf-idf* technique [17] to analyze the sampled data sources. We consider the geospatial objects of a data source in relational table format as a document and the measurement of *tf-idf* on top of this document gives us a score s_{table} for each table t_i corresponding to d_i as follows.

$$s_{table}(t_i) = \sum_{w \in w_1..w_p} tf \cdot \log_2 \frac{N}{n} \; ,$$

where tf is the frequency of word w in d_i, N is the number of documents, and n is the number of documents where word w appears at least once.

Next, we compute the score at the record level. For a given keyword w, the score assigned to a record is based on the following three steps:

- Single-attribute IR-style relevance score function $s_{att}(a_i)$ for each textual attribute a_i under *AND* semantics.
- A function $s_{record}(r_j)$ that combines the single attribute score into a final score for the record r_j and,
- A function $s_{final}(r_j)$ that takes into account other relevant tables.

In the first step, we use the state-of-the-art IR definition [18] as a single-attribute scoring function as follows.

$$s_{att}(a_i) = \frac{1 + \ln(1 + \ln(tf))}{(1-s) + s \cdot \frac{dl}{avdl}} \cdot \ln \frac{N+1}{df} \; ,$$

where, tf is the frequency of word w in a_i, df is the number of tuples in a_i's relation with word w in this attribute, dl is the size of a_i in characters, $avdl$ is the average attribute-value size, N is the total number of tuples in a_i's relation, and s is a constant (usually 0.2). We now combine the single attribute scores into one score $s_{record}(r_j)$ for r_j under the *AND* semantics. We use the following summation function.

$$s_{record}(r_j) = \sum_{a_i \in r_j} s_{att}(a_i)$$

The final score for the record should be in comparison with other tables. Hence, we use $s_{table}(t_i)$ as a weighting term to the record-level score. It is given as follows.

$$s_{final}(r_j) = s_{table}(t_i).s_{record}(r_j)$$

The final score of d_i matching all keywords in region q is given by,

$$s_{d_i}^{ic} = \sum_{w \in w_1..w_p} \sum_{d_i \cap q} s_{final}(r_j),$$

where for each keyword w we sum the score of each sample record r_j of d_i matching the query.

4. TRAVERSAL POLICY

In this section we discuss a virtual quad traversal algorithm and its heuristics.

4.1 TRAVERSAL ALGORITHM

The algorithm (Table 3) takes as input a set of data sources D and the query region q, and outputs an integrated dataset D_q. The while-loop checks if the difference between current time and start time reaches the user-defined time threshold t as a stopping condition. All the overlapping data sources $D_{overlap}$ with q (root) are ranked using the function *BestDatasource* (Table 3), which uses the scoring function discussed in the previous sections and the best data source q_{best}^d for q is identified. The assumption here is that there exist sample objects collected from each data source, which allow us to compute the ranking function *RankDatasource*. Then the algorithm traverses q's children using the function *Traverse* (Table 3). The best data source in each of q's children is identified and then the quadrants are ranked using the function *RankQuadrant.* The ranked quadrants are pushed into a priority queue sorted according to their score, and its head is popped for subsequent traversal.

The integrated dataset D_q is obtained using the algorithm presented in Table 4. The algorithm starts with the root. If the root is not expanded into quadrants, the function returns and outputs the data by doing a range query on the root's *BestDatasource*. Otherwise, the function recursively checks each of the quadrant's children and traverses up to the leaf quadrants. The algorithm now outputs the data by doing a range query on the respective leaf quadrant's *BestDatasource*. We only consider leaf quadrants, because, the *BestDatasource* identified in a non-leaf quadrant may be subsequently replaced by other data sources in its children. We combine some of the quadrants at the leaf and non-leaf levels, if they have the same *BestDatasource*, provided the combination still results in a bounding box. This is to minimize the range query operations. After the objects are obtained as a result of a range query, some of them, which do not satisfy the query keywords, are filtered.

4.2 HEURISTICS FOR RANKING QUADRANTS

A good traversal policy will improve the quality of integrated datasets quickly. The question is how to choose a quadrant for deeper traversal. Our heuristics to choose the best quadrant is based on the following observations:

Table 3. Virtual quad traversal algorithm.

Input: $D = \{d_1, d_2, ..., d_n\}$, query region q, and time threshold t.
Output: Integrated dataset D_q
Initialize: Priority Queue, $p_q = \phi$, best rank, $b_r = 0.0$

```
While (CurTime − StartTime ≤ t)
    D_overlap = D ∩ q
    q^d_best ← BestDatasource(D_overlap, q)
    Travserse(q)
End While
IntegrateDatasets(q)
```

```
BestDatasource(D, q) {
  For Each d_i in D
    If RankDatasource(d_i, q) ≥ b_r
      b_r = RankDatasource(d_i, q)
      q^d_best = d_i
    End If
  End For
  Return q^d_best
}
```

```
Travserse(q) {
  Q_c ← GetChildren(q)
  For Each q_i in Q_c
    D_overlap = D ∩ q_i
    q_i^d_best ← BestDatasource(D_overlap, q_i)
  End For
  p_q ← RankQuadrant(q_c)
  Travserse(head ← p_q)
}
```

Table 4. Algorithm for unifying the datasets.

```
IntegrateDatasets(q) {
  Q_c ← GetChildren(q)
  For each q_i in Q_c
    If q_i has no children
      d_i ← q_i.BestDatasource
      OutPut ← OutPut + RangeQuery(d_i, q_i)
    Else
      IntegrateDatasets(q_i)
    End if
  End For
}
```

- Since the user is interested in spatial coverage of the query region, the overall spatial coverage of the data sources in a quadrant should be one of the criteria.
- The user is also interested in the information content and hence the overall information content of a quadrant should be another criterion.

- The third criterion is quality refinement. If there are more overlapping data sources in a quadrant, there is more scope for refinement.

The heuristics we propose is a weighted combination of each of the above criteria. An example in Figure 3(a) illustrates that data sources falling in q_3 have better spatial coverage than other quadrants. Figure 3(b) illustrates that q_1 has more scope for refinement since its overlapping area is larger compared to the others.

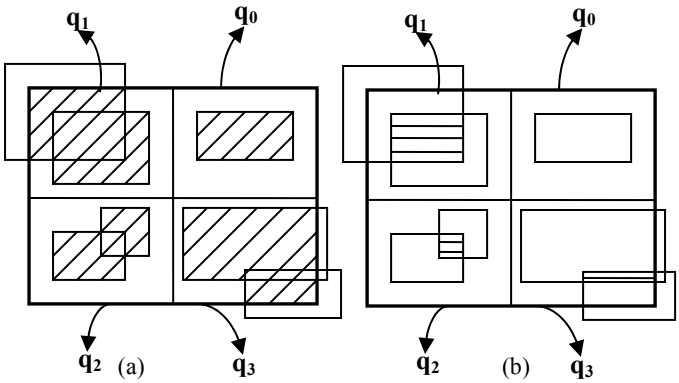

Figure 3. Quadrants showing overall (a) spatial coverage and (b) overlap for refinement.

To compute the overall spatial coverage $s_{overall}$ of a quadrant q, we consider the set of data sources $D = \{d_1, d_2, .., d_n\}$ overlapping with q. $s_{overall}$ is given by the union of the overlapping areas of all d_i with q over area of q.

$$s_{overall} = \frac{\bigcup_{i=1}^{n} Area(d_i \cap q)}{Area(q)}$$

The overall refinement $r_{overall}$ is given by the total intersection area of all d_i overlapping with q over area of q.

$$r_{overall} = \frac{\bigcap_{i=1}^{n} Area(d_i \cap q)}{Area(q)}$$

The overall information content $ic_{overall}$ q is given by the following expression.

$$ic_{overall} = \frac{\sum_{i=1}^{n} Info(d_i, q) - \sum_{i=1}^{n}\sum_{j=i}^{n} Area(d_i \cap d_j)\left[\frac{\sum_{i=1}^{n} Info(d_i, q)}{\sum_{i=1}^{n} Area(d_i \cap q)}\right]}{Area(q)},$$

where $Info(d_i, q)$ computes the information content of d_i in q. The second expression in the above equation multiplies the total area by the average information content present in the overlapping area to discount the effect of redundant information content. The weighted sum of the above three equations gives the ranking function for q.

$$RankQuadrant(D, q) = w_s . s_{overall} + w_r . r_{overall} + w_i . ic_{overall},$$

where $w_s, w_r, and\ w_i$ are suitable user-defined weights.

5. EXPERIMENTS

In this section we present our experimental results. We first compare the effectiveness of our modified *tf-idf* technique on top of GIS datasets with the current metadata ranking mechanism. Later we show experiments that compare our heuristics with other naïve techniques. We first describe the settings of our experiment and the datasets we used.

We first used school datasets with point geometry available from the ESRI Data Catalog. There were two different school datasets, one containing information about schools in the USA and the other containing information mixed with other geographic landmark features such as churches and hospitals. A rough content analysis of these two datasets led to the conclusion that there were a lot of different kinds of information, and one dataset contained more schools than the other. We restricted our analysis to mainly the region displayed in Figure 4, which consisted of the states of New York, Pennsylvania, and New Jersey. We clipped the datasets to match the above regions. The intuition for selecting three adjacent states as analysis regions is due to the fact that the query region may span multiple jurisdictions. A disaster occurring at the border of these states would require datasets to be integrated from different sources such as local, county, state, and federal levels. We also obtained datasets for these regions from the respective state GIS data clearinghouses and other sources. Finally, we got 35 school datasets from different sources that matched the analysis region. The maximum size of the datasets consisted of 25,000 records and the minimum consisted of 2,000 records. The sizes of the datasets we used were a mix of large, medium, and small sizes. For all practical purposes, we considered these datasets as our remote data sources, even though all our analysis were done locally. For each dataset we took a random sample of 15-20% of its objects and all our analyses were performed on these sampled objects.

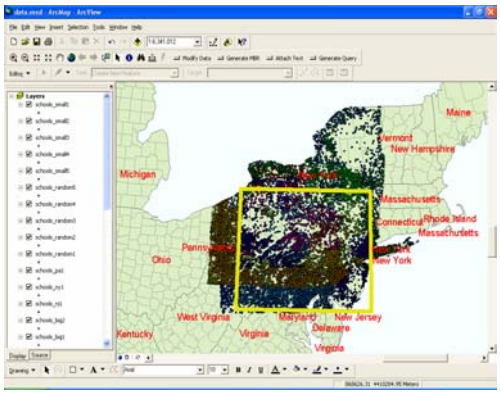

Figure 4. Different overlapping datasets in the analysis region.

5.1 Improved Dataset Ranking

We conducted experiments to evaluate our improved *tf-idf* model on the GIS datasets. We indexed all the keywords present in the dataset samples. We used the s_{table} function (see Section 3.4) to find the score for each dataset and ranked them according to the score. The results in comparison with metadata ranking using s_{meta} for query keywords "`junior schools`" are shown in Table 5. In order to avoid bias due to spatial coverage, we used five datasets that had roughly the same spatial coverage. It is shown that the metadata ranking hardly differentiated any of the

datasets. However, our table ranking differentiated the datasets quite well.

Table 5. Metadata ranking vs. information ranking.

Dataset	Metadata Ranking	Table Ranking
D1	1	3
D2	1	5
D3	1	1
D4	1	2
D5	1	4

5.2 Progressive Approximate Algorithm

We implemented our progressive approximation algorithm and ran the algorithm on the school datasets for different query sizes. Our initial query range was 1000 miles by 1000 miles and the keywords used were "`junior schools`". We uniformly placed range queries of the same size over the entire analysis region and the results were averaged over all the queries. The results are shown in Figure 5. The x-axis is the number of iterations, and the y-axis is the normalized weighted score. It shows the results of three techniques in selecting the nodes for traversal after each iteration. In the first technique, we used our virtual quad partition and applied our heuristics in selecting the nodes. In the second technique, we used a grid partition. In the grid partition, we divided the query region initially into 4, and then recursively divided into multiples of 4 such as 16, 64, 256, and so on. In the third technique, we used virtual quad and randomly selected a node to traverse. The results show that our heuristics works superior in comparison with the other two. The grid method performs slightly better than the random method. This is because the grid based method behaves like a breadth first search and gets an overall gain by not traversing any deeper partition. This experiment clearly shows that our heuristics always picked a node that can potentially maximize the score.

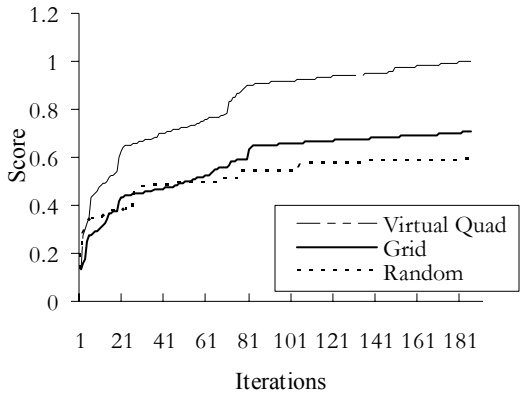

Figure 5. Comparison of iteration versus score.

The proof of Theorem 1 (see Section 3.3) can also be seen from Figure 5. As we went deeper into spatial regions (more the iterations, deeper the analysis is), the quality measured as weighted score also increased.

We plotted and compared the time taken and the number of nodes processed by the three methods. The results are shown in Figure 6. The grid technique processed more nodes compared to the virtual quad method and the random method. However, the virtual quad method achieved a very high score even at the initial stages. The random or grid method processed more nodes, and the virtual quad method achieved a quite significant score. The previous graph shows that it took around 80 iterations for the heuristic technique to achieve the highest score for which the time corresponds to about 5 seconds. It should be noted that these different techniques pick nodes that are different in size during the traversal stage. It is interesting to note that the difference in iterations among the three methods is negligible for this time, but their scores vary widely. We varied the query size and observed similar effects with respect to graphs shown in Figures 5 and 6.

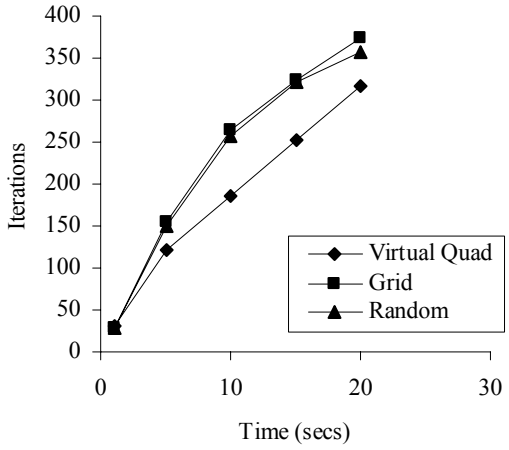

Figure 6. Comparison of time versus number of iterations.

6. RELATED WORK

GIS dataset ranking has been conventionally based on metadata [3, 11]. A multi-dimensional metadata ranking system for metadata documents based on thematic, spatial and temporal characteristics of datasets is proposed in [11]. In [3] this idea is improved by considering additional factors such as spatial hierarchy and spatial neighborhood of datasets. It also provided ontological framework for improving user interactivity. Ontology helps in better retrieval of datasets as opposed to traditional keyword matching, since different users use different terminologies for specifying the same theme during their search. In this paper, we improve their approaches by ranking datasets based on keywords present at the data level (i.e., relational table). Keyword-based search on relational tables has been proposed by a number of projects such as in DBXplorer and DISCOVER [1,2]. Their main idea is an IR-style method of retrieving rows matching user specified keywords from different tables without user knowing the schema of the table. Our approach also uses similar techniques, but focuses more on integrating data sets.

One perspective of GIS data integration is to automatically conflate heterogeneous datasets such as imagery, maps, and vector data [5, 6]. The idea is to overlay non-geo-referenced aerial images on geo-referenced vectors or maps, by automatically generating control-pairs using image processing techniques. Image datasets carry better visual information and vector datasets carry better attribute information. The combination provides a powerful information enhancement for analysts. However, this

technique is different from our work and may not be useful for integrating quality-varying datasets. This paper does not deal with the alignment problem, but rather integrates multiple source datasets and provides users a single unified dataset with a high quality. Other important GIS data integration projects such as [4, 5] develop geographic mediation systems for querying online heterogeneous GIS databases. Their work concentrates on how to process user queries using such mediation systems.

7. CONCLUSIONS

GIS data comes from different sources with varying quality and redundancy. There is an increasing need for analysts to obtain GIS data in a timely manner and in an integrated fashion without worrying about the sources and post-processing overhead. In this paper, we addressed this concern by proposing approximate methods by using space-partitioning techniques. The quality of result refines as we do deeper analysis. We showed experimental results of our techniques and compared it with other naïve methods.

As future research directions, we can develop optimal space-partitioning techniques for the local analysis based on some characteristics of the datasets. Another possible research direction is to incorporate ontology-based search on GIS datasets. Since datasets come from different sources, their representation and meaning of data differ widely. Hence ontology becomes critical for retrieving relevant data for analysts. It will be interesting to investigate different sampling techniques that can closely represent a data source in terms of spatial and keyword distribution.

8. ACKNOWLEDGEMENTS

This work is supported by the National Science Foundation under Award Number 0331707.

9. REFERENCES

[1] S. Agrawal, S. Chaudhuri, and G. Das. DBXplorer: A system for keyword-based search over relational databases. In Proceedings of the 18th International Conference on Data Engineering, San Jose, CA, 2002.

[2] A. Balmin, V. Hristidis and Y. Papakonstantinou. ObjectRank: Authority-based keyword search in databases. In Proceedings of the 30th VLDB Conference, Toronto, Canada, 2004.

[3] M.K. Beard and V. Sharma. Multidimensional ranking in digital spatial libraries. Special Issue of Metadata. Journal of Digital Libraries, Vol.1, No. 1, 1997.

[4] O. Boucelma, M. Essid, Z. Lacroix, J. Vinel, J-Y. Garinet, and A. Betari. Virgis: Mediation for geographical information systems. In Proceedings of the 20th International Conference on Data Engineering, Boston, MA, 2004.

[5] C-C. Chen, C.A. Knoblock, C. Shahabi and S. Thakkar. Automatically and accurately conflating ortho imagery and street maps. In Proceedings of the 12th ACM International Workshop on Geographic Information System, ACM-GIS, Washington, DC, 2004.

[6] C-C. Chen, C. Shahabi, C.A. Knoblock and S. Thakkar. Automatically and accurately conflating satellite imagery and maps. In Proceedings of the International Workshop on Next Generation Geospatial Information, NG2I, Boston, MA, 2003.

[7] M. Essid, O. Boucelma, F-M. Colonna and Y. Lassoued. Query processing in a Geographic Mediation System. In Proceedings of the 12th ACM International Workshop on Geographic Information System, ACM-GIS, Washington, DC, 2004.

[8] Federal Geographic Data Committee (FGDC), www.fgdc.gov.

[9] H. Garcia-Molina, Y. Papakonstantinou, D. Quass, A. Rajaraman, Y. Sagir, J. Ullman, V. Vassalos and J. Widom. The TSIMMIS approach to mediation: data models and languages. Journal of intelligent information systems, 1997.

[10] Geospatial-one-stop, www.geodata.gov.

[11] S. Gobel and P. Klein. Ranking mechanisms in meta-data information systems for geo-spatial data. In Proceedings of EOGEO, Ispra, Italy, 2002.

[12] A. Guttman. R-trees: a dynamic index structure for spatial searching. In Proceedings of ACM/SIGMOD Annual Conference on Management of Data, Boston, MA, 1984.

[13] V. Hristidis, L. Gravano and Y. Papakonstantinou. Efficient IR-style keyword search over relational databases. In Proceedings of the 29th VLDB Conference, Berlin, Germany, 2003.

[14] I. Lazaridis and S. Mehrotra. Progressive approximate aggregate queries with a multi-resolution tree structure. In Proceedings of SIGMOD Conference, Santa Barbara, CA 2001.

[15] J. Radke, T. Cova, M.F. Sheridan, A. Troy, M. Lan and R. Johnson. Application challenges for geographic information science: implications for research, education, and policy for emergency preparedness and response. URISA Journal, Vol. 12, No. 2, Spring 2000.

[16] H. Samet. The design and analysis of spatial data structures. Addison-Wesley, Reading, MA, 1990.

[17] G. Salton. Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison Wesley, 1989.

[18] A. Singhal. Modern information retrieval: a brief overview. IEEE Data Engineering Bulletin, Special Issue on Text and Databases, Vol. 24, No. 4, December 2001.

[19] S.K. Thomas, S.L. Cutter, M. Hodgson, M. Gutekunst, and S. Jones. Use of spatial data and geographic technologies in response to the September 11 terrorist attack. Quick Response Report #153, Natural Hazards Center, University of Colorado, 2002.

[20] The RESCUE Project, http://www.itr-rescue.org/.