Issues in Location-Based Indexing for Co-operating Mobile Information Systems

Wendy Osborn[1] and Annika Hinze[2]

[1] University of Lethbridge, Canada
wendy.osborn@uleth.ca
[2] University of Waikato, New Zealand
a.hinze@cs.waikato.ac.nz

Abstract. Mobile information systems need to collaborate with each other to provide seamless information access to the user. Information about the user and their context provides the points of contact between the systems. Location is the most basic user context.

TIP is a mobile tourist information system that *also* provides location-based access to documents in the digital library Greenstone. This paper identifies the challenges for providing efficient access to location-based information using the various access modes a tourist requires on their travels. We discuss our extended 2DR-tree approach to meet these challenges.

1 Introduction

Mobile information systems need to collaborate with each other to provide seamless information access to the user. The intention of a combined information system is that a user, traveling with an Internet-connected mobile device such as a pocket PC, is actively presented information based on their location (automatically detected through GPS) and a user profile that records their interests (such as architectural follies). When a passing detail seems particularly pertinent or piques their interest, the user seamlessly taps into the "deeper" resources managed by the digital library that can better satisfy their quest for both more details and related information. Location-based search provides access to a set of rich materials with cross references between different digital library collections and the tourist information system. Location is the most basic context of a user. Location-based access to static spatial data has been proposed for Geographic Information Systems (GIS). Efficient handling of location context of mobile users still creates challenges to be addressed.

Our goal is to identify the challenges for *efficient* location-based and context-aware mobile access to digital libraries and other information systems. Section 2 provides an introduction to the mobile tourist information system TIP, the Greenstone digital library system, and the TIP/Greenstone Bridge. Section 3 explores the different forms of location-based access that are necessary from the user's point of view and the system's perspective. Section 4 details background information on location-based indexing. Section 5 introduces our approach to efficient index-based access to location-based data.

R. Meersman, Z. Tari, P. Herrero et al. (Eds.): OTM 2007 Ws, Part I, LNCS 4805, pp. 226–235, 2007.
© Springer-Verlag Berlin Heidelberg 2007

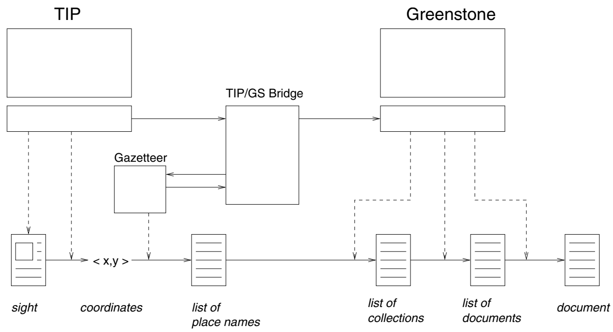

Fig. 1. Data flow in the TIP/Greenstone bridge

2 Background: TIP and Greenstone

This section gives a brief introduction on the Tourist Information Provider (TIP) and Greenstone. TIP [4] delivers location-based information to mobile users based on their context, such as current location, interest in particular semantic groups of sights and topics, and travel history. Sights and topics are captured in a user's profile. Examples of sights include art, buildings and beaches, while examples of topics are history and architecture. A user's travel history includes locations and sights that the user visited and their feedback about these sights. Here, we focus on user location, and briefly discuss inclusion of other contexts.

Greenstone is an open source digital library toolkit [12]. Examples of digital libraries that have been created with it are on historic newspapers, books on humanitarian aid, first editions of works by Chopin, and scientific repositories. The TIP/Greenstone bridge [5] provides TIP users with location-based access to documents in a digital library. Fig. 1 shows the conceptual interplay between TIP and Greenstone as they are connected via the Bridge, using the Gazetteer. A Gazetteer provides information about locations, such as population, province, and country. The lower part of Fig. 1 shows the data that are retrieved.

3 Location-Based Access and Types of Queries

We first describe location-based retrieval from the user's perspective and then analyze which types of queries are necessary to support the user.

3.1 Location-Based Retrieval from the User Perspective

User at location. The user is situated at a location. We consider them to be static in their location in regards to the queries. At a new location the user will issue a separate query. We identify three types of user requests.

– A: *Retrieve documents about this sight or place.* The user is situated at a sight or is accessing information about a sight in TIP. A sight example is

Fig. 2. Examples of query types

the 'University of Waikato', while place examples include 'Hamilton' and 'Rotorua'. Starting from the given sight or place, the user wants to find related documents or collections within a Greenstone digital library. The input may be a point `<x,y>` (Fig. 2(a)) or a 2D area (Fig. 2(b)).

– B: *Retrieve documents related to location* `<x,y>`. Starting from location `<x,y>`, either in person (GPS location) or virtually (e.g. on a map), the user wants to retrieve related documents or collections within Greenstone. This situation is depicted in Fig. 2(a). Note the issues of layering and hierarchy: given the coordinates `<x,y>`, a number of spatial objects and concepts may be identified. An example is shown in Fig. 3. Some of the objects belong to a common hierarchy, such as the library building being part of the 'University of Waikato'. Some objects are layered, as they belong to different concepts. For example the 'North Island' as a geo-spatial territory shares a subspace with 'New Zealand' as the political concept of a country.
– C: *Retrieve data related to a place mentioned in a Greenstone document.* The user follows a hyperlink that has been created by the TIP/Greenstone bridge for a place name in a Greenstone document. The target of the link is TIP, the Gazetteer, or other Greenstone documents and collections that mention this place. This cross-referencing function allows the user to link to information from across the co-operating sources.

In addition to these queries, there exist the 'typical' location-based queries in TIP: based on the current coordinates `<x,y>` of the user the system retrieves information about sights near by. These queries are currently answered by TIP's postGIS database. Note that only those sights are retrieved that match the user's other context conditions, such as their interest profile and travel history.

Virtual user travel. In this situation, a query is triggered by the user's travel planning on the map. The user draws a line that defines the trajectory they will take, and requests to see everything that is within a distance of ϵ (ϵ-range) of their chosen trajectory. The value of ϵ is set by the user. For example, if the user prefers sights or documents that are nearby, a small ϵ value is chosen. If the user wants to know about sights that are further away, a larger value needs to

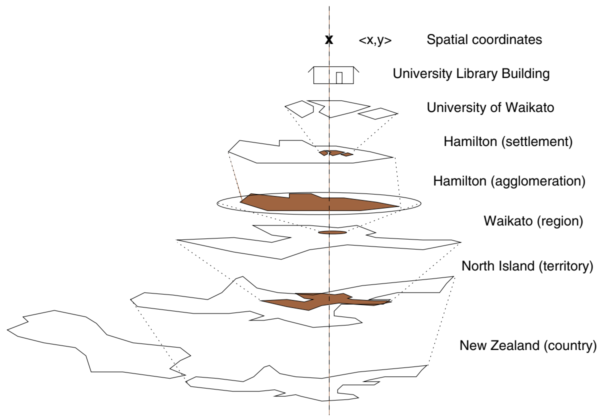

Fig. 3. Hierarchy and Layering of geo-spatial and conceptual objects

be specified. An example is depicted in Fig. 2(d). Note the similarities to area queries (see Fig. 2(b)). We identify two types of queries:

– A: *Retrieve all sights in TIP that are within ϵ-range of this trajectory*
– B: *Retrieve all documents and collections in Greenstone referring to place names that are within ϵ-range of the trajectory*

Mobile user (changing locations). Here, the user moves along a trajectory that is unknown beforehand, and continuously queries the system for information. An example is shown in Fig. 2(c).We identify two query types:

– A: *Retrieve all TIP sights within ϵ-range of the stream of locations*
– B: *Retrieve all documents and collections in Greenstone referring to places within ϵ-range of the stream of locations*

3.2 Location-Based Retrieval from the System Perspective

Given the different types of location-based queries required by the user, the following queries need to be supported:

1. **Point queries.** A point query takes a single coordinate and query for items (documents, sights, files) that are associated with this location (see Fig 1). Point queries are not very typical as they refer to single `<x,y>` coordinates. The granularity of GPS data is too fine for items to be exactly at given coordinates. In addition, travelers typically are close to objects of interest but not standing directly on them. Therefore, most point queries are transformed into region queries with a radius of ϵ to cover objects in close proximity to the given location. Thus, most point queries are executed as 2D range queries.
2. **2D range queries.** As argued before, most point queries are executed as 2D range queries. In addition, genuine range queries are triggered, for example, by travel planning on the map (c.f. virtual user travel). Range queries refer to

polygon shapes and retrieve all objects linked to points within the polygon. Area queries are simple 2D range queries. 2D range queries are typical in situations where the user is stationary. Efficient access for 2D range queries requires location-based indexes.
3. **Incremental 2D range queries.** These queries follow a trajectory that is evolving as the mobile user is traveling (i.e., a stream of location coordinates). They can also be viewed as range queries starting from a nearby point; see Fig. 2(c). Efficient access for incremental queries requires preservation of information on the spatial relationships (e.g. north, northeast, east, etc.) between objects within the index.

3.3 Issues Identified

From our analysis, we identify the following challenges that need to be addressed for the support of efficient location-based mobile information access:

1. Efficient execution of 2D range queries and incremental range queries. Index structures need to be developed that provide direct and efficient support for the identified query types. Efficiency is especially important in a mobile environment where the user's location may be constantly changing.
2. Support for a wider notion of context, such as time or user preferences. As argued before, information access depends on a variety of context data. Indexes that ignore the context dimensions will encounter mapping problems.
3. Index structures need to support a distributed setting where location-based data may be provided by various providers, depending on the location and available connectivity of the mobile users.
4. Location-based indexing for mobile applications needs to explicitly support hand-held mobile devices. In addition to efficient retrieval, location-based and context-aware caching on devices with limited capacity or support for caching is also an important issue.

4 Background: Spatial Indexing and Retrieval

This section introduces some background on spatial data, spatial queries and spatial indexing, that is necessary for our approach to location-based retrieval in TIP. Extensive introductions can be found in [2,6,9,11].

Spatial data is represented in multidimensional space. Examples include simple points, lines, polygons and complex objects composed of sub-objects. For example, the region of Waikato contains towns (i.e. points), cities (i.e. regions), and roads (i.e. multiple connected lines, or linestrings). Each feature has a location in space. In addition, all features are spatially related to each other (for example, the city of Rotorua is located southeast from the city of Hamilton).

Many types of spatial search exist [2], including point and region searches. The point search finds all points or objects that overlap a query point. An example is a point search that identifies objects higher up in the hierarchy, such as towns or countries (see Fig. 3). The region search finds all objects and points that overlap

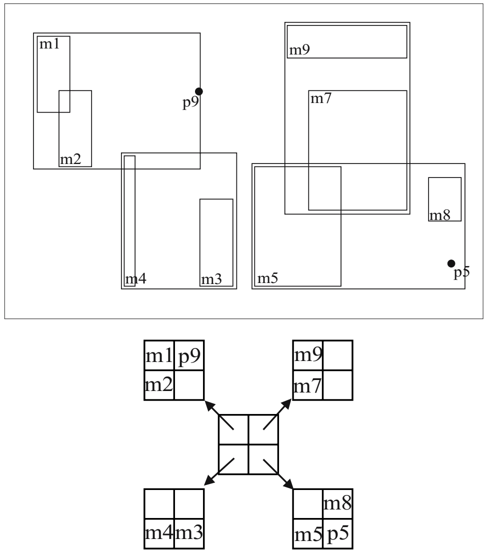

Fig. 4. Order 2*2 2DR-tree

a search region. An example of a region query is the retrieval of documents within the ϵ-range of a user (see Fig 2(b)).

Many spatial indexes are proposed that are based on the R-tree [3], the first multidimensional variant of the B$^+$-tree [1]. Objects and points are indexed in a uniform manner, by storing approximations for objects in the leaf nodes, and approximations for subregions that contain objects in the non-leaf nodes.

A spatial index that has been recently proposed is the 2DR-tree [6,7,8]. Unlike most spatial indexes, the 2DR-tree uses 2-dimensional nodes, which are the same dimensionality as the search space. Objects can be placed in nodes so that all spatial relationships (e.g. north, east, northeast) between them are preserved. Therefore, alternative search strategies can be employed, such as a binary or greedy search. Also, an incremental range query that traverses across nodes in an efficient manner is a viable option. For these reasons, we employed a 2DR-tree in our strategy to be described in Section 5. We summarize how spatial relationships are defined, followed by an example of a 2DR-tree. Other details on the 2DR-tree are available in [6].

Each node has X index values along the x-axis, and Y index values along the y-axis. For each location (i,j) in the node that contains an object or subregion, defined as $MBR_{(i,j)}$, each other location that is south, east, or southeast of (i,j) contains an MBR that is south, east or southeast of $MBR_{(i,j)}$, while each location that is northeast of (i,j) contains an MBR that is northeast of $MBR_{(i,j)}$. Similarly, each location that is north, west or northwest of (i,j) contains an MBR that is north, west or northwest of the centroid for $MBR_{(i,j)}$, while each location that is southwest of (i,j) contains an MBR that is southwest of $MBR_{(i,j)}$.

Fig. 4 shows a 2DR-tree that preserves all spatial relationships for the given points and object approximations (taken from [2]). In the leaf node (m4,m3), m3 is located southeast of m4, so m3 is located east of m4 in the node. In node (m5,p5,p8), p5 is located southeast of m5, and m8 is located northeast of m5 and northwest of p5. Therefore, p5 is stored east of m5 while m8 is stored northeast of m5 and north of p5. The remaining two leaf nodes, (m7,m9) and (m2,m1,p9) preserve the spatial relationships between their objects. Spatial relationships between subregions in the root are also preserved.

5 Spatial Indexing in TIP

This section addresses the first of the four issues identified in Section 3.3. We first present the proposed TIP indexing structure, followed by our preliminary approach for processing an incremental range query using the index. We also present extensions to our approach, and other issues under consideration.

5.1 The TIP Index

The TIP index will consist of three parts. The first is the spatial index. A 2DR-tree [6,7,8] will be constructed using the coordinates from the Gazetteer. The second part will consist of locations in the Gazetteer. Each point in a leaf node will reference a list of all locations at the point. For example, one point can have Hamilton Gardens, Hamilton, and Waikato associated with it. Also, for each location, the information provided in the Gazetteer, such as population, can also be stored with the location. The third part consists of a list of Greenstone collections. Each location will reference a corresponding list of collections.

Fig. 5 depicts a portion of a TIP indexing structure that includes Hamilton and the area surrounding it. Focusing on the leaf level of the 2DR-tree, one node contains 3 pairs of <x,y> coordinates. One pair references a list of all locations accessible from the point. These locations are the spatial objects and concepts depicted back in Fig. 3. Each location will reference a list of Greenstone collections that are related to the location. For example, 'Hamilton' will refer to the 'Plant and Garden' and the 'Old NZ maps' collections.

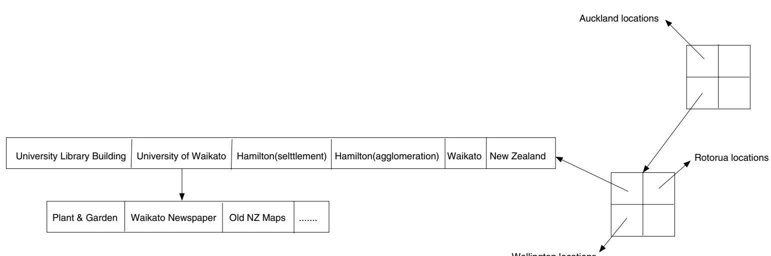

Fig. 5. Spatial indexing for TIP

5.2 Incremental Search

Our preliminary approach to the incremental range query extends the existing 2DR-tree binary search strategy (see [6,7] for details) with node traversal. The incremental range query works as follows. When the user begins their trip, the coordinates of their initial location are provided to the TIP index. A binary search is performed to locate both a node and a location within the node. This serves as the starting point for the incremental the range query. From here, the query is guided by the user's movements. For example, if the user moves in the northeast direction, the range query will proceed in the northeast direction within the node. Likewise, if the user moves in a southeast direction, the range query will proceed to the east within the node. Similarly, the user's movements will guide the movement of the range query within the node.

For example, we consider the user at the University of Waikato. When they switch to the TIP/Greenstone bridge, the coordinates of their location are provided to the spatial index. A binary search is performed, which leads to the list of locations given in Fig. 5. After the user selects Hamilton, the corresponding lists of Greenstone collections are retrieved and displayed to the user. If the user proceeds southeast towards Rotorua, then the range query will move to the east within the original leaf node. Then, all locations accessible from the next coordinates can be retrieved and displayed to the user. Similarly, the set of locations related to Wellington can be retrieved simply by traversing the leaf node to the south from the coordinates for Hamilton.

This guided search is possible because the points in the 2DR-tree are organized based on their relationships with each other. In addition, as long as the incremental range remains within the node, we do not need to perform a new query from the root of the tree every time the user moves.

5.3 Extensions

Incremental search is expected to work well if the user stays in a small area. Unfortunately, if the user travels any significant distance, additional binary searches will be necessary because the bounds of the current node will be reached. For example, if our user travels east or southeast of Rotorua, or northwest of Hamilton towards Auckland, a new binary search would need to be performed in order to locate a new starting point for the incremental range query.

One extension to the spatial index for TIP that is under consideration is to relate the leaf nodes with links. This idea is similar to the leaf node links used in the $B^{+} - tree$ [1]. Fig. 6 depicts an example link between two leaf nodes. Here, if the user proceeds northwest from Hamilton to Auckland, the range query can be moved to the next node by following the link. This would eliminate the need for an entire binary search to be performed.

5.4 Other Considerations

Leaf Node Links. When adding links between leaf nodes, it must be done so that spatial relationships are preserved between points that exist across

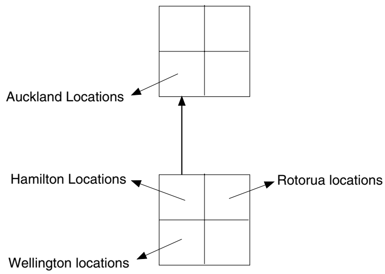

Fig. 6. Link between two leaf nodes

different nodes. Because the TIP index incorporates the 2DR-tree, all spatial relationships are preserved between objects or points in each node at the leaf level, and between all regions that contain subjects of objects or points in each node at the non-leaf levels [6]. Therefore, adding links at the leaf level looks promising.

However, the current method of constructing a 2DR-tree is by inserting each object or point one at a time. Although spatial relationships between objects within a node are maintained, the spatial relationship between two objects that belong to two different nodes cannot be easily determined. We suggest a different technique for constructing the TIP index, namely a bottom-up tree construction strategy. This strategy will require a strategy for clustering coordinates that considers the spatial relationships between them. Then, the clustered coordinates are organized into a 2DR-tree. It is expected that the resulting tree structure will lend itself to the linking of its leaf nodes.

Size of a Range Query. The proposed incremental range query assumes that the search region does not extend past the bounds of the current leaf node that is being traversed. This can be guaranteed for a point query. However, as mentioned earlier, the user rarely, if ever, stands on top of the area that they needs information about. Therefore, range queries will be used in the TIP index. We need to address how information is retrieved across multiple leaf nodes. Initially, this will be accomplished with a binary search. The problem arises when traversing across a node - it is possible that the edge of the range query travels outside the edge of the node. This problem also needs to be addressed.

Node Traversal. If a user is traveling north, south, east, west, northeast or southwest, the corresponding movement in the current leaf node is straightforward. If the user is traveling northwest or southeast, the next move is not as obvious, since for each there are three options to choose from. For example, if the current location in a node is $(1, 1)$, and the user wants to move southeast, moves to $(1, 0)$, $(2, 0)$, or $(2, 1)$ are possible. Methods to determine the right movement must be considered.

6 Conclusion

In this paper, we analyse the co-operating systems TIP and Greenstone for challenges for efficient access to context-aware information. We focus on location-based access to digital libraries. Four challenges were identified: (1) 2D-range and incremental-range queries, (2) context dimensions, (3) distributed indexes, (4) mobile devices. We presented our 2DR-tree indexing extension and approach for incremental range queries to address the first challenge. The challenge of further context dimensions will be addressed in the near future by extending the 2DR-tree into a nDR-tree.

References

1. Comer, D.: The ubiquitous B-tree. ACM Computing Surveys 11, 121–137 (1979)
2. Gaede, V., Günther, O.: Multidimensional access methods. ACM Computing Surveys 30, 170–231 (1998)
3. Guttman, A.: R-trees: a dynamic index structure for spatial searching. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 47–57. ACM Press, New York (1984)
4. Hinze, A., Buchanan, G.: The challenge of creating cooperating mobile services: Experiences and lessons learned. In: Estivill-Castro, V., Dobbie, G. (eds.) ACSC 2006. Twenty-Ninth Australasian Computer Science Conference. CRPIT, Hobart, Australia, vol. 48, pp. 207–215. ACS (2006)
5. Hinze, A., Gao, X., Bainbridge, D.: The TIP/Greenstone bridge: A service for mobile location-based access to digital libraries. In: Proceedings of the 10th European Conference on Digital Libraries, pp. 99–110 (2006)
6. Osborn, W.: The 2DR-tree: a 2-dimensional spatial access method. PhD thesis, University of Calgary (2005)
7. Osborn, W., Barker, K.: Searching through spatial relationships using the 2DR-tree. In: Proceedings of the 11th International Conference on Internet and Multimedia Systems and Applications (2006)
8. Osborn, W., Barker, K.: An insertion strategy for a two-dimensional spatial access method. In: Proceedings of the 9th International Conference on Enterprise Information Systems (2007)
9. Rigaux, P., Scholl, M., Voisard, A.: Spatial Databases: with Application to GIS. Morgan-Kauffman, San Francisco (2001)
10. Scholl, M., Thilliez, M., Voisard, A.: Location-based mobile querying in peer-to-peer networks. In: Proceeings of the First International Workshop on Context-Aware Mobile Services, pp. 166–175 (2005)
11. Shekhar, S., Chawla, S.: Spatial Databases: A Tour. Prentice-Hall, Englewood Cliffs (2003)
12. Witten, I.H., Bainbridge, D.: How to Build a Digital Library. Elsevier Science Inc., Amsterdam (2002)