TIP Spatial Index: Efficient Access to Digital Libraries in a Context-Aware Mobile System

Wendy Osborn[1] **Annika Hinze**[2]

[1] Department of Mathematics and Computer Science,
University of Lethbridge
4401 University Drive West,
Lethbridge,
Alberta, T1K 3M4,
Canada
Email: wendy.osborn@uleth.ca

[2] Department of Computer Science,
University of Waikato,
Private Bag 3105,
Hamilton 3240,
New Zealand
Email: hinze@cs.waikato.ac.nz

Abstract

We present a framework for efficient, uniform, location-based access to digital library collections that are external to a context-aware mobile information system. Using a tourist Information system, we utilize a spatial index to manage the context of location. We show how access to resources from within and outside of the tourist information system can be carried out in a seamless manner. We show how the spatial index can be navigated to continually provide information to the user. An empirical evaluation of the navigation strategy versus traditional spatial searching shows that navigation is efficient and outperforms traditional spatial search. In conclusion, our work provides a strategy for context-aware mobile systems to co-operate with digital libraries in a seamless and efficient manner.

Keywords: context-aware mobile systems, location-based, spatial indexing, digital libraries

1 Introduction

A mobile tourist information system provides information to a traveller based on several factors, including location. In addition to providing the user with information managed by the system itself, information that is available from local, external repositories should be made available to the user. In particular, digital libraries contain a wealth of information for the user to access. The overall goal of our information system is as follows: While travelling with a mobile device such as an iPod, the user is continually presented with information from not only the mobile information system, but also from digital libraries, based on their location and their user profile.

To accomplish this requires efficient location-based organization and retrieval of information that is both internal and external to a mobile tourist information system. Two issues arise around the location-based indexing of information: 1) There exists many municipalities in the world, each of which has its own co-ordinates. (For example, World Gazetteer (http://www.world-gazetteer.com/ contains over 300,000 co-ordinates.) 2) in addition to existing municipalities, a use may want information from a location that is not specific to a municipality (e.g. an iceberg that has migrated to a location off the coast of New Zealand). Given both sources of co-ordinates, there is the potential of having to index close to a million co-ordinates. Therefore, information must be organized by its location in such a way that efficient information retrieval based on the user's current location is possible.

In this paper, we present an efficient location-based index structure. We show the use of of index when accessing digital libraries that are external to a context-based mobile system. Our primary goals are twofold. The first is to provide a framework for efficient and uniform access from a context-aware mobile information system to resources contained externally in digital libraries, as well as resources within the mobile information system itself. This is accomplished by utilizing a spatial index. The second is to propose an algorithm for efficiently navigating the spatial index to continually provide the user with information whenever it exists. An empirical evaluation of the navigation algorithm versus traditional spatial searching shows that the navigation algorithm is efficient and outperforms repeated spatial searching.

Initially, we focus on the location context in this work. We also focus on external resource access from a mobile tourist information system. However, it is expected that a self-contained module will evolve from this work that could be utilized by any mobile information system to access external resources. We also note that, although we present some visualization in order to provide context for the paper, the presentation/visualization of any resources that are fetched by TIP is outside the scope of the work described in this paper.

The remainder of this paper is structured as follows. Section 2 provides a brief summary on TIP (Tourist Information Provider), digital libraries and the mqr-tree spatial index, all of which are required

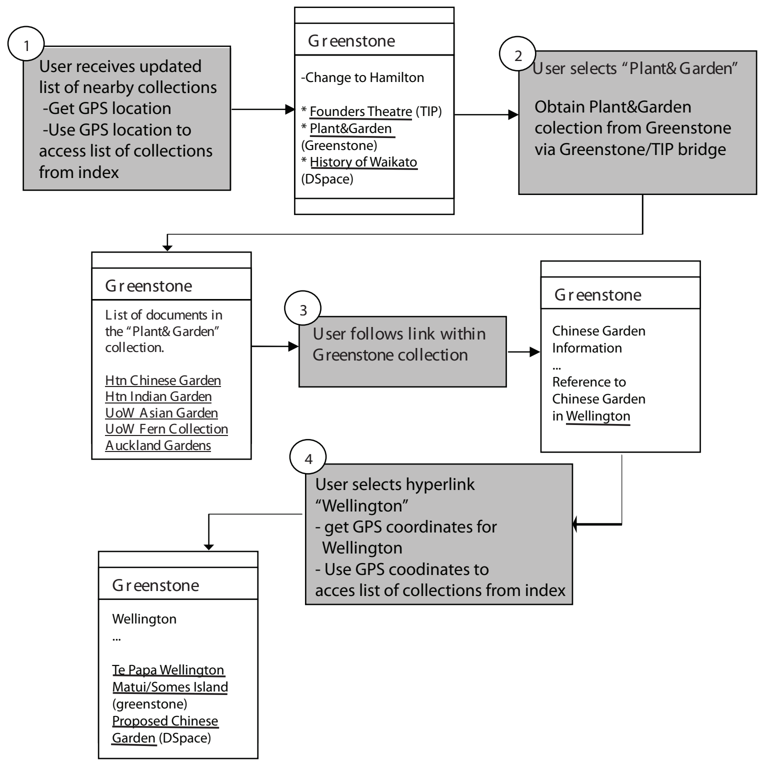


Figure 1: TIP/Greenstone Interaction Sequence

for our work. Section 3 summarizes related work and its limitations. In addition, this section will present our contributions. Section 4 presents our first user scenario that demonstrates uniform access to both internal and external resources. Section 5 presents the required modifications to the TIP architecture, while Section 6 presents the TIP tree, its organization of internal and external resources, and the algorithm for continuous navigation of the index. Section 7 presents our second scenario that demonstrates how the TIP tree is continuously navigated in order to provide updated information to the user. Section 8 presents our empirical evaluation of the TIP tree. Finally, Section 9 concludes the paper and discusses outstanding issues.

2 Background

This section presents a brief introduction on TIP, digital libraries, spatial indexing and the mqr-tree, all of which are required for this work. TIP (Hinze et al. 2009) provides the user with information based on their contexts such as current location, interests, and travel history. It is organized by separating the core system logic and TIP data from the various services that interact with it. Therefore, this architecture would allow us to: 1) modify how the location context is handled, and 2) extend it to provide access to external resources. These features are the reasons why we chose TIP for our work.

A digital library (Witten et al. 2009) is a collection of documents that are digitized, individually catalogued, and organized for online dissemination and access. Digital library software systems provide the means for searching and browsing. Some digital libraries can be searched and browsed using different metadata. Many digital library software systems exist, including Greenstone (New Zealand Digital Library Project 2008, Witten et al. 2009), DSpace (Tansley et al. 2006) and Fedora (Lagoze et al. 2006).

Spatial indexing (Shekhar & Chawla 2003) provides an efficient mechanism for accessing data using location, such as longitude and latitude co-ordinates. Many spatial indices have been proposed in the literature (see (Gaede & Günther 1998, Shekhar & Chawla 2003) for surveys). A spatial index that has been proposed recently, the mqr-tree (Moreau et al. 2009), which organizes point and object data using the inherent spatial relationships that exist between them (e.g. one object is northeast of another). This provides support for efficient navigation within the spatial index itself. In addition, performance comparisons versus other spatial indices (Moreau et al. 2009) shows that the mqr-tree supports a one-path search when indexing point data such as co-ordinates – a feature that most other spatial indices cannot achieve. For these reasons we chose to modify the mqr-tree for our work.

3 Related Work

Many mobile tourist information systems have been proposed, including AccesSights (Klante et al. 2004), CATIS (Pashtan et al. 2003), CRUMPET (Poslad et al. 2001), Cyber-Guide (Abowd et al. 1997), Guide (Cheverst et al. 2002), Gulliver's Genie (O'Hare & O'Grady 2003) and TIP (Hinze et al. 2009). Most of these systems focus on providing structured information and recommendations. Additionally, most of these systems only provide the user with information that is directly maintained by the system itself. They do not provide information from sources that are maintained in external sources, such as digital libraries.

The first system to be proposed that integrates a mobile tourist information system with a external digital library is TIP. The TIP/Greenstone

Bridge (Hinze et al. 2006) provides the functionality for TIP to connect to and retrieve information from a Greenstone digital library (Witten et al. 2009). In addition to the information that is provided by TIP for a specific location, a user can obtain further information on items and features in a Greenstone digital library that are related to the same location. Some limitations of the TIP/Greenstone bridge include the following: 1) it only retrieves external documents from Greenstone collections, 2) it only works with geographically aware digital libraries (i.e. place names must be identified and marked up), 3) TIP locations are defined by place names, while a mobile device defines location using GPS co-ordinates, extra steps are required to obtain a place name and have the user select an appropriate one before the user is able to access a Greenstone collection, and 4) information needed to access internal TIP information and external Greenstone information are kept separate, and cannot be presented to the user for consideration at the same time.

Contributions of the paper. Based on our analysis of related work, this project provides the following contributions to improve upon the limitations of existing systems:

1. Incorporating a spatial index into TIP for managing access to both internal and external resources. We will organize known sources of information in a spatial index by their GPS coordinate location instead of their place name.

2. Access to any digital library collection. Our system is not restricted to only accessing digital library collections managed by Greenstone. Links to collections managed by other digital library software systems such as DSpace (Tansley et al. 2006) and Fedora (Lagoze et al. 2006) can be added to the index as well.

3. One-pass access to relevant collections. Because GPS co-ordinates can be obtained directly from a mobile device, and the information sources are organized by their GPS co-ordinates, we can directly access both internal and external sources directly by directly accessing the spatial index.

4. No requirement of a "geographical aware digital library". In order for the TIP/Greenstone Bridge to work, a collection needed to be preprocessed to identify place names (i.e. locations) in its documents. This is no longer a strict requirement. Any resource (TIP or external) can now be added by specifying their GPS location. The GPS location can be obtained by the user from a World Gazetteer.[1] However, place name mark-up provides an additional advantage. It allows for place names to be selected (as a hyperlink) and further index accesses to take place, although an extra translation step to obtain GPS co-ordinates is required. But it will not be required in all situations when the index is accessed.

5. Handling collections that reference multiple locations. If a digital library collection contains documents that mention multiple locations, the collection can be added to the index multiple times, once per every location that is mentioned.

6. Navigating the index for continuous information dissemination to the user. We base the TIP index on an efficient spatial index, which could be searched from the beginning every time a user moves along with their mobile device. However, we feel this is unnecessary. The spatial index can also be efficiently navigated to continually provide information to the user, and thus extra searching is avoided.

[1] such as http://www.world-gazetteer.com

4 Usage Scenario 1 – Retrieval of Resources

In this section we demonstrate how TIP can be utilized to access both internal and external resources. Here, our user arrives in Hamilton, NZ, and wants to access information on the city using TIP. In addition, our user finds links to other locations and is interested in looking into these further. Figure 1 shows the sequence of interactions that will take place.

First, TIP determines the user's current GPS coordinates from their mobile device. Using these coordinates, TIP searches its index to identify all resources – both internal and external to TIP – that are near the user's current location. TIP presents three links to the user – the first is information on Founders Theatre that is maintained by TIP itself, the second is a link to a Greenstone collection on plants and gardens, and the third is a DSpace collection on the history of the Waikato region. Our user is interested in gardens, and decides to follow the link to the Greenstone collection. TIP uses the Greenstone service (i.e. TIP/Greenstone Bridge) to obtain documents from the Plant&Garden collection.

Next, the user follows the link for the Hamilton Chinese Garden document. In addition to information on the garden itself, there is a link to the location of Wellington. Our user plans to visit Wellington in the near future, and decides to click on the place name for more information. TIP obtains the GPS coordinates that corresponds to Wellington and searches the index for all internal and external resources that it has on the location of Wellington and displays these to the user for further follow up. This is depicted at the end of Figure 1.

5 Modified TIP System Architecture

In this section we present our modifications to the existing TIP architecture to facilitate location-based access to TIP and external resources.

Figure 2 shows the conceptual design of TIP with the required extensions. In particular, the following components need to be added: 1) a spatial index that organizes TIP and external resources using their location co-ordinates, and 2) services to handle the retrieval of collections and documents managed by different digital library software systems. Our extended architectures shows services to handle retrievals from external Greenstone, DSpace and Fedora digital libraries. TIP already has a Greenstone service that we can use here (Hinze et al. 2006). In addition, this architecture can be easily extended to provide access to digital library collections managed by other software systems by adding the appropriate retrieval services.

6 TIP Tree: Spatial Index with navigation

In this section we present the TIP tree, which extends with mqr-tree to handle navigation of the location context. The goals of the TIP tree are:

1. To provide location-based access to information from both TIP and external digital libraries in a uniform manner; and

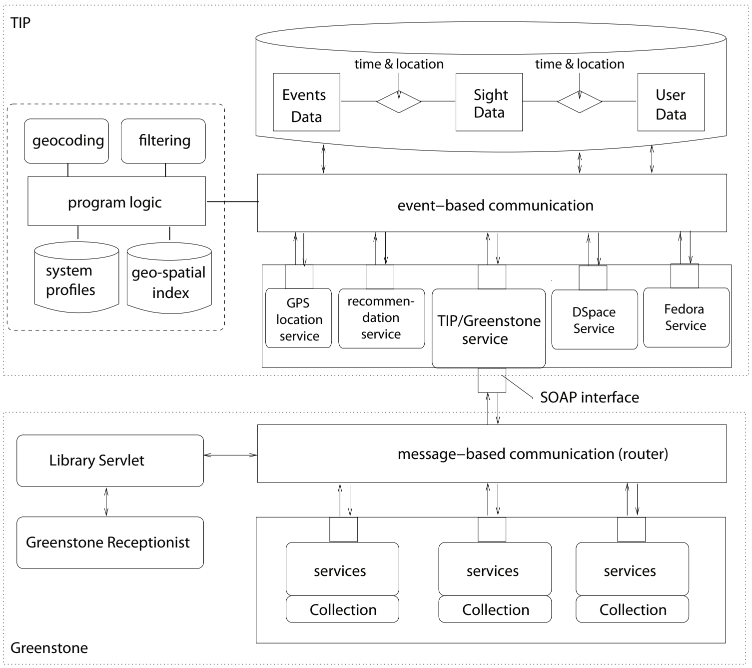


Figure 2: Modified TIP architecture and link to Greenstone

2. To provide efficient navigation of the location context so that up-to-date information is provided continuously to the user.

Each node in the TIP tree represents a bounded region in space. Each bounded region can contain a co-ordinate for a particular location and/or other bounded regions covered by child nodes. For lower-level and leaf nodes, these regions will be smaller than those represented by higher-level nodes, with the root node encompassing all regions covered by all lower-level nodes. Each co-ordinate in the TIP tree represents a location for which TIP information and/or collections in external digital libraries exists. Therefore, a leaf node stores information on each item, including its name, whether it is internal or external to TIP, and if it is external, the name of the digital library that manages it and instructions on how it is accessed (i.e. usually via a URL).

Figures 3 and 4 depict an example of the TIP tree. Here, the TIP tree is indexing resources from different locations in New Zealand. For example, the co-ordinate representing Auckland provides access to resources on All Blacks games (located in TIP), a Greenstone digital library on the Auckland Museum (external to TIP), and a DSpace collection containing documents from the University of Auckland (also external to TIP).

6.1 Tree Construction and Updating

The TIP tree is constructed in the following way. For each resource to be added (i.e. either TIP information or an external digital library) and corresponding co-ordinate[2] that indicates its location, a search is performed to see if the co-ordinate already exists in the TIP tree. If so, then the resource is added to the corresponding list. If not, then a new point representing the co-ordinate is inserted into the tree, and a new leaf node is added that contains the resource.

Updates to a resource in the TIP-tree can be performed by deleting the existing resource record, and re-inserting it with the updated information and/or location co-ordinate. Updates to a location co-ordinate can be performed by first preserving the existing leaf node it is linked to, then deleting the existing co-ordinate, inserting a new co-oridinate, and linking the new co-ordinate to the existing leaf node.

The insertion and updating algorithm used by TIP is adapted from the mqr-tree – more information on the details of insertion can be found in (Moreau et al. 2009).

Note that although the example index (Figure 4) organizes resources based on its actual physical location, it is possible to also add a collection based on a location being mentioned in a collection. For example, if a collection at the University of Auckland mentions Hamilton, this collection can be added to the corresponding list for Hamilton as well by specifying the co-ordinate that represents Hamilton.

6.2 Navigation

Figure 5 shows the pseudocode for navigating the TIP tree. Navigation begins by performing an initial search to find a starting point in the index. This puts the search point within a bounded region of space that is covered by (i.e. represented with) a node in the tree.

Within the bounded region, the search point will be located in a particular quadrant (NW, NE, SE SW). If a co-ordinate exists in the current quadrant, the corresponding list of digital library collections and TIP information will be retrieved and sent to the user, along with the distance the user is from the co-ordinate. Otherwise, nothing is displayed back to the user - if no co-ordinate exists at this location, then TIP currently had no information containing to this subregion, and therefore has nothing further to present the user.

The user will then move, which causes the co-ordinates of the search point to be updated. This will result in one of the following situations:

1. The search point remains in the current quadrant. In this situation, nothing changes other

[2] obtained from a world gazetteer

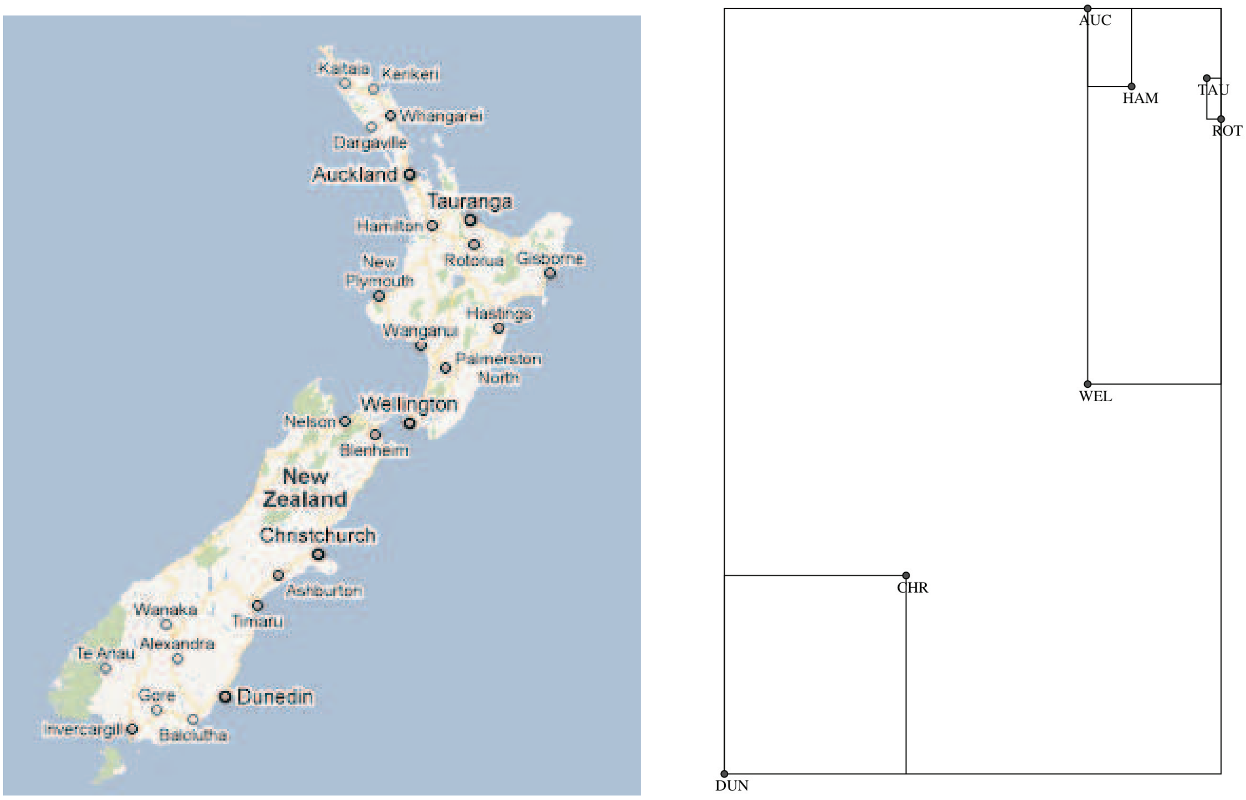


Figure 3: The TIP spatial index

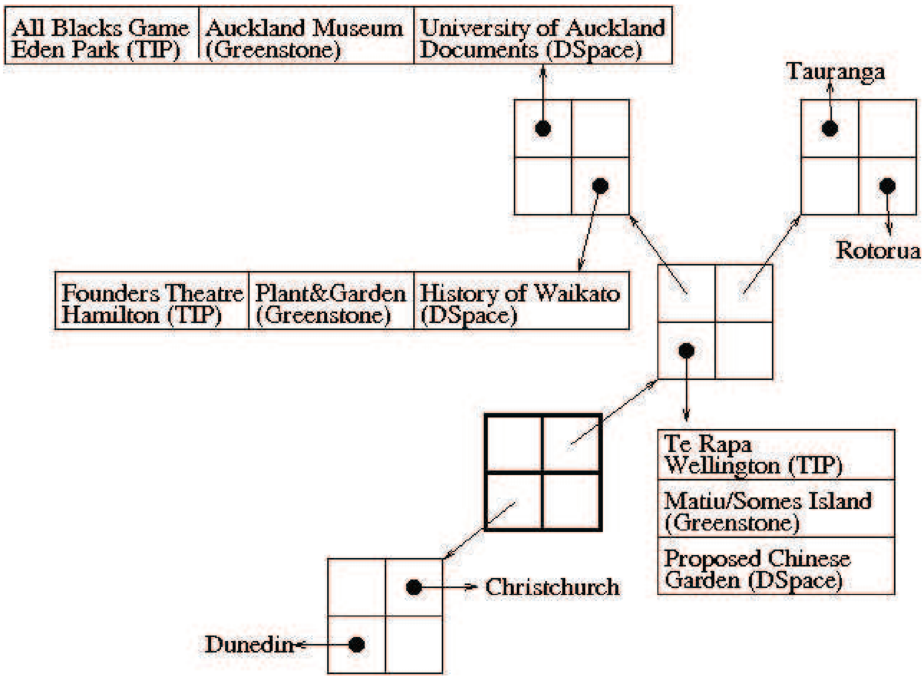


Figure 4: The TIP Tree

than updating the distance from the existing co-ordinate in the quadrant, if any. Any digital library collections and TIP information that is currently fetched for the user is still available to them.

2. The search point moves to a different quadrant. If a co-ordinate exists, then its corresponding collection and information are displayed to the user. If the quadrant references a child node and the search point is within the bounded region of the child node, then the search point is sent to the child node. Otherwise, nothing occurs until the search point is updated again - because no collections or information is indexed at this point, nothing is presented to the user.

3. The search point falls outside of the current bounded region. In this situation, the search point is sent to the parent node and re-evaluated using the above criteria.

This process repeats until the search point falls outside of the space covered by the index, at which point navigation is terminated.

7 Usage Scenario 2 – Navigation

We present an example that demonstrates TIP tree navigation. Here, the TIP user will arrive at Auckland Airport and make their way to Wellington. Along the way, TIP will notify our user of any nearby information of interest. Figure 6 shows the locations (location indicated by coloured points, referenced by numbers) where TIP will display information to the user, and the corresponding locations (using the same coloured points and numbers) in the TIP tree.

The user will begin receiving information from the TIP tree on arrival at Auckland Airport (Latitude -37.00806, Longitude 174.79167, 1 red point). An initial search of the index will set the search point in the bounded region and quadrant (NW) containing the co-ordinate for Auckland. Corresponding to the co-ordinate for Auckland are the digital library collection for the Auckland Museum, a digital library collection from the University of Auckland, and information on the upcoming All-Blacks rugby game at Eden Park. The first two collections are managed by Greenstone and DSpace respectively, and both reside on servers that are external to TIP. These collections are fetched from their respective servers using the corresponding retrieval service (see Figure 2) and made available to the user. The third item is fetched from within TIP and also presented to the user.

As the user makes their way to Hamilton (2 green point), the quadrant that the search point is in will be updated (SE) but the search remains in the same bounded region. The information available for Hamilton will now be fetched and displayed to the user. As the user continues along the west side of the North Island (3 pink point), the search point will be transferred to the parent node and its NW quadrant. Since the TIP tree contains no information on this area, nothing is retrieved for the user. Finally the user makes their way to Wellington (4 cyan point, SW quadrant, same bounded region) and information on Wellington is now displayed to them.

8 Evaluation

In this section we present our empirical evaluation of the TIP tree. We compared the performance of the TIP tree against the original mqr-tree. The TIP tree uses the navigation strategy presented in the paper to process the user's trajectory, while the mqr-tree processes a user's trajectory by performing a search on every co-ordinate in the trajectory.

```
traversing = true;
search point (x,y) = findStartingNode(x,y);
while(traversing) {

   (x,y) = update_co-ordinates(x,y);

    /*is search point within the bounded
       region covered by current node?*/

    if( (x,y) within node_region(X) ) {
          Quad = determineQuad(X);

         /*does quad contain a co-ordinate?*/
         if(contains_co-ordinate(Quad)) {
             Sights = access_all_sites(Quad);
             display_all_sights(Sights);
         }
         /*should we traverse in subtree?*/
         else
         if(with_node_region_of_child(Quad)){
            X = child_node(X);
         }
         /*otherwise, there is no information
           to display,  so nothing is done
           until co-ordinates are updated.*/
    }

    /*if search point is outside the node
       region covered by current node,
       but within space covered by index,
       then traverse in the parent node.*/

    else if( has_parent_node(X) )  {
         X = parent_node(X);
    }
    /*otherwise, outside of space covered by
      index, so terminate search*/
    else {
         traverse = false;
    }
}
```

Figure 5: TIP tree Navigation Pseudocode

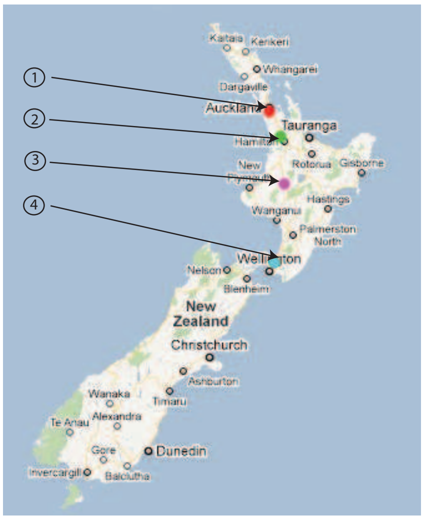
We first present an overview of our evaluation methodology. Then, we present and discuss the results of our tests, and finish with further discussion that arose from our evaluation.

8.1 Overview of Methodology

Here, we present the data sets, evaluation criteria and a brief overview of the tests that were performed.

Data sets. The evaluations used three sets of co-ordinates. Two of the three sets each contain 10,000 co-ordinates and consist of both real and synthetic points that represent various locations. The real co-ordinates represent locations of existing communities in New Zealand and were obtained from the World Gazetteer.[3] The synthetic co-ordinates represent other locations in New Zealand and are randomly generated. The rationale for using additional randomly created locations is that a collection of documents may refer to a region that does not exist in the World Gazetteer (e.g. iceberg photos from off the coast of New Zealand) or may be located in a remote area that is not maintained in the World Gazetteer.

[3] http://www.world-gazetteer.com



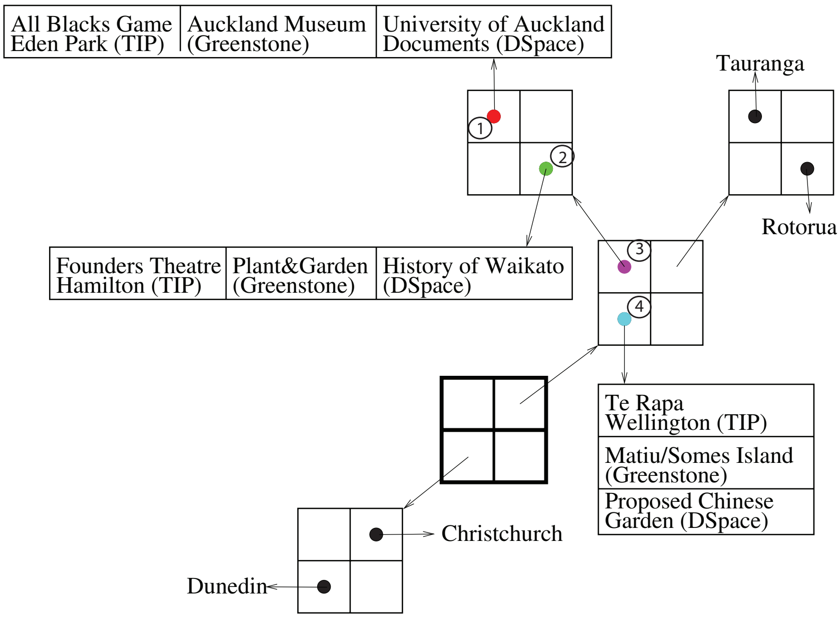


Figure 6: Navigation using the TIP tree

The third set of points represents a trajectory of 712 co-ordinates that represent a user's 16km route from Hamilton to Cambridge, New Zealand. The trajectory was recorded using a mobile device while travelling by vehicle on the most direct route between the two cities (GPS trail capturing was done using the everytrail application on android[4]).

Test criteria. The primary performance criteria in all of our tests is the number of disk accesses required to locate whether data of interest exists at a specific location or not. We evaluate the worst case scenario – no caching is used. Therefore, every time a node is required, it must be fetched from secondary storage, which requires one disk access. A secondary performance criteria is the CPU time. However, we discovered that this value is negligible. We discuss the CPU results further in Section 8.5 below.

[4] http://www.everytrail.com

Test setup. We performed four tests in our evaluations. The first test studies the effect of the number of points in the index (TIP tree and mqr-tree) on the number of disk accesses required to locate information. The second test is similar to the first, except the points are drawn over a smaller geographical region. This leads to an increased geographical density in points, while the same number of points is represented in the index. The third test studies the effect of the number of co-ordinates along the trajectory on the number of disk accesses required to locate information. The fourth test is similar to the third, but use the overall set of points drawn over a smaller geographical region. Each test is described in more detail and the results are presented below.

8.2 #Overall Indexed Points

For the first test, we used a set of 10,000 points that consists of locations around the North Island of New Zealand. We performed ten test runs. Each run constructed both a TIP tree and an mqr-tree using a subset of the points. Then, both the TIP tree and mqr-tree processed the trajectory of 712 co-ordinates. The average number of disk accesses per co-ordinate was calculated for each data structure. Each of the ten test runs used 1000, 2000, 3000, up to 10,000 points to create the trees.

Figure 7 depicts the results of the test runs. From the graph, we observe that on average, the TIP tree only needs to execute one disk access per co-ordinate in the user's trajectory. For the mqr-tree, the average number of disk accesses is significantly higher and increases as the number of points in the mqr-tree increases. For 1000 points, the average number of disk access is approximately 4.8 and for 10,000 points it is approximately 6.2. The reason for the improved performance of the TIP tree is because each "search" begins at the last node visited by the previous co-ordinates in the trajectory. If several subsequent co-ordinates are close to each other, then it is very likely that the same node will be visited repeatedly, and no further search to find the correct node is required. Because the original mqr-tree performs all searches from the root, this will result in a higher number of nodes being accessed.

8.3 Density of Points in a Region

For the second test, we evaluated the effect of density on the average number of disk accesses required to locate information. Here, we used a set of 10,000 points as in the first test. However, all points are locations from in and close to the Waikato Region, which is a subsection of the North Island of New Zealand. Therefore, we are looking at the effect of increasing density on the number of disk accesses required to locate information.

We perform the same test runs for the second test as performed for the first test. Figure 8 depicts the outcome of the second test. Again, we find that on average the TIP tree requires only one disk access per co-ordinate on the trajectory. However, we also find that, although the TIP tree has a lower average disk access over the mqr-tree, the improvement is not as significant as that found in the first test since the average number of disk accesses required by the mqr-tree is between 2 and 2.25.

This is a surprising result, since we had assumed that the higher the density of points, the more disk accesses that would be required. What is happening is the following. The mqrtree and TIP tree achieve a significantly lower amount of coverage of space when a dense set of points are being indexed. This means that more points are being placed higher in the index, and fewer rectangles that represent coverage of space are being stored. This results in points (and therefore, information) being located much faster by the mqr-tree than in an mqr-tree that is indexing points that are more spread out geographically.

8.4 #Co-ordinates on Traversal Path

The last two tests we performed evaluated how the number of co-ordinates on a user's trajectory affects the average number of disk accesses. Will too many or too few co-ordinates cause more disk accesses than necessary?

For the first of two tests, we used the set of 10,000 points that represents the North Island to construct the TIP tree and mqr-tree. For the second, we used the denser set of 10,000 points that represents the Waikato Region. For both tests, we performed ten test runs, using approximately 10%, 20%, 30% ..... up to 100% of the co-ordinates in the Hamilton to Cambridge trajectory.

Figure 9 presents the results of the third test, while Figure 10 presents the results of the fourth test. We find that the TIP tree achieves a lower average number of disk accesses for each co-ordinate over the mqr-tree. However, in both cases, we also find that the average number of disk accesses is constant, regardless of the number of points on the trajectory that are used for locating information. This is a very significant result, especially with respect to efficiency of the TIP tree – fewer points does not result in an increase in the number of disk accesses, and therefore traversal remains efficient.

8.5 Further Discussion

Over all tests, we found that the TIP tree and its traversal algorithm performed better than the mqr-tree with respect to the average number of disk accesses per co-ordinate. We performed the worst-case evaluation here – specifically, that every time a node is checked, a disk access is required. We finish this section with related discussion on other issues that are related to this evaluation.

We also recorded the CPU execution time for each co-ordinate "search". We found that the average execution time was less than 1ms. This time may be slightly higher if network transmission time was considered. However, given that minimal mobile web pages with a few links would be sent back to the user in the worst case, we feel that in many cases that network transmission time will be negligible too. Therefore, we feel that disk access time is the crucial factor in how well this system performs. However, further testing should be done to confirm this.

We also did not perform a best-case empirical evaluation at this time for a number of reasons. First, we found that the worst-case evaluation produced efficient results. Also, we found that in many cases, the same nodes were being accessed repeatedly. For the TIP tree, this would be one node. For the mqr-tree, this would be the same path of nodes to get to the one node that several co-ordinates will require. If the nodes were cached in memory, this would result in an average disk access of almost zero. Checking the cache may increase the execution time, depending on which cache management strategy is utilized. However, if caching takes place on the mobile device instead of on the server, this would eliminate most of the network transmission cost, the disk access cost and the execution time on the server. If the TIP tree is used for managing access to external resources, then most of the time only one node would be to be stored by the mobile device. If the mqr-tree is used, then cache size and cache management would be an important factor in how well the index performs in the "best case". Caching is future work that we will be looking into.

9 Conclusion and Remaining Issues

In this paper, we present a framework for accessing both internal information and resources from external digital libraries from within a mobile information system. Using the TIP framework, we show how resources can be organized using the TIP tree so that uniform access to both TIP information and external information from collections managed by different digital library software packages can be achieved. We also present an algorithm for traversing the TIP tree

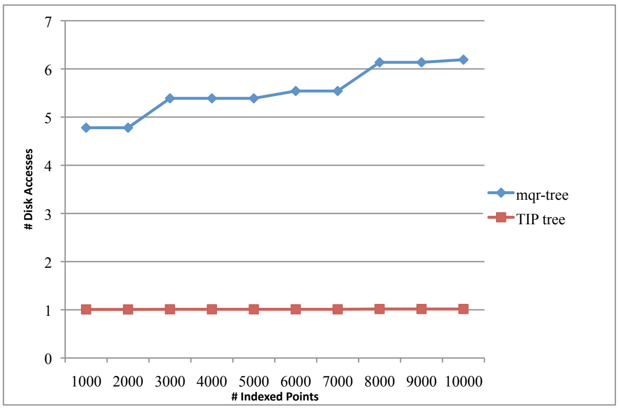


Figure 7: Overall Index Size

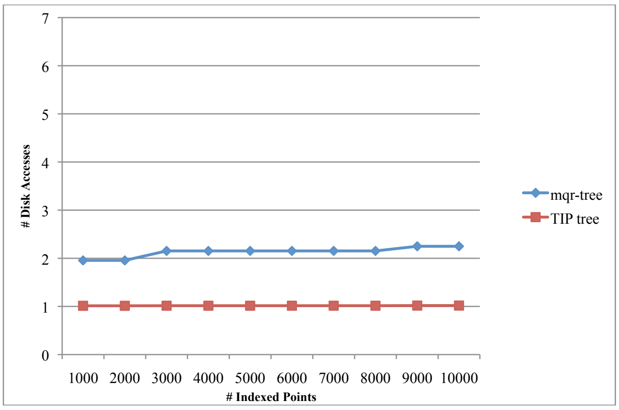


Figure 8: Density of Points in a Region

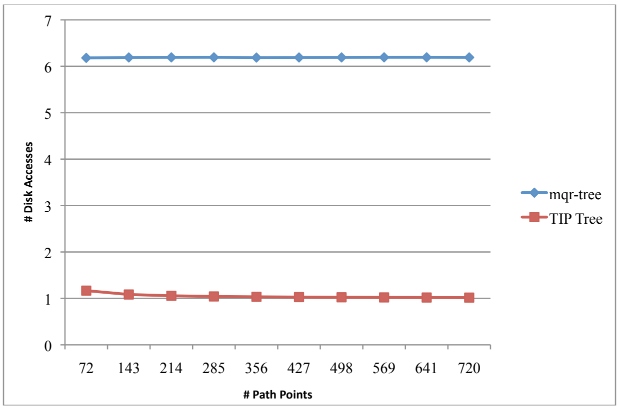


Figure 9: Size of Traversal Path

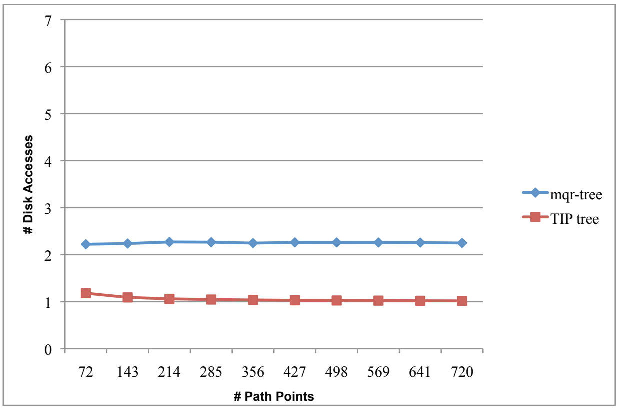


Figure 10: Size of Traversal Path (density)

to continually provide information to the user, so that repeated searching from the beginning is not necessary. An empirical evaluation shows the efficiency of the TIP tree and the traversal algorithm for handling a user's trajectory as they fetch information from digital libraries based on their location.

Other contributions of this work include: 1) one-pass access to any digital library collection, 2) no requirement of a "geographically aware digital library collection, and 3) the ability to handle collections that reference multiple locations. In conclusion, our work provides a strategy for context-aware mobile systems to co-operate with digital libraries in a seamless and efficient manner.

From our current work, there are some outstanding issues that need to be addressed. First, navigation only works for co-ordinates. Two search additions that we want to include are:

- A region search where all information within a specified radius or rectangular region is fetched. An issue with the existing co-ordinate based search is the "oscillation" between two different result sets if the user crosses continuously over a node or quadrant boundary. For example, if the user crosses back and forth over the boundary that separates Auckland from Hamilton, the information that is presented to the user will also switch back and forth. Incorporating a region search will solve this problem by presenting either both sets of information to the user, or a subset of it. In either case, the information that is presented to the user in this situation. will not be in constant change. Although the TIP tree supports region searching (which is inherited from the mqr-tree), it currently does not support the navigation of a search region. This needs to be addressed.

- incorporating searching on other forms of context, such as user interests. This could easily be incorporated by performing an addition search on the results of the location context search. However, it is more desirable to include other types of context into the indexing structure so that only one search is required.

Second, the ability to have more general to more specific searching (i.e. searching higher up versus lower down in the tree) is desired. Both of these require modifications to the current TIP tree structure so that all leaf nodes are on one level of the tree and all data resides in the leaf level.

Third, the use of node caching and the best-case scenario need to be explored and evaluated. As mentioned in the evaluation, several nodes appear to be fetched repeatedly, which if cached would lead to savings in disk access, CPU and communication costs. However, because a mobile devices has less storage than a server, and the same device is also running other programs and services, we cannot assume that all nodes that are fetched can be cached on the device. Therefore cache management strategies for nodes needs to be explored.

These and other issues that may arise will be addressed in future work.

References

Abowd, C., Atkeson, C., Hong, J., Long, S., Kooper, R. & Pinkerton, M. (1997), 'Cyber-guide: a mobile context-aware tour guide', *ACM Wireless Networks* **3**, 421–433.

Cheverst, K., Mitchell, K. & Davies, N. (2002), 'The role of adaptive hypermedia in a context-aware tourist guide', *Communications of the ACM* **45**(5), 47–51.

Gaede, V. & Günther, O. (1998), 'Multidimensional access methods', *ACM Computing Surveys* **30**, 170–231.

Hinze, A., Gao, X. & Bainbridge, D. (2006), 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.

Hinze, A., Voisard, A. & Buchanan, G. (2009), 'Tip: Personalizing information delivery in a tourist information system', *J. of IT & Tourism* **11**(3), 247–264.

Klante, P., Krshe, J. & Boll, S. (2004), Accessignts - a multimodal location-aware mobile information system, *in* 'Proceedings of the 9th International Conference on Computers Helping People with Special Needs (ICCHP 2004)', Paris, France.

Lagoze, C., Payette, S., Shin, E. & Wilper, C. (2006), 'Fedora: an architecture for complex objects and their relationships', *International Journal on Digital Libraries* **6**(2), 124–138.

Moreau, M., Osborn, W. & Anderson, B. (2009), The mqr-tree: improving upon a 2-dimensional spatial access method, *in* 'Proceedings of the 4th IEEE International Conference on Digital Information Management'.

New Zealand Digital Library Project (2008), 'Greenstone digital library software'. Website. Last Visited June 2010. http://www.greenstone.org.

O'Hare, G. & O'Grady, M. (2003), 'Gulliver's genie: A multi-agent system for ubiquitous and intelligent content delivery', *Computer Communications* **26**(11), 1177–1187.

Pashtan, A., Blatter, R. & Heusser, A. (2003), CATIS: A context-aware tourist information system, *in* 'Proceedings of the 4th International Workshop on Mobile Computing', Rostock, Germany.

Poslad, S., Laamanen, H., Malaka, R., Nick, A., Buckle, P. & Zipf, A. (2001), Crumpet: Creation of user-friendly mobile services personalized for tourism, *in* 'Proceedings of the IEE 3G2001 Mobile Communication Technologies Conference', London, UK.

Shekhar, S. & Chawla, S. (2003), *Spatial Databases: A Tour*, Prentice Hall.

Tansley, R., Bass, M. & Smith, M. (2006), Dspace as an open archival information system: Status and future directions, *in* 'Proc. of the 10th European Conference on Digital Libraries'.

Witten, I. H., Bainbridge, D. & Nichols, D. (2009), *How to Build a Digital Library*, Elsevier Science Inc.