Spatio-Temporal Database Research at the University of Melbourne

Egemen Tanin*
Department of Computer Science and Software Engineering
University of Melbourne

Rui Zhang
Department of Computer Science and Software Engineering
University of Melbourne

Lars Kulik
Department of Computer Science and Software Engineering
University of Melbourne

ABSTRACT

The spatio-temporal database research group at the University of Melbourne focuses on introducing new techniques for distributed systems such as mobile and ubiquitous systems as well as P2P networks. Our approach is to exploit the spatial and temporal nature of data and queries such as motion characteristics. In this article, we discuss four major themes and projects of the group: (i) nearest neighbor queries, (ii) temporal data processing and continuous queries, (iii) P2P spatial data management, and (iv) location privacy. The group is supported by funding from the Australian Research Council and the National ICT Australia Victoria Research Laboratory.

1. INTRODUCTION

The Department of Computer Science and Software Engineering at the University of Melbourne has a long history of information retrieval and data mining research. As a young research group at the Department, we are focusing our efforts on spatio-temporal databases to contribute to this long history in data management. Our group consists of three members of the faculty, two postdoctoral research fellows, two software engineers, and ten students.

The current focus of the group is on distributed systems such as mobile, ubiquitous, and P2P systems. Location and spatio-temporal context information have become a prominent part of data that form today's databases. Locational information is commonly collected in a distributed manner. Access to this information is also mostly available over distributed systems. As a well-established example, mobile phones are creating vast amounts of spatio-temporal information in the form of geographical references.

We tap into the spatial as well as temporal properties of data and queries to develop new techniques in data management. Spatial data should not be viewed as just another type of multidimensional data. Each dimension in a spatial data set shares the same unit, and distances between entries on one dimension rarely makes sense without other dimensions. When concatenated with time, correlations become even stronger. For example, two cars moving towards each other on a road network commonly leave a data trail that is hard to encounter in other multidimensional data sets. This view leads us to investigate databases from a different angle. We exploit spatio-temporal characteristics of data and queries, e.g., our techniques use characteristics such as the speed and direction of moving objects for performance improvements.

In the following subsections of this article, we discuss four major themes and projects from our group: (i) Nearest Neighbor Queries, (ii) Temporal Data Processing and Continuous Queries, (iii) P2P Spatial Data Management, and (iv) Location Privacy.

2. NEAREST NEIGHBOR QUERIES

The k nearest neighbor (kNN) query has consistently attracted interest from the database community over a long period of time. Recently, the proliferation of online location-services and equipment has raised strong interest in variants of kNN queries.

In traditional NN query processing, both the query object(s) and the data objects are static. In recent years, variants with moving objects and various query constraints have been taken into account. For example, the query object may move and can trigger a kNN query to be continuously processed as its location changes. In another kNN query variant, we may need to consider obstacles, and only objects that are visible to the query object are of interest. In our group, we have investigated two kNN query variants: moving kNN queries and visible kNN queries.

2.1 Moving K Nearest Neighbor Queries

The moving k *nearest neighbor* (MkNN) query finds the k nearest neighbors of a moving query point continuously. This query is useful when a user is traveling with a GPS equipped

*Contact Information:
Department of Computer Science and Software Engineering
University of Melbourne, Victoria 3010, Australia
Tel: +61 3 8344 1350
Fax: +61 3 9348 1184
Email: egemen@csse.unimelb.edu.au

device and looking for some points of interest in the vicinity, e.g., the five nearest restaurants.

Techniques based on the concept of a *safe region* have been quite successful in processing MkNN queries. In a safe-region-based technique, an answer is returned with a region. As long as the query point stays in this region, the answer remains the same. When the query point moves out of the region, another answer with an associated region is returned. Therefore, a safe-region-based method always (that is, continuously) provides accurate answers without the need for frequent sampling.

A classic example of safe-region-based techniques is the *Voronoi Diagram*. This technique divides the space into regions called *Voronoi cells* where each cell corresponds to a set of points where the nearest neighbor does not change (or the set of k nearest neighbors remains the same for high-order diagrams). Then finding the kNN set is basically equivalent to identifying the Voronoi cell the query object is in. However, Voronoi Diagrams have some significant drawbacks such as expensive precomputation, no support for dynamically changing k values, and inefficient update operations.

We proposed a technique called the *V*-Diagram* [10]. The V*-Diagram can be seen as a local or incremental method that requires no precomputation. It can incrementally compute answers and efficiently adapt to changes – such as insertions and deletions of objects. It can also work with dynamically changing values of k.

The key novelty of the V*-Diagram is to compute a safe region based on not only the data objects, but also the query point and the current knowledge of the search space. This is different from previous safe-region-based techniques that compute safe regions based on the data objects only. The experimental results show that the V*-Diagram regularly outperforms the best existing technique by two orders of magnitude. A thorough analysis on the performance of the V*-Diagram and related techniques is provided in [9]. A detailed analysis of the spatial-network adaptation of the V*-Diagram technique is also available from this article.

2.2 Visible K Nearest Neighbor Queries

Visibility has long been an area of interest in computer graphics. Researchers were interested in efficient ways to render large scenes with many objects. We have introduced visible kNN queries from a spatial databases point of view. A visible k nearest neighbor (VkNN) query is only about k objects with the smallest visible distances to a query object and the rest of the scene or rendering are not relevant. Basically, this query type is most useful when visibility is necessary in finding nearest neighbors.

For example, a tourist can be interested in locations where views such as a building or mountain are available. In an interactive online game, a player commonly needs a map that shows enemy locations that are in line of sight from her position. We introduced the VkNN query in [8] and then proposed algorithms that can incrementally retrieve visible nearest neighbors. Our work focuses on I/O and is optimal in terms of index node accesses for commonly used indices.

In [7], we provided more detailed analysis on the query and studied a more general version of it, aggregate VkNN (AVkNN) queries. An AV1NN finds a data object that minimizes an aggregate distance (e.g., sum distance) to a set of query objects. An example application is finding a site (data object) to install an antenna to provide network access to a number of other sites (query objects), and the distance between the query and the data objects need to be minimized to provide good service. The AVkNN query is an extension of the AV1NN query where we are interested in multiple data objects with the smallest aggregate distances to a set of query objects. We provide efficient algorithms for incrementally finding aggregate visible nearest neighbors [7].

3. TEMPORAL DATA PROCESSING AND CONTINUOUS QUERIES

Time is becoming an increasingly important feature in many spatial and non-spatial databases. We study time from two perspectives: (i) methods that deal with querying temporal attributes, and (ii) continuous queries.

3.1 Temporal Data Processing

The finance industry is an important source of temporal data. The stock market generates huge volumes of data such as stock prices, stock orders, and trading transactions on a daily basis. These records arrive at a high rate as time series. Prompt detection of stock price changes is a task of high priority. Directly observing stock prices usually leads to delayed reports of changes. We have proposed an alternative way of detecting stock price changes, i.e., through the detection of *distribution change* in the number of stock orders [5]. It is based on the well-established findings in financial research that private information (e.g., a company is going bankrupt) available to a small group of traders causes abnormal trading behavior and changes the distribution of the number of stock orders preceding the stock price change. We presented in [5] a technique that can detect the distribution change of stock orders more promptly and accurately than existing techniques.

Indexing and retrieving records according to their temporal attributes are basic functionalities for managing temporal data. While there are many temporal indices proposed, it is shown in [6] that how the TSB-tree, a well-known temporal index, is implemented in a commercial database and retains a performance close to a non-temporal one, the B$^+$-tree. This involves: (i) unique designs of version chaining and treating index terms as versioned records to achieve the TSB-tree implementation with backward compatibility with B+trees, (ii) a data compression scheme that substantially reduces the storage needed for preserving historical data, and (iii) dealing with technical issues such as concurrency control, recovery, handling uncommitted data, and log management.

3.2 Continuous Queries

Beyond the realm of continuous NN queries, many applications utilize query types that need to provide continuous answers to users. We studied some of these query types in our group.

Intersection join is an expensive operation even when data objects are static. It is more expensive when the objects

are moving. We introduced the concept of *time constrained query processing* in [16] to shorten the time range when the query needs to be processed, which reduces the workload significantly. We also proposed a suit of techniques to improve the join algorithm. We presented an algorithm that outperforms the current practice by several orders of magnitude and makes it possible to provide continuous join answers on large datasets in almost real time.

Another important query type on spatial objects is the window query, which returns all objects that fall into a given spatial range. In augmented reality applications, virtual 3D objects are added to the view of a user (through a head-mounted display or a mobile device) according to the current position and viewing direction of the user. This can be seen as a continuous window query on 3D objects. The data retrieval in this setting is an overwhelming problem due to limited wireless bandwidth, especially when the view changes at a high speed and hence causes a large number of 3D objects to be retrieved.

We provided a systematic solution [2] to this problem based on the key insight that the user is only interested in and capable of absorbing high level information in the view when the view is moving at a high speed. We use wavelets to represent 3D objects in multiple resolutions and only retrieve the necessary information to display the required resolution. We also propose a buffer management scheme with motion prediction. In addition, we introduce an efficient index to handle 3D objects taking into account the movement of the view. Our system improves the performance significantly over a system that uses existing techniques and enables a smooth visualization of the 3D objects as the view moves.

4. P2P SPATIAL DATA MANAGEMENT

One of the main research projects that we have been pursuing for the last few years is the P2P virtual worlds project. With this work, we have introduced one of the first spatial indexing mechanisms for P2P networks [11].

Finding a music file given a filename was the main form of use for the early P2P systems and was fundamentally performed in two ways. One way is using a pseudo-decentralized system where the data is stored in the network over many devices and the index regarding who stores which files is available from a dedicated server(s). Second, both the index and the data are available only in decentralized form. The second approach gained popularity over time and made many database researchers excited about research problems in P2P data management in the early 2000s.

Indexing data without having a server or query processing without a center were interesting visions from a data management point of view. Distributed data management approaches of that day did not offer true decentralization acceptable by the P2P community or could not scale to the levels that are required by a P2P system.

In collaboration with the University of Maryland at College Park, we have started making spatial data available on P2P systems in 2003. We have observed that although the P2P paradigm promised a bright future, its application domain was quite restricted, and indexing and query processing were limited to a few types of data. We have taken upon the challenge of making applications such as P2P versions of eBay and online virtual worlds possible. These required dealing with more complex queries and data types.

In particular, distributed hashing is accepted as the main form of structured P2P data indexing and search. IP addresses and available files in a network of PCs can both be hashed onto a virtual address space. If each PC in a given P2P network uses the same virtual address space and hash function, then we can use this function for finding files in that P2P system. It is shown that with each machine storing only O($log\ n$) entries in its routing table, it is easily possible to find a file given a filename in O($log\ n$) hops for n machines with a high probability. However, one cannot trivially perform range queries with such a scheme. Hashing-based schemes cannot be easily used for even the simplest spatial query, i.e., the spatial range query.

We have used the fact that recursive space partitioning in a quadtree creates a set of buckets with unique centroids, i.e., each quadrant of a quadtree has a unique center point among all the other quadrants, independent of the fact that some quadrants are small and some can be quite large. The centroid coordinates can be used with a hash function. Thus, spatial objects, which do not have names like filenames, can be associated with buckets which can later be hashed onto a P2P network of computers. Regardless of the quadtree type used, we can create a mapping between buckets of space and a virtual name space. Spatial range queries can then follow this distributed quadtree index to locate the objects of interest, without a given name but by only using the range query specification itself. Details of this work is now available from [12]. In this paper, we have practiced with an experimental P2P real-estate system where spatial range queries can access a P2P spatial index to locate houses available for sale in a region of a city. We have also introduced a caching mechanism to improve the behavior of our index, reaching realistic query processing times for our experimental application domain.

We focused our implementation efforts on *P2P virtual worlds* with a 3D version of our index [13] and also patented a multi-rooted version of it. This project is now being developed by National ICT Australia (NICTA) Victoria Research Laboratory (VRL) for P2P massively multiplayer online gaming. The main idea is to let user avatars travel in a virtual world using a P2P index, thus jumping from one player's PC to another player's PC seamlessly without using a central server.

Our P2P virtual world vision basically divides the virtual environment that avatars occupy among the machines available in the P2P network of players. Each machine is responsible for maintaining a part of the world. This, however, creates a problem when traveling between different parts of the world. One has to locate which machine to contact to without using a centralized indexing mechanism for travel. This is where our P2P spatial index comes into play.

Attached to this project, we have also investigated load balancing strategies for P2P systems. Unlike many distributed environments, load balancing in a P2P system cannot resort to one dedicated server and it has to scale easily. Thus,

peer-to-peer load trading is the main method that we used on this front. We have shown that using spatial characteristics of a problem at hand, e.g., motion characteristics of moving objects, we can achieve a good level of load balance in a P2P system [1]. For example, given a traffic pattern of moving objects, it is desirable not to divide the load of maintaining a fleet of moving objects among peers, as this will only increase message passing between peers, slowing down the system.

5. LOCATION PRIVACY

Location-based services (LBSs) enable the access of information based on an individual's location. They comprise a large range of applications such as emergency services, mobile e-commerce, care for the elderly, navigation services, or traffic monitoring. Current location-based services are able to continually sense an individual's location and provide updated information services based on that location. If an individual misses a turn while following navigation instructions, a *location-based service provider* (LSP) that is continually monitoring a user's location can immediately respond and recalculate a modified set of instructions. LBSs have a tremendous potential that will be amplified with new technologies such as Google Maps' ability to locate a mobile phone without using GPS.

More generally, there are two types of queries for LBSs: (a) snapshot queries that are based on the individual's position send as a single request for information to an LSP without requiring any further updates; (b) continuous queries that instantaneously update the supplied location-based information based on the individual's current location. Examples for snapshot queries are queries for the closest points of interest (such as shoe shops) that could be highlighted on a map, whereas a continuous query would include real-time navigation instructions as well.

Due to their high degree of convenience, LBSs are likely to become a central part of our daily life. However, they also have privacy risks: an LSP that tracks the movement of all of its users with a high spatial and temporal fidelity, is able to generate a complete history of each user's movement including the time and type of accessed service. An individual's location, however, is personal and sensitive information. For example, an individual might want to restrict others from knowing or inferring certain illnesses. *Location privacy* studies how to safeguard a person's privacy. The protection of an individual's location is a key prerequisite for the wide acceptance of real-time LBSs.

Our research on privacy in snapshot queries led to the development of a new approach called *obfuscation* that protects a person's location privacy by degrading the quality of information about the person's location [3]. Using obfuscation, people can reveal varying degrees of information about their location (for example, suburb, block, street, or precise coordinate-level information). Higher levels of obfuscation lead to greater location privacy, but could lead to a service with decreased quality. Our approach provides a computationally efficient mechanism for successfully balancing the need for high-quality information services against an individual's need for location privacy. We use negotiation to ensure that a location-based service provider receives only the location information it requires to provide a service of satisfactory quality. This means in particular that an LSP does not receive precise coordinates.

Access to location information inherently requires some level of trust. Most approaches adopted a central architecture, which uses a location anonymizer that removes identifying information (such as the person's location or ID in the form of a telephone number). It is well known that central architectures have some disadvantages, in particular security threats if information is stored in a single place. Our research [4] has proposed a decentralized approach to distribute the trust among all peers in a decentralized network. We exploit the capability of mobile devices to form wireless ad-hoc networks in order to hide a user's identity and position. These local ad-hoc networks enable us to separate an individual's request for location information, the query initiator, from the individual that actually requests this service on its behalf, the query requestor. A query initiator can select itself or one of the $k-1$ peers in its ad-hoc network as a query requestor. As a result the query initiator remains k-anonymous, even to the mobile phone operator.

To facilitate the next generation traffic monitoring systems that provide real-time information about traffic and road conditions current systems aim to collect significant amounts of real-time information about individuals. This even includes individuals who do not require any service. Correspondingly, privacy concerns might increase if data collection and tracking of individuals intensify. In our approach to balance the need for real-time data and the concerns of individuals about their privacy, we also proposed to collect *aggregated data* instead of individual data. To estimate the current traffic flow in a road network, it is not necessary to track the movement of each individual driver. It would suffice to record the number and speed of cars at dedicated observation points and track the saturation flow rate at intersections (the maximum number of vehicles passing through an intersection during an hour if the signal is always green).

We show in our work [15] that simple count (aggregated) information stored in a spatial data structure can be used to answer a surprisingly large range of queries. We store counts stored in a spatial data structure that is called the Distributed Euler Histogram (DEH) to achieve this. The DEH can answer not only queries about the total traffic in an area but can be used for other queries, for example how many unique cars entered an area, which could be used to estimate available parking spaces. We have extended this work and proposed a Privacy Aware Monitoring System (PAMS) for traffic monitoring applications [14] that solves a larger range of aggregate queries without the need of true identities. This system is based on an extension of the DEH: the *Euler Histogram based on Short ID (EHSID)*, which allows us to answer even more queries while safeguarding a motorist's privacy. The use of periodically changing short IDs enables us to recognize a road user without actually identifying her.

6. FUTURE

We plan to focus our future efforts on three projects. First, we plan to investigate new moving nearest neighbor query types in metric spaces. Second, we will invest in future

paradigms for location privacy protection. Third, we will work on active environments using RFID technology in ubiquitous systems.

We see current moving nearest neighbor search as a precursor to future short-trip planning queries. Unlike the well established area of offline trip planning, these queries can be addressed online but are hard to answer using existing nearest neighbor queries. For example, a user may want to visit a friend's house but is planning to pick a pizza on the way to the house. In this scenario, nearest neighbors of today's techniques are not the first choices of interest for the user as they may take the user away from the ultimate destination. In addition, we are also interested in cases where the location of spatial objects are not well defined or are imprecise. For such cases, absolute results make little sense for ranking neighbors.

In location privacy, we plan to extend our work in two ways. Protecting location privacy for continuous queries will be the first challenge we will address. Simply hiding the exact position using an imprecise location such as a region cannot ensure privacy for continuous queries: continuous disclosure of regions enables an adversary to follow an individual's movement path. Even an individual who anonymously accesses a service can be identified once the current location refers to an identifiable place such as an office or a home address. Second, the current work on statistical data analysis, known as negative information theory, is an another important area that we will investigate. In negative information surveys, individuals select a category to which they (or a phenomenon) do(es) *not* belong. If the number of categories is large, then this technique can avoid the disclosure of sensitive or private data. This technique allows us to obtain precise aggregate but not individual information about observed phenomena.

Finally, we see RFID technology as a major component in future ubiquitous systems. We envision active environments where massive deployments of RFID tags and readers are used to compute and reveal different types of information to the users. Thus, we may witness a dramatic expansion of localization techniques using RFID technology. Current techniques cannot supply high granularity information and thus are limited in their use. However, we see future systems where small items such as books can be tagged and tracked remotely using passive RFID tags. We plan to investigate future uses of this technology for spatio-temporal data management.

7. ACKNOWLEDGMENTS

We would like to acknowledge the efforts of our students as well as our co-authors. In particular, we thank our students Sarana Nutanong, Muhammad Umer, Tanzima Hashem, Mohammed Eunus Ali, Hairuo Xie, Dana Zhang, Martin Stradling, Parvin Asadzadeh-Birjandi, Mei Ma, Pu Zhou, Elizabeth Antoine, and our research fellows Dr Jie Shao, Dr Xiaoyan Liu. We also thank our co-authors Professor Hanan Samet, Professor Elisa Bertino, Dr David Lomet, Professor Ben Shneiderman, Dr Aaron Harwood, Dr Dan Lin, and Dr Matt Duckham. We would like to acknowledge colleagues at the Department, in particular, Professor Rao Kotagiri for his mentoring and Associate Professor Chris Leckie for his valueable comments. Finally, we thank agencies Australian Research Council, A. E. Rowden White Foundation, and NICTA VRL for funding multiple grants and projects.

8. REFERENCES

[1] Mohammed Eunus Ali, Egemen Tanin, Rui Zhang, and Lars Kulik. Load balancing for moving object management in a P2P network. In *Proc. of DASFAA*, pages 251–266, 2008.

[2] Mohammed Eunus Ali, Rui Zhang, Egemen Tanin, and Lars Kulik. A motion-aware approach to continuous retrieval of 3D objects. In *Proc. of ICDE*, pages 843–852, 2008.

[3] Matt Duckham and Lars Kulik. A formal model of obfuscation and negotiation for location privacy. In *Proc. of Pervasive*, pages 152–170, 2005.

[4] Tanzima Hashem and Lars Kulik. Safeguarding location privacy in wireless ad-hoc networks. In *Proc. of Ubicomp*, pages 372–390, 2007.

[5] Xiaoyan Liu, Xindong Wu, Huaiqing Wang, Rui Zhang, James Bailey, and Kotagiri Ramamohanarao. Mining distribution change in stock order streams. In *Prof. of ICDE*, 2010.

[6] David B. Lomet, Mingsheng Hong, Rimma V. Nehme, and Rui Zhang. Transaction time indexing with version compression. *Proc. of VLDB*, 1(1):870–881, 2008.

[7] Sarana Nutanong, Egemen Tanin, and Rui Zhang. Incremental evaluation of visible nearest neighbor queries. *To appear in IEEE Trans. Know. Data Eng.*

[8] Sarana Nutanong, Egemen Tanin, and Rui Zhang. Visible nearest neighbor queries. In *Proc. of DASFAA*, pages 876–883, 2007.

[9] Sarana Nutanong, Rui Zhang, Egemen Tanin, and Lars Kulik. Analysis and evaluation of V*-kNN: An efficient algorithm for moving kNN queries. *To appear in VLDB Journal.*

[10] Sarana Nutanong, Rui Zhang, Egemen Tanin, and Lars Kulik. The V*-Diagram: A query-dependent approach to moving kNN queries. *Proc. of VLDB*, 1(1):1095–1106, 2008.

[11] Egemen Tanin, Aaron Harwood, and Hanan Samet. A distributed quadtree index for peer-to-peer settings. In *Proc. of ICDE*, pages 254–255, 2005.

[12] Egemen Tanin, Aaron Harwood, and Hanan Samet. Using a distributed quadtree index in peer-to-peer networks. *VLDB Journal*, 16(2):165–178, 2007.

[13] Egemen Tanin, Aaron Harwood, Hanan Samet, Deepa Nayar, and Sarana Nutanong. Building and querying a P2P virtual world. *GeoInformatica*, 10(1):91–116, 2006.

[14] Hairuo Xie, Lars Kulik, and Egemen Tanin. Privacy aware traffic monitoring. *To appear in IEEE Trans. Intel. Transp. Sys.*

[15] Hairuo Xie, Egemen Tanin, and Lars Kulik. Distributed histograms for processing aggregate data from moving objects. In *Proc. of MDM*, pages 152–157, 2007.

[16] Rui Zhang, Dan Lin, Kotagiri Ramamohanarao, and Elisa Bertino. Continuous intersection joins over moving objects. In *Proc. of ICDE*, pages 863–872, 2008.