ANNATTO: Adaptive Nearest Neighbor Queries in Travel Time Networks

Wei-Shinn Ku, Roger Zimmermann, Haojun Wang and Trung Nguyen
Computer Science Department
University of Southern California
Los Angeles, California 90089
[wku, rzimmerm, haojunwa, tnnguyen]@usc.edu

Abstract

Nearest neighbor (NN) searches represent an important class of queries in geographic information systems (GIS). Most nearest neighbor algorithms rely on static distance information to compute NN queries (e.g., Euclidean distance or spatial network distance). However, the final goal of a user when performing an NN search is often to travel to one of the search results. Based on this observation, finding the nearest neighbors in terms of travel time is more realistic than the actual distance. In the existing NN algorithms dynamic real-time events (e.g., traffic congestions, detours, etc.) are usually not considered and hence the pre-computed nearest neighbor objects may not accurately reflect the shortest travel time. In this demonstration we present ANNATTO, a novel adaptive nearest neighbor query model for travel time networks which integrates both spatial networks and real-time traffic event information. The ANNATTO system includes the implementation of a global-based adaptive nearest neighbor algorithm and a local-based greedy nearest neighbor algorithm that both utilize real-time traffic information to provide adaptive nearest neighbor search results.

1 Introduction

Nearest neighbor (NN) queries are of significant interest for applications that work with spatial data. A sample query could be to "find the nearest Japanese restaurant from my current location." Previous work [5, 2] has resulted in efficient techniques to compute NN queries in Euclidean space. More recently, novel algorithms [6, 4] have been proposed to compute NN queries in spatial networks. These methods extend NN queries by considering the spatial network distance, which provides a more realistic measure for applications where object movements are constrained by underlay networks. However, these existing techniques only consider static models of spatial networks: pre-defined road segments with fixed road conditions (e.g., speed limits) are used in computing nearest neighbors. Thus, any real-time events (e.g., detours, traffic congestions, etc.) affecting the spatial network cannot be reflected in the query result. For example, a traffic jam occurring on the route to the computed nearest neighbor most likely elongates the total driving time. More drastically, the closure of a restaurant which was found as the nearest neighbor might even invalidate a query result. This motivates the need for new algorithms which extend existing NN query techniques by integrating real time event information.

Recent advances in personal locater systems (e.g., GPS), wireless communication technologies (e.g., IEEE 802.11x), and peer-to-peer networks (P2P) have created an innovative environment that allows the exchange of real time traffic information between peers. By leveraging ad-hoc networks, traffic information can be shared in a P2P manner among mobile hosts (MH) and thus local traffic information (e.g., driving speed of vehicles) can be considered when computing NN queries. Furthermore, cellular communication enables remote *traffic information server* (TIS) access such that collecting and disseminating traffic information for a much wider area becomes possible.

In this demonstration we present ANNATTO (Adaptive Nearest Neighbor queries in trAvel Time neTwOrks), a prototype of the system in [3] which implements an adaptive nearest neighbor query model in conjunction with a travel time network. In particular, the ANNATTO system exhibits the following distinguishing characteristics:

- **Adaptive nearest neighbor query execution.** The ANNATTO system demonstrates a novel adaptive nearest neighbor query algorithm which computes the nearest neighbor in a best-first manner with global traffic information.
- **Excellent scalability.** The ANNATTO system also performs a local-based greedy nearest neighbor query algorithm which leverages P2P data sharing to achieve scalability in terms of the number of peers. A higher

density of peers improves its efficiency.

- **Realistic movement on road network.** The movement of mobile hosts in the ANNATTO system is constrained to real world road networks. Mobile hosts in ANNATTO autonomously proceed on road networks and the velocity of the movement is determined by the real-time speed of underlying road segments.

- **Playback of real traffic events.** The traffic events occurring in the ANNATTO system are provided by the city of Los Angeles Department of Transportation (LADOT)[1]. The system regenerates these real traffic events for verifying the efficiency of the two proposed algorithms.

2 The ANNATTO Infrastructure

Figure 1 illustrates the system infrastructure of ANNATTO. We are considering mobile hosts with abundant power capacity, such as vehicles, that are equipped with a Global Positioning System (GPS) for obtaining continuous position information. Furthermore, we assume that two tiers of wireless connections are available on each mobile host. The cellular-based networks (such as utilized by the OnStar service) allow medium range connections to base-stations that interface with the wired Internet infrastructure. A second type of short-range ad hoc communication protocols (e.g., IEEE 802.11x) are also supported to communicate between neighboring peers. In addition, mobile hosts also maintain the road network data and the set of *points of interest* (POI) in local memory. The road network data (e.g., the US Census TIGER data set) covers the road segments of highways, primary roads, rural roads, etc. These different road types are defined as road class attributes in the TIGER data set.

Mobile hosts can either broadcast requests of traffic information to peers within the communication range (local solution) or send requests to the traffic information server (TIS) directly (global solution). Currently there are many real-time traffic event providers (e.g., California Highway Patrol[2], SIGALERT.com real-time traffic information[3], etc.) which supply traffic information of many urban areas. These web sites can be easily integrated with TIS servers through Web service interfaces in the future. A local travel time network in each mobile host is thus built by integrating the information of traffic events from peers or a TIS and the local stored road network for processing nearest neighbor queries.

[1]http://www.lacity.org/LADOT/
[2]http://cad.chp.ca.gov/
[3]http://www.sigalert.com/

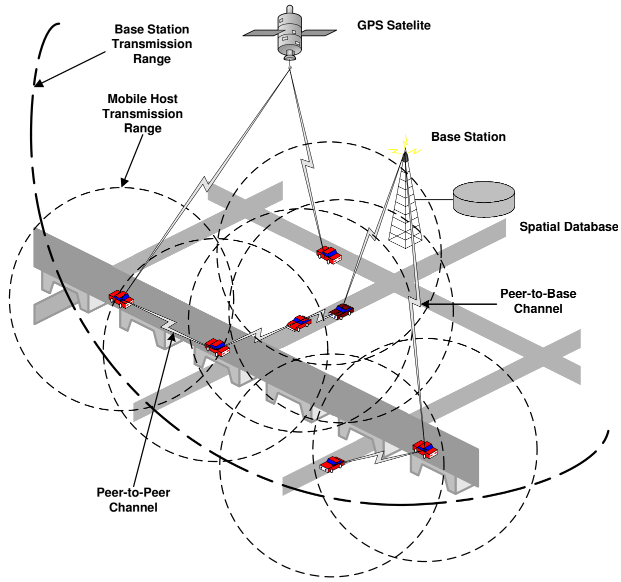

Figure 1. The ANNATTO infrastructure.

3 The ANNATTO Components and Algorithms

In this section we describe the modular implementation of the ANNATTO system and the simulation parameter set such as the points of interest and mobile host density.

3.1 System Modules

The ANNATTO system consists of four main components: (1) the *adaptive NN query simulation module*, (2) the *multiple peer simulation module*, (3) the *traffic information server module*, and (4) the *visualization module*. Adaptive NN queries are demonstrated by navigating mobile hosts to their destinations on the road network.

- The **adaptive NN query simulation module** navigates user selected vehicles with either of our two novel NN query algorithms to their destinations.
- The **multiple peer simulation module** concurrently models a predefined number of mobile hosts. It implements all the functionality of a single mobile host and provides the communication facilities among peers.
- The **traffic information server module** is responsible for updating mobile hosts with real-time traffic events which are provided by transportation agencies.
- The **visualization module** provides a rendering of the navigation process of our two novel nearest neighbor query algorithms and a traditional spatial network NN query algorithm for comparison purpose. Users can arbitrarily select a mobile host and launch an adaptive NN query within the simulation region.

3.2 Traffic Events and Simulation Parameters Collection

The City of Los Angeles Department of Transportation operates an Automated Traffic Surveillance and Control (ATSAC) system[4] for alleviating traffic congestion. The ATSAC system is a computer-based traffic signal control system which monitors real-time traffic conditions of the City of Los Angeles. The main functionalities of the system are selecting appropriate signal control strategies and performing equipment diagnostics with alert functions. LADOT deployed numerous sensors in streets for detecting the passage of vehicles, vehicle speed, and the level of congestion (Figure 2). The sensor collected information is received on a second-by-second basis and is analyzed on a minute-by-minute basis at the ATSAC Operations Center. To date, ATSAC has been implemented at 3,100 of 4,300 City of Los Angeles signalized intersections.

The traffic event data (e.g., vehicle speed, congestion, etc.) of our system is collected by the ATSAC system and these events can be replayed on road segments within the simulation region. Mobile hosts can detect the traffic events of road segments which they are travelling on.

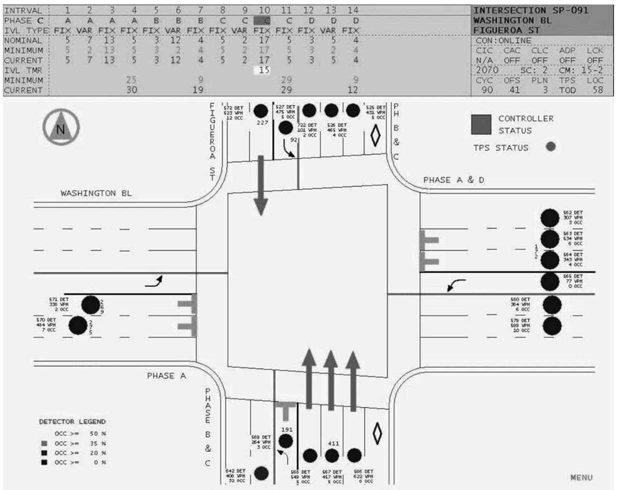

Figure 2. Real time traffic data collected by LADOT sensors.

ANNATTO models the density of POIs (currently gas stations and restaurants) in the Greater Los Angeles area via data available from two online sites: GasPriceWatch.com[5] and CNN/Money. ANNATTO also imports vehicle statistics of the Greater Los Angeles area from the Federal Statistics web site to initialize the mobile host density. However, users are not limited to utilizing these density presets. Many parameters – the density of interest objects and mobile peers among them – can be changed via the multiple peer simulation module.

3.3 Road Network Generation

We generated the underlying road network from the TIGER/LINE street vector data set available from the U.S. Census Bureau. The current ANNATTO system stores the road network of several Southern California counties. The road segments are differentiated into several road classes, such as freeways, primary highways, secondary and connecting roads, and rural roads. Road segments of different road classes are associated with different driving speed limits. One of the challenges when integrating road segments into a complete road network is to isolate paths that cross and determine if they are indeed intersections. For example, freeways generally project many intersections in two-dimensional space, however, many of them are overpasses or bridges. Our solution is to detect intersection points with the help of their endpoint coordinates. In addition, differing road classes let us distinguish overpasses from intersections.

3.4 Travel Time Networks

A travel time network (TTN) utilizes the travel time between nodes as the graph edge weight, rather than the network distance as in spatial networks. With travel time networks, real-time events can be readily integrated into affected road segments by converting their effect to time. For example, if the driving speed of a road segment is notably slower than its speed limit because of a traffic congestion, the travel time between its start and end points can be dynamically updated to reflect the congestion. In case of a road segment closure for a traffic accident, the road segment can be temporarily removed from its TTN. Since turn restrictions (e.g., right turn prohibition) can be modeled by including extra nodes in the spatial network, we do not consider them as real-time events in our research.

3.5 Local-based Greedy Nearest Neighbor Queries

Since the cellular-based communication is much more expensive than the short-range ad hoc communication, traffic information can be exchanged in a local area with a much lower cost. In addition the travel time network enables the integration of real time traffic information into a spatial representation. Based on this observation, we propose a Local-based Adaptive Nearest Neighbor query algorithm (LANN). LANN relies on exchanging local traffic information between mobile hosts for updating the TTN of each peer. With LANN, a mobile host m first computes

[4] http://trafficinfo.lacity.org/
[5] http://www.gaspricewatch.com

the nearest POI as the destination d on the travel time network. Next, m incrementally updates local traffic information and selects a road segment correspondingly as the shortest path to d. As the mobile host begins to navigate on a road segment, it broadcasts requests to collect local traffic information from peers within the ad hoc communication range. Only the traffic information of the surrounding road segments is requested. In the case that the traffic information of some road segments cannot be collected from peers, the default speed limits of the corresponding road segments are used. A local travel time network, which evaluates the weight of each surrounding road segment as the sum of the actual travel cost and a heuristic travel cost, is hence built up. The weight calculation utilizes the travel time of immediate surrounding road segments as the actual cost a and the heuristic cost h is computed as the Euclidean distance from the end of each immediate surrounding road segment to d divided by a heuristic travel speed (e.g., the average travel speed on the TTN). m selects the road segment with the lowest weight as the shortest path to d and starts to navigate on that road segment.

LANN is executed in an incremental manner: when m reaches the end of the selected road segment, it broadcasts again to collect local traffic information from peers and update its local TTN. Next, m selects a road segment to continue navigation based on the updated TTN. The mobile host keeps executing the algorithm until it reaches d.

3.6 The Traffic Information Server

In order to support the Global-based Adaptive Nearest Neighbor Queries, a traffic information server has to maintain valid traffic events for mobile hosts to access. Traffic events can be broadly classified into four categories:

- Category 1 - *Congestion Events*: The real traffic speed of a road segment is notably lower than the speed limit. This condition is usually caused by traffic accidents or traffic jams.
- Category 2 - *Detour Events*: A road segment is closed and the mobile host has to detour.
- Category 3 - *Closure Events*: The selected POI is closed and the mobile host has to search for another nearest POI.
- Category 4 - *Recovery Events*: A mobile host can recover its local TTN from previous events: a traffic congestion has been relieved, a detour has been removed, or a POI is reopened.

Since mobile hosts send requests to the TIS for acquiring new traffic events, they can simultaneously upload the speed of their current road segments and report any real-time traffic events. Consequently the TIS aggregates traffic congestions, accidents, and other real-time traffic events. In addition, transportation and law enforcement agencies can report road construction and accident information to the TIS. Commercial businesses (e.g., gas stations) can also report closure events to the TIS. Every traffic event is time-stamped and a mobile host can synchronize traffic information with the TIS by checking the latest time-stamp in its local memory.

3.7 Global-based Adaptive Nearest Neighbor Queries

We propose a Global-based Adaptive Nearest Neighbor query algorithm that computes the nearest neighbor in a best-first manner with global traffic information. At the start of a trip a mobile host m executes the GANN algorithm to compute a nearest POI as the destination d and the shortest path to d as the selected route S_{route}. Afterwards the mobile host follows S_{route} for traveling to d before updating traffic events with the server. When m receives new traffic information T_{info} from TIS, it needs to determine if T_{info} has any influence (e.g., traffic jams usually slow down the traffic on road segments) upon S_{route}. If T_{info} has no influence on the current route of m, the mobile host only needs to integrate T_{info} into its local TTN for future usage. However, if T_{info} is related to the current journey of m, the mobile host has to execute more methods. As discussed in Section 3.6, there are four traffic event categories. With category 1 and 2, mobile hosts have to update their local TTN (remove the edge of the closed road segment for category 2) and recalculate the Dr_{time} from the current location to d. Then it launches a travel time network NN query with Dr_{time} as the upper search bound S_{bound}. Afterwards, GANN chooses the shortest driving time POI within S_{bound} as the new destination and navigates the mobile host there, if any new NN has been found. In category 3, a selected POI (d) can be closed unexpectedly after a mobile host starts its trip. When receiving a POI closure event which is the current destination, a mobile host has to launch a travel time network NN query for finding a new nearest POI. With category 4, if the recovery is about a traffic congestion and related with the route to the current d, the mobile host only needs to update its local TTN. Otherwise, the mobile client launches a travel time network NN query with the current driving time Dr_{time} as the search upper bound S_{bound} on the updated TTN.

4 Demonstration

ANNATTO is implemented in Java and Figure 3 shows its visualization interface. The left window frame shows the simulated service region of ANNATTO visualizing all mobile hosts and POIs. Mobile hosts move on the road network autonomously [1] while observing the real-time speed of each underlay road segments. On the right pane, the simulator displays the configuration parameters of current simulation, such as the service region dimensions, and the number of mobile hosts. Users are able to select (via mouse click) any mobile host to launch an adaptive NN query with our two novel algorithms. ANNATTO visualizes the execution of these two algorithms with a traditional spatial network NN query as comparison.

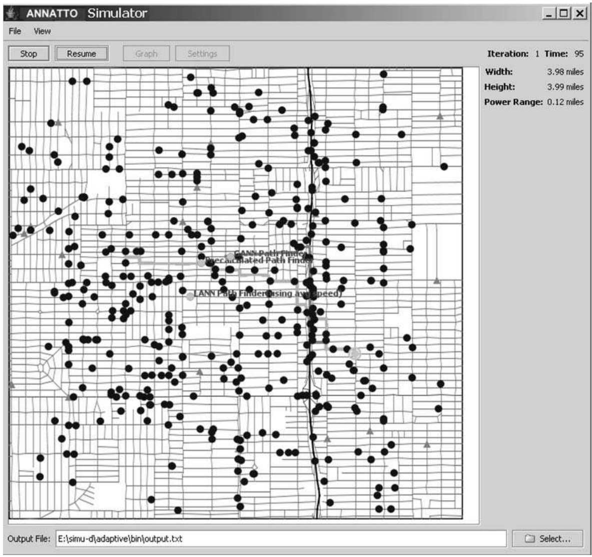

Figure 3. The ANNATTO system visualization interface.

Figure 4 illustrates the configuration interface of ANNATTO. Users are able to define the system performance characteristics through parameters, such as accident event count, peer count, and POI count. ANNATTO is also able to load recorded system parameters such as the traffic event data collected by the ATSAC system. Our demonstration has two main objectives:

- Demonstrating and clarifying our novel adaptive nearest neighbor query algorithms.
- Providing hands-on experience for the audience.

5 Conclusions

We have described ANNATTO, a system to aid in the study of adaptive nearest neighbor queries in travel time networks. We implemented two adaptive NN query algorithms to for comparing with static NN solutions. We also implemented a road network with realistic traffic events to constrain the movement of peers in the mobile environment. The objective of ANNATTO is to provide a platform for the evaluation of our ongoing research of travel time networks and related query algorithms. ANNATTO demonstrates the excellent scalability and effectiveness of our current algorithms in high density mobile environments.

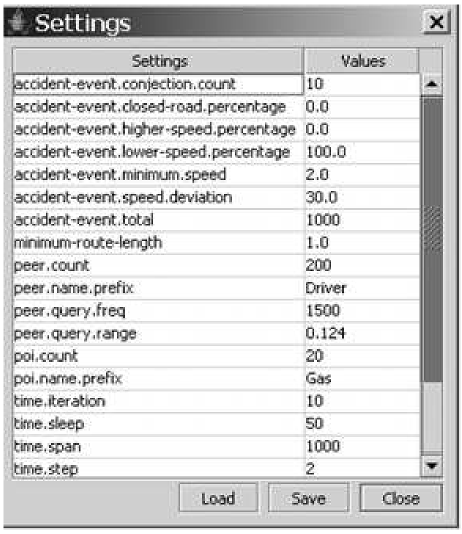

Figure 4. The ANNATTO system parameter setting panel.

6 Acknowledgments

This research has been funded in part by NSF grants EEC-9529152 (IMSC ERC), CMS-0219463 (ITR), and equipment gifts from the Intel Corporation, Hewlett-Packard, Sun Microsystems and Raptor Networks Technology.

References

[1] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. In *Proceedings of the 4th ACM/IEEE MobiCom*, pages 85–97, 1998.

[2] G. R. Hjaltason and H. Samet. Distance browsing in spatial databases. *ACM Trans. Database Syst.*, 24(2):265–318, 1999.

[3] W.-S. Ku, R. Zimmermann, H. Wang, and C.-N. Wan. Adaptive Nearest Neighbor Queries in Travel Time Networks. In *Proceedings the 13th ACM International Symposium on Geographic Information Systems.*, pages 176–183, 2005.

[4] D. Papadias, J. Zhang, N. Mamoulis, and Y. Tao. Query Processing in Spatial Network Databases. In *Proceedings of the International Conference on Very Large Databases*, pages 790–801, 2003.

[5] N. Roussopoulos, S. Kelley, and F. Vincent. Nearest Neighbor Queries. In *Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data*, pages 71–79, San Jose, CA, May 22-25, 1995.

[6] C. Shahabi, M. R. Kolahdouzan, and M. Sharifzadeh. A Road Network Embedding Technique for k-Nearest Neighbor Search in Moving Object Databases. In *Proceedings of the Tenth ACM International Symposium on Advances in Geographic Information Systems*, pages 94–100, McLean, Virginia, November 2002.