TransDec: A Data-Driven Framework for Decision-Making in Transportation Systems

Ugur Demiryurek, Farnoush Banaei-Kashani, and Cyrus Shahabi
University of Southern California
Department of Computer Science
Los Angeles, CA 90089-0781
[demiryur, banaeika, shahabi]@usc.edu

ABSTRACT

In this paper, we present an end-to-end data-driven system, dubbed TransDec (short for Transportation Decision-Making), to enable decision-making queries in transportation systems with dynamic, real-time and historical data. With TransDec, we particularly address the challenges in visualization, monitoring, querying and analysis of dynamic and large-scale spatiotemporal transportation data. TransDec fuses a variety of transportation related real-world spatiotemporal datasets including massive traffic sensor data, trajectory data, transportation network data, and points-of-interest data to create an immersive and realistic virtual model of a transportation system. Atop such a virtual system, TransDec allows for processing a wide range of customized spatiotemporal queries efficiently and interactively. TransDec has a three-tier architecture, including a multimodal spatiotemporal database, an expressive and efficient query-engine and an interactive map-mashup presentation tier. With this paper, first we describe the components of the TransDec architecture. Subsequently, as proof-of-concept we present a number of decision-making queries enabled by TransDec.

INTRODUCTION

Real-time data-driven framework of the transportation systems (i.e., a system driven by real data collected from the field) enables development of *data-driven* Intelligent Transportation Systems (ITS) for realistic and effective decision-making, planning, and management of the transportation systems. A data-driven ITS system must be able to handle various data types such as automatic vehicle location data (AVL), traffic congestion information, traffic incidents reports, road construction notices, driver information, CCTV video streams and snapshots, etc. Considering the large size of the transportation data, variety of the data (different modalities and resolutions), and frequent changes of the data, the integration, visualization, querying and analysis of such data for large-scale real-time systems are intrinsically challenging *data management* tasks. Due to these challenges, most of the current ITS applications only support limited data monitoring and analysis capabilities. Moreover, transportation researchers and policy developers, who mostly focus on developing algorithms and policies to enhance the efficacy of the transportation systems, often are restricted to simulation test-beds driven by synthetic or simplified data, and/or expensive and time-consuming field surveys to analyze the transportation systems and evaluate the proposed solutions.

In this paper, we present a data-driven framework, dubbed TransDec, which enables real-time visualization, monitoring, querying, and analysis of dynamic transportation systems. We build TransDec on top of a three-tier architecture (presentation tier, query-interface tier, and data tier) that allows users to create customized spatiotemporal queries through an interactive web-based map interface in support of decision-making. With this architecture, we particularly address the fundamental data management and visualization challenges in 1) effective handling of dynamic and large-scale transportation data, and 2) efficient processing of real-time and historical spatiotemporal queries on transportation networks. To evaluate TransDec, we utilize a rich set of real transportation data which we have obtained from RIITS[1] **(**Regional Integration of Intelligent Transportation Systems). The RIITS dataset is collected by various organizations based in Los Angeles County including Caltrans D7, MTA-Metro, LADOT, and CHP. This dataset includes both inventory and real-time data (with update rate as high as every 1 minute) for freeway and arterial congestion, bus location, events, and CCTV snapshots. Moreover, in order to support diverse ITS applications, the TransDec data tier contains the transportation network of the entire US, as well as a wide variety of points-of-interest data provided by Navteq [2].

In addition to offering realistic and immersive virtualization of the traffic information and moving assets (e.g., busses, trains) through a web-based application, TransDec allows decision makers to issue various real-time and historical spatiotemporal queries about a) traffic at specific segments or sensor stations (with any user-defined level of aggregation at any desired timeframe), and b) moving assets and their navigational (e.g., speed, time-to-destination), route, and location-based information. We categorize these spatiotemporal queries into three groups: monitoring queries, analysis and mining queries, and planning queries (such as route planning queries) and location-based queries (which include nearest neighbor, range and trajectory-based queries).

The remainder of this paper is organized as follows. First, we will review the related work about data-driven monitoring and simulation systems for transportation systems. Next, we will describe the three-tier architecture and components of TransDec in detail. Later, we will elaborate on the real-time and historical spatiotemporal query types supported by TransDec. We will mention the underlying challenges of each query type and our solutions in each case. Finally, we will conclude the paper with conclusion and future work.

RELATED WORK

The latest developments in wireless technologies as well as the widespread usage of sensors have led to the recent prevalence of transportation network monitoring and simulation systems. The main functionality of such systems, in general, is to collect and archive data from distributed sensors and analyze the archived data for transportation applications such as planning and mobility measurement. Among the numerous transportation network monitoring and simulation systems that have been developed in

---

[1] http://www.riits.net/

[2] http://www.navteq.com/

the last decade, we review two of the most relevant systems, namely PeMS [Pems08] and ADMS [Adms08].

The freeway Performance Evaluation Monitoring System (PeMS) developed by UC Berkeley collects and stores data from loop detectors operated by Caltrans. The main goal of PeMS is to convert this freeway sensor data into intuitive tables and graphs that show traffic patterns on highways. Moreover, PeMS is used to spot bottlenecks, measure the efficiency of highways, and estimate travel times for highway segments based on the historical travel time data.

Similarly, ADMS developed by Smart Travel Lab in University of Virginia is a system which utilizes archived ITS data to provide information services to measure the performance and operation of Virginia transportation systems. ADMS, in addition to monitoring highways, enables users to query the database for real-time (and historical) weather and incident information for specific routes and segments.

Our work is fundamentally different from the aforementioned systems in several ways. First, while the scope of these systems is limited to collection, archival and analysis of the sensor data, with TransDec we fuse the sensor data with various spatiotemporal data (e.g., transportation network data, moving object data, point of interest data) to be able to support end-to-end ITS applications more realistically. Second, PeMS and ADMS, for analysis of huge datasets, use traditional Database Management System (DBMS) specific analytical query processing tools. DBMSs can only support a set of predefined analytical queries. However, with TransDec we utilize more sophisticated query processing techniques that allow for ad hoc and complex analytical queries as well. Finally, in addition to real-time traffic monitoring and analysis, TransDec enables various other real ITS applications such as real-time moving asset tracking, route planning and location-based querying.

TransDec ARCHITECTURE

Similar to many state-of-the-art database driven applications, TransDec adopts a *three-tier architecture* [Eckerson95] where presentation, query-interface, and data management tiers are logically separated. With the three-tier architecture of TransDec, the query is initialized at the presentation tier interactively and sent to the query-interface tier where each request is formulated as structured query language (SQL) before interacting with the data tier. Apart from the usual advantages of modular system with well defined interfaces, the three-tier architecture is intended to allow any of the three tiers to be upgraded or replaced independently as requirements or technology change. Figure 1 shows the three-tier architecture of TransDec. Below we elaborate on each tier in detail and describe the interaction between them.

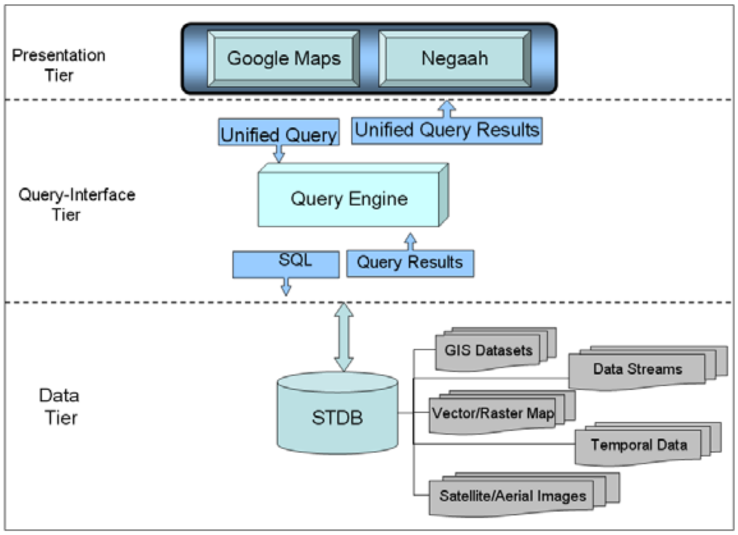


Figure 1. The TransDec architecture

Presentation Tier

One of the key distinguishing features of TransDec is the provision of an immersive environment that enables users to interact with the system and perform a wide range of ITS related spatiotemporal queries intuitively. The current graphical user interface (GUI) provides users with query flexibility by allowing them to query the spatiotemporal datasets based on a user-defined area and time interval. Specifically, users can selectively query and display different layers of information on desired regions, and move forward or backward in time for various query types. To implement the presentation tier, we have integrated a new generation *web-based map application*, Google Map™, into TransDec as graphical user interface. As a second choice, we have also developed our proprietary interface (termed *Negaah*) that provides custom spatiotemporal queries not supported by typical web-based mapping applications.

The advantages of utilizing Google Map™ at the presentation tier are as follows: a) appealing and highly responsive visualization options, b) ability to zoom in and out to the desired scale and to pan to specific locations, c) flexible environment for as many layers of information as necessary to provide the level of detail required, and d) easy access through Internet via any browser (without any software or hardware installation). Additionally, given that Google Map™ is already ported to PDAs, porting TransDec to mobile computing devices is not a far-fetched future work. To support querying capability on mobile computing devices, we have a head start by studying multi-resolution vector data compression techniques [Khoshgozaran06] that effectively compresses the result of query windows, taking into account the client's display resolution.

Query-Interface Tier

With the query-interface tier, TransDec allows several independently developed GUIs to interact with our spatiotemporal data tier transparently. Our query-interface tier offers a universal standard for specifying the type of query (e.g., shortest path, range aggregate, etc.) and its parameters, as well as the returned results. Specifically, depending on the query type received from the GUIs, the query-interface tier constructs and sends the query to our data tier through low-level structured query language (SQL) commands for evaluation. In most cases, the query results are formulated into an industry standard XML[3] file before sending to presentation tier.

In addition, this tier includes a data fusion engine which continuously acquires sensor readings from RIITS web-services and stores them into our data repository. Data fusion engine utilizes WSDL[4] (provided by RIITS) in combination with SOAP[5] to retrieve the dynamic sensor data through a secure communication channel over the Internet.

Data Tier

The data repository of TransDec is a spatiotemporal database management system (STDBMS), namely Oracle 10g. STDBMS stores a variety of both dynamic (frequently changing) and static datasets such as highway sensor data, road network information (i.e., vector data), moving object trajectory data, point of interest data (e.g., hospitals, restaurants), terrain data, satellite and aerial imagery, and raster maps. Most of the data stored in the repository are labeled by both space and time to allow for a wide range of spatiotemporal (both time and space based) queries.

In the data tier, our main focus is on developing index structures to answer spatiotemporal queries efficiently. Index structures are pre-computed data structures that are built off-line to speed up the evaluation of the queries issued on-line. Towards this end, in addition to incorporating numerous off-the-shelf indexing methods (e.g., R-Tree), we have introduced various versions of Voronoi Diagrams (VD) [Kolahdouzan04, Sharifzadeh06] as index structures for enabling efficient spatiotemporal querying on different types of data hosted in TransDec. The major data components of TransDec are as follows:

Sensor Data: TransDec, through RIITS, acquires traffic sensor data from approximately 1500 sensors located on 18 highways at the boundaries of Los Angeles County. The arrival rate of the data from each sensor is 1 reading/sensor/min. The storage space required for this streamed dataset is approximately 350 MB/day (120 GB/year).

Each sensor is a separate data source that generates records with several attributes including sensor id, measure (value of reading), space (coordinates), and time (timestamp). The traffic measures captured by the sensors are speed, HOV speed, volume (number of vehicles) and occupancy (ratio of time vehicle is detected). Based on their spatial placement, we group the sensors as follows: a) individual sensors, b) spatially

---

[3] http://www.w3.org/XML/

[4] http://www.w3.org/TR/wsdl

[5] http://www.w3.org/TR/soap/

continuous sensors (segments), and c) spatially continuous segments (sections). Also, we consider the following time intervals to interact with the sensor data: a) a time point/range within a day, and b) day of week. These hierarchical categorizations with both space and time enables TransDec to process all roll-up and drill-down queries on traffic measures efficiently by navigating between various hierarchies of the time and space dimensions. For example, when computing the average speed at a network segment within five past Mondays from 08:00am to 08:30am, the weekend traffic data and unrelated segments can be discarded easily.

Transportation Network Data: TransDec contains a wide range of transportation network data (provided by Navteq ™) including highways, major and secondary roads, streets, railroads, bridges, etc., for the entire US. Each network segment is represented in the vector data format and described by more than 20 attributes such as direction, speed limit, zip code, paved, etc. We utilize transportation network dataset primarily when processing route planning and location-based queries.

Trajectory Data: Another spatiotemporal dataset of TransDec consists of the trajectory information collected from moving objects whose location in the space changes over time. The position of a moving object is sampled at discrete times, and a series of straight lines connecting successive positions represents the trajectory of the object. Currently, TransDec collects live location data from GPS equipped USC trams and student cell phones (GeoSim, 2008). All time variant vehicle and cell phone coordinates are stored and archived in the data tier.

Points-of-Interest (POI) Data: Point of interest data stored in TransDec (also provided by Navteq^{TM}) includes list of public locations, such as bus satiations, parking structures, hospitals, restaurants, etc. Each POI data record is represented as spatial point location in the database and includes various attributes such as phone number, street address, type, etc.

SPATIOTEMPORAL QUERY PROCESSING

In this section, we discuss various spatiotemporal queries enabled by TransDec, and describe how we address the query processing and the data management challenges associated with each query type. We categorize the current queries adopted by TransDec into following three types: 1) Monitoring Queries on Streaming Data, 2) Analysis and Mining Queries on Historical Streaming Data, and 3) Planning Queries, which include route planning queries and location-based planning queries.

Monitoring Queries on Streaming Data

With monitoring queries on streaming data, the data is in the form of constantly received data streams rather than finite stored datasets, and the queries are continuous as opposed to one-time queries. For example, consider a query that continuously reports the speed and occupancy information from each highway sensor every ten seconds. TransDec currently includes two types of streaming data collected from a) highway traffic sensors in LA County and b) GPS enabled moving objects (USC trams and student cell phones [GeoSim08]). Considering the size and streaming nature of the data, query processing with streaming data is a challenging task, because the streaming sensor data is unbounded (infinitely received) and streaming queries are continuous and periodic.

To enable monitoring queries on the streaming traffic sensor data, we have materialized the on-the-fly sensor data (received from RIITS) into our data repository by maintaining the real-time (daily) and the historical datasets concurrently. While the real-time dataset is utilized for monitoring queries, the historical dataset is mainly used for analysis and mining queries (discussed in the next section). In order to keep the results up-to-date for continuous monitoring, the result sets are updated incrementally with the arrival of new data tuples rather than querying the entire datasets in defined intervals. To enable monitoring queries on the moving objects, we have stored in the database the real-time position information and the motion vector (which describes the position as a function of time) for each moving object. Similarly, the real-time position information is utilized for monitoring queries and the motion vector data is used for analysis and mining queries.

Figure 2 shows an example of monitoring query on streaming traffic sensor data. Each color-coded (based on the speed measure) icon on the map represents a highway traffic sensor that provides speed measure, location, and last-update timestamp. Once new readings from the sensors arrive in to the system, the measure values and color codes are updated automatically.

Vehicle ID: 701

Start 12/15/2008 End 12/18/2008

I-5 North
SLAUSON

Lanes	Current Speed	Avg Speed
General Lane	24	58

Speeds

- 🟥 < 19 mph
- ▬ 20 - 39 mph
- 🟩 40 - 59 mph
- 🟩 > 60 mph

Figure 2. Real-time traffic monitoring

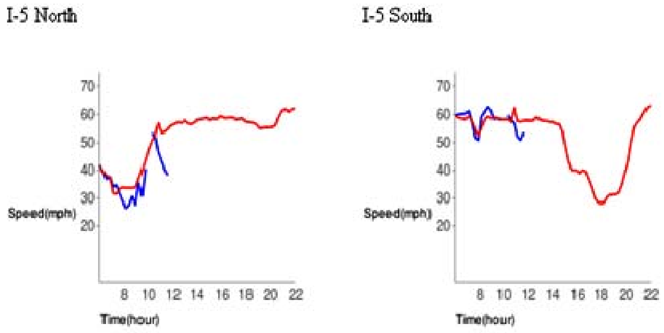
Analysis and Mining Queries on Historical Streaming Data

With TransDec, the goal of analysis and mining queries is to discover useful knowledge and extract patterns from historical data. To exemplify, consider a query that reports the average speed of a segment (e.g., on I-5 from post mile 293 to 300) within last five Mondays between 8:00am to 8:30am.

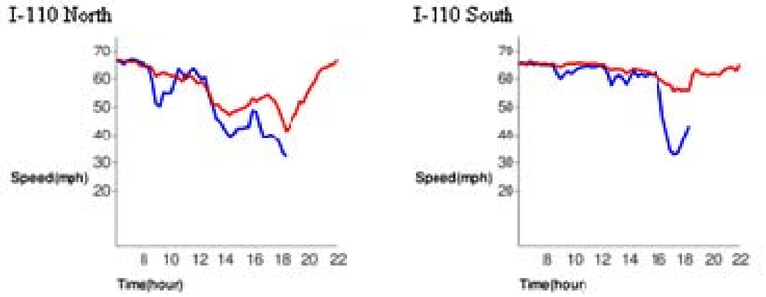
To perform such analysis on the historical datasets, two alternative solutions are traditional database management systems (with standard SQL operations) and online analytical query processing (OLAP) tools. The main shortcomings of these tools with processing OLAP queries are as follows. Traditional database solutions are mainly designed for transactional rather than online analytical query processing. Therefore, standard SQL operations on huge datasets (with multiple levels of aggregation) would suffer from intolerably long response times. Even though various OLAP tools have been developed (some embedded to the commercial database systems) for overcoming this limitation of traditional database solutions, they heavily rely on precalculating the query results to enable online query processing. Consequently, they can support only a limited set of predefined (rather than *ad hoc*) analytical queries online.

To cope with these challenges, we utilize the wavelet based online analytical query processing techniques which we have developed in the past [Shahabi04, Jahangiri07]. Wavelet based techniques store small approximate sketches of data (i.e., wavelet-coefficients), and compute point and range aggregate queries with sufficiently precise approximation based the stored sketches. Specifically, TransDec leverages the wavelet-coefficients of the historical data and the query [Schmidt02a, Schmidt02b] to provide exact, approximate, and progressive answers to OLAP queries.

Figure 3 depicts four speed plots that are drawn based on the results of a range aggregate query (an OLAP query) supported by TransDec. The red plot represents the historical speed measure average (for each 15 minutes between 7:00am to 10:00pm) of a particular highway segment within a particular day, and the blue plot shows the real-time speed reading on the same segment.



a) I-5



b) I-110

Figure 3. Real-time and historical speed plots

Figure 3a depicts the speed pattern of a particular segment on I-5 on a weekday where the travel speed is at its minimum between 7:30am and 9:30am on I-5 North and 4:30pm and 7pm on I-5 South. Similarly, Figure 3b illustrates the speed plot of a specific segment on I-110 on a weekend day where there is big deviation between the daily real-time speed measure and its historical average on I-110 South. Assuming the sensors on this segment functioning properly, this deviation may indicate a potential traffic accident to operators.

Planning Queries

With TransDec, we categorize planning queries into two main areas, namely, route planning queries and location-based planning queries. We describe each category in detail below.

Route Planning Queries: With the route planning problem, one needs to find a shortest path from a starting point to a destination on a transportation network such that the total travel time is minimal. The traditional route planning approaches been studied extensively in the past. With TransDec, we introduce a new class of route planning query namely, *time-dependent route planning* (TDRP).

The major drawback of the traditional route planning approaches is that the recommendation provided to the users at any given time is based on *one-time (fixed)* congestion information for the travel and disregards the highly dynamic changes of the congestion situation during the travel time. However, in practice, segment travel times can change rapidly over the time, especially at the boundaries of the traffic peak periods where optimal planning is indeed most useful to the drivers (since at other times, either during the rush hour or while the road network is not congested, the optimal path usually is fixed and/or has a known pattern). With time-dependent route planning, we consider the future congestion/travel-time information for each segment. As a result, the travel time at each road segment in the network become time-varying, which in turn causes a continuous change in the total travel cost as the time goes by. For example, consider the scenario in Figure 4 where a driver wants to determine the shortest travel path from point A to point B starting his/her journey at 8:00am. As shown, a segment (S) may have multiple travel times (TT) based on the arrival time to starting node of that segment. Assuming that the travel time in S1 is 15 minutes at 8:00am, the driver needs to select one of the segments, S2 or S3, in the upcoming intersection at 08:15am. The dynamic route planner would select S2 or S3 based on the accumulative *time-dependent travel time* information of all segments in all possible paths (i.e.: S1-S2-S5-S7, S1-S2-S4-S6-S7, S1-S3-S6-S7, S1-S3-S4-S5-S7) considering the arrival times to each intersection. Continuing our example from above, after retrieving the travel times of S2 and S3 as 25 and 20 minutes respectively, the algorithm would then consider the future travel times for S4 (two-way segment) at 8:40am and 8:35am when accessed from S2 and S3 respectively. The computation would continue in the same manner until all possible paths are exhausted by reaching point B, and the algorithm returns the upcoming segment in the optimal (minimum total travel time) path.

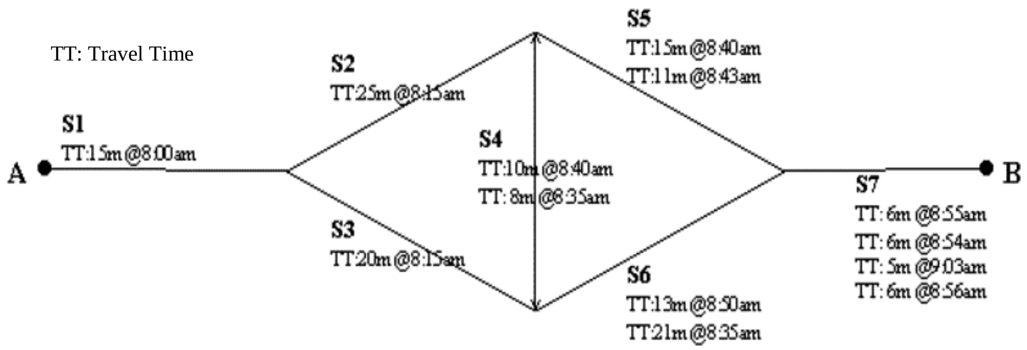


Figure 4. An example of transportation network with time-dependent travel times

Table 1 represents the optimal route selection computation for the scenario depicted in Figure 4. The timestamp (e.g., 8:00am) next to travel cost (e.g., 15m) represents the timestamp when the travel is expected to start for the corresponding segment.

Path	Travel Time Cost (Segment Travel Start Times)
S1-S2-S5-S7	15m (8:00am)+25m (8:15am)+15m (8:40am)+6m (8:55am) = 61min.
S1-S2-S4-S6-S7	15m (8:00am)+25m (8:15am)+10m (8:40am)+13m (8:50am)+5m (9:03am) = 68min.
S1-S3-S6-S7	15m (8:00am)+20m (8:15am)+21m (8:35am)+6m (8:56am) = 62min.
S1-S3-S4-S5-S7	15m (8:00am)+20m (8:15am)+ 8m (8:35am)+11m (8:43am)+6m (8:54am) = 60min.

Table 1. Time-dependent route computation

As presented, the iterative nature of the time-dependent route planning computation makes the problem very hard and computationally expensive in real-life transportation networks as the search space to find the shortest path may grow exponentially and the computations may be redundantly repeated due to the time-dependent nature of the problem. We address these challenges by extending A* [Korf90] algorithm. Specifically, we treat each entry in the priority queue as travel-time function (that we derived from historical traffic data) instead of a single travel-time and determine earliest departing time interval (time refinement) for each node in the graph. Once we obtain such an interval and corresponding function, we utilize them to find the next node in A* algorithm.

Figure 5 illustrates an example of TDRP where blue route is computed based on the time-dependent edge travel times and green route is computed with fixed edge travel times for a user-defined departure time from the start point (green icon) to the destination (red icon). The blue slider on the left menu enables users to move forward or backward in time by allowing them to select various departure times from the start point.

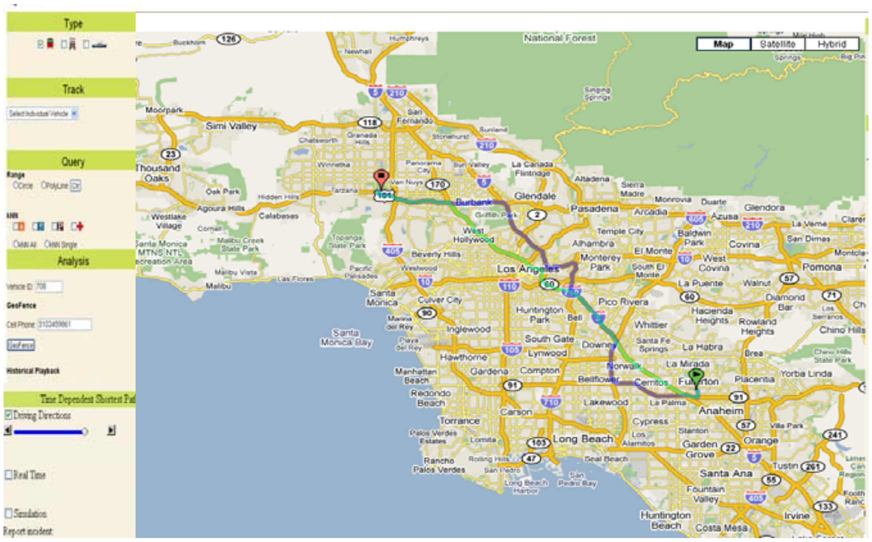
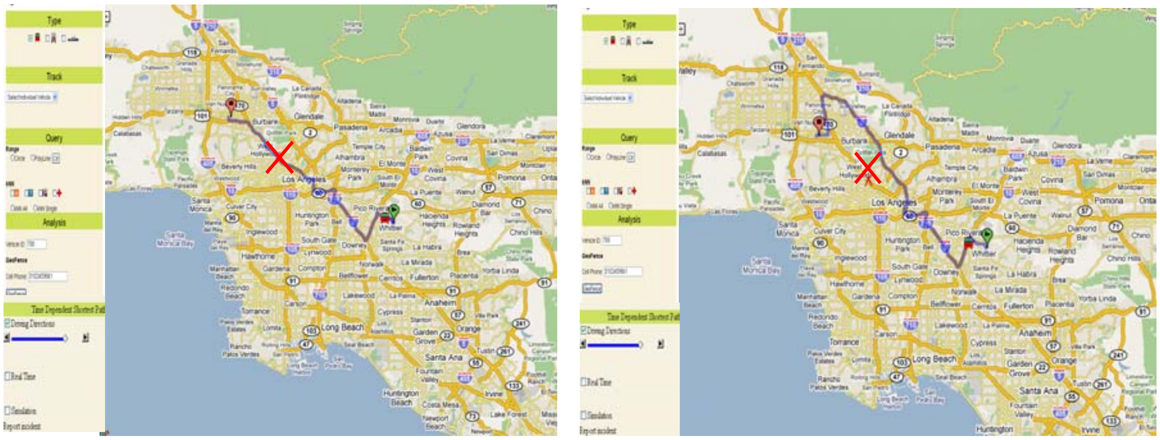


Figure 5. Time-dependent route planning

In addition, the cost for each network segment may change in real-time due to recent incidents such as accident or natural disaster. With TransDec, we take into consideration such real-time changes that make the problem even more challenging. With our approach, we re-adjust TDRP incrementally by utilizing some information form the original computation. Figure 6a shows an example of accident simulation on one of the links (marked with a cross on the map) on the optimal route. Figure 6b illustrates the re-adjusted path which avoids the accident simulated link.



a) Accident simulation b) TDRP recalculation

Figure 6. Accident simulation and TDRP recalculation

To exemplify the utilization of TDRP as an ITS tool, imagine a service which knows the current location of the vehicles (bus, emergency vehicles such as police, fire, ambulance, etc.) in the transportation network and their stations (start and end points) and can dynamically plan the routes between stations based on the predicted and real-time congestion data. Moreover, the TDRP can be used to alert the travelers about the best departure time and the expected travel time for a given start and destination point.

Location-Based Planning Queries: The location-based queries are more traditional spatial queries, including range, and *k* nearest-neighbor queries that look for desired points-of-interest to a referred object or location. To enable location-based queries, TRASIM integrates various real points-of-interest data (provided by Navteq™) such as hospitals, bridges, ports, train stations, etc. The location-based queries adopted by TransDec are as follows.

k-Nearest Neighbor (k-NN) Query: The *k*-NN query is a spatial query that is utilized to find *k* POI that have the nearest network distance to a given query point. With *k*-NN queries, both the POI and the query points could be either static or dynamic (moving), which results into combination of four cases depicted in Table 2.

Query Point	**POI**
Static	Static
Static	Dynamic
Dynamic	Static
Dynamic	Dynamic

Table 2. Query and POI movement types

The main challenges in processing *k*-NN queries are as follows: a) the mobility of query points and/or POI, b) fast road network distance (shortest path) calculation, and c) efficient update of dynamic index structures.

To cope with these challenges, we have implemented a novel technique based on a *filter-and-refine* method [Demiryurek08]. With our approach, we use filtering mechanism to rapidly find a set of candidate POI based on their Euclidean distance from the query object and then refine by computing their network distance. In addition, depending on the movement types, we utilize several other query optimization and efficient indexing techniques which we have developed in the past. These include Voronoi Based *k*-NN Search in Network Databases [Kolahdouzan04**]** and Road Network Embedding Technique for K-Nearest Neighbor Search [Shahabi03].

Figure 7 illustrates an example of continuous k-NN query where the nearest hospital (circled in blue) to a moving vehicle (circled in red). As the vehicle moves along its path, the query result (the nearest hospital) is continuously updated.

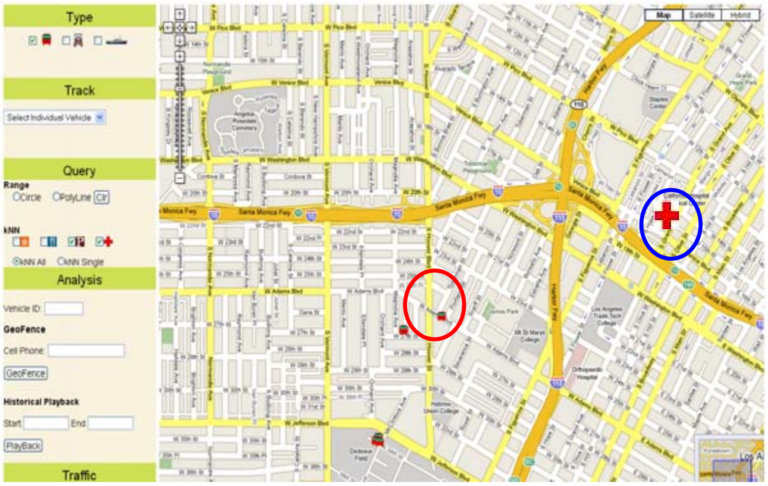


Figure 7. An example of *k*-NN query on transportation networks

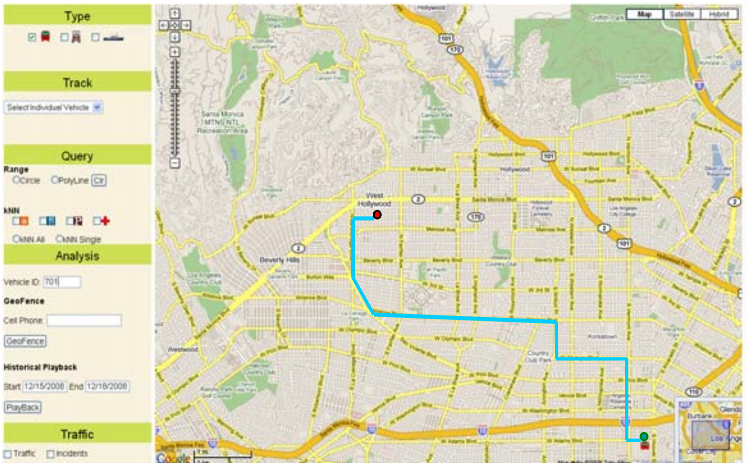
Range Query: A range query is a spatial query which returns all POI inside a user-specified bounding box such as circle or polygon. For example, a user may draw a circle with a center any point on the map interface and ask the system to return buses within one miles radius of that center.

As with other spatial queries, the spatial range query differs from conventional SQL queries as they a) use geometric data types such as lines and polygons (rather than the character and number data types), and b) consider the spatial relationship between these geometric data types. Therefore, implementing a range query requires different data and index structures which support multidimensional selection and join operations by spatial criteria. Toward this end, we utilize a popular spatial data structure namely, R-tree [Guttman84] which hierarchically organizes spatial objects based on their minimum bounding rectangles. Specifically, we create a R-tree index on spatial attributes (coordinates) of POI and utilize this index structure for efficient pruning of the space.

An advanced range query type supported by TransDec is *GeoFence* query. With GeoFence query, users can create a virtual boundary on a geographic area on the map. When that boundary is crossed by a moving object (in either way), the intrusion is recognized as an event and the user is notified by a SMS text or email message. With this type of particular query type, one can formulate more complex queries involving several areas.

Trajectory-Based Query: The trajectory-based queries are built on the current and the past trajectory information (derived from sampled location points at discrete times) of moving objects. The trajectory-based queries are usually combined with *k*-NN and range queries hence inheriting the similar challenges. To exemplify, consider the queries such as “show me the trajectory of vehicle A between 8:30am and 9:30am today”, and “find all the buses inside the boundaries of Los Angeles between 9-12am yesterday”.

Figure 8 depicts an example of trajectory query where the historical trajectory of a vehicle is graphically displayed based on a given time interval (8:30am to 9:30am, December 7, 2008).



Vehicle ID: 701

Start 12/15/2008 End 12/18/2008

Figure 8. An example of historical trajectory query

In addition, TransDec supports user-defined trajectory queries which allow users to specify their queries on a custom drawn trajectory on the map. For example, a user can create a custom trajectory from his/her home to work by interactively drawing it on the map and query the system for the real-time or historical traffic and incident information on the given trajectory.

CONCLUSION AND FUTURE WORK

In this paper, we introduced TransDec, a data-driven end-to-end decision making system that enables interactive and extensive querying of transportation related spatiotemporal datasets including traffic sensor data, trajectory data, transportation network data, and points-of-interest data. Particularly, we explained the design and implementation details of TransDec's three-tier architecture. We also introduced a set of advanced spatiotemporal queries supported by TransDec. We elaborated on the challenges with each query type and presented our solution in each case.

We intend to pursue this work in three directions. First, we plan to extend the capabilities of the query-interface tier to support more complex ad hoc analytical queries. Second, we intend investigate various real ITS applications to extend the set of queries supported by TransDec. We expect for the set of the supported queries to eventually evolve to a complete minimum set that allows for formulation of generic decision-making queries in all typical ITS applications. Finally, we plan to improve the presentation tier of TransDec to port it to mobile computing devices.

ACKNOWLEDGEMENTS

This research has been funded in part by NSF grants IIS-0238560 (PECASE), IIS-0534761,IIS-0742811 and CNS-0831505 (CyberTrust), and in part from the METRANS Transportation Center, under grants from USDOT and Caltrans. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[Adms08] ADMS Smart Travel Lab. http://cts.virginia.edu/stl_adms.htm, Last visited December 21, 2008.

[Dijkstra59] Dijkstra, E. (1959). "A Note on Two Problems in Connection with Graphs", Numerische Mathematik 1.

[Demiryurek08] Demiryurek, U. and et al. (2008). "ER-CkNN: Efficient Continuous KNN Query in Spatial Networks using Euclidian Restriction", Submitted for review.

[Eckerson95] Eckerson, W. (1995). "Three-tier Client/Server Architecture: Achieving Scalability, Performance, and Efficiency in Client Server Applications", OIS.

[Geosim08] GeoSIM: "An Urban Sensing System for Social Image Mapping of Urban Geolocations". http://infolab.usc.edu/projects/GeoSIM/, Last visited Dec. 20, 2008.

[Guttman84] Guttman, A. (1984). "R-Trees: A Dynamic Index Structure for Spatial Searching", SIGMOD.

[Jahangiri05] Jahangiri, M., and et al. (2005). "SHIFT-SPLIT: I/O Efficient Maintenance of Wavelet-Transformed Multidimensional Data", SIGMOD.

[Jahangiri07] Jahangiri, M and Shahabi, C. (2007). "Wolap: Wavelet-Based Range Aggregate Query Processing", Tech. Report, Dept. Computer Science, USC, 2007.

[Khoshgozaran06] Khoshgozaran, A. and et al. (2006). "A Multi-Resolution Compression Scheme for Window Queries Road Network Databases", SSTDM.

[Kolahdouzan04] Kolahdouzan, M. and Shahabi, C. (2004). "Voronoi-Based K Nearest Neighbor Search for Spatial Network Databases", VLDB.

[Korf 90] Korf, R. E. (1990). "Real-time Heuristic Search", Artificial Intelligence.

[Pems08] Freeway Performance Measurement System. https://pems.eecs.berkeley.edu/, Last visited December 21, 2008.

[Riits08] RIITS (Regional Integration of Intelligent Transportation Systems). http://www.riits.net/, Last visited December 25, 2008.

[Schmidt02a] Schmidt, R. and Shahabi, C. (2002). "Propolyne: A Fast Wavelet-Based Technique for Progressive Evaluation of Polynomial Range-Sum Queries", EDBT.

[Schmidt02b] Schmidt, R. and Shahabi, C. (2002) "How to Evaluate Multiple Range-Sum Queries Progressively", PODS.

[Shahabi03] Shahabi, C. and et al. (2003). M. "A Road Network Embedding Technique for K-Nearest Neighbor Search in Moving Object Databases", Geoinformatica.

[Shahabi04] Shahabi, C. and Schmidt, R. (2004). "Wavelet Disk Placement for Efficient Querying of Large Multidimensional Data Sets", Tech. Report, Dept. of Comp. Sc., USC.

[Sharifzadeh06] Sharifzadeh, M. and Shahabi C. (2006) "The Spatial Skyline Queries", VLDB.