Dynamic Spatio-temporal Integration of Traffic Accident Data

Ove Andersen
Aalborg University
Dept. of Computer Science
Aalborg, Denmark
xcalibur@cs.aau.dk

Kristian Torp
Aalborg University
Dept. of Computer Science
Aalborg, Denmark
torp@cs.aau.dk

ABSTRACT

Up to 50% of delay in traffic is due to non-reoccurring events such as traffic accidents. Accidents lead to delays, which can be costly for transport companies. Road authorities are also very interested in warning drivers about accidents, e.g., to reroute them. This paper presents a novel and efficient approach and system for uncovering effects from traffic accidents by dynamic integration of GPS, weather, and traffic-accident data. This integration makes it possible to explore and quantify how accidents affects traffic. Dynamic integration means that data is combined at query time as it becomes available. This is necessary, because data can be missing (weather station down) or late arriving (accident not officially reported by the police yet). Further, the integration can be parameterized by the user, e.g., distance to accident, which is important due to inaccuracy in reporting. We present the integrated data on a map and show the effectiveness of the integration by allowing users to interactively browse all accidents or pick a single accident to study it in very fine-grained details. Using information from 31 433 road accidents and 38 billion GPS records, we show that the proposed dynamic data integration scales so very large data sets.

CCS CONCEPTS

• **Information systems → Information integration**;

KEYWORDS

Data Integration, Spatio-temporal, GPS, Traffic Accidents, Weather

ACM Reference Format:
Ove Andersen and Kristian Torp. 2018. Dynamic Spatio-temporal Integration of Traffic Accident Data. In *26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL '18), November 6–9, 2018, Seattle, WA, USA.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3274895.3274972

1 INTRODUCTION

Traffic accidents are a huge expense for both the individual user and the society. The annual cost of traffic accidents in the USA is estimated 871B USD (2010) [9].

In Denmark, information on traffic accidents contains rudimentary spatio-temporal information such as the time and the place of an accident [1]. However, the weather condition at the time of the accident is typically not reported and there is no information on how an accident affected the traffic.

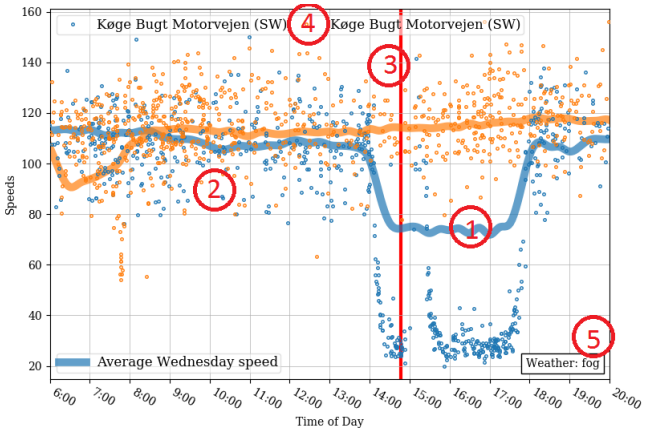

Figure 1: Wednesday Afternoon Accident on a Motorway

Drivers, transport companies, and road authorities are interested in analyzing the impacts of accidents on traffic, e.g., how much traffic is delayed and how many vehicles are impacted? Such questions can be answered when accident data is integrated with GPS data.

Figure 1 shows the impact of an accident on a motorway a Wednesday afternoon at 14:47. The bold lines (item 1) show the average travel speed on Wednesdays (in opposite directions). The blue and orange markers (item 2) show the travel speeds of individual vehicles that passes the accident site on the day of the accident. Both the average speed (item 1) and the individual speeds (item 2) are computed from GPS trajectory data.

From Figure 1 it is clear that the travel speed is significantly below the average travel speed in one direction. While the traffic in the other direction is not affected. The red line (item 3) shows the official reported time of the accident. At item 4 map information is shown and at item 5 the local weather information is listed.

Figure 1 is created by integrating four data sources: (1) data from 31 433 accidents, (2) a country sized road network, (3) 38 billion GPS records, and (4) weather data. A complete back-end and front-end implements all the features described in this paper.

The four data sources are unaligned, i.e., data becomes available with different delays, from a few minutes when an accident is reported to several days when weather information is available. Further, the spatial accuracy varies significantly, e.g., weather information is coarse grained whereas GPS data is fine grained.

[1] Of ethical reasons all other information related to accidents is removed

SIGSPATIAL '18, November 6–9, 2018, Seattle, WA, USA

ACM ISBN 978-1-4503-5889-7/18/11.
https://doi.org/10.1145/3274895.3274972

Because the data sources are unaligned a dynamic data integration is required. This is done using user-defined parametrized queries, thus facilitating a user-driven data exploration.

Existing work exists in studying the impact of accidents on traffic [2, 6, 8]. However, none of these uses GPS data and can therefore not make detailed analysis of all individual roads as with this approach.

The paper is organized as follows. We describe the data foundation, data model, dynamic data integration, and system architecture in section 2. In section 3, we describe query processing and optimization. In section 4, we study query performance. In section 5, we present the demonstration use cases. Finally, section 6, summarizes the paper.

2 ACCIDENT PLATFORM

First the data foundation is introduced. Next, a logical data warehouse model is presented. Then the dynamic data integration techniques are presented. Finally, the software architecture is shown.

2.1 Data Foundation

31 433 accidents [1] includes all recorded accidents in Denmark from January 2016 to July 2018. The accident data is modeled as $a = \langle id, date, time, coordinate \rangle$ where id is a unique id, $date$ and $time$ describe when an accident is reported, and $coordinate$ is the location of the accident.

An OpenStreetMap (OSM) extract of Denmark is modeled as a directed graph, $G = \langle V, E \rangle$. The OSM highway tag is used as $category$ and $driving\_direction$ tells whether two-way traffic is allowed. The graph consists of 816 641 vertices, V, and 1 834 051 directed edge, E.

The GPS data is recorded at 1 HZ. The GPS data is modeled as $GPS = \langle vid, date, time, position \rangle$ where vid is a vehicle id and $date$, $time$, and $position$ describe when and where the vehicle is located. In total, 38 026 127 859 records are logged in the period February 2016 to July 2018 by between 14,814 and 66,851 daily drivers.

The GPS records are map matched using a state-of-the-art algorithm [3] and stored as 4 116 660 920 rows. In total 1 180 432 597 traveled kilometers have been logged over 15 521 699 driving hours resulting in 44 595 479 distinct trips/trajectories.

Weather data is retrieved from NOAA [4]. Denmark is covered by 77 active weather stations. The weather is modeled as $W = \langle date, hour, location, condition \rangle$ where $date$ and $hour$ describe which hour the measurement is valid for and $location$ the location of the station. The $condition$ describes weather class, e.g., dry, snow, or rain.

2.2 Data Model

To efficiently query the integrated data a star-schema data model is designed. Figure 2 shows the logical model. Attributes in italic are spatial OGC [5] features, either points or linestrings. The tables and relations are colored by their relations. Fully marked lines define static and predefined relations and dashed lines define dynamic relations that are configured using parameters at query time.

2.3 Dynamic Spatio-Temporal Integration

Independent data sources are updated with different frequencies, e.g., a traffic accident is report almost immediately via Twitter or

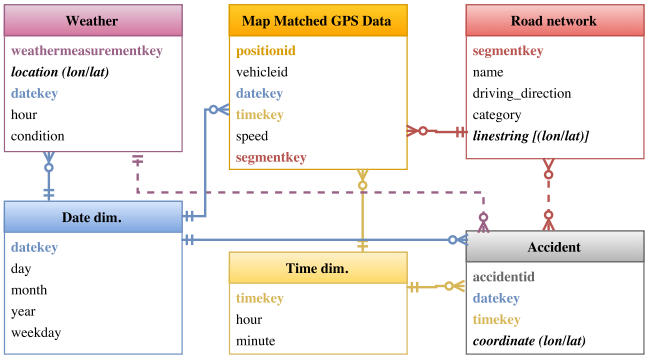

Figure 2: Logical Data Model

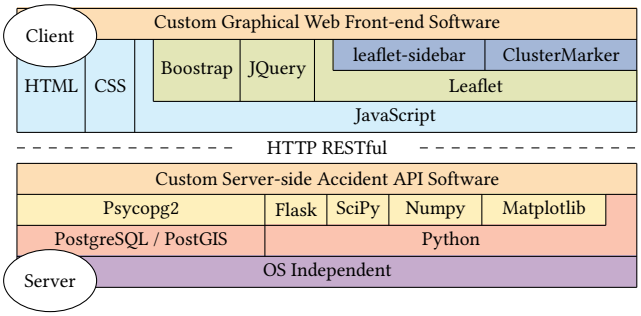

Figure 3: System Architecture

TMC (radio), GPS data is bulk loaded nightly, and curated weather information takes several days before being available. This means a several days delay, e.g., on estimating how travel time is affected by the combination of a traffic accident and heavy rain.

Another practical problem with integrated data sources is that values may be late-arriving or it is uncertain if data is missing or will arrive later, e.g., weather data for some stations may be delayed, hence integration with nearest weather station changes as data becomes available.

To overcome these problems, we use a dynamic spatio-temporal integration where the join conditions are determined at query time and not when the schema is created. This approach to integration is generic and has several advantages.

- It supports data exploration very well as a data analysts can change integration parameters such a temporal period or spatial relationship at query time.
- It support seamless integration of multiple data sources as soon as they are available, e.g., a traffic accidents effect on the GPS trajectories can be done without the weather information being available.
- It supports that the nearest available data can be used when a data source has missing or late-arriving data, e.g., if the nearest weather station malfunctions for a period the second-nearest is used instead.

To enable the same possibilities with static integration would require exorbitant large quantities of data for all combinations of spatial and temporal granularities.

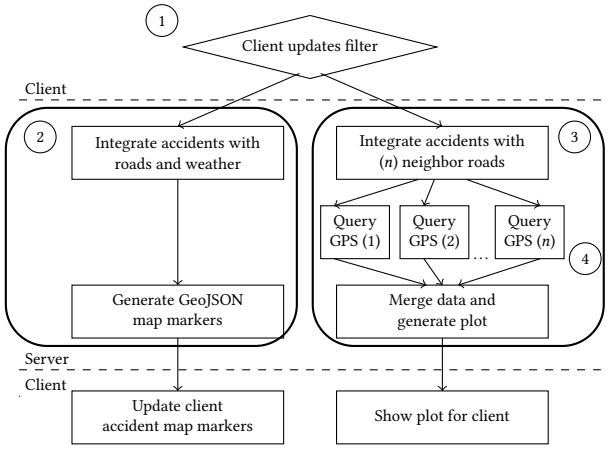

Figure 4: Execution Flow

2.4 System Architecture

The system has a client-server architecture as shown Figure 3. The architecture is open-source based and is independent on operating system and web browser. The client is a web-based graphical front-end implemented using JavaScript and HTML. The JavaScript library Leaflet serves an interactive map. The back-end is implemented as a RESTful API using Python 3.6, PostgreSQL 9.6, and PostGIS 2.3. Plots are generated using the matplotlib library.

3 QUERY PROCESSING AND OPTIMIZATION

In this section, the implemented query processing technique is introduced and a concrete example presented. Next, query optimization techniques are discussed.

3.1 Query Processing

The demonstration in Figure 5, is an interactive web application, where read-only users can browse and explore accidents data. Users can dynamically modify various parameters and filters related to spatial, temporal, road, and weather data. Data is updated in batches every night, when data becomes available. A nightly batch load of 400 million GPS data, the equivalent of five days data, takes approximately 3 hours.

Figure 4 shows the execution flow when users update any parameters or filters in the left pane in Figure 5. At (1) in Figure 4 the user changes one value. This leads to two parallel requests for information, (2) updating the markers on the map, and (3 and 4) refreshing the plot.

The processing (2) is a single SQL query integrating accidents, roads, weather, date-, and time dimensions. In (3), all neighbor road segments (n segments), are first determined. At (4) the n roads found are distributed into n parallel queries, to compute the average speeds on the roads. Finally, the output from the parallel queries are combined into a single plot.

3.2 Example of Parameterized Query

An example of a parameterized query is shown in Listing 1. This SQL query is used at (2) in Figure 4 and enables filtering and preparing data for visualization. The coloring of the query in Listing 1 is identical to the coloring in Figure 2.

Listing 1: Dynamic Integration using Parameters

```
select distinct a.id, a.coordinate,
  r is not null as has_roads, w is not null as has_weather
from accident a
join dimdate d on d.datekey= a.datekey
  and d_from <= d.date <= d_to
  and d.weekday in d_weekday and d.month in d_month
join dimtime t on t.timekey = a.timekey
  and  t_start <= t.time <= t_end
left join roads r on dist(a.loc,r.linestring) <= r_dist
left join weather w on w.weatherkey=
  (select w.weatherkey from weather w2
  where w2.hour=t.hour and w2.datekey=d.datekey
  order by dist(w2.location,a.coordinate) limit 1)
where coalesce(r.category,'missing') in r_cat
  and coalesce(w.condition,'missing') in w_cond
```

Table 1: Query Time and Result Set Size for Overview

	Query Time [ms]				Found Roads [rows]			
M	1 wk	1 mth	3 mth	12 mth	1 wk	1 mth	3 mth	12 mth
10	164	664	2094	8224	639	2529	7383	22 875
25	173	706	2234	8787	941	3745	10 972	33 863
50	192	787	2496	9872	1526	6034	17 295	52 147
100	243	1005	3193	12 649	3088	12 304	34 389	95 240
200	388	1611	5130	20 370	8059	30 157	76 562	182 727

The parameters d_{from} and d_{to} describe a date interval, $d_{weekday}$ and d_{month} enable filtering on weekdays and months, and t_{start} and t_{end} enable filtering on time-of-day.

The $dist()$ function in Listing 1 computes the Euclidean distance. All italic fields are parameters, specified at query time. The r_{dist} parameter specifies the spatial range for searching for roads and r_{cat} enables filtering on road categories. The left outer join and coalesce on $r.category$ enables querying on accidents even though no road or GPS data is available. To find weather information we search for the nearest weather station, hence an inner select is used to order weather stations by the Euclidean distance to an accident.

3.3 Query Optimizations

Index-only scans [7] is a technique, where an index covers all columns utilized by a query (also called a covering index). Such a covering index can greatly enhance performance. In Listing 1, a covering index on the columns *weatherkey* and *condition* is utilized when joining *weather w*.

Horizontal table partitioning is used on the GPS data to reduce table and index sizes. Partitioning improves query performance when only few partitions are used by a query.

4 PERFORMANCE STUDY

The performance of the query-time data integration is important to ensure that users can browse accident data interactively. The performance of the query presented in Listing 1 depends primarily on the spatial and temporal ranges. We study the performance of the query for varying ranges in this section.

In Table 1, the spatial search range varies from 10 to 200 meters. The temporal range varies from 1 week to 12 months. The left side in Table 1 shows the query times and the right side shows the number of roads found. The system scales almost linear with the ranges

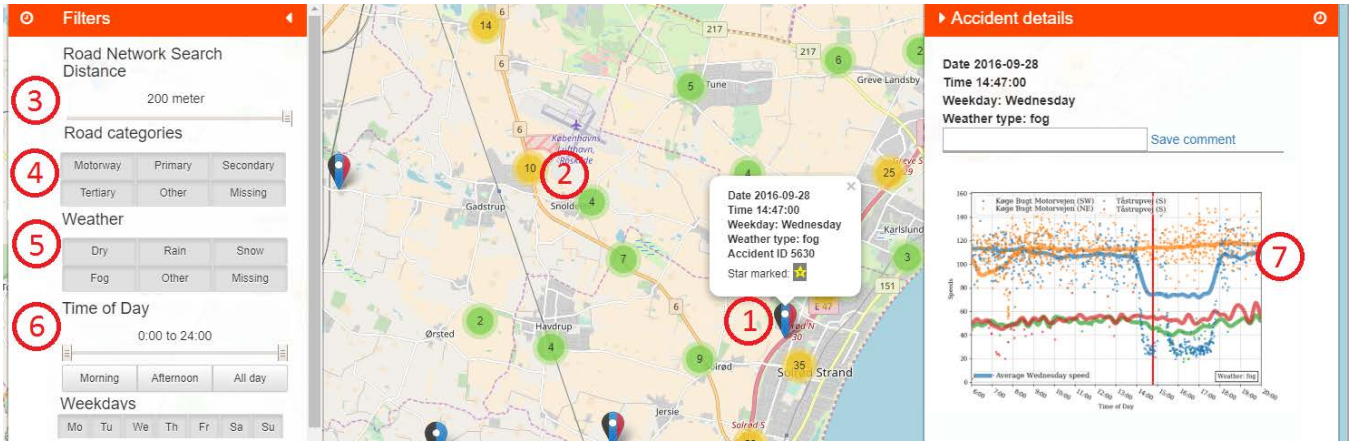

Figure 5: Accident Impact

Table 2: Query Time and Result Set Size for Plot

	Query Time [s]				Found Speed Measures [rows]			
M	+/-1 h	+/-3 h	+/-6 h	all day	+/-1 h	+/-3 h	+/-6 h	all day
10	2.6	2.6	2.6	2.6	148	395	613	814
25	3.1	3.2	3.2	3.2	217	669	1205	1720
50	3.1	3.1	3.1	3.3	217	669	1205	1720
100	3.0	3.0	3.1	3.2	217	669	1205	1720
200	4.1	4.6	4.6	5.1	231	704	1250	1800

and the query-time data integration is efficient. A closer analyses reveals that most of the time is spent on spatially integrating the accidents with the edges on the map.

The time to generate a plot like the one shown in Figure 1 depends on the temporal range (the x-axis) and the spatial range. This is shown in Table 2. Increasing the temporal range from one hour to all day has limited impact. Increasing the spatial range only slightly decreases the performance.

5 DEMONSTRATION

The developed graphical web front-end is shown in Figure 5. This front-end demonstrates the power of the dynamically integrated accident, map, GPS, and weather data. A back-end processes the dynamic integration with the data warehouse.

The accidents are plotted on a map (item 1) and when the user zooms out the accidents are clustered in groups (item 2) to provide an overview of the number of accidents in an area. A colored pointer (item 1) indicates an individual accident. Black color is an accident, blue means a road segment is found, and red means weather is present. Selecting an accident will pop up a window (item 1) providing details on the accident. To the left and right two panes can be unfolded (already done in Figure 5). In the left pane the user can alter the spatial range data (item 3) and it is possible to filter on road and weather contexts (item 4 and 5). A set of temporal filters are also available (item 6). The plot at (item 7) shows an accident's impact on nearby roads, depending on (item 3) and (item 4). The plot shows the same accident as Figure 1 though with a larger search distance, hence two more roads are included here.

The demonstration uses the following three scenarios as an outset. The first scenario demonstrates how the user can browse all traffic accidents. It is demonstrated how the various filters are applied to focus the browsing, e.g., looking only at motorways (item 4 in Figure 5) and how the spatial range can be changed at query-time. The second scenario focuses on how analyses are usable even when having late-arriving facts. It will be demonstrated how the system provides valuable information at very early stages when GPS and weather data is not available. The third scenario shows how dynamic data integration can be valuable when analyzing a single accident by adjusting the spatial range (item 3 in Figure 5).

6 SUMMARY

A data model has been presented that integrates accident, GPS, map, and weather data. Because these data source are upgrade with different frequencies a query-time integration is used. The integrated data is presented to the end-user through a web front-end. A performance study shows that the proposed data model and software architecture scales to very large data set using country-size road networks. A number of scenarios demonstrates the usefulness solution, in particular how accidents may delay traffic is highly relevant to both drivers and road authorities. Natural future work directions is real-time detection of accidents and automatically determine spatial and temporal scope from accidents.

REFERENCES

[1] Danish Road Directorate. [n. d.]. http://www.vejdirektoratet.dk/EN/
[2] Bo Xu et al. 2016. Real-time Detection and Classification of Traffic Jams from Probe Data. In *ACM SIGSPATIAL GIS*. 79:1–79:4.
[3] Paul Newson and John Krumm. 2009. Hidden Markov Map Matching Through Noise and Sparseness. In *ACM SIGSPATIAL GIS*. ACM, 336–343.
[4] North Oceanic and Atmospheric Administration. [n. d.]. http://www.noaa.gov/
[5] Open Geospatial Consortium Inc. 2011. *OpenGIS Implementation Standard for Geographic information - Simple feature access* (1.2.1 ed.).
[6] Bei Pan, Ugur Demiryurek, Cyrus Shahabi, and Chetan Gupta. 2013. Forecasting Spatiotemporal Impact of Traffic Incidents on Road Networks. In *IEEE ICDM*. 587–596.
[7] PostgreSQL Wiki. [n. d.]. Index-only scans. https://wiki.postgresql.org/wiki/Index-only_scans Checked 2017-09-29.
[8] Benjamin Romano and Zhe Jiang. 2017. Visualizing Traffic Accident Hotspots Based on Spatial-Temporal Network Kernel Density Estimation. In *ACM SIGSPATIAL GIS*. 98:1–98:4.
[9] USA TODAY. 2015. https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/812013