Event Processing and Real-time Monitoring over Streaming Traffic Data

Kostas Patroumpas[1] and Timos Sellis[1,2]

[1] School of Electrical and Computer Engineering
National Technical University of Athens, Hellas
[2] Institute for the Management of Information Systems, R.C. "Athena", Hellas
{kpatro,timos}@dbnet.ece.ntua.gr

Abstract. Tracking mobility of humans, animals or merchandise has recently given rise to a wide variety of location-based services and monitoring applications. In this paper, we particularly focus on real-time traffic surveillance over densely congested road networks in large metropolitan areas. In such a setting, streaming positional updates are being frequently relayed into a central server from numerous moving vehicles (buses, taxis, passenger cars etc.). Our analysis concerns two important aspects. First, we outline characteristics of a robust processing engine that is capable to efficiently manage such massive, transient, and perhaps noisy geospatial data. Our objective is to provide online aggregates and reliable estimates regarding the current traffic situation at multiple levels of resolution. At a second step, we design a framework for effective multi-modal dissemination of derived information to the end users, in the form of interactive maps for intuitive visualization as well as instant notifications via message feeds. As a proof of concept, we also report on our ongoing development of EPOPS; in its current version, this functional prototype of the proposed scheme is able to deliver cross-platform geographic, textual, and even multimedia content through web and smartphone interfaces.

Keywords: cross-platform dissemination, events, geostreaming, multi-resolution, traffic analytics.

1 Introduction

Widespread deployment of sensor networks and cost-effective communication technologies have recently led to a surge in Intelligent Transportation Systems (ITS). Along highways, vehicle counts get collected via stationary detectors or inductive loops placed on the road surface. Surveillance cameras monitor traffic conditions and can provide data for speed and travel times from post-processed video images. But installation of such systems over dense networks in urban areas with millions of inhabitants seems neither feasible nor affordable, because expensive equipment is set up at fixed positions on major roads.

Instead, modern positioning devices (like GPS, RFID, GSM) can actually turn moving vehicles into active contributors of high-quality, real-time traffic information. Effectively, such floating car data (FCD) tracks successive locations for a fleet of registered vehicles at distinct time instants. At a relatively low cost, FCD is then transmitted into a central server, where it gets analyzed for extracting traffic statistics [6,22,24]. Apart from control centers, this data might further assist to travel planning for commuters, support coordination of public transport, and also integrate with car navigators or traffic telematics.

In this paper, we advocate for a real-time evaluation of vehicle traces over road networks and their collective representation as *traffic data streams*. Similar platforms have emerged lately [1,7,10,11,18], as vehicle positions and derived statistics are inherently fluctuating, potentially intermittent, and ever more voluminous to be hosted by a traditional DBMS. In a spirit close to ours, the objective is to turn quantitative samples (raw positions) into qualitative estimates (e.g., average speed, expected travel times). Such approximate, yet reliable analytics is derived online, employing stages of filtering, aggregation and export.

Nevertheless, our framework is mostly geared towards a *multi-resolution representation* of traffic streams and props up *multi-modal diffusion* of custom, succinct analytics to potential users. We stress that our goal is not to suggest novel practices to traffic engineers; unsurprisingly, sophisticated algorithms already abound. What we really attempt is a *geostreaming* approach to road traffic data, enhanced with methods for extracting privacy-preserving driving patterns and evolving phenomena network-wide. Interesting events may not just be issued from external sources (e.g., reported incidents), but could be promptly detected by observing trends in traffic conditions (e.g., increasing delays).

Therefore, we introduce an efficient and robust combination of: (i) suitable *traffic semantics* that can capture spatiotemporal phenomena and evolving events along road networks; (ii) the advanced data management capabilities offered by *stream processing engines*; (iii) the skillful dissemination power of *web and mobile technologies*; and (iv) rich visualization tools from modern *geospatial infrastructure*. Implementation of our prototype platform (EPOPS) confirms that this methodology can offer up-to-date traffic information at varying levels of detail and greater range, and also achieve its versatile portrayal through interactive maps. Our contribution can be summarized as follows:

- We outline an abstract model for representing traffic data streams.
- We suggest a solid methodology for online processing of floating car data.
- We combine on-the-fly traffic measurements with contextual information in order to provide timely, reliable, and multi-faceted analytics.
- We develop interfaces for visualizing, inspecting and diffusing results through web and smartphone applications.

The remainder of this paper is organized as follows. In Section 2, we discuss fundamental notions and modeling of traffic data streams. In Section 3, we develop a processing framework for translating vehicle positions into online traffic analytics. Section 4 presents methods for cross-platform delivery of results to the end-users. In Section 5, we report our experience from implementing EPOPS, a functional prototype of the proposed scheme. In Section 6, we survey related work. Section 7 offers conclusions and indicates possible future extensions.

2 Fundamentals of Traffic Data Streams

In this section, we analyze design considerations and modeling guidelines concerning how streaming positional data is acquired, represented, and eventually transformed into meaningful traffic aggregates.

2.1 Data Acquisition

Although we do not rule out accepting traffic surveillance data feeds from public authorities (Traffic Police, Transportation Offices, etc.), we mainly focus on processing positional information issued from the moving vehicles. Indeed, monitoring a few thousand (e.g., 5000) location-aware cars moving in a city can provide a fairly accurate FCD sample for assessing actual traffic conditions. Assuming a proper combination of vehicle types (buses, taxis, trucks, passenger cars etc.), not only can we capture various driving patterns (fixed itineraries, ad-hoc routes, detours etc.), but also get traffic indications network-wide. Such raw tracking data could be acquired through diverse means: GPS-equipped vehicles, mobile phones of the drivers, wireless networks etc.

Positional updates are reported to a central server, but not necessarily at fixed periods, as sampling rates may not be standard for the entire fleet (e.g., buses with known itinerary may report less frequently than taxis). Reporting frequency might also be varying even for a single vehicle so as to reduce communication cost; for instance, it could relay a new location when making a turn or upon significant change in its speed. On the server side, positional information is periodically correlated with the geographic representation of the underlying network and certain rules in order to derive online traffic analytics. In brief:

Guideline 1 *Timestamped positional updates from moving vehicles are being transmitted to a centralized processor in a streaming fashion.*

2.2 Data Semantics

We adopt an object-relational schema with spatiotemporal extensions for representing static and dynamic data. Static tables store information about entities that rarely change (e.g., road network and its classification, vehicle types), whereas dynamic tables retain streaming data (e.g., vehicle positions, speed profiles for roads) always associated with timestamp values. A rich combination of semantics is employed to capture interesting traffic phenomena:

Spatial Semantics. It goes without saying that a detailed *road network* would provide a solid basis for the entire framework. Typically:

Guideline 2 *The road network is abstracted as a 2-d graph with links and nodes.*

Links represent centerlines of road segments as 2-dimensional polyline vectors, seamlessly interconnected at nodes that denote crossroads, junctions, or dead-ends. On the other hand, vehicle positions are abstracted as 2-dimensional *point locations*, ignoring other parameters such as the volume of each car or the altitude. Other geographic entities are clearly optional, since they do not directly interfere with traffic analysis; for example, points of interest (terminal stations, sporting venues, shopping malls etc.) or zone boundaries (limited-access areas, neighborhoods, or districts) might prove useful for specific calculations (e.g., access to a station), but they are not deemed indispensable features.

Temporal Semantics. Each incoming positional update, as well as every resulting aggregate must hold time indications [15]. Accordingly:

Guideline 3 *Timestamping of stream items achieves ordering and simultaneity.*

The former property guarantees the sequential nature of positional feeds and traffic analytics, and also allows reconstruction of the historical trace for each vehicle. The latter enables correlation of events occurring at the same time, provided that they carry identical timestamps. Not only does timestamping come for free because location-aware devices always emit *valid time* (i.e., when a position was actually measured), but it should be retained throughout evaluation stages for properly signaling aggregates (estimated speeds, travel times etc.). In case of unsynchronized clocks, *transaction time* of data admission to the system could be alternatively used, instead of temporarily buffering delayed messages.

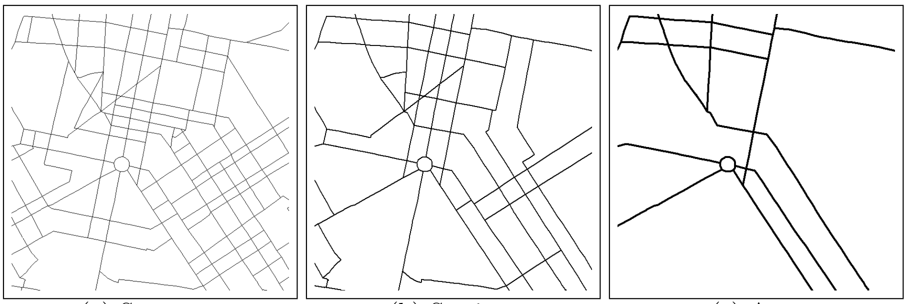
Motion Semantics. Spatiotemporal notions such as displacement, travel time, speed, heading etc. are important when dealing with moving objects [9]. Derived values per vehicle may be either instantaneous (based on its two most recent recordings) or averages (over larger intervals). Yet, it is crucial to realize that:

Guideline 4 Atomic measurements *per moving vehicle should eventually translate into* collective anonymized aggregates *along roads.*

In our case, the current speed or direction of each individual vehicle are entirely insignificant, no matter their accuracy. On the contrary, averaging speed measurements for vehicles moving along a specific road gives an approximate, but still reliable indication about traffic conditions there. From a privacy perspective, such aggregates can also effectively hide the identity of each driver among similarly moving others, and thus cannot disclose its whereabouts.

Traffic Semantics. Apart from their geographic representation, roads are key features for traffic estimation. Therefore, they should include properties such as length, national and international codes, speed limits, number of lanes, direction of traffic flow, hierarchy (highways, primary and secondary arterials, or collector roads) etc. Furthermore, roads can be dynamically classified according to their observed level of congestion, thus distinguishing currently saturated links from those with normal flow. Most importantly:

Guideline 5 *The road network may be organized in tiers of gradual resolution.*



(a) Segments (b) Sections (c) Axes

Fig. 1. Bottom-up hierarchical organization of road network in tiers.

At the finest tier, consecutive geometric *segments* belonging to the same road can form a unified *section* (e.g., between signaled crossroads); further, a series of sections with similar characteristics can represent an entire *axis* at the coarser tier. As shown in Fig. 1, smaller and less important roads may be suppressed from higher levels. Optionally, this logical grouping could further lead to multi-scale geographic representations of the network: at smaller scales the map could only display geometrically simplified axes, whereas zooming into a particular area should render all segments at the finest detail. As we discuss later, such modeling is advantageous not only for its richer semantics, but also for its decisive impact on data processing and map visualization of traffic phenomena.

Event Semantics. As vehicles circulate across the network, several events may occur, such as road accidents, demonstrations, bad weather conditions etc., which may have local or more severe implications to current traffic flow. Hence:

Guideline 6 *Support for event detection and online notification is essential.*

Event handling is two-fold: (i) for *declaring certain incidents*, like announcements issued from the Traffic Police for accidents or urgent warnings; and (ii) for *discovering driving patterns online* (e.g., a sharply reduced speed well below normal levels along a road may indicate an accident; a long-observed deterioration of traffic flow on a highway may cause queues stretching over several kilometers etc.). As we explain in Section 3.5, this latter class of events requires special-purpose algorithms or complex continuous queries against the traffic streams.

2.3 Data Manipulation

As each vehicle reports its position at distinct time instants, the server is able to trace its movement as an evolving *trajectory* [9]. But in our case, neither is it affordable nor even necessary to maintain the historical trace per vehicle as a 3-d "polyline" vector. This representation uses vertices at successive timestamped locations, and interpolates to approximate positions in between sampled readings.

Since our objective is to monitor network traffic and not individual vehicles, we must identify those road segments traversed by each car across time. Therefore:

Guideline 7 *Vehicle positions incrementally create* trajectory feeds *as sequences of road segments, judiciously selected from the underlying network.*

This way, the route of each vehicle is constructed as a series of road segments derived from a "map-matching" process, as we discuss in Section 3.1. Note that geometric accuracy and connectivity are essential properties of the network, since error-prone GPS readings should be mapped into suitable road segments (i.e., where a car actually moves on) minimizing spatial mismatches. In terms of achieving smooth traffic estimates, such modeling of trajectories is certainly beneficial, as it can effectively cope with diverse reporting frequencies and possible communication delays, while it also avoids double-counting of vehicles that keep moving along the same segment. Admittedly, some trajectory feeds may appear occasionally disconnected with gaps between selected segments (Fig. 3a). However, this is not a serious flaw, as vehicle traces are eventually turned into approximate traffic estimates. These traffic analytics are computed at coarser resolutions of the network, usually against sections or axes, each stretching over several segments. Hence, estimates for speed, link saturation, expected travel times etc. are actually aggregates from values contributed by trajectories of vehicles that have recently "charged" a given road.

Guideline 8 *Traffic aggregates are computed from trajectory feeds for selected tiers of the road network.*

We stress that all derived data is implicitly georeferenced against the road network. Every trajectory measurement or traffic aggregate is always associated with a particular road entity, so the identifier of the respective linear feature suffices and must be attached to that data item. There is absolutely no need to copy geometry features into any derived dataset, as these can be readily obtained from a simple join operation thanks to common identifiers. Of course, results must also be timestamped with respect to time indications of the input.

3 Traffic Stream Processing

Figure 2 illustrates the processing components of the suggested framework. Rectangles stand for domain data; each domain may be physically implemented with multiple relational tables of static records or streaming items. Diamonds represent processing tasks applied against incoming data according to specifications, rules, and parametrization depicted with ovals. Thick arrows indicate data flow from raw positional input to output traffic information passing through intermediate modules. Next, we discuss each component in detail.

3.1 Map-Matching

This pre-processing stage attempts to associate vehicle locations with road segments of the underlying network. As already mentioned, tracking data from

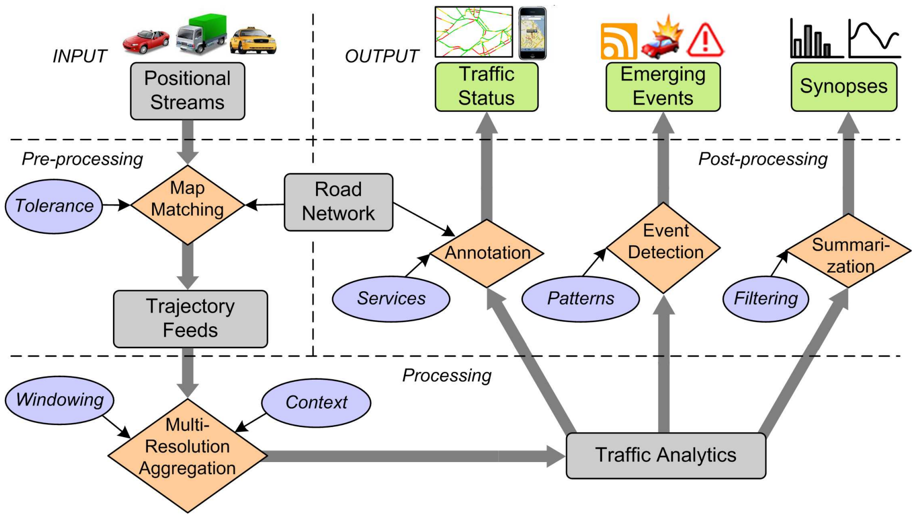


Fig. 2. Data flow diagram.

moving vehicles has limited positional accuracy. Apart from the sampling error caused by the reporting frequency, there is also a measurement error from the positioning device itself (e.g., GPS) [3]. Besides, the vector representation of the road network can never be perfectly accurate, due to digitization errors, scale of the source map or imagery resolution, lack of recent updates etc. As a result, it is not always straightforward to snap a positional item to the nearest node or link of the network. It may occur that locations could fall several meters off the road centerlines, and thus might be wrongly assigned (dotted segments in Fig. 3a). In addition, raw positional data are inherently noisy and must be properly cleansed before attempting any map-matching. For instance, vehicle positions should be ignored in case they might indicate manoeuvres for on-street parking followed by immobility, entrance into a private garage or parking space etc.

This problem is definitely crucial, but orthogonal to our approach. Thus, we can make use of any state-of-the-art algorithm like [3,5,19,20] that minimizes erroneous assignments under proper tolerance metrics (such as Fréchet or Hausdorff distance), also taking into account the recent vehicular movement as well as network features (e.g., direction of traffic flow, road hierarchy etc.). Depending on the technique, apart from timestamped points, other features like bearing or speed could be used for more accuracy. We integrate spatial access methods for indexing and fast retrieval of road entities, since identification of relevant segments must be performed for every incoming position. The result of map-matching is a sequence of road segments which the vehicle has supposedly traversed. Yet, we need something more: we also have to know when this vehicle entered each particular road segment, assuming that it kept moving at an estimated speed along each segment. As shown in Fig. 3b, interpolation of timestamp values from the original samples along involved road segments can

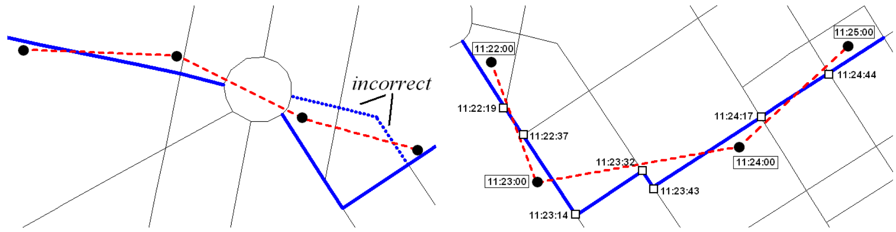


Fig. 3. (a) Mismatching and occasional gaps. (b) Map-matching of positional samples.

provide a fair approximation of such entrance times. Note that when a car turns into another road, intermediate segments should get included into the sequence, so we utilize a shortest-path algorithm between successive samples.

Consequently, such incremental trajectory feeds contain tuples that denote when a given vehicle entered a particular road segment and what was its actual velocity. This latter feature involves simple manipulations with the latest displacement of each moving vehicle, by comparing its current and previously reported position. Apart from the speed, we also need to know the direction of movement, which is quite significant for bidirectional roads.

3.2 Multi-resolution Traffic Analytics

The main processing task deals with online aggregation against map-matched trajectory feeds. Although such spatiotemporal computations are not particularly sophisticated, they must be repetitively applied across an extended network with thousands of links, and also keep track of the most recent traces for numerous vehicles. Such a task cannot be accomplished by a traditional DBMS, not only because of the unsustainable burden of frequent transactions, but also due to the necessity for incremental response to continuous traffic analytics.

We advocate for a lightweight, special-purpose stream processing mechanism that works in main memory and can offer qualitative, real-time estimates for:

- *Average speed*. Since every trajectory tuple for a given vehicle always states which road segment it currently traverses and at what speed, this aggregation is just a grouping of items by their road identifier.
- *Expected travel time* can be simply expressed as quotient of the geometric length of each road entity over its estimated average speed.
- *Vehicle counts*. This measure of traffic load is actually based on samples, since monitoring all vehicles moving in a city is not realistic by today's standards. Still, such counts might be useful as a rough indication of flow saturation, but primarily for assessing the accuracy of other traffic statistics.

Preferably, traffic estimates make sense for larger sections or axes (i.e., top tiers of road network) in order to attain sufficient samples. If only a few vehicles are currently moving along a large stretch of a road, then this sample may not be representative, so doubtful measurements should not get publicized. As a means of coping with such inherent deficiencies and also offering succinct traffic information, we opt for computing *multi-resolution analytics* according to:

– *Spatial context.* Data can be progressively aggregated at multiple network tiers (mostly axes and sections) so as to offer more insight into actual traffic conditions. For instance, flow may appear almost regular along an axis, but at a certain section the situation may be worsening. Similar roll-ups or drill-downs at varying levels of detail can help explaining certain traffic phenomena. Depending on application requirements, more options are possible: travel times may be estimated for several frequent routes (e.g., from/to the airport); low speed in designated zones may indicate congestion (around major junctions, near terminal stations or commercial districts) etc.
– *Temporal context.* Aggregating over the latest measurements only (e.g., instantaneous speed of each vehicle) can hardly give a truthful glimpse of traffic status. In most cases, data should be examined at varying time horizons (during past 5, 10, 30 minutes or more) for smoothing fluctuations caused by waiting times at traffic lights, impulsive reactions from drivers etc.
– *Traffic context.* Analyzing data by vehicle type can give precious clues about driving patterns for buses, taxis, trucks, passenger cars, etc. Besides, for selected arterials or road classes, this breakdown can also provide the presumed traffic composition (i.e., % of each vehicle type over the total traffic).

With respect to time frames, it matters not merely the period of interest for the latest streaming data, but also how frequently analytics should get refreshed. As in most stream processing engines, we stipulate *sliding windows* [15] for specifying the *range* ω (e.g., samples received during past 10 minutes), the *sliding step* β (e.g., results get reevaluated every minute) and the *initiation time* τ_0 of this computation. Note that parametrization of time frames is based on timestamps, in order to accomplish incremental production of results at regular intervals. Certainly, the exact window specifications are application-dependent, but they can be fixed in case of periodic evaluations (e.g., rush hours), or ad-hoc for capturing specific events (such as traffic near a sporting venue).

3.3 Derived Annotations

As soon as a new batch of traffic analytics is produced, this information should be forwarded to users. But is it worth returning the exact statistics for average speed or travel time per road, since these are just presumed estimates? We suggest that such approximate data should better be translated into inferred, *qualitative* indications. Indeed, knowing that the observed road speed is 12 km/h may seem too detailed and hardly precise due to limited samples, whereas an indication of "slow speed" is fairly reasonable, succinct and clear.

We employ a post-processing stage that interprets traffic analytics into *annotated* information according to its intentional use, particularly in web and wireless platforms. Determined by the type of services to be offered (e.g., maps, charts, statistics etc.), a variety of lookup tables may be utilized to transform numerical into categorical data. For instance, distinguishing speed values as "slow", "moderate" or "fast" is easily comprehensible, especially when depicted on maps with suitable symbology. Deviations of observed travel times from historical figures could simply characterize the congestion level of each road feature as "heavy", "normal" or "light". Note that such classification could be done dynamically according to time-dependent lookups (e.g., ranges may vary for rush hours, holidays etc.) and domain expert knowledge of local traffic conditions.

3.4 Updateable Synopses

Although online manipulation of incoming updates is our main focus, it should be noted that certain historical data may also prove valuable to traffic analysis. Raw positional data should not be maintained, not only because of privacy concerns but also due to their accumulating bulk. Nevertheless, properly summarized information about speed profiles per axis, vehicle counts or travel time estimates for traversing each road segment, can all be permanently stored in tables for both online use and offline historical comparisons.

Such concise synopses come from further aggregation of traffic analytics at *multiple granularities* [14]. Taking into account some expert knowledge, applied windows may span time periods varying from a few minutes to an entire day. Such post-processed data are incrementally appended into summaries and thus essentially maintain several evolving timeseries for critical parameters (speed, flow level, travel time etc. per road). In addition, the most fresh batch of such aggregates may also be used to offer online services, particularly for finding time-dependent shortest paths according to fluctuating traffic conditions.

3.5 Event Detection

Timely recognition of emerging incidents across the road network is of great importance for traffic surveillance. Such interesting phenomena include:

Link saturation : When the observed speed along a road steadily approaches or drops below a threshold (deduced from statistics over this time period), there are high chances that this link might soon become overloaded. As we discussed in [14], computing average speed values against nested time frames (say, spanning 5, 10 and 20 minutes) could instantly give a sign for unsatisfactory flow.

Deteriorating traffic flow : We can make use of a multi-level linear regression algorithm for discovering trends in traffic patterns across selected roads. The method we introduced in [14] involves multi-granular windows for online estimation of linear fits over varying time ranges in the recent past. When the current slope of these trendlines exacerbates or deviates substantially from the historical pattern of vehicle circulation on a road, traffic controllers can be notified to investigate the possible causes of that unusual situation (e.g., an accident, an inundated underpass after heavy rain, lengthy queues towards the stadium etc.).

4 Cross-platform Content Delivery

The wealth of traffic data gathered and analyzed through the processing engine makes its availability to the public a challenging opportunity for versatile, customized, and timely dissemination of information through web and mobile platforms. Offering online traffic analytics and event warnings can be accomplished with a user-friendly combination of rich visualization tools, interoperable data formats and modern technological outlets, as we explain next.

Advanced Visualization. Interactive maps are the perfect means for conveying traffic information. World-scale mapping infrastructure (e.g., GoogleMaps, OpenStreetMap etc.) or commercially available cartographic data could be used as the backdrop for real-time traffic monitoring. Properly annotated analytics for average speed, expected delays, and incident locations across the network can be exported in XML, RSS etc. Furthermore, correlating results with the underlying road network can also dynamically produce geographic features in GML/KML formats. Taking advantage of a multitude of APIs and open-source libraries, such dynamic layers can then be easily superimposed over the basic map with standardized symbology (colors, symbols, legends). On the other hand, continuously maintained speed profiles and traffic composition for roads could be illustrated with diagrams, histograms or charts. Last but not least, geographic annotation could integrate multimedia content (e.g., images linked to event locations, or streaming video from surveillance cameras), as well as eye-catching animation (e.g., flashing heavily congested links).

Customized Presentation. Traffic controllers, public authorities, and drivers often have differing perceptions of the traffic status. A common driver is primarily concerned about congestion along her way, while the Police usually want to avert bottlenecks and keep a regular flow mostly throughout arterial roads. Thus, results should be made available via *localized views* at various levels of detail (city, district, neighborhood) or depending on proximity to the current location of the driver.

Apart from standard information available for free, subscribers of the service may be offered a *personalized view* of the map with more detailed content. For example, user preferences could affect the desired level of resolution, refresh periods, filters for particular types of events, specific zones of interest etc. In addition, registered users may be allowed to post their remarks or hints to other drivers (e.g., a tweet to avoid a congested route).

Multi-modal Availability. Nowadays, the Web is the primary medium for making data easily accessible to the public. Hence, a user-friendly portal is indispensable for presenting and searching traffic information, combining maps, text, charts or multimedia. In addition, web services can be offered to public authorities or subscribed companies, such as fleet management agencies, taxi associations, delivery firms, advertising, radio/TV stations, etc. Such services can provide custom information including privacy-preserving statistics, traffic composition, speed profiles for roads etc.

Development of mobile applications for smartphones, tablets and PDAs would prove by far the most popular solution, as drivers can become aware of the traffic conditions while on the move. From instant notification about incidents, to interactive maps, to alternative route planning or a suite of online facilities (e.g., route recalculation, message tweets, image posts), the potential is enormous.

Broadcasting notifications can also be achieved in various ways. Depending on their preference settings, subscribers could receive urgent alerts with SMS texts or monitor periodically refreshed RSS feeds for incidents of their particular interest. Traffic message boards along major roads can display warnings to the drivers, whereas touch screens in parking lots or gas stations around the city could be used to depict interactive maps and news feeds. Finally, traffic message channel technology (TMC) via radio frequencies is currently operational worldwide for delivering traffic and travel information.

5 The EPOPS Prototype

EPOPS* is the acronym of our framework for **E**vent **P**rocessing and **O**nline monitoring of **P**ositional **S**treams. We have begun implementing a traffic monitoring platform using the open-source TelegraphCQ stream engine [17] and publicly available APIs. A central processor is continuously "listening" for streaming elements, whereas annotated analytics and events are periodically materialized into XML formats, thereafter available through web and mobile interfaces.

5.1 Data Management

Due to lack of real-time traffic information, EPOPS is currently tested against synthetic datasets that represent random itineraries of 10 000 vehicles circulating at the road network of greater Athens. We simulate online updates, enforcing a very frequent arrival rate of a new location per moving vehicle issued every 15 seconds. We assume no stream imperfections (e.g., missing or delayed data), so all positional items can be ordered by their original timestamp values.

Although TelegraphCQ is a research prototype with certain limitations (e.g., no support for nested subqueries or stream self-joins), it has been built on top of PostgreSQL. Hence, it comes readily equipped with built-in spatial operators, functions and data types (point, path, polygon, etc.), offering a great benefit for expressing continuous queries over geospatial streams as we demonstrated in [13]. Using SQL-like commands, we have defined a schema for maintaining

---

* ἔποψ is the ancient Greek name of the Hoopoe (*Upupa epops*), a colorful bird with a distinctive "crown" of feathers. In Aristophanes' comedy "The Birds" (ὄρνιθες, 414 BC), two Athenians are in search of prince Tereus, whom the Olympian gods have turned into a hoopoe and charged as all-seeing ruler (παντεπόπτης) over the birds.

static data (spatial and non-spatial tables) as well as properly organized streaming data (concerning incoming locations and derived features). At the least, each moving vehicle relays tuples ⟨`vehicleID, x, y, ts`⟩ with its geographic position in lon/lat coordinates at timestamp `ts`. This data resides at stream table `Positions` in main memory. The road network of Athens is stored in spatial table `RoadSegments` and its polyline features are indexed with R-tree. Thanks to a three-part coding scheme, higher tiers are defined as external views (`RoadSections`, `RoadAxes`) over the detailed road segments.

We developed stored procedures in PL/pgSQL for coping with map-matching of observed locations into road network features. As positional items are turned into trajectory sequences that feed a stream table `Trajectories`, windows are employed in order to periodically derive traffic analytics at varying levels of resolution (sections and axes). Streamlined results are always emitted with the latest timestamp value, thanks to the `wtime(*)` function of TelegraphCQ.

Specifically, a tumbling window [15] fetches the most recently received items and assists in computing an indicative traffic load as *vehicle counts*. Note that the cutoff threshold of 10 samples could be varying for each road section:

```
SELECT T.SectionID, COUNT(T.VehicleID) AS Num_vehicles, wtime(*) AS ts
FROM Trajectories AS T [RANGE BY '15 SECONDS' SLIDE BY '15 SECONDS'
                        START AT '2011-11-28 12:00:00']
GROUP BY T.SectionID
HAVING COUNT(T.VehicleID) > 10;
```

Instead, for computing the *average speed* per axis, a sliding window is employed to aggregate instantaneous speed values over the past 5 minutes. Results are updated every 15 seconds, in pace with newly arriving measurements:

```
SELECT T.AxisID, AVG(T.Speed) AS Avg_speed, wtime(*) AS ts
FROM Trajectories AS T [RANGE BY '5 MINUTES' SLIDE BY '15 SECONDS'
                        START AT '2011-11-28 12:00:00']
GROUP BY T.AxisID;
```

Expected *travel times* over road features can be also expressed with an adequate sliding window. Correlating the geometric length of each axis (as obtained from view `RoadAxes`) with the respective speed estimate during the past 10 minutes, easily provides such time-dependent figures every minute:

```
SELECT T.AxisID, R.Length/AVG(T.Speed) AS Travel_time, wtime(*) AS ts
FROM Trajectories AS T [RANGE BY '10 MINUTES' SLIDE BY '1 MINUTE'
                        START AT '2011-11-28 12:00:00'], RoadAxes AS R
WHERE T.AxisID = R.AxisID
GROUP BY T.AxisID;
```

Incremental results from this continuous query could be appended into a synopsis that may serve online routing requests from users. Indeed, by updating network cost values with these evolving travel times and making use of the pgRouting module [16] for PostgreSQL, we managed to provide a *shortest path facility* for resolving routes between arbitrary origin/destination points. We have

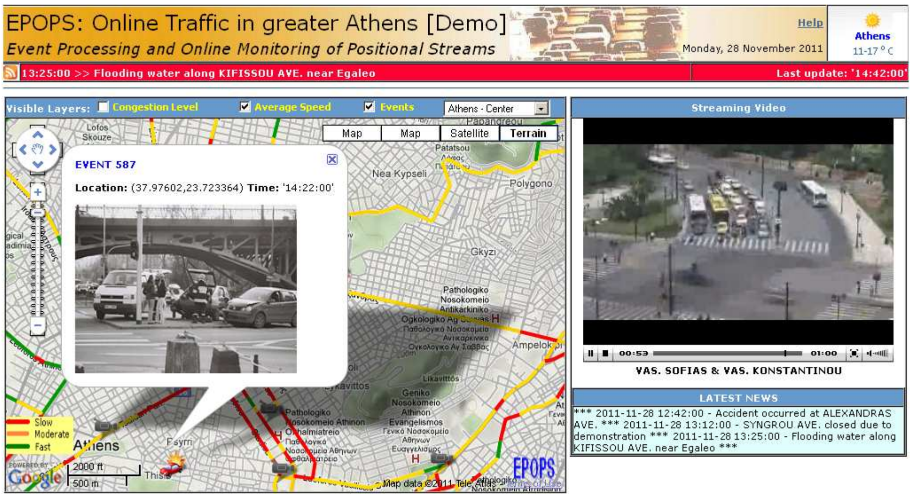


Fig. 4. Web interface for the EPOPS prototype.

also successfully experimented with a scenario for online toll charging against drivers that habitually aggravate congestion levels during rush hours.

In terms of event handling, we currently support issuing of incidents from authorized users only. Tuples like ⟨`eventID, x, y, ts, type, message, media`⟩ get inserted into a stream table `Events`, stating occurrence of an incident (car crash, road works, etc.), along with a textual message and multimedia content (image or video). We are working towards integration of our standalone algorithms [14] for detecting events directly from traffic streams (cf. Section 3.5).

5.2 Multi-modal User Interfaces

In terms of content delivery, EPOPS intends to facilitate user interaction with traffic information in an intuitive fashion. Implemented interfaces for *web* (with JavaScript and php scripts) and *smartphone* platforms (in Android SDK) offer functionalities that can handle maps, notifications and multimedia content.

More specifically, both interfaces are centered around geographic maps using GoogleMaps API. Typical cartographic operations are built-in, so our effort focused on proper rendering of dynamic layers (speed, delays, events) with suitable symbology. Reported events and annotated aggregates from online analysis in TelegraphCQ get periodically posted at the server in various formats (XML, KML, RSS). Derived elements become available at multiple resolutions, expressed in different map scales and event rankings. For example, zooming out the map to the entire city only displays speed levels along major axes and a handful of the most important incidents. But zooming into a neighborhood retrieves more detailed information for road sections and all events occurred recently in that area. To help reduce communication costs and response times for render-

Fig. 5. Screens of Android application for the EPOPS prototype.

ing, we have chosen to provide lightweight KML files with properly generalized shapes (simplified polylines) depending on the actual map scale on screen.

Aside from visual inspection of traffic phenomena, the implemented web interface (Fig. 4) also offers: localized views when the user chooses an area of interest, a widget for actual weather conditions, as well as scrolling messages for events as they occur (accidents, road works, demonstrations, etc.). Furthermore, users may display photos for chosen incidents and even watch videos from surveillance cameras at major junctions. Because access to streaming video is not yet authorized for our tests, we currently show archived footage only.

The Android application for EPOPS (Fig. 5) offers a basic screen for map rendering of traffic layers with custom symbols, always at a spatial resolution implicitly controlled by the actual scale. The user is able to personalize reception of results according to her preferences; configuration is now limited to vehicle types and time horizons, but more options are possible. Finally, the mobile application connects to the server and accepts XML/RSS feeds with event notifications.

6 Related Work

Stream-based processing of geospatial data, also known as *geostreaming*, offers a novel paradigm for spatiotemporal management, particularly for online monitoring of location-aware vehicles. Spatially-enabled stream engines have emerged, covering a wide range of topics related to ITS, such as data acquisition, map-matching, aggregation, and prediction. CarTel [10] is a mobile sensor computing platform designed to collect, process, deliver, and visualize data from sensors located on automobiles. IBM InfoSphere Streams [1] shares a lot of features with our approach, as it aims at scalability, timeliness and versatility of derived information in ITS. This component-based distributed platform utilizes a declarative language SPADE for specifying data flow graphs as well as for deploying applications at runtime. Accordingly, a prototype ITS application was developed for traffic monitoring in the city of Stockholm, using real vehicle GPS readings and road network maps. Besides, an evaluation framework equipped with data stream mining algorithms is proposed in [7] specifically for traffic applications. In a case study, cooperative cars exchange messages through cellular infrastructure and the proposed techniques may detect road segments with queue-ends. Finally, GeoInsight [11] is based on Microsoft SQL Server StreamInsight platform and leverages continuous query execution with spatial and temporal capabilities, as well as ad hoc analytical extensions for online refinement and prediction. Their demo scenario handles streaming data from traffic sensors in Los Angeles county, and correlates it with historical statistics in order to predict future trends. As opposed to such sophisticated, full-fledged commercial engines, our prototype takes advantage of open source processing software and endorses mobile technology features for cross-platform dissemination of traffic statistics.

Regarding map-matching of sensed vehicle locations, one class of algorithms is mostly geared towards maximizing *accuracy* of the resulting road identifications. Having a traffic engineering flavor, such approaches (e.g. [19,20]) attempt to minimize any mismatched links and reduce error propagation. A second class primarily focuses on maximizing *throughput* without compromising quality, thus taking a data management perspective. Throughput-oriented techniques can be further characterized either as: (a) *incremental* based solely on positional samples [8,5], when they examine edge distances and orientation of movement so as to greedily expand the existing path with an additional edge for fast response, or (b) *global* [2,3,4,12,21,23], in case that they check against all possible trajectories to find the most similar to the actual movement with better accuracy. More specifically, [4] proposed an ϵ-road-snapped trajectory construction algorithm in a weighted graph representation that returns a path, whose Hausdorff distance to the vehicle trajectory does not exceed location-sensor error ϵ. Based on a combination of spatial, topological and temporal features, a global ST-Matching algorithm in [12] considers GPS trajectories of low sampling rate. Exploiting the evolving trajectories of vehicles, algorithms introduced in [3] consider the entire path of a vehicle instead of its current position only. In essence, they attempt to minimize the Fréchet distance between the trajectory and the constructed sequence of road links; the computed distance also serves as a quality guarantee for the result. Such an approach, possibly enhanced with error estimates and output sensitivity [23], seems suitable for real-time tracking. Without excluding global techniques, an incremental one suits better to our scenario, as the cost of checking with potentially large trajectories would be prohibitive for increased stream rates of GPS readings. Since we presently handle synthetic datasets, for simplicity we take the shortest path between consecutive GPS measurements along the network as a close estimation of the vehicle trajectory, as in [1].

Aside from popular web sources (e.g., Google Maps, Yahoo Maps, Bing Maps, ESRI ArcGIS Online) that mostly visualize digested data, various methods have been proposed for travel time and shortest path estimation specifically for floating car data. Indicatively, algorithms in [6] are based on neural networks and pattern matching and offer short-term predictions. Characterizations of unique traffic patterns per road are addressed in [24], whereas [22] correlates GPS points with neighboring ones in space and time, in order to assess traffic status. For travel time estimation, [18] combines archived data with live traffic feeds in a streaming-oriented mode. In contrast, we opt for multi-resolution, succinct traffic statistics intended for easily perceptible portrayal and quick notifications.

7 Conclusions and Further Extensions

Overall experience from our functional prototype is more than encouraging. Implementation of EPOPS is still a work-in-progress, but already integrated modules offer concrete evidence for its robustness and low latency. The stream model for processing positional updates, as well as the hierarchical organization of network features have proven decisive factors for achieving timely, concise and multi-grained results. However, the most promising prospect comes from swift integration of modern geospatial, mobile, and web technologies for online delivery of traffic analytics. We anticipate that similar platforms will soon become the chief providers of real-time information for road traffic and safety.

In terms of scalability, it appears that even a full-fledged stream engine shows certain limitations when faced with increasing data volumes and arrival rates. In our extensive tests, TelegraphCQ was proven able to emit results promptly, but we believe that monitoring applications may need to accept information from tens of thousands of vehicles or more. A successful system should adequately fuse complementary tracking data that emanates from diverse devices (inductive loops, cameras etc.) and flows through heterogeneous networks. Not only could such a deployment provide better estimates, but it also opens prospects for custom web services to partners in related domains (logistics, transport, car navigation systems etc.) and social networking (user feedback, quick suggestions etc.). In that respect, traffic stream computing in the cloud would provide a flexible, highly-distributed solution.

We further envisage a probabilistic treatment for addressing the inherent uncertainty, gradual ageing, and transmission delays of positional streams. Traffic forecasts at short-term horizons (like 15, 30, or 60 minutes ahead) could be issued, gracefully weighing online analytics with offline statistics. Developing approximation algorithms in order to get traffic estimates with error guarantees seems a promising and quite challenging topic for research.

Acknowledgements. We would like to thank M.Sc. students Kyriakos Georgoussis and Chrysaida Papadopoulou for their collaboration in developing demo applications for online toll charges and shortest paths over this framework.

References

1. A. Biem, E. Bouillet, H. Feng, A. Ranganathan, A. Riabov, O. Verscheure, H. Koutsopoulos, and C. Moran. IBM InfoSphere Streams for Scalable, Real-time, Intelligent Transportation Services. In *SIGMOD*, pp. 1093-1104, 2010.
2. E. Bouillet and A. Ranganathan. Scalable, Real-Time Map-Matching Using IBM's System S. In *MDM*, pp. 249-257, 2010.

3. S. Brakatsoulas, D. Pfoser, R. Salas, and C. Wenk. On Map-Matching Vehicle Tracking Data. In *VLDB*, pp. 853-864, 2005.
4. H. Cao and O. Wolfson. Nonmaterialized Motion Information in Transport Networks. In *ICDT*, pp. 173-188, 2005.
5. A. Civilis, C.S. Jensen, and S. Pakalnis. Techniques for Efficient Road-network-based Tracking of Moving Objects. *IEEE TKDE*, 17(5):698-712, 2005.
6. C. de Fabritiis, R. Ragona, and G. Valenti. Traffic Estimation and Prediction Based on Real Time Floating Car Data. In *IEEE ITSC*, pp. 197-203, 2008.
7. S. Geisler, C. Quix, and S. Schiffer. A Data Stream-based Evaluation Framework for Traffic Information Systems. In *IWGS*, pp. 11-18, 2010.
8. J. Greenfeld. Matching GPS Observations to Locations on a Digital Map. In *81th Annual Meeting of the Transportation Research Board*, Washington, DC, 2002.
9. R.H. Güting, M.H. Böhlen, M. Erwig, C.S. Jensen, N.A. Lorentzos, M. Schneider, and M. Vazirgiannis. A Foundation for Representing and Querying Moving Objects. *ACM Transactions on Database Systems*, 25(1):1-42, 2000.
10. B. Hull, V. Bychkovsky, Y. Zhang, K. Chen, M. Goraczko, A. Miu, E. Shih, H. Balakrishnan and S. Madden. CarTel: A Distributed Mobile Sensor Computing System. In *SenSys*, pp. 125-138, 2006.
11. S.J. Kazemitabar, U. Demiryurek, M. Ali, A. Akdogan, and C. Shahabi. Geospatial Stream Query Processing using Microsoft SQL Server StreamInsight. In *VLDB*, pp. 1537-1540, 2010.
12. Y. Lou, C. Zhang, Y. Zheng, X. Xie, W. Wang, and Y. Huang. Map-Matching for Low-Sampling-Rate GPS Trajectories. In *ACM GIS*, pp. 352-361, 2009.
13. K. Patroumpas, E. Kefallinou, and T. Sellis. Monitoring Continuous Queries over Streaming Locations. In *ACM GIS*, pp. 521-522, 2008.
14. K. Patroumpas and T. Sellis. Multi-granular Time-based Sliding Windows over Data Streams. In *IEEE TIME*, pp. 146-153, 2010.
15. K. Patroumpas and T. Sellis. Maintaining consistent Results of Continuous Queries under Diverse Window Specifications. *Information Systems*, 36(1):42-61, 2011.
16. pgRouting Project. `http://www.pgrouting.org/`
17. TelegraphCQ Project. `http://telegraph.cs.berkeley.edu/telegraphcq/v2.1/`
18. K. Tufte, J. Li, D. Maier, V. Papadimos, R. Bertini, and J. Rucker. Travel Time Estimation using NiagaraST and latte. In *SIGMOD*, pp. 1091-1093, 2007.
19. N. Velaga, M. Quddus, and A. Bristow. Developing an Enhanced Weight-based Topological Map-matching Algorithm for Intelligent Transport Systems. *Transportation Research*, Part C 17:672-683, 2009.
20. W. Wang, J. Jin, B. Ran, and X. Guo. Large-Scale Freeway Network Traffic Monitoring: A Map-Matching Algorithm Based on Low-Logging Frequency GPS Probe Data. *Journal of Intelligent Transportation Systems*, 15(2):63-74, 2011.
21. M. Weber, L. Liu, K. Jones, M.J. Covington, L. Nachman, and P. Pesti. On Map Matching of Wireless Positioning Data: A Selective Look-ahead Approach. In *ACM GIS*, pp. 290-299, 2010.
22. L. Wei, W. Peng, C. Lin, and C. Jung. Exploring Spatio-Temporal Features for Traffic Estimation on Road Networks. In *SSTD*, pp. 399-404, 2009.
23. C. Wenk, R. Salas, and D. Pfoser. Addressing the Need for Map-Matching Speed: Localizing Global Curve-Matching Algorithms. In *SSDBM*, pp. 379-388, 2006.
24. J. Yoon, B. Noble, and M. Liu. Surface Street Traffic Estimation. In *MobiSys*, pp. 220-232, 2007.