Scalable Detection of Traffic Congestion from Massive Floating Car Data Streams

Győző Gidófalvi
Division of Geoinformatics
Department of Urban Planning and Environment
KTH Royal Institution of Technology, Sweden
gyozo@kth.se

Can Yang
Division of Geoinformatics
Department of Urban Planning and Environment
KTH Royal Institution of Technology, Sweden
cyang@kth.se

ABSTRACT
Motivated by the high utility and growing availability of Floating Car Data (FCD) streams for traffic congestion modeling and subsequent traffic congestion-related intelligent traffic management tasks, this paper proposes a grid-based, time-inhomogeneous model and method for the detection of congestion from large FCD streams. Furthermore, the paper proposes a simple but effective, high-level implementation of the method using off-the-shelf relational database technology that can readily be ported to Big Data processing frameworks. Empirical evaluations on millions of real-world taxi trajectories show that 1) the spatio-temporal distribution and clustering of the detected congestions are reasonable and 2) the method and its prototype implementation scale linearly with the input size and the geographical level of detail / spatio-temporal resolution of the model.

Categories and Subject Descriptors
H.2.8 [Database Applications]: Data mining, Spatial Databases and GIS

General Terms
Algorithms, Performance

Keywords
Congestion Detection, FCD, Trajectory Data Mining, Intelligent Transport Systems

1. INTRODUCTION
Congestion is a major problem in most metropolitan areas. The relevant definition and the correct detection of congestion is a prerequisite of any congestion modeling and service that tries to accomplish congestion-related intelligent traffic management tasks, e.g., congestion prediction, congestion notification and congestion-avoidance routing. Given the current and growing availability- and the task-specific utility of Floating Car Data (FCD), this paper proposes a novel definition for congestion, a grid-based, time-inhomogeneous model and method for the detection of congestion, and a simple but effective, scalable implementation of the method using off-the-shelf relational database technology. Empirical evaluations show that 1) the spatio-temporal distribution and clustering of the detected congestions are reasonable given the common notions about where congestions present themselves and how they evolve in space-time, and 2) the proposed method and its implementation using high-level declarative languages is computationally scalable w.r.t. the input size and the spatio-temporal resolution of the model.

The unique features of the proposed model and method as well as the contributions of the paper are as follows:

(i) Grid-based model: The proposed grid-based modeling without loss of generality has several advantages. First, it does not necessitate the need for complete and accurate road network information and map matching. Second, the model can easily be scaled to any geographical (spatio-temporal) level of detail.

(ii) Representation of direction: The proposed model effectively represents the direction of movement and flow within the grid-based framework.

(iii) Time-inhomogeneous model: The proposed method defines and extracts parameters for normally observed traffic conditions for different days of the week and different hours of the day.

(iv) Novel congestion definition: The proposed method defines a grid cell to be congested if a) there is enough evidence for this, b) the evidence is reasonably unanimous, c) the evidence is statistically different from the normally observed traffic conditions, and d) the measured traffic conditions reflected in the evidence are relatively annoying for the drivers compared to the normally observed traffic conditions.

(v) Simple, scalable and portable implementation: The proposed prototype uses off-the-shelf RDBMS technology, which can be implemented in a few lines of SQL code and can be ported to Big Data processing frameworks.

The rest of this paper is organized as follows. Section 2 presents the related work. Section 3 formally defines the problem of directional congestion detection. Section 4 proposes a novel definition for congestion, a grid-based, time-inhomogeneous model and scalable stream-based method and implementation for the detection of congestions. Section 5 presents the evaluations and Section 6 concludes and points to future research directions.
2. RELATED WORK

In previous studies on congestion detection, traffic data have been collected from various sources including loop detectors [8], cameras [1], video [17] and GPS equipped vehicles [7,9,13,16,20,24]. In principle, congestion is detected on a road link by comparing the observed traffic condition with the light congested or free flow condition. Since the traffic condition would vary with road type, width, traffic signals, etc., there is no standard measurement to determine congestion [9] and the metric is frequently chosen empirically. The most commonly way to detect congestion is to simply define a uniform threshold on link travel speed, which is used in [7,8,18,24]. Other speed related metrics can be the ratio of average travel speed over the link’s speed limit in [9,13]. In [17], speed is used together with density to detect congestion. Human perception is introduced in congestion detection in [20] where instantaneous velocities reported directly from GPS device together with human perception of congestion levels are fed into a decision tree learning model to determine congestion level. Link travel time has also been used to detect congestion. In [1] a threshold is defined for the ratio of the observed- and the expected link travel time. In [16], link travel time distribution from two consecutive periods are compared to detect and measure congestion.

The results of congestion detection have been used for congestion prediction [7] and pattern analysis [1,13,24]. In [7], a Markov model is then built to predict the future flow based on current flow that is represented as density on a road link and estimated from GPS trajectory. Congestion pattern are analyzed by clustering congested links and periods according to some criteria. For example, in [1], congested links at specific periods are extracted as episodes, which are then clustered according to their spatial and temporal overlapping and finally groups of congestions are identified. Similarly, spatial and temporal distance is defined for clustering in [24] to detect recurrent congestion on a network. In [13], several other patterns of congestion are defined such as drop pattern indicating the traffic congestion is more serious than downstream objects and propagation pattern indicating one link is likely to be congested following another link.

The approaches used in aforementioned studies can be classified as network-based, which have a common limitation that a detailed network is usually required for the analysis. In practice, the network may not always be readily available. In [7], the network is drawn from traces of GPS probes, which can be arduous. An alternative solution is to use a grid-based model, which divides the study area into uniform grids. Grid-based models are both geographically and computationally scalable and can easily be applied to other study areas. Grid based models have been used to study 1) macroscopic regional mobility patterns such as origin destination patterns [18, 23] and popular regions [10] and 2) microscopic mobility patterns such as popular [21] and frequent [5] routes. However, to the authors’ best knowledge, the feasibility of grid based model for congestion detection and prediction has not been tested and is hence the subject of the present paper.

3. DEFINITIONS

Let $\mathcal{G}$ denote a grid with grid cells $g_1, g_2, \ldots$ with side length $glen$ that uniformly partition the 2D Euclidean space $\mathbb{R}^2$. Let the time domain be denoted by $T \equiv \mathbb{N}_0$. Let $O = \{o_1, \ldots, o_M\}$ be a set of moving objects, i.e., vehicles, that periodically send timestamped status reports which reference grid cells in the grid $\mathcal{G}$. Let a status report $r$ be one of three types (position_update, speed_update, or stopped) and contain the following information: the object ID of the object $o$ that submitted the report, the timestamp $t$ of the report, the grid cell $g$ that the object $o$ was inside at time $t$, the speed $s$ of object $o$ at time $t$, the direction $d$ from which object $o$ has entered grid cell $g$, and the type $T$ of the report. Let $SSR$ denote the time-ordered stream of status reports, i.e., an unbounded ordered sequence $(r_1, r_2, \ldots)$ of status reports such that for all $i > 1$, $r_i.t \geq r_{i-1}.t$. Let $SSR_{awin}$ denote the subsequence of $SSR$ that contains all the reports up to time instance $t_c$. Then the directional congestion detection task is defined as follows:

**Definition 1. Directional Congestion Detection:** Given a sequence of status reports $SSR_{awin}$ for a set of objects $O$ up to time $t_c$ and a temporal analysis window size $\Delta_{awin} \in \mathbb{N}^*$ find all directional congestions $(g, \text{dir})$, i.e., grid cell-direction combinations, such that the speeds of the objects that have entered grid cell $g$ from direction dir during the temporal analysis window $[t_c, t_c+\Delta_{awin}, t_c]$ is significantly and substantially bellow “normal”.

The above task can naturally be extended for the entire stream of status reports by performing the directional congestion detection task periodically.

4. CONGESTION DETECTION

As it is foreshadowed in the definitions, the proposed methodology adopts a grid-based discretization of space, which by changing the resolution of the grid allows the system to scale in terms of its computation cost (time and storage) and the geographical level of detail of traffic information that it manages. Given this grid-based framework, the outline of the directional congestion detection method is as follows.

1. Map the directional movement / flow of objects in $SSR$ to the grid-based framework.
2. Form tumbling windows over the mapped input stream and treat them as temporal analysis windows.
3. Extract *Current Directional Flow Statistics* (CDFS) from the Recent Trajectories (RT) that are within the current tumbling / temporal analysis window.
4. Incorporate the CDFS into *Historical Directional Flow Statistics* (HDFS) for different temporal domain projections.
5. Detect a grid cell $g$ to be congested from a particular direction dir if the current mean speed of vehicles that have entered the grid cell $g$ from the direction dir is significantly and substantially below the normal according to the temporally relevant HDFS.

The subsequent paragraphs present theoretical and implementation details of the above described processing stages.

4.1 Grid-based Directional Flow Statistics

While the grid-based discretization of space is inherently non-directional, the movements and flows of objects on the underlying road network is inherently directional. To capture the directional aspects of movement and flow, the proposed model defines directional movement and flow in terms
of a grid cell and its eight immediate cell neighbors. Specifically, the proposed model / method, without loss of generality, assumes that grid-based trajectories are spatially contiguous and defines the direction dir for entering a grid cell g from one of g’s eight immediate neighboring grid cells n I as the positive angle in degrees from n to g with respect to North. Using this definition, the proposed method for each observed grid cell-direction combination (g, dir) extracts three basic directional flow statistics from the grid-based trajectories of objects: the number-, the average speed, and the standard deviation of the speeds of objects that enter grid cell g from direction dir.

4.2 Stream Processing Model

The online processing of SSR, i.e., the detection of directional congestions, is facilitated by adopting a commonly used temporal sliding window model for streams:

Definition 2. Temporal Sliding Window Model: Given a stream of ordered time-stamped elements, \( S = \{(e_1, t_1), (e_2, t_2), \ldots\} \), and temporal sliding window parameters, \( \text{window size, } wsize \in \mathbb{N}, \) \( \text{and window stride, } wstride \in \mathbb{N}, \) the Temporal Sliding Window Model (TSWM) at every window slide time instance, \( tslide = tsc + wsize \) where \( a \in \mathbb{N}^0 \), processes the elements of the stream that are within the time interval of the window \( [tslide - wsize, tslide] \). Consequently, a TSWM is defined by the pair \( \text{TSWM} = (wsize, tslide) \).

Given the above definition of the TSWM, the stream of status reports SSR is processed in tumbling windows, according to \( \text{SSR} = (tsize, tslide, awin) \), as follows. At every window slide / tumble time the current tumbling window is equated to the temporal analysis window, i.e., \( tsize = awin \), and is used to perform the directional congestion detection task based on the current and the long-term, historical directional flow statistics. Given the fact that the directional flow statistics are derived from windows of size \( tsize = awin \), i.e., the statistics are implicitly assumed to be valid for a period of \( awin \) future, i.e., the period of the succeeding tumbling window is treated as the prediction horizon. Consequently, the above described processing model implicitly assumes a short-term congestion prediction model that predicts congestions to last during the prediction horizon, i.e., the succeeding tumbling window.

4.3 Incremental Historical Summary Statistics

To be able to efficiently extract long-term, historical directional flow statistics from the stream of status reports SSR, the proposed method takes advantage of the fact that the Current Directional Flow Statistics (CDFS) that are extracted from tumbling windows are based on non-overlapping subsets \( X \) and \( Y \) of SSR and hence can be combined in an incremental fashion according to the following equations [22]:

\[
\mu_{X \cup Y} = \frac{n_x \mu_x + n_y \mu_y}{n_x + n_y} \quad \text{(1)}
\]

\[
\sigma_{X \cup Y} = \sqrt{\frac{n_x \sigma_x^2 + n_y \sigma_y^2}{n_x + n_y} + \frac{n_x n_y}{(n_x + n_y)^2} (\mu_x - \mu_y)^2} \quad \text{(2)}
\]

where \( n, \mu, \) and \( \sigma \) denote the size-, mean- and standard deviation of a given sample. Using Equations 1 and 2, the CDFS are incrementally combined and compressed into long-term Historical Directional Flow Statistics (HDFS).

4.4 Temporal Domain Projections

Human mobility exhibits a large degree of regularity [19] that movement models try to capture. There are at least three different types of regularities in movement: temporal, periodical, and sequential [11]. To capture the temporal and periodical regularities that may arise in the flows of objects, the proposed method extracts HDFS for different values of the day-of-week and hour-of-day temporal domain projections.

4.5 Directional Congestion Detection

Let \( \dot{\mu}, \dot{\sigma}, \dot{\mu}, \dot{\sigma} \) respectively denote the CDFS and HDFS of a given grid cell g from a given direction dir. Then, the proposed method defines and detects the grid cell g as being congested from direction dir when all of the following four criteria are satisfied:

1. Sample size criterion: \( \dot{n} \geq \min_{veh} \)
2. Sample dispersion criterion: \( \dot{\sigma}/\dot{\mu} < \max_{cv} \)
3. Statistical power criterion: \( (\dot{\mu} - \mu)/\sigma/\sqrt{\dot{n}} < \max_{z} \)
4. Speed difference criterion: \( (\dot{\mu} - \mu)/\sigma < \max_{relspeed} \)

In other words, the criteria require that the recent status reports for grid cell g from direction dir are sufficiently many (1), and the reported speeds in them are in close agreement with one another (2), and (according to a z-test) significantly- (3) and substantially relatively (4) lower than the historical (“normal”) speeds. To account for the time-inhomogeneity of directional flow statistics, in addition to using the global (atemporal) HDFS, the above criteria are also separately evaluated using the HDFS for the day-of-week and the hour-of-day of the CDFS for (g, dir), and the direction congestion \( (g, dir) \) is detected if the criteria holds for either the global-, the day-of-week projected-, or the hour-of-day projected congestion model.

4.6 SQL-based Implementation

A prototype system that performs the described congestion detection task can be conveniently and effectively implemented using the power of off-the-shelf Relational Database Management Systems (RDBMS), e.g., PostgreSQL, and the simplicity of declarative programming languages, e.g., SQL. The paragraphs bellow explain the details of such a prototype implementation whose performance is empirically evaluated in Section 5. The aims of the detailed explanation are to illustrate the simplicity of the proposed solution and to highlight the portability of the proposed solution to Big Data processing paradigms that in a scalable manner support the basic relational algebra operators, e.g., MapReduce-based data processing frameworks like Apache\textsuperscript{TM} Hadoop\textsuperscript{®} [3] and main-memory, streaming variants like Apache Spark\textsuperscript{TM} [4].

4.6.1 Relational Database Schema

The prototype implementation stores recent trajectories and current and historical directional flow statistics in the following three database tables:

- RT = \( <\text{oid}, \text{dgid}, \text{spd}> \)
- CDFS = \( <\text{dgid}, \text{nr, mu, sig}> \)
- HDFS = \( <\text{dgid}, \text{nr, mu, sig}> \)

The information stored in the three tables are as follows. The RT table records the status reports that have been received from the clients during the most recent tumbling window. More specifically, a row in RT stores the information...
that at the time of the report the vehicle with object ID oid entered the grid cell with grid cell coordinates (gx, gy) in the direction dir—which is uniquely encoded as the integer concatenation of the three values (gx, gy, dir) into a directional grid ID dgid = gx_gy_dir—with the speed spd. In addition, the prototype implementation assumes that consecutive status reports from a given vehicle that refer to the same grid cell (i.e., an initial position_update status report is followed by one or more speed_update status reports) are aggregated into one status report that has the timestamp and speed information of the most recent speed_update status report. In an operational setting, all the information in RT can be calculated by the clients of the system (i.e., a software on a position aware computing device, e.g., navigation system or mobile phone, in the vehicles) provided some conventions for grid-based trajectory reporting.

The CDFS table stores for each directional grid ID dgid = gx_gy_dir the number of vehicles nr and the mean mu and standard deviation sig of the speeds of these vehicles that, during the current tumbling window, have entered the grid cell (gx, gy) in the direction dir.

The HDFS table stores long-term, historical aggregates of the statistical values of the CDFS table.

With the exception of the columns spd, mu, and sig, which are of type float, all other columns in the tables are of type int or bigint. Unlike in conventional relational table schema notation, in the above list the underlining denotes that the given columns have a hash index to speed up the join, selection, and aggregation operations during the processing of the queries that implement the directional congestion detection task. It is once more worth to emphasize the design choice for the column dgid. As explained before, dgid contains the unique concatenation of the planar / projected grid coordinates gx and gy and the direction of movement dir that results in an integer. Effectively, given that all subsequently described queries that implement the directional congestion detection task only involve equijoins on dgid, the 1-dimensional hash index on dgid efficiently indexes information about movement/flow in the 2-dimensional space.

### 4.6.2 Calculation of CDFS

As it can be seen in code listing SQL 1, the CDFS are computed based on simple aggregations of a single source of information, namely, the recent grid-based partial trajectories of the vehicles in table RT. SQL 1, as well as all SQL-code in the subsequent sections, show the bodies of SQL functions that at definition time the Query Planer and Optimizer (QPO) of the RDBMS compiles into executable query plans. During the processing of each tumbling window, the plan for SQL 1 is executed and its results are stored in the table CDFS, as described in Section 4.6.1. The logic implemented in SQL 1 is straight forward: the query groups all recent status reports in the current tumbling window by the

```sql
SQL 1 FUNCTION calc_CDFS()
1 SELECT dgid, count(*) AS nr, avg(spd) AS mu,
2 COALESCE(stddev(spd),0) AS sig
3 FROM RT
4 GROUP BY dgid;
```

directional grid ID dgid and for each dgid selects the corresponding CDFS that conform the table schema of CDFS and its application semantics that are described in Section 4.6.1. Note that CDFS can easily be constrained to be based on the most recent status reports of the objects by simply inserting a (WHERE seqnr = 1)-condition in the query, ultimately resulting in more recent but potentially less robust directional flow statistics, which rather than referring to the entire current tumbling window period, would refer more towards the end of that period. For the short-term directional congestion detection, the current implementation favors the robustness of statistics over their recency.

### 4.6.3 Incremental Calculation of HDFS

As described in Section 4.6.1, the table HDFS stores long-term, historical aggregates of the statistics of the CDFS table. These historical statistics are, according to the formulas in Section 4.3, incrementally updated in two phases: first statistics for previously observed directional flows are incrementally updated, then statistics for previously not observed directional flows are recorded. These two phases are illustrated in SQL 2. In particular, the UPDATE-query (Lines 1-9), according to Equations 1 and 2 updates the statistics (Lines 2-7) for the previously observed directional flows in HDFS for the directional flows that are also found in CDFS (Line 9). Subsequently, the INSERT-query, based on a left join between tables CDFS and HDFS selects the currently observed directional flows and statistics from CDFS that are not present (Lines 14-15) among the previously observed directional flows in HDFS and inserts them into HDFS.

```sql
SQL 2 FUNCTION ud_HDFS()
1 INSERT INTO HDFS (dgid, nr, mu, sig)
2 SELECT c.dgid, c.nr, c.mu, c.sig
3 FROM CDFS AS c
4 WHERE gh.dgid = c.dgid;
5 UPDATE HDFS AS gh
6 SET nr = (c.nr+gh.nr),
7 mu = (c.nr*c.mu+gh.nr*gh.mu)/(c.nr + gh.nr),
8 sig = sqrt((gh.nr*gh.sig^2 + c.nr * c.sig^2) / (gh.nr + c.nr) * (gh.nr + c.nr))
9 WHERE gh.dgid = c.dgid;
10 INSERT INTO HDFS (dgid, nr, mu, sig)
11 SELECT c.dgid, c.dir, c.nr, c.mu, c.sig
12 FROM CDFS AS c
13 LEFT JOIN HDFS AS gh ON (gh.dgid = c.dgid)
14 ON (gh.dgid = c.dgid)
15 WHERE gh.dgid IS NULL;
```

4.6.4 Calculation of Directionally Congested Cells

The CongCells(min_veh, max_vcx, max_z, max_relsdpd) function in SQL 3, provided the current- and the long-term, historical directional flow statistics (Line 2), as the proposed methodology suggests in Section 4.5, identifies all directional grid cells dgid (Line 1) where 1) the sample size criterion (Line 4), 2) the sample dispersion criterion (Line 5), 3)
the statistical power criterion- (Line 6), and 4) the speed difference criterion (Line 7) are satisfied.

4.6.5 Temporal Domain Projections

To preserve clarity, the above description of the SQL implementation of the prototype system does not contain the temporal domain projection aspects of the proposed model. However, these aspects have been implemented as follows. First, clients calculate and submit with each status report the day-of-week (dow) and hour-of-week (hod) projections of the timestamp of the status report. These temporal domain projected values are stored in- or are propagated throughout the computations to each of the three tables of the relational database schema, i.e., each table has dow and hod as int-type columns that in the case of the table HDFS is also indexed. All temporal domain projected, long-term, HDFS are stored in HDFS. The value of -1 for dow and hod are used to denote the “any” value for the domain projections, in general, and is used to distinguish between dow-projected-, hod-projected-, and global statistics. While the current directional flow statistic query (SQL 1) is modified to additionally return the current values of dow and hod from RT, the query for maintaining historical summary statistics (SQL 2) is extended to UPDATE and INSERT statistics for the current values of dow and hod, and the directional congestion detection query (SQL 3) is modified to contain additional conditions so that they relate temporally domain projected historical information that match the current values of dow and hod. Specifically, the directional congestion detection query combines the different temporally domain projected information as a disjunction (logical OR) in their respective decision criteria. That is, a directional congestion is detected if the statistical power criterion and the speed difference criterion are satisfied either based on the dow-projected-, hod-projected- or the global statistics.

5. EMPIRICAL EVALUATIONS

5.1 Test Environment

The empirical evaluations have been carried out on a personal laptop with Intel® Core™ i7-5600U CPU with 16 GB of main memory and a 512 GB solid state drive running a 64-bit Ubuntu 14.04 LTS installation with PostgreSQL 9.3.9.

5.2 Real-world Data Set

The proposed method is evaluated on a six day long (Mon, Tue, Thu, Fri, Sat, Sun) sample of the near real-time stream of raw GPS positions of around 11,000 taxis moving on the streets of Wuhan, China [14]. In this sample, positions of moving vehicles are read approximately every 20 to 60 seconds, totaling about 85 million records. The time-stamped readings include vehicle ID, location, speed and heading. After removing obvious outliers, sampling gaps of longer than 120 seconds are used to identify trips in individual trajectories. To adapt the raw GPS data set to the proposed framework, two consecutive Cartesian coordinate locations within a trip are linearly interpolated by approximating the interpolating line with a sequence of contiguous grid cells and corresponding speeds that are calculated by a modified Bresenham line algorithm [6]. After eliminating short trajectories (less than 300 seconds or 10 grid cells), approximately 2.26 million trips have been identified that are within an 18km-by-18km rectangular boundary that is centered at the mean coordinates of the measurements which approximates the city center. The identified trips have an average length of 1265 seconds and 82 grid cells and refer to 24783 100-meter grid cells. The resulting data set contains approximately 185 million 100-meter grid based status reports. A heat map of the trips is shown in Figure 1(a). The average length of trips in grid cells or the number status reports in the data set and the number of grid cells that are referenced therein increase approximately linearly with the inverse of Glen, which is also termed as the geographical level of detail or spatio-temporal resolution of the model[3].

5.3 Experiment Setup

The empirical evaluations are divided in two large groups of experiments: quality assessment (Section 5.4) and scalability assessment (Section 5.5) experiments. To be able to evaluate the quality of the detected congestions and scalability of the proposed method on a large spatio-temporally dense data set, depending on the given experiment, trip trajectories are temporally aligned so that they occupy the same relevant spatio-temporal region and yet are reasonably representative for the given experiment scenario. In particular, for quality assessment experiments and scalability assessment experiments w.r.t. the spatio-temporal resolution of the model the six days worth of trajectory data is temporally aligned so that the trajectories take place on the same “fictional” day at the time that is indicated by their original timestamp. This data alignment is referred to as the hod-alignment. For some scalability assessment experiments w.r.t. the number of vehicles that are managed by the system, trajectories are temporally aligned to start at the same time instance of the same “fictional” day. This data alignment is referred to as the fixed-alignment. Unless otherwise stated in a given experiment, the default parameter values in the experiments are as follows: temporal analysis window size / prediction horizon $\Delta_{\text{t}_{\text{win}}} = \Delta_{\text{t}_{\text{pred}}} = 60$ seconds, minimum number of current status reports $\text{min}_{\text{veh}} = 2$, maximum sample dispersion $\text{max}_{\text{cv}} = 0.5$, maximum negative z-score $\text{max}_{z} = -1.65$ (which for a left-sided z-test represents a significance level of $\alpha = 0.05$), maximum negative relative speed difference $\text{max}_{\text{relsppdiff}} = -0.5$.

5.4 Quality Assessments

5.4.1 Traffic and Congestion Indicators

The premise of the proposed methodology and the evaluations is that the selective directional congestion notifications need to be computer and evaluated for realistic / representative directional congestions. In lack of ground truth information on congestions, to verify the validity of

---

**SQL 3 FUNCTION CongCells(min_veh, max_cv, max_z, max_relsppdiff)**

```
1 SELECT c.dgid AS dgid
2  FROM HDFS AS gh, CDFS AS c
3  WHERE gh.dgid = c.dgid
4    AND c.nr >= min_veh
5    AND c.sig / c.mu < max_cv
6    AND (c.mu - gh.mu) / (gh.sig / sqrt(c.nr)) < max_z
7    AND (c.mu - gh.mu) / gh.mu < max_relsppdiff;
```
5.4.1 Spatial Distribution of Congestions

This premise, directional congestions are extracted using all the trajectories, hour-of-day projections, and default directional congestion detection parameters. Figures 1 and 2 respectively show the spatial and temporal distributions of traffic- and congestion indicators that are described as follows. TL measures the traffic level as the average number of vehicles that are present during a given period. NrC is the number of times a (non-directional) grid cell is congested during a given period. RelCL and AbsCL respectively measure the relative and absolute congestion level as the sum of the relative and absolute deviation in speed w.r.t. the normal values (i.e., congestion induced slowdown / delay) that vehicles in congested cells experience during a given period. For the examination of the spatial distributions the indicators are calculated for a 24-hour day (Figure 1 and Table 1), while for the examination of the temporal distributions the indicators are calculated for an average temporal analysis window during the given hour-of-day (Figure 2).

5.4.2 Spatial Distribution of Congestions

The spatial distributions of the traffic- and congestion indicators are shown in Figure 1. As all of the indicators have a long tailed distribution with basic distribution statistics given in Table 1, for better visibility of the spatial aspects of the congestion indicators the indicators are shown on a logarithmic scale in Figure 1. As the maps show, congested grid cells seem to be clustered and concentrated along main (well integrated) arteries and intersections of the road network. Although the clustering and concentration at intersections is suspicious and could potentially be due to falsely detecting red-light periods of signaled intersections, the scale (Figure 1(b)) and distribution (Table 1) of NrC provides reasonable insight for the refutation of this possibility. Namely, during the course of detecting grid cells as being congested from eight possible directions based 1440 one-minute long temporal analytic windows, the grid cell with the highest number of detections, from all possible directions, is detected 270 times compared to the 1440 \times 8 = 11520 possible times that a (non-directional) grid cell can be detected during a 24-hour day.

5.4.3 Temporal Distribution of Congestions

The temporal distributions of the traffic and congestion indicators in Figure 2 also seem plausible and even provides a rather interesting insight into the usage / operation of taxis. First, congestions mostly occur and affect vehicles during the days (7am to 7pm) even though a high level of taxi activity is observable during the late night and the very early morning hours. Second, the highest levels of congestion (both absolute and relative) occurs during two main peak periods: from 7am to 12am and from 2pm to 7pm. Third, while the traffic load of taxis is high and relatively even from 7am to midnight, there is a clear dip in the taxi traffic load from 5pm to 6pm which exactly co-occurs with the highest level of congestion. This co-occurrence is interesting because even though the number of taxis in operation is relatively low during this period, the number of taxis that are affected by the congestions is highest, i.e., the average number of vehicles that are within a congested grid cell at the time of detection is 523.58 compared to the overall average of 231.88 (not shown in Figure 2). A possible explanation for this is that both taxi drivers and potential customers are aware of how likely it is to get into a serious congestion during this period and therefore they both avoid treacherous trips where a large portion of the fare can potentially be wasted on idling and waiting in congestions.

5.4.4 Clustering of Congestions

One’s notion about how congestion evolves in space-time suggests that correctly detected congestions in grid cells

---

Table 1: Basic distribution statistics of traffic- (TL) and congestion (NrC, RelCL, AbsCL) indicators.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>TL</th>
<th>NrC</th>
<th>RelCL</th>
<th>AbsCL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum</td>
<td>2</td>
<td>5</td>
<td>5.00</td>
<td>17.00</td>
</tr>
<tr>
<td>Median</td>
<td>172.82</td>
<td>10</td>
<td>18.05</td>
<td>176.76</td>
</tr>
<tr>
<td>Mean</td>
<td>7288.3</td>
<td>19.69</td>
<td>39.48</td>
<td>383.22</td>
</tr>
<tr>
<td>90th percentile</td>
<td>95934</td>
<td>100</td>
<td>367.41</td>
<td>3059.38</td>
</tr>
<tr>
<td>Standard deviation</td>
<td>19679</td>
<td>20.88</td>
<td>67.42</td>
<td>625.92</td>
</tr>
<tr>
<td>Maximum</td>
<td>291910</td>
<td>270</td>
<td>1149.2</td>
<td>15226</td>
</tr>
</tbody>
</table>

---

The phrase “vehicles in congested cells experience” means that the relative and absolute speed deviation in a congested grid cell is multiplied by the number of vehicles that are in the grid cell during the detected congestion.
should form relatively contiguous spatio-temporal clusters. The examination of the temporally- and spatially projected spatial and temporal distributions of congestion indicators does not allow one to detect or verify any such spatio-temporal clustering. Additionally, for the detected congestions to be truly interesting, one would expect their distribution and clustering to be statistically significantly different from the underlying population, i.e., the density of vehicles which also tends to be highest along the main arteries and intersections of the underlying road network. Consequently, as a final test to validate the representativeness of the detected congestions, the strength and statistical significance of the spatio-temporal clustering of the detected congestions are assessed by calculating the Mantel statistic (a measure of space-time clustering of events) [15] of the detected congestions and, through Monte Carlo simulations, comparing it to the distribution of the statistics that is adjusted for the inhomogeneity of the distribution of the underlying background population, i.e., the spatio-temporal density of vehicles, using the general framework of [12]. In particular, directionally congested grid cells are treated as spatio-temporal events and for a set of events \( E \) the Mantel test statistic \( M = \sum_{i \in E} \sum_{j \in E} X_{ij} Y_{ij} \) is calculated, where \( X_{ij} \) and \( Y_{ij} \) respectively measure the spatial (i.e., grid-) and temporal distance between events \( i \) and \( j \). To adjust for the inhomogeneity of the distribution of the underlying population and assess the statistical significance the test statistic of the detected 19753 directional congestions, the distribution of the test statistic for the underlying population is estimated by randomly drawing 100 equal-sized subsets of space-temporal events from the 185 million status reports. The so simulated test statistic for the underlying population has an approximately normal distribution that has a mean value of 1.2998 \times 10^5 and a standard deviation of 3.9890 \times 10^5. In comparison, the test statistic for the detected directional congestions has a value of 1.3079 \times 10^{11} which has a \( P \)-value of 0.021026 for the one-tailed, one-sample \( z \)-test that is significant at \( \alpha = 0.05 \). These results show that the spatio-temporal clustering of the detected directional congestions is significantly weaker (i.e., has a higher test statistics which represents that events have a larger scattering) than one would expect if the events were randomly generated according to the distribution of the underlying population.

5.5 Scalability Assessments

In two large sets of experiments the scalability of the proposed model, method, and implementation has been tested for varying values of the two most important parameters: the input size (i.e., the number of vehicles that the system manages) and the geographical level of detail / spatio-temporal resolution of the model. To examine the scaling behavior w.r.t. the input size the fixed-alignment data set is used and congestions are modeled and detected using only global directional flow statistics. To examine the scaling behavior w.r.t. the spatio-temporal resolution of the model the hod-alignment data set is used and congestions are modeled and detected using both global- and hod-projected directional flow statistics. In both sets of experiments, scalability is measured in terms of the time and the storage (i.e., number of rows in tables) that different computation phases use. With the exception of the HDFS storage measure in the spatio-temporal resolution scalability experiment, where the measure refers to the total number of global- and hod-projected HDFS that are extracted during the course of the 24-hour long fictional day, the time and storage scalability measures refer to the results of processing an average temporal analysis window.

Figures 3(a) and 3(b) show the results of the input size scalability experiments. It is clear that in the examined range of input sizes, the computational time and storage scale linearly with the number of vehicles. This linear trends can be reasonable extrapolated for larger number of vehicles, although, the processing time and storage for updating HDFS and detecting congestion (which are mainly pattern dominated tasks) are expected to grow slightly slower than linear with the number of vehicles provided that the vehicles occupy the same spatio-temporal region and the total number of observed HDFS saturates as the vehicle density increases. Provided these trends and the 60-second real-time processing limit that is dictated by the size of the temporal analysis window, the proposed global HDFS based system is expected to be able to manage approximately 60/5 \times 0.2 = 2.2 million vehicles. It is also clear that the total processing time (5 second for 200 thousand vehicles) is dominated by the load time (4 seconds). This bottleneck however can likely be reduced using main-memory based stream processing frameworks and is as planned future work. Such improvements can potentially allow the global HDFS based system to scale far beyond the limits of this prototype.

Figure 3(c) and 3(d) show the results of the spatio-temporal resolution scalability experiments. Again, approximately linear processing time and storage scaling behavior is observable w.r.t. the spatial resolution of the model. This is surprising at first, as one expects that the number for directional flow statistics should grow quadratically with the resolution given the quadratically increasing number of cells.
in the grid. The unintuitive linear trend is due to the fact that the vehicles move on the underlying linear road network that occupies only approximately linearly as many grid cells as the resolution is increased. The processing times of the systems are impressive even for high resolutions (33.3 meter grid cells) when the total number of hod-projected HDFS that the system manages is close to 10 million. Given the linear trends and the fact that in the hod-alignment the number of vehicles in an average temporal analytical window in 54 thousand, the hod-projected HDFS based system (discounting the load time) is expected be able to manage in real-time 2 million and 700 thousand vehicles for $glen = 100$ meters and $glen = 33.3$ meters, respectively.

6. CONCLUSIONS AND FUTURE WORK

This paper proposed a grid-based, time-inhomogeneous model, method, and a simple, effective, and portable SQL-implementation of the method for the detection of congestion from large FCD streams. Empirical evaluations on millions of real-world taxi trajectories have shown that 1) the spatio-temporal distribution and clustering of the detected congestions are reasonable and 2) the method and its prototype implementation scale linearly with the input size and the geographical level of detail / spatio-temporal resolution of the model.

Future work is planned along several directions. First, the quality of the detected directional congestions will be further analyzed using an interactive, spatio-temporal visual analytic tool that is under development. Second, the detected directional congestions will be used to devise local and global holistic congestions models. Third, a road network based implementation and evaluations of the method will be conducted. Finally, the method will be implemented in a main-memory based stream processing framework.

Acknowledgements

Although the present research departs from Rui Zhu’s master’s thesis that the present author supervised [25], it was solely performed by the present author and represents a significant improvement in all aspects over its departure. The provision of trajectory preprocessing utilities is gratefully acknowledged to Christian Borgelt. Helpful discussions on the definition of congestion are also acknowledged to Adrian C. Prelipcean.

7. REFERENCES