Journal of Computer Languages 53 (2019) 53–62

Contents lists available at ScienceDirect

Journal of Computer Languages

journal homepage: www.editorialmanager.com/cola/default.aspx

AtoMixer: Atom-based interactive visual exploration of traffic surveillance data

Guodao Sun[a,*], Yin Zhao[b], Dizhou Cao[a], Jianyuan Li[c], Ronghua Liang[a], Yipeng Liu[a,*]

[a] College of Computer Science and Technology, Zhejiang University of Technology, HangZhou, Zhejiang Province, China
[b] College of Information Engineering, Zhejiang University of Technology, HangZhou, Zhejiang Province, China
[c] Enjoyer CO., LTD, HangZhou, Zhejiang Province, China

ARTICLE INFO

Keywords:
Human-centered computing
Visual analytics
Visual query

ABSTRACT

Massive traffic surveillance data extracted from vehicle detectors such as cameras provide essential information for revealing urban traffic pattern. However, most existing tools only allow users to analyze the data in specific time periods and regions with particular requirements. In this paper, we work closely with traffic domain experts and investigate a novel way of reframing visual traffic analysis tasks into the combinations of various atom categorical/numerical features and visual presentation. The categorical features contain primitive attributes such as vehicle type, O/D status and driving direction, and the numerical features contains information such as vehicle frequency and speed. The combination of above features includes four basic operations, namely *and, or, xor* and *not* to support diversified user requirements. Basic and advanced visualization methods such as trajectory view and flow distribution view are provided to demonstrate the combination results. Through interactive assembling of various atom operations, analysts could derive different query conditions to meet existed and potential upcoming analysis requirements such as locating suspicious vehicles (e.g., fake plate vehicles). Furthermore, AtoMixer, a visual analytic system is developed to support spatio-temporal investigative tasks for traffic surveillance data. We evaluate the effectiveness and scalability of our approach with real world traffic surveillance data.

1. Introduction

The rapid deployment and development of various surveillance camera and computer vision technologies enable us to collect an increasing amount of structurized spatio-temporal data [1]. These data play an important role in decision making and problem solving in the field of urban transportation, induction, and control. However, due to the growing amount of data and ever-changing of analysis demand, investigating appropriate approaches to satisfy existed and unanticipated analysis requirements in time have proven challenging [2,3]. Providing users with proper and scalable visual analysis approaches to reveal existed and uncover unexpected patterns in spatio-temporal data is essential [4,5].

Although practically useful and necessary, satisfying existed user requirements and foreseeing future needs is a daunting challenge in empirical research [6,7]. For example, identifying suspicious fake-licensed car and verifying them is a regular work of traffic police, and mining and visualization of the surveillance camera data can assist them realize this requirement straightforwardly. However, in most visual analysis system, the design and implementation are proposed to meet specific tasks (e.g., identifying fake-licensed vehicles) or general ones (e.g., simply visualizing traffic flow), and current systems may fail to answer new added demands. Thus, it is natural to come up with a system that is scalable enough to satisfy unanticipated user requirements and offer appropriate visualizations. Nevertheless, it is challenging to do so because there are several major challenges to overcome. First, from the perspective of data mining, how to integrate different user requirements in a logical and seamless way into a system to generate a solution for new user requirements depends on meticulous analysis of existing requirements. Second, from the perspective of visual design, different requirements may correspond to different visualizations, how to visually present an accurate and insightful form for the requirements is challenging. Last but not least, how to explore existed and unanticipated requirements in an intuitive and interactive way is the third challenge we have to face.

In recent years, empirical studies have been conducted to examine the spatio-temporal patterns existed in traffic data such as taxi-data, vehicle surveillance data. These studies focused either on the

* Corresponding author.
E-mail addresses: godoor.sun@gmail.com (G. Sun), liuyipeng@zjut.edu.cn (Y. Liu).

https://doi.org/10.1016/j.cola.2019.03.001
Received 16 January 2018; Received in revised form 25 January 2019; Accepted 1 March 2019
Available online 09 May 2019

visualization of specific and existed tasks such as traffic flow analysis and traffic situation monitoring, or on improving the efficiency of processing traffic data such as trajectory data [8–11]. There are several limitations with existing studies. First, these studies merely target at specific and well-defined user requirements, and may not be able to answer unanticipated demands. Second, these studies are not scalable and intuitive enough to introduce new visualizations. Not only developers need to design specific visualization for new requirements, but also the target users need to get familiar with the input, output and pipeline of the new visualizations.

In this paper, we first investigate and summarize typical user requirements and analysis tasks in the traffic surveillance data analysis field. The investigation follows a participatory design process under a close cooperation with domain experts. The user requirements are abstracted into different element operations, i.e., atom operations including categorical atoms such as vehicle plate atom, and numerical atoms such as speed atom. These atom operations could be freely assembled into various forms to generate an essential subdataset for existed and unanticipated requirements. However, an intuitive and effective approach for interacting with above atoms and visualizing the result subdataset is missed. We resolve this problem by proposing AtoMixer, an atom-based visual analysis system that supports visual exploration and sense-making through a rich set of user interactions. With AtoMixer, users are allowed to interactively combine categorical atoms and numerical atoms freely to create tree-like visualization, and visualization atoms can be attached to the nodes of the trees to generate corresponding visualizations. In AtoMixer, we present typical and advanced visualization components such as heat maps, RadarFlow view and temporal view that could be integrated with above combinations to visually summarize the dynamics of traffic surveillance data and obtain insight from the data. Major contributions of our work are summarized as follows.

- We propose a query taxonomy on traffic surveillance analysis tasks, which splits each task into a combination of categorical atoms, numerical atoms and corresponding visualization atoms.
- We design and develop AtoMixer, a visual analysis system for interactive visual exploration of existed and potentially unanticipated analysis requirements of traffic surveillance data.
- We provide empirical findings based on real world traffic surveillance data to test the effectiveness and usability of the visual analytics system.

2. Related works

This section reviews related works on traffic data visualization and visual query language.

2.1. Traffic data visualization

Extensive efforts have been made to analyze and visualize various kinds of traffic data such as taxi data [11,12], vehicle surveillance data [13] and ship travel data [14]. Chen et al. distinguished traffic data into three categories by working modes of sensors in traffic vehicles or monitors installed along the roads, i.e., location-based mode, activity-based mode, device-based mode [15]. Location-based data can be captured by fixed monitoring device, such as surveillance cameras at a road cross, high way and tunnel entrance. When a vehicle passes, the equipment will not only record the video information but also can collect their vehicle, speed and direction information. These surveillance data allow analysts to investigate interesting regular spatio-temporal patterns or cause of the traffic accident, and many innovative approaches for visual analytics of surveillance data have been proposed. For example, Harald and Matthias designed a system called AlVis, which can increase situation awareness in the surveillance of road tunnels [13]. They employ three views including present state view, history view, and future view. With those views, AlVis can reduce the occurrence of unimportant traffic information to save time in identifying situations that may cause an accident by setting the priority for the traffic information. In addition to location-based data, trajectory data is one of the most common form of traffic data. Scheepens et al. design an interactive visualization of multivariate trajectory data with density maps [14]. Wang et al. present an interactive visualization system for visual analysis of urban traffic congestion, users can analyze traffic jams propagation by jam propagation graphs which is based on traffic speed calculation [16]. TripVista use three perspectives (spatial, temporal and multidimensional views) to help users find traffic patterns and abnormal behaviors [17]. AL-Dohuki et al. design a novel system named SemanticTraj [12], which allows user to create different queries by the descriptive terms to perform advance data exploration. Nivan et al. propose a new model which allows user to create query by combining temporal information, spatial information and pickup/dropoff information to filter taxi data [8]. In general, most of the existed work either focus on the visualization of raw traffic data with specific requirements [18,19], or on improving the data model to improve spatial-temporal traffic index and query efficiency [8,20]. Thus, we propose AtoMixer, which is scalable and can deal with different regular and unanticipated user requirements, and various traditional and advanced visualizations can be integrated with different requirements.

2.2. Visual query language

Designing intuitive and effective visual query language has been research hot spots [21–24]. Draper et al. presents an interactive canvas with radial user interfaces for querying multivariate data, and the visual operation will be converted into a SQL language and passed to the database [25]. DataMeadow [26] is presented to analyze multidimensional data with a visual query language, and provide filtering in high-dimensional space. D3 presents a low-level grammar-based language scheme to create customized visualization [27], while Vega-Lite presents a high-level grammar to describe data and generate interactive data visualizations [28]. In addition, assembling existing and basic visualization component (with term atom) can lead to advanced, complex and powerful visualization result [29–32]. Dörk et al. design VisGets, which can interactively create combinations of different query terms of web-based information [33] to achieve advance requirement. VisGets provides meta information of web resources and can visually filter the data. ScatterBlogs2 build atom-based visualization filter according to the information which are already available and allow user to customize appropriate query [34]. Claessen et al. created a atom-based visualization approach for creating axis-based visualization techniques FLINA [35]. FLINA enables users to combine numerical, categorical and tabular data to define a wide variety of different visualizations. Gratzl et al. propose a multiform visualization technique-domino [36]. Similar to FLINA, domino allow users to explore a combination of numerical and categorical data connected via various item types, while Domino can be applied to a broader spectrum of tasks by providing comprehensive tools to arrange, combine, and extract subsets. In this paper, we absorb the idea of the visual query language and integrate it with domain experts' knowledge and the regular tasks to explore the spatio-temporal patterns in traffic surveillance data.

3. System overview

3.1. Data description

The traffic surveillance data in this paper are collected with surveillance camera installed at the road cross or along a road. The surveillance data consist of 120 million records (one month), which take up 10GB storage in database. Each record contains following attributes: vehicle plate, which is recognized with computer vision technology, longitude and latitude of the camera, capture time when a vehicle passes by, vehicle color, instant speed, and the direction of vehicle. We use key-value database Mongodb[1] to store the data due to its efficient query operation in big data with the appropriate index. We employ 2dsphere indexes[2] in the Mongodb, thus the query time for the range of 2-dimensional geospatial space is significantly reduced.

Due to malfunction of the surveillance device, there may exist error data in the dataset, such as the vehicle not being identified, vehicle type and vehicle color identification error. We first perform a data cleaning process. The process deletes the records in which vehicle plate is not identified, and unifies the color and type in different record of same trajectory.

3.2. System pipeline description

The pipeline of our system comprises three parts, namely task analysis, data processing, and visualization, as illustrated in Fig. 1. The first step of the pipeline aims to study the basic and potential analytical tasks existed in the traffic surveillance data. Typical analytical tasks include traffic flow analysis, illegal behaviour mining and vehicle clue mining, which will be elaborated in Section 4. The next step of the pipeline cleans the surveillance data and create trajectory data with surveillance data. The amount of diffusion flow among different road crossing is also obtained in this stage. The visualization step is fed with the output of data preprocessing. In this step, a main component of our system, i.e., visual query analysis component, allows user to combine categorical, numerical and visualization atoms into a query to create data filters and visualization results. Different atomic combinations will lead to different result set and visualizations. The spatio-temporal visualization component is further fed with the output of the visual query analysis component. Various spatio-temporal visualizations such as RadarFlow view, stacked graph, and heatmaps allow user to analyze data from different aspects.

4. Analytical tasks

To better understand the essential and potential analysis requirement in the traffic surveillance data analysis field, we have conducted interviews with domain expert users who have engaged into the analysis of traffic surveillance data for years. The daily tasks of the domain users are to discover the potential illegal objects (e.g., fake-licensed vehicles), monitor specific objects (e.g., tracking a vehicle), and reveal and verify overall traffic running situation. Through computer data processing and data mining techniques, the efficiency of investigating illegal behavior, locating potential illegal clues and dealing with them can be improved. The final goal of the domain users is to understand the patterns, principles and trends of the urban traffic situations, and ensure the regularization, standardization and securitization of urban traffic. We follow a participatory design process under a close cooperation with domain experts, and collect and compile a list of analytical tasks from domain experts as follows.

R1. Statistical analysis of the traffic surveillance data. Users should be able to quickly grasp the basic statistical information of the data, which mainly covers four main aspects.

(1). A detailed trajectory record of a vehicle should be provided to the users (e.g., locate a vehicle to its occurring space and time) to help track illegal behavior.

(2). The overall distribution of the vehicle information in a city should be presented such as the number of city vehicles, the temporal changes of the city vehicle amount, the proportion between personal cars and public cars.

(3). There are many illegal driving behaviours in a city, thus an overall statistic of the illegal driving behaviours should be presented. For example, the distribution of illegal driving behaviours, illegal event-prone areas, and illegal vehicles of high frequency.

(4). In addition, all the data above rely on the good condition of the running status of vehicle detectors such as road cameras. Thus, the quality of vehicle detectors such as the distribution and temporal changes of data loss rate (compared to the amount of average data retrial) should be presented. The changes of weather or light may also affect the running condition of the cameras, thus temporal changes of the data quality with respect to different periods of the day should be provided.

R2. Traffic flow analysis Overall traffic flow patterns should be provided to help perceive the city movements to verify regular spatio-temporal patterns and reveal abnormal patterns, which mainly covers three main aspects.

(1). O/D (origin/destination) behaviour analysis. The domain users are interested in investigating the dynamic traffic flow (e.g., flow amount and direction) based on a specific location. For example, how do the traffic flow origins from a location evolve over time, and what is the distribution of the flow origins from the location?

(2). Regional and temporal traffic flow Analysis. The overall statistics such as the amount and temporal aspect of the entry/exit traffic flow within an area, the time periods when the peak flow happens are essential to understand city movements.

(3). Travel delay analysis. A color encoded index should be provided to represent the degree of travel delay to help communicate the instant traffic flow situation. The index could be computed with the ratio between current travel time and the one in a free-flow traffic state.

R3. Illegal traffic behaviours mining Illegal traffic behaviours mining are important to domain users since it can help identify potential illegal vehicles in an intelligent way, improve the work efficiency of police and administration, and help ensure the public safety. The most important task in illegal traffic behaviours mining is to find fake plate vehicles (i.e., a vehicle owned a plate that is not licenced by the police or has been licenced to another vehicle). With traditional methods, domain users can use computer-assist approaches to obtain potentially suspicious vehicles, and retrieve the trajectory and surveillance image of each vehicle to perform further verification, which is tedious and inefficient.

R4. Vehicle clue mining The mining of illegal traffic behaviours is to locate well-known illegal vehicles directly, however, some vehicles on the road may be identified suspicious due to their abnormal travel behaviours. Different abnormal travel behaviours lead to different clues. We collect typical clues needed to be identified through surveillance data as follows.

(1). “Night vehicles clue mining” is to discover the vehicles that are less frequent appeared during the day, but relatively active during the night. Domain users can compare these clues of night vehicles with the vehicles in the blacklist, to prevent potential illegal behaviour in advance.

(2). “High frequent vehicles mining” refers to the discovery of vehicles that is highly active in the urban areas, except the public cars such as taxis and buses. The overview of activity area should be presented to the domain users.

(3). “Accompanying vehicles mining” is to discover two or more vehicles that are travelling together with each other simultaneously. This clue is particularly useful in the analysis of team committing crime such as fraud and robbery.

(4). “Hidden vehicles mining” refers to the discovery of vehicles that existed before a time period, but disappears after that time or appear for very few times. This clue is particularly useful in revealing the escape behaviour of traffic accident vehicles.

(5). We also collect other analysis tasks such as non-local vehicles mining, only entering and no exiting vehicles mining, and only exiting and no entering vehicles mining. These tasks are all crucial for domain users to locate suspicious vehicles and focus on the travel patterns and

[1] http://www.mongodb.com.

[2] http://docs.mongodb.com/manual/core/2dsphere.

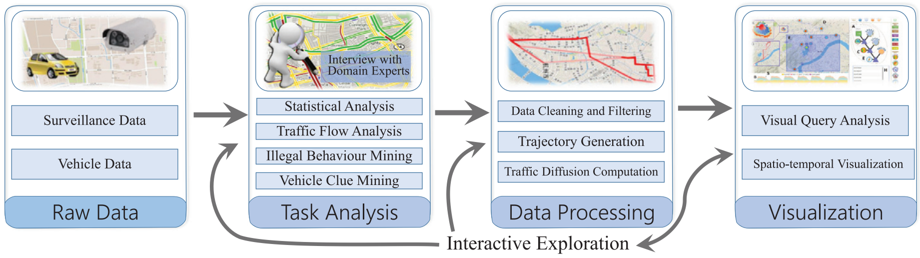


Fig. 1. The pipeline of the AtoMixer system, which includes three major parts: task analysis, data processing and interactive visualization.

activity area of them.

To address above tasks, we work closely with domain experts and extract their preferences and opinions on crafting an interactive visual analysis system with high availability. The above diverse tasks could easily complicate the analysis process, therefore, we need to decompose them into understandable and operable actions, which are more convenient for domain experts to follow and perform.

5. Visual design

This section introduces our visualization techniques. Fig. 2 shows our user interface which has three main views: AtoMixer view for presenting available categorical atoms, numerical atoms and visualization atoms, and allows users to select and combine them freely (Fig. 2(A)), a temporal visualization on the bottom-left part for showing the overall trend of selected attributes with stacked graph (Fig. 2(B,P)), and a spatial view on the up-left part for showing the map context as well as spatial visualization component for display the overall traffic distribution, traffic flow and vehicle trajectory (Fig. 2(D)).

5.1. Design considerations

Considering the visual design for tackling user requirements, we need to meet following criteria. Firstly, the domain experts do not have much knowledge on advanced visualization, thus, simple and intuitive visualization should be adopted into the system. Secondly, we must

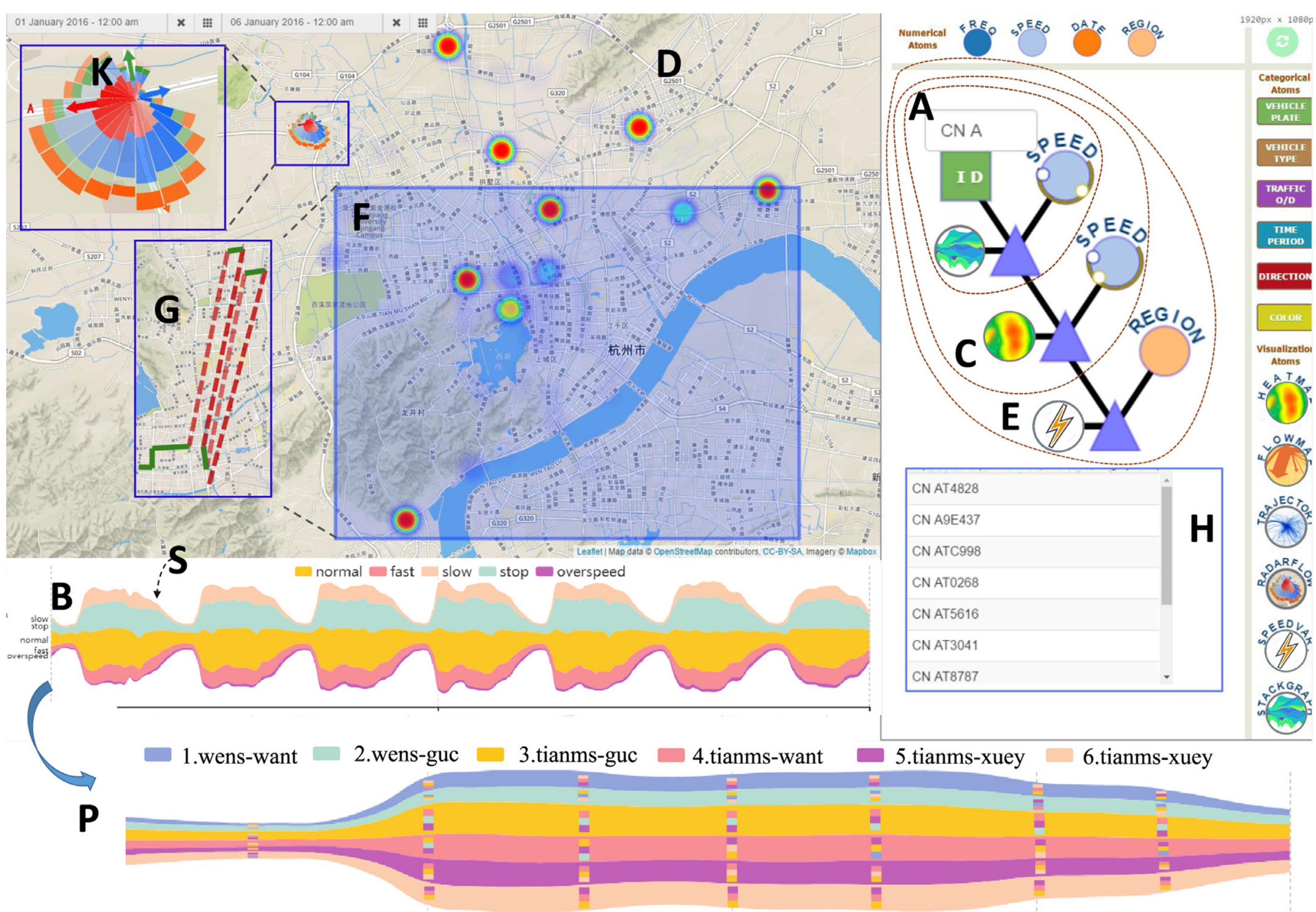


Fig. 2. The overview of our system, which mainly consists of three parts: AtoMixer view, Spatial view and Temporal view. AtoMixer view on the right part provide the space for combining different categorical, numerical and visualization atoms to perform complex analysis tasks. The Spatial view on the up-left part is used to present visualization such as RadarFlow, heatmaps to explore the spatial patterns. The Temporal view on the bottom-left part is used to present temporal visualization with categories determined by the combination results of AtoMixer. The patterns in this figure demonstrate the process of identifying suspicious fake plate vehicle through an iterative combination of various atoms.

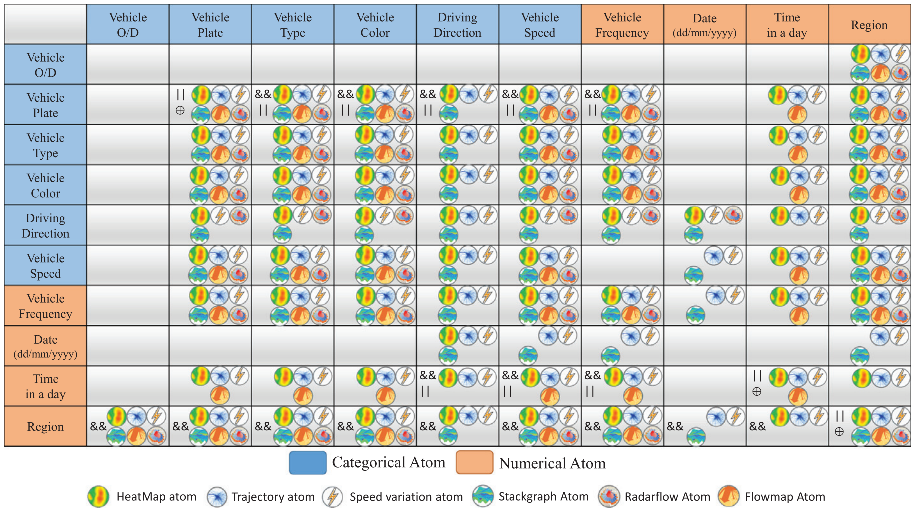


Fig. 3. The query taxonomy table of combinations of categorical, numerical and visualization atoms. For each cell, the cell in blue background represents categorical atom, and the one in orange background represents numerical atom. The && symbol in the cell means that two corresponding atoms could be perform *and* operation, the || symbol represent the *or* operation, and the ⊕ symbol is for xor operation. Blank cell means that the two corresponding atoms could not perform any meaningful logic operation. Possible visualization atoms for different combinations are attached in each cell. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

ensure that the interactive visual operation is flexible enough to meet a variety of analysis requirements. It requires us decouple above analytical tasks into basic domain-specific actions. Thirdly, we consider that when multiple actions are displayed together in the interface, the query structure should be clearly presented.

Therefore, to address the tasks in Section 4, we work closely with domain experts, and decompose the analytical tasks into different atomic operations, namely, categorical and numerical atomic operations. For categorical atoms, we identify six atoms, namely, vehicle O/D (origin/destination), plate, type, color, direction and speed selection atoms. For numerical atoms, we identify four atoms, i.e, vehicle frequency, date (in dd/mm/yyyy format), time (in a day) and region selection atoms. Different atoms could be combined to create various results. For example, combining category of vehicle O in vehicle O/D atom and region selection atom, we can obtain the traffic flow that origin from selected geographical regions. With the combination of vehicle plate atom (e.g., starting with a certain character) and frequency atom (e.g., 1000 times), we can obtain all the taxi (the plate of taxi always stars with “CN A”) with frequency of 1000 times.

Regarding the design consideration of the combination layout, we consider that the visual query view should be as intuitive as possible. The tree structure is a well-established form that can visually imply the hierarchical combination of different nodes. However, multiple combination of nodes or trees may lead to visual clutter, thus, visual clutter and link crossing reduction should be performed.

5.2. Atomixer visualization

Visual Encoding In the Section 4, we investigate the query taxonomy of traffic surveillance data analysis. In our design, we need to demonstrate an intuitive way for the users to freely combine the categorical atoms, numerical atoms and visualization atoms to satisfy exploration tasks. Integrating self-design expression language such as SQL language [37] is an alternative solution and have enough scalability, however, the domain experts express that the style and format of expression language is difficult for them to grasp and communicate with others. In addition, the domain experts do not have much knowledge on advanced visualization, and prefer simple and intuitive visualization methods. Thus, we proposed AtoMixer, a tree-like visual interaction component, that allows users interactively select different atoms to create different combinations and visualizations. In Fig. 2, a rectangle in the AtoMixer View is used to represent a categorical atom (e.g., vehicle plate, type), a smaller circle on the top is used to represent a numerical atom (e.g., frequency, time), and a larger node on the right represent a visualization atom. Users can freely drag a categorical/numerical atom into the empty space of AtoMixer, and the atom will stay in that space. If a categorical atom is dragged, users could set or enter the specified category. If a numerical atom is used, an arc will appear and surround the circle representing this atom to represents the range of corresponding attribute. The point on the arc can be dragged along the arc to select a specific range for this numerical attribute, users could also enter the corresponding value through the input box to ensure the accuracy of range setting. Then, a categorical/numerical atom can be dragged onto the atom to create a simple and direct logic operation (i.e., *and, or, xor* and *not*), and the result of the operation is represented by a triangle. Two triangles could be further combined into one child triangle to satisfy complex analysis tasks. A visualization atom can be dragged onto the triangle to provide specific visualization for the combination results, and the circle representing the visualization component is attached to the triangle to help better correlate the visualization and combination result.

However, we face several challenges when realizing the AtoMixer. First, not all of the categorical and numerical atoms could be combined to create meaningful results. For example, when performing an *or* operation with vehicle atom and region atom, the system will obtain the traffic data that origin from anywhere, which will return all the vehicles from database. This is not a meaningful operation, and leads to visual clutter and add burden to the server. Thus, we further establish a query taxonomy table (See Fig. 3) that provides meaningful logic operations between two atoms. For each cell, the cell in blue background represents categorical atom, and the one in orange background represents numerical atom. The && symbol in the cell means that two corresponding atoms could be perform *and* operation, the || symbol represent the *or* operation, and the ⊕ symbol is for xor operation. Blank cell means that the two corresponding atoms could not perform any meaningful logic operation. In addition, since not all of the visualization components can be applied to the combination results. We add possible visualization atoms for different combinations in the table (See Fig. 3).

Second, since users can freely drag and combine categorical and numerical atoms to create complex tree-like visualization, determining an intuitive and readable tree structure is essential. For example, in the

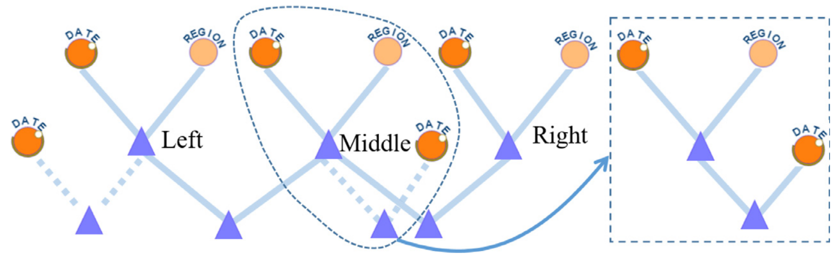


Fig. 4. Illustration of node adding algorithm to avoid link crossing. Adding *Date* node to *Middle* node leads to link crossing, thus a new separated tree is created besides the original tree, while adding to the *Left* node remains no intersection.

dashed line area of Fig. 4, the *Middle* node has a sibling node on both sides, thus if we drag the *Date* atom onto the *Middle* atom, it may lead to intersection with other links. To resolve this problem, we propose to create a separated tree besides the original tree. If the atom is dragged onto the *Left* or *Right* atom, which leads to none intersection, thus the algorithm directly attaches the new added node to the target node. Algorithm 1 illustrates the operating sequence of adding a node to an existing tree.

In Fig. 5, we illustrate an example of the combination process of different atoms to help discover hidden vehicles. As mentioned in the Section 4, hidden vehicles refer to the ones that existed before a time period, but disappears after that time or appear for very few times. The hidden vehicles are potentially the ones that escape from a road accident, and hide in unknown locations (usually, the car plate is not captured or cannot be recognized at the accident location). In order to find hidden vehicles after a road accident, we need to first analyze the hidden patterns behind this event. We assume that a hidden vehicle may appear and be captured by the surveillance camera frequently before the time of accident, and disappear or be captured by the camera for few times after that time point. Thus, we first drag the frequency atom to the AtoMixer view area and set the frequency to a larger value (e.g., f_1 times captured by the camera), and drag another date atom (the date range starts from the beginning of the dataset, and ends at the time of the accident, e.g., d) onto the frequency atom to create an *and* operation, which leads to Result A. This operation could identify all the vehicles that appeared and captured by the camera for at least f_1 times before that time. We then perform a further *and* operation of a frequency atom (e.g., with a smaller value such as f_2 times captured by the camera) and date atom (the date range starts from the beginning of the accident, e.g., d, and ends at current time), which leads to Result B. This operation could identify all the vehicles that appeared and captured by the camera for at most f_2 times after that time. Finally, we perform an *and* operation on Result A and B, which may help identify the vehicles that are hidden after an accident event at the time d. The combination of Result A and B can be further integrated with a region selection atom to perform spatial exploration, and visualization such as heatmaps and trajectory visualization could be further employed to display these vehicles on the map to perform deeper explorations.

5.3. Spatial visualization

Since we are dealing with traffic surveillance data, which is a kind of typical spatio-temporal data. Therefore, intuitive spatial visualization is needed to help users maintain spatial awareness of the data. Different combination of the atoms may lead to different results, thus we need to map them into related visualization methods. Due to the design goal of domain users' preference of simple and intuitive visualization, we investigate possible visualization for different combinations (see Fig. 3), and found most of them could be satisfied by existing traditional visualizations. Regarding the spatial distribution of vehicles, heatmaps is employed to present the information since it is the most widely used visualization for spatial distribution.

Regarding the trajectory view of a vehicle, we support two kinds of trajectory views, i.e., the dashed-line based speed variation view and solid-line based route view (see Fig. 6). With the speed variation view, we directly link each point of the vehicles which are captured by the camera to obtain a quick glance of the vehicle movement route. The color transparency of each link is used to encode the speed uncertainty of each moving behavior between two points. The larger the speed is, the higher uncertainty of the trajectory is.

In the formula (1) and (2) of Fig. 6, $\bar{s}$ and σ stand for the average and the standard deviation of the speed of vehicles moving between two certain locations, respectively. We adopt the confidence interval concept in probability theory to derive reasonable range of vehicle speed. The confidence interval is set to 95% in our scenario. Through checking the Normal Distribution Table we know that the $z_{\frac{\alpha}{2}}$, i.e., $z_{0.025}$ corresponds to 1.96. Thus, we can obtain the reasonable maximal speed c for vehicles in different areas, and consider that the speed smaller than value c is credible and can reflect the situation in the real world. When the speed is larger than value c, we adopt a sigmoid function to compute the confidence level of a trajectory. The input variable for this sigmoid function is $(s - c)/s$, which could imply the degree of uncertainty of a trajectory. The reason that we feed this variable into the sigmoid function is that we can enhance the discrimination degree of the input value, and map the input to the range of 0–1, which could be applied during the transparency mapping.

This visualization is particularly useful to help confirm certain interesting patterns like fake licensed plate vehicles. If real licensed plate vehicles and fake licensed plate vehicles occurred in different locations and at same time periods, the link connecting each occurring points may be swinging on the map, which seems to be abnormal to domain users. In addition, domain users wanted to grasp how a vehicle may travel in the city since surveillance cameras only capture a vehicle in a specific location, and the actual path of the vehicle is unknown. Thus, we propose route view, in which the actual travel path along city roads of a vehicle between two points is estimated with OpenSource Routing Machine API.[3]

For a specific location with surveillance camera, domain users also have the demand of understanding the temporal traffic that flows outwards from that location. The design of this visualization is challenging due to following reasons. First, to better maintain users' mental map, the designed visualization should be as close as to the location with surveillance camera on the map. Second, since the traffic may flow to different roads (e.g., four roads in a cross), and different directions may have different semantic meanings (e.g., entering or exiting downtown). Third, the traffic flow evolves over time, which adds difficulty to the visual design. During the close collaboration with domain experts, we have learned that they often use radar charts to understand data distribution. Thus, inspired by the radar charts, we propose a traffic flow visualization named RadarFlow (see Fig. 7 (a)), to present the daily spatio-temporal patterns of the vehicle data captured by a camera. In the RadarFlow, we use four color (i.e., red, blue, orange and green) to represent the traffic flow towards the four different roads. For

[3] http://http://project-osrm.org/ .

```
Input: nt refer to the new tree,et refers to the existing tree
Output: New combined tree
calculate en level position      // en refers the node of et which will be combined by nt
lp=level position                // lp refers position of en at its tree level
if lp==none then                 // none refers a position without any brother nodes
    Calculate the number of nodes on left and right sides of en
else if lnumber > rnumber then Place nt to the right
else Place nt to the left

if lp==right then Place nt to the right
if lp==middle then               // middle represents the position that is neither left nor right
    Extracte the combination of nt and et to other area to avoid crossing
```

Algorithm 1. AtoMixer Layout Algorithm.

example, in Fig. 7 (a), the red sectors represents the temporal traffic flow that travels towards west direction (represented by the sign A). Each sector represents the traffic flow towards one direction during one hour, and several sectors are stacked to intuitively demonstrate the total traffic flow at a road cross during one hour. In order to present the traffic flow direction and reduce visual clutter, we employ existed visual channel in the existed design, namely, the color. The transition of the color from low saturation to high saturation encodes the traffic direction. For example, the red sectors stands for the temporal traffic flow that flow towards road *A* (see the red arrow in Fig. 7 (a)). With RadarFlow, the spatio-temporal traffic flow of a road cross can be recognized and compared intuitively.

In addition, to help correlate the interaction among different road crosses and regions, we further enhance the RadarFlow with flow map. When users click a RadarFlow component at a road cross, the movement of traffic flow from this road cross to other regions is presented. The thickness of the edge represents the amount of flow between a source road cross and a target road cross, and visual clutter is reduced by bundling edges.

5.4. Temporal visualization

To reveal the overall temporal trend of traffic situation, we adopt a stacked graph to present the time-varying attributes of the different traffic statistics. However, determining the category that each strip represents is non-trivial since different combination of the atoms may lead to different results. Thus, we propose a heuristic three steps atom-based scheme to determine the category to be represented. When users drag a categorical atom *c* onto another atom (regardless of categorical or numerical atom), each strip is then used to represent temporal evolvement of each entry in categorical *c*. When users drag a numerical atom *n* onto another atom (regardless of categorical or numerical atom), the range of the numerical atom *n* is divided into *s* segments and each segment is represented by a strip in a stacked graph. By default, the parameter *s* is set to 5 to ensure that the visual clutter of stacked graph is reduced. For example, when a speed atom is dragged onto another atom, the numerical atom of speed is divided into 5 segments (see Fig. 2(B), i.e., the number of vehicles with velocity of overspeed, fast, normal, slow and stop), with each segment represented by a strip in the stacked graph and the height of each strip encodes the number of vehicles of corresponding category.

In addition, we extend the stacked graph to include more information with visual glyph. Fig. 2 (P) shows a extended stack graph view that presents the traffic flow visualization with rectangular glyphs attached on each strip. The glyph of three-color rectangles on each strip at each time stamp represent the top three in-coming sources of the traffic flow. The usage of the extended stacked graph with rectangular glyphs will be further introduced and explained in the case study section (Section 5.5.2).

5.4.1. User interaction

AtoMixer supports various advanced and basic interactions, such as panning, zooming, highlighting of atoms, maps and stacked graph.

Overview first and details-on-demand We strictly follows this visual exploration pipeline. When users select two or more atoms to create combinations, the system provides a temporal overview of selected categories with stacked graph to assist users in finding interesting and critical time points. Different combinations will lead to stacked graph with different categories to demonstrate multiple perspectives of the traffic data. Users can examine the detailed spatial patterns by directly clicking on the stack graph at corresponding time. In the spatial visualization, users can further investigate the detailed traffic distribution, flow and trajectories.

Comparative analysis support Users could hover over a sector in the RadarFlow vis to highlight the sectors with same time periods in different locations. When users have created a combination using

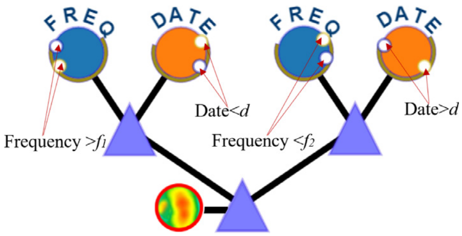


Fig. 5. An example illustrating the combination process of different atoms to help discover hidden vehicles. We assume a hidden vehicle may appear and be captured by the surveillance camera frequently before the time of accident, and disappear or be captured by the camera for few times after that time point.

different atoms, he/she can further drag more atoms to create combinations for other tasks. All the tree-like atom combinations stay in the AtoMixer view, and users can click on any of them and restore corresponding spatial and temporal patterns. Moreover, users can create several combinations with same categorical and numerical atoms but with different ranges and visualization atoms, and further switch among different combinations.

Detailed spatio-temporal patterns examination Users can click on a road cross, and a detailed RadarFlow will pop up to present a visual summary of temporal traffic flow towards different directions at a given time period. Users are allowed to enter a vehicle plate name or a kind of vehicle type, and detailed distribution, trajectory, and temporal occurrence can be examined in the coordinated view. We also provide a detailed list view to offer detailed information of the analysis results such as the instant speed, color, plate name of suspicious vehicles.

5.5. User evaluation

5.5.1. Participants and questionnaire

To evaluate the usefulness and effectiveness of the proposed system, we designed a questionnaire based on typical tasks achieved by our system (Fig. 8) and recruited nine volunteers to conduct a laboratory study. The nine volunteers all have basic experience in traffic data analysis or visualization. The questions are classified into three categories, namely, choosing right query combination for specific tasks, choosing propose description for specific query combination, and rating the flexibility and usability of AtoMixer. In the beginning of the laboratory study, we demonstrate visual encoding and basic usage of our system until they are familiar with each feature. When the participants finished all of the questions, we collected the answers they selected and the time they spent on each question. The entire questionnaire took around 15 minutes for each participant.

5.5.2. Results

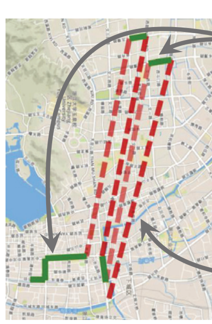
The results indicated that our system could well satisfy the need of typical traffic data analysis. For example, the accuracy rate of the task of searching for overspeed vehicles and night vehicles were 100%, and on average, took 30 (± 7.87) seconds and 34 (± 20.72) seconds respectively for participants to provide right answer on average.

However, when it came to some complex problems such as searching for most concentrated area of overspeed trucks or looking for nonlocal vehicles during a certain period of time in a specific location, the accuracy rate was relatively low. After discussing with the participants, we found that this issue was due to the lack of decomposition of question, users need to decompose the problem into sub-problems first, and then combine the results of each subset step by step, and on the other hand, the participants may need more time to get used to our system, thus some of them were confused about how to obtain right query combination. Task that took longest time to finish is how to find shared vehicles, participants spent 94 (± 21.48) seconds on average with 55.6% accuracy, we speculated that this task is not a straightforward combination of existing atoms, but also needs to understand the real world scenario. We also received overall rating of the Atomixer in terms of participants satisfaction which ranging from 1 (strongly disagree) to 7 (strongly agree). Based on the results, our system seemed to be easy to learn and use with an average rating 5.8 (± 1.20), and the query method receives highest scores of 6.1 (± 0.78) which respect to the flexibility of the system.

5.6. User feedback

To get further professional advice, we interviewed three domain experts in traffic management area. One expert PA is engaged in traffic regulation and control field for more than ten 10 years, and two (PB and PC) are technician in mining and discovering routine and abnormal traffic events. The analysis tasks in Section 4 are derived from above domain experts follows a participatory design process under a close cooperation with them. We first described the interface, different atoms and visual encoding, and user interactions in AtoMixer, then demonstrated the patterns that were observed in the case studies. Their feedback is summarized as follows.

Visualization Design The visual design of AtoMixer received positive feedback from three domain experts. They agreed that the tool is inspiring, interesting, intuitive, engaging and easy to grasp. They were very impressed by the free drag interaction features of different atoms, and the visualization responds to the combinations of different atoms. PA said that the user interactions are smooth and helpful, and particular like our effect of overlaying a thumbnail of corresponding visualization to the visualization atoms, which help them better relate the atom they may need to choose to the final visualization. Regarding the visualization on the map, they also agreed that despite being simple, the spatial visualization can intuitively convey spatial patterns and is effective in communicating the visualization result to others since the visual encoding is not complex. PB particularly liked the visual design of RadarFlow, and he added that, "*such component is convenient and valuable for me to understand the traffic situation at a road cross using this one single diagram. The direction of the traffic flow is not difficult to perceive with the color transition.*" However, PC mentioned that "*I found that for*



route view, displayed by green solid lines, shows real vehicle trajectories along the roads

$$c = \bar{s} + \frac{\sigma}{\sqrt{n}} \cdot z_{\frac{\alpha}{2}} \quad (1)$$

$$transparency = \begin{cases} 1 & 0 \le s \le c \\ 2 - 2 \cdot \dfrac{1}{1 + e^{-\frac{s-c}{s}}} & c < s \end{cases} \quad (2)$$

speed variation view, displayed by red dashed lines and encoded by transparency, shows the speed uncertainty of a trajectory

Fig. 6. Two kinds of trajectory views, i.e., speed variation view and route view, are supported in our system. The selection of the two views is automatically decided by above formula.

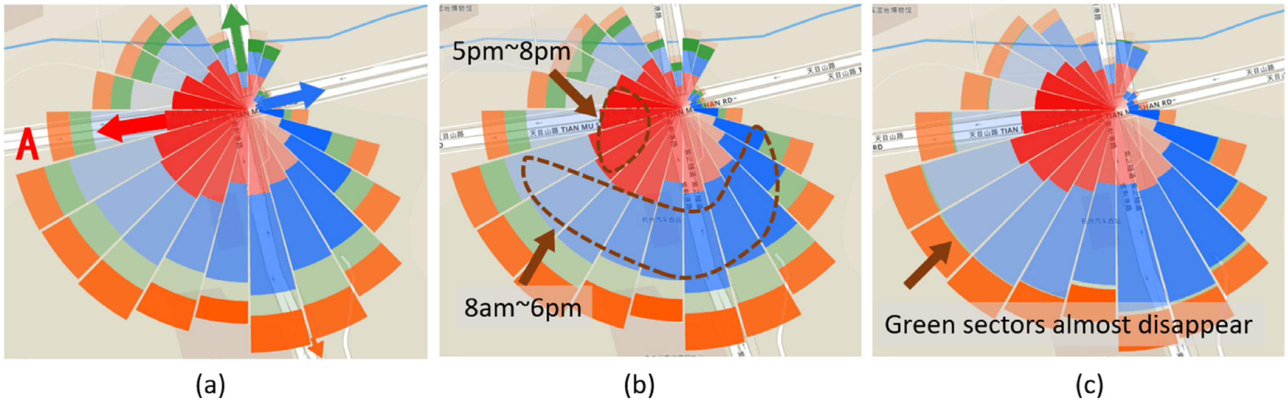


Fig. 7. RadarFlow showing the temporal traffic flow distribution in a road cross. (a) The illustration of the RadarFlow, four color (i.e., red, blue, orange and green) is used to represent the traffic flow towards the four different roads. (b) The main flow of this road is from west to east during day hours, and the flow from east to west increase significantly during evening hours. (c) The flow towards north (represented by the green sectors) almost disappears due to the temporary close of this road. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

some road crosses with huge imbalance of traffic flow, some sectors maybe too small to perceive, thus I need to exclude the direction that is easier to perceive and derive the direction for small sectors. Overall I like this design since it forms a sense that traffic flow to the direction when color transit to".

Usability All of the users confirmed the usefulness and effectiveness of the system, and they all expressed their eagerness for the system to be deployed into their real-world practice. PA mentioned that the process for exploring patterns in such way of "from overview to detail" is reasonable, and that the user interactions were smooth and easy to master. All of the users appreciated the effect of automatically dividing the range of a numerical atom into different segments to help identify temporal data distribution. PC noted that "*system is not only useful for data analysis, but also helpful for communicating their findings to colleagues or a wider audience, or even for case-based teaching of illegal event discovering*".

Suggestions The users provided valuable suggestions for improving the system. PB suggested that when selecting the range for a numerical atom, he is not aware of the distribution of the data (for example, the frequency distribution or speed distribution), thus he needs to explore the data by interactively dragging the small circle on the atom. Providing a chart of distributions of different numerical atoms may help him locate relatively accurate range. Three students suggested that the system should support the function of exporting the analysis data to text or excel files, such as temporal patterns of different categorical data. In addition, PC noted that "*It will be perfect that by default the system can provide a set of existed combinations for exploring routine tasks, so we can directly drag the combinations to the AtoMixer view, rather than combining the atoms again*".

6. Discussion

Our study has two important implications. First, we dig into the abstraction of the tasks in traffic data analysis. The abstract of the tasks is theoretically significant and practically necessary in various scenarios in traffic data mining. For example, in the field of traffic data cleaning, we may come across different kinds of expected and unexpected dirty data like data missing or position shifting. It will be tedious to develop corresponding components to address various analysis tasks. However, the query taxonomy of traffic analysis tasks and visualization atoms in this paper can be reused in other traffic data mining to help identify the patterns hidden in the data with different perspectives. Moreover, the framework of the abstraction of the tasks is easy to be scaled, and can be extended with more atomic tasks to satisfy future requirements.

Second, the proposed the visual analytic system, AtoMixer, is of great practical values in different applications. For example, the AtoMixer can act as a tool for other data analysis field, where the analysis tasks can be abstracted into the combinations of different atom operations like occurrence frequency and traditional visualization components. In terms of municipal administration, decision makers could employ our tool to study how urban infrastructure construction such as subway station or tunnel construction may affect the traffic flow. Last but not least, the AtoMixer can be employed by news reporters without much technical knowledge/skills to vividly communicate spatio-temporal patterns of traffic during evolution of urban construction process.

The proposed visual design strictly follows a participatory design process under a close cooperation with domain experts, who prefer simple and easy-to-use visualizations. This top-down visualization strategy works effectively for exploring the spatio-temporal patterns.

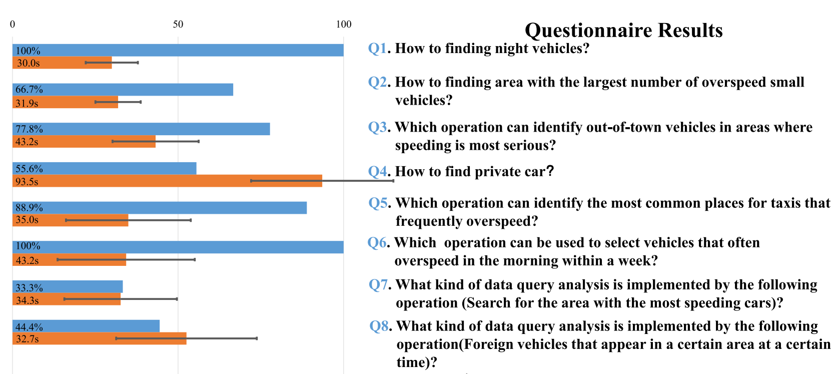


Fig. 8. Evaluation questions and analysis of the accuracy and time.

Although the visualization is intuitive, the underlying techniques are non-trivial (i.e., the layout generation in the AtoMixer view). The present work, however, has some limitations. For example, when domain users try to drag plenty of atoms to create complex analysis tasks, the AtoMixer View may become congested and users need to scroll up/down or left/right to navigate between different parts of the tree, which makes the exploration less efficient. Thus, a level-of-detailed presentation of the combined atoms may be needed to better help users recognize the exploration history.

7. Conclusion

In this study, we present AtoMixer, a visual analysis system that couples various categorical, numerical and visualization atoms to explore and analyze the spatio-temporal patterns of traffic surveillance data. The categorical and numerical atoms are derived from the analysis tasks proposed by the domain experts, and the visualization atoms including traditional and advanced displays such as RadarFlow View and Speed Variation View are integrated into the system to help identify and validate the patterns. AtoMixer provides free interactions that allow users to assemble various atoms into different query conditions, and explore regular and unanticipated analysis requirements. AtoMixer is scalable to integrate additional atoms to satisfy future needs.

In the future, we plan to provide an application program interface for our AtoMixer, thus developers from other fields can define and create their own atoms to analyze their data freely, visually and interactively. We will also collect more traffic datasets such as Taxi data and work with the experts to understand spatio-temporal diffusion in a more diverse fields.

Acknowledgments

The work is partly supported by Zhejiang Provincial Natural Science Foundation of China (LY19F020026), National Natural Science Foundation of China (61602409) and Teaching Reform Project of Zhejiang University of Technology (PX-62181545).

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.cola.2019.03.001.

References

[1] Y. Zheng, L. Capra, O. Wolfson, H. Yang, Urban computing: concepts, methodologies, and applications, ACM Trans. Intell. Syst.Technol. (TIST) 5 (3) (2014) 38.
[2] D. Liu, D. Weng, Y. Li, J. Bao, Y. Zheng, H. Qu, Y. Wu, Smartadp: visual analytics of large-scale taxi trajectories for selecting billboard locations, IEEE Trans. Visual. Comput.Graph. 23 (1) (2017) 1–10.
[3] R. Scheepens, C. Hurter, H. Van De Wetering, J.J. Van Wijk, Visualization, selection, and analysis of traffic flows, IEEE Trans. Visual. Comput.Graph. 22 (1) (2016) 379–388.
[4] G. Andrienko, N. Andrienko, S. Wrobel, Visual analytics tools for analysis of movement data, ACM SIGKDD Explorations Newsl. 9 (2) (2007) 38–46.
[5] G. Andrienko, N. Andrienko, U. Demsar, D. Dransch, J. Dykes, S.I. Fabrikant, M. Jern, M.-J. Kraak, H. Schumann, C. Tominski, Space, time and visual analytics, Int. J. Geog. Inf.Sci. 24 (10) (2010) 1577–1600.
[6] D. Keim, J. Kohlhammer, G. Ellis, F. Mansmann, Mastering the information age solving problems with visual analytics, Eurograph. Assoc., 2010.
[7] K.A. Cook, J.J. Thomas, Illuminating the path: the research and development agenda for visual analytics, Technical Report, Pacific Northwest National Laboratory (PNNL), Richland, WA (US), 2005.
[8] N. Ferreira, J. Poco, H.T. Vo, J. Freire, C.T. Silva, Visual exploration of big spatio-temporal urban data: a study of new york city taxi trips, IEEE Trans. Visual. Comput.Graph. 19 (12) (2013) 2149–2158.
[9] Z. Wang, T. Ye, M. Lu, X. Yuan, H. Qu, J. Yuan, Q. Wu, Visual exploration of sparse traffic trajectory data, IEEE Trans. Visual. Comput.Graph. 20 (12) (2014) 1813–1822.
[10] X. Huang, Y. Zhao, C. Ma, J. Yang, X. Ye, C. Zhang, Trajgraph: a graph-based visual analytics approach to studying urban network centralities using taxi trajectory data, IEEE Trans. Visual. Comput.Graph. 22 (1) (2016) 160–169.
[11] M. Lu, C. Lai, T. Ye, J. Liang, X. Yuan, Visual analysis of route choice behaviour based on gps trajectories, Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on, IEEE, 2015, pp. 203–204.
[12] S. Al-Dohuki, Y. Wu, F. Kamw, J. Yang, X. Li, Y. Zhao, X. Ye, W. Chen, C. Ma, F. Wang, Semantictraj: a new approach to interacting with massive taxi trajectories, IEEE Trans. Visual. Comput.Graph. 23 (1) (2017) 11–20.
[13] H. Piringer, M. Buchetics, R. Benedik, Alvis: Situation awareness in the surveillance of road tunnels, Visual Analytics Science and Technology (VAST), 2012 IEEE Conference on, IEEE, 2012, pp. 153–162.
[14] R. Scheepens, N. Willems, H. van de Wetering, J.J. Van Wijk, Interactive visualization of multivariate trajectory data with density maps, Pacific Visualization Symposium (PacificVis), 2011 IEEE, IEEE, 2011, pp. 147–154.
[15] W. Chen, F. Guo, F.-Y. Wang, A survey of traffic data visualization, IEEE Trans. Intell. Transp.Syst. 16 (6) (2015) 2970–2984.
[16] Z. Wang, M. Lu, X. Yuan, J. Zhang, H. Van De Wetering, Visual traffic jam analysis based on trajectory data, IEEE Trans. Visual. Comput. Graph. 19 (12) (2013) 2159–2168.
[17] H. Guo, Z. Wang, B. Yu, H. Zhao, X. Yuan, Tripvista: triple perspective visual trajectory analytics and its application on microscopic traffic data at a road intersection, Pacific Visualization Symposium (PacificVis), 2011 IEEE, IEEE, 2011, pp. 163–170.
[18] R. Scheepens, N. Willems, H. Van de Wetering, G. Andrienko, N. Andrienko, J.J. Van Wijk, Composite density maps for multivariate trajectories, IEEE Trans. Visual. Comput. Graph. 17 (12) (2011) 2518–2527.
[19] C. Tominski, H. Schumann, G. Andrienko, N. Andrienko, Stacking-based visualization of trajectory attribute data, IEEE Trans. Visual. Comput. Graph. 18 (12) (2012) 2565–2574.
[20] X. Wang, S. Zhao, L. Dong, Research and application of traffic visualization based on vehicle gps big data, Proceedings of the Second International Conference on Intelligent Transportation, Springer, 2017, pp. 293–302.
[21] B. Yu, C.T. Silva, Visflow-web-based visualization framework for tabular data with a subset flow model, IEEE Trans. Visual. Comput. Graph. 23 (1) (2017) 251–260.
[22] H. Blau, A visual query language for relational knowledge discovery, Comput. Sci. Dep. Faculty Publi.Ser. (2001) 105.
[23] R. Shadoan, C. Weaver, Visual analysis of higher-order conjunctive relationships in multidimensional data using a hypergraph query system, IEEE Trans. Visual. Comput. Graph. 19 (12) (2013) 2070–2079.
[24] J. Zhao, C. Collins, F. Chevalier, R. Balakrishnan, Interactive exploration of implicit and explicit relations in faceted datasets, IEEE Trans. Visual. Comput. Graph. 19 (12) (2013) 2080–2089.
[25] G. Draper, R. Riesenfeld, Who votes for what? a visual query language for opinion data, IEEE Trans. Visual. Comput. Graph. 14 (6) (2008) 1197–1204.
[26] N. Elmqvist, J. Stasko, P. Tsigas, Datameadow: a visual canvas for analysis of large-scale multivariate data, Inf. Visual. 7 (1) (2008) 18–33.
[27] M. Bostock, V. Ogievetsky, J. Heer, D^3 data-driven documents, IEEE Trans. Visual. Comput. Graph. 17 (12) (2011) 2301–2309.
[28] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, J. Heer, Vega-lite: a grammar of interactive graphics, IEEE Trans. Visual. Comput. Graph. 23 (1) (2017) 341–350.
[29] C. Stolte, D. Tang, P. Hanrahan, Polaris: a system for query, analysis, and visualization of multidimensional relational databases, IEEE Trans. Visual. Comput. Graph. 8 (1) (2002) 52–65.
[30] J. Mackinlay, P. Hanrahan, C. Stolte, Show me: automatic presentation for visual analysis, IEEE Trans. Visual. Comput. Graph. 13 (6) (2007) 1137–1144, https://doi.org/10.1109/TVCG.2007.70594.
[31] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, H. Pfister, Upset: visualization of intersecting sets, IEEE Trans. Visual. Comput. Graph. 20 (12) (2014) 1983–1992.
[32] W. Javed, N. Elmqvist, Explates: spatializing interactive analysis to scaffold visual exploration, Computer Graphics Forum, vol. 32, Wiley Online Library, 2013, pp. 441–450.
[33] M. Dörk, S. Carpendale, C. Collins, C. Williamson, Visgets: coordinated visualizations for web-based information exploration and discovery, IEEE Trans. Visual. Comput. Graph. 14 (6) (2008).
[34] H. Bosch, D. Thom, F. Heimerl, E. Püttmann, S. Koch, R. Krüger, M. Wörner, T. Ertl, Scatterblogs2: real-time monitoring of microblog messages through user-guided filtering, IEEE Trans. Visual. Comput. Graph. 19 (12) (2013) 2022–2031.
[35] J.H. Claessen, J.J. Van Wijk, Flexible linked axes for multivariate data visualization, IEEE Trans. Visual. Comput. Graph. 17 (12) (2011) 2310–2316.
[36] S. Gratzl, N. Gehlenborg, A. Lex, H. Pfister, M. Streit, Domino: extracting, comparing, and manipulating subsets across multiple tabular datasets, IEEE Trans. Visual. Comput. Graph. 20 (12) (2014) 2023–2032.
[37] C. Wu, Q. Zhu, Y. Zhang, Z. Du, X. Ye, H. Qin, Y. Zhou, A nosql–sql hybrid organization and management approach for real-time geospatial data: a case study of public security video surveillance, ISPRS Int. J. Geo-Inf. 6 (1) (2017) 21.