Multi-attribute interactive visualization of three-dimensional trajectory sets

Jing He[1], Haonan Chen[2,*]

[1]*School of Journalism and Communication, Tsinghua University, Beijing, China*

[2]*College of Geoscience and Surveying Engineering,*
China University of Mining and Technology-Beijing, Beijing, China

**Corresponding author: harman@student.cumtb.edu.cn*

Abstract

Rapidly advancing location-awareness technologies and services have collected and stored massive amounts of moving object trajectory data with attribute information that involves various degrees of spatial scales, timescales, and levels of complexity. Unfortunately, interesting behaviors regarding combinations of attributes are scarcely extracted from datasets. Further, trajectories are typically dependent on the environment of three-dimensional space, and another issue of interest to us is to preserve spatial-location visualization while guaranteeing the description of temporal information. Therefore, we developed a novel analytics tool that combines visual and interactive components to enable a dynamic visualization of three-dimensional trajectory multi-attribute behaviors. Under the context of spatiotemporal analysis, this approach integrates multiple attributes into one view to efficiently explore the attribute visualization problem of multi-attribute combination without over-plotting. To assess the feasibility of our solution, we visualized and analyzed multi-attribute information of moving object trajectories using a real mining truck dataset as a case study.

Keywords: Attribute combination; data processing; interaction; trajectory attributes; visualization.

1. Introduction

Trajectory attribute data are typically large-scale vector data of space-time or high dimensions, whose complex data features require powerful analytical techniques. The space-time cube (STC) is a common method for visualizing trajectory data; however, this approach may result in confusing displays when analyzing large amounts of trajectory data gathered over long periods. Tominski et al. (Tominski *et al.*, 2012) modified the absolute time on the z-axis into a time sequence, thereby effectively extending the STC. This approach smoothly resolves the problem concerning the overlap or intersection of routes; hence, it can be used for the analysis of big trajectory data. In graphic mapping systems, the consensus is that visual variables such as colors and textures should remain constant during interactive operations such as zooming, panning, and rotating to provide excellently rendering effects and to constrain attribute information. Based on this concept of color constancy, we propose a solution for analyzing the trajectory data involving attributes and attribute combinations to achieve the multi-attribute visualization of three-dimensional trajectories. Our research results are as follows:

- We cluster spatially similar trajectories, and describe their multi-attribute correlated behaviors in a trajectory set (TS);
- We establish an elevation model for multi-attribute trajectory sets, thereby extending the 3D representation of 2D stacking trajectory visualization into a 4D-like representation of 3D stacking trajectory visualization;
- We develop a visualization tool with strong interactivity, enriching processing methods for moving object trajectory (MOT) data in both static and dynamic visualizations.

2. Related works and data

2.1 Related works

Trajectory visualization has progressed from points to lines, from two to three dimensions, and from space-time data to attribute data. As a consequence, visualized point representations are now vivid and intuitive. For example, OpenDataCity (OpenDataCity, 2013) provides point-based animations of conference participants' positions based on records of wireless network signals; however, the approach is generally not suitable for comparative analysis. In contrast, visualizations using lines facilitate data analysis, but the excessive storage of data and increasing numbers of attribute types raise the requirements for such visualization approaches. The MobilityGraph designed by Von Landesberger et al. (Von Landesberger *et al.*, 2015) displays MOT data with a long span in a space-time diagram that is capable of reducing the disorder within the data but applies only to high-quality data. The STC is a classic approach for displaying spatiotemporal data in which the motion states of individual trajectories concerning high-speed motions, halts, encounters and separations can be observed. Zhang et al. (Zhang & Lin, 2019) chose a rational trajectory segmentation strategy and displayed trajectories in space-time cube. Filho et al. (Filho *et al.*, 2020) studied the space-time cube in the field of immersive Analytics and plan to replace their methods with real-world space-time datasets. This approach compares trajectories without data quantity limitation, but cannot represent multiple attributes.

It is uncommon to witness trajectory visualizations for multiple attributes over a decade. Huang et al. (Huang *et al.*, 2016) abstracted taxi trajectories into a traffic graph where a vertex represents a street or a region and an edge indicates taxi traffic between streets or regions; Al-Dohuki et al. (Al-Dohuki *et al.*, 2017) summarized trajectories by transforming them into text with reverse geo-coded POI name and speed meta data. However, all of these studies are concerned with the intermediate locations between origins and destinations. The OD connections are hard to perceive due to the data abstraction, aggregation, or the high level of detail. Pei et al. (Pei *et al.*, 2018) suggested an urban traffic visual analysis based on bus sparse trajectories to explore multiple attributes in trajectory data and display an overview of massive data. Based on a statistical view of small multiples, Zhu et al. (Zhu *et al.*, 2020) designed a multi-view combination visualization approach to depicting the spatial and temporal distribution of trajectory data. Such versatility and flexibility help analysts focus on various aspects of the data but bring more comprehension the burden to users.

Reviewing relevant works conducted worldwide, our research objectives lay on integrating multiple attributes into a single view to accurately and efficiently explore visualization issues concerning multi-attribute combinations of three-dimensional TSs in the corresponding space-time context.

2.2 Data and preprocessing

Tominski et al. (Tominski *et al.*, 2012) proposed that the space-time and attributive methods (e.g., space aggregation (S), time aggregation (T), and attribute aggregation (A)) of aggregating movement data can be applied not only to a single dimension but also to multiple dimensions (e.g., space-time aggregation (S×T), time-attribute aggregation (T×A), space-attribute aggregation (S×A), and space-time-attribute aggregation (S×T×A)). In this paper, multiple attributes are denoted as ”A+”. The interrelationships among the S, T, and A+ components in MOT data constitute multi-attribute behavior concerning space and time: S×T→A+. The issues related to this approach are as follows:

- *Behavior description.* Monitor the distribution of A+ values across the entire or partial S and T and characterize the A+ behavior, such as the collaborative changing trends and common sharp variations, within a specific space-time region.

- *Behavior search.* Search for outliers of A+, detect occurrence areas of specific behaviors of interest to analysts, and locate them in S and T. For example, when the attribute values (direction, speed, acceleration, etc.) of all the vehicles in an area of a certain road are simultaneously anomalous, a traffic accident may have occurred in that area.

- *Behavior comparison.* Compare the A+ behaviors in different areas of S over different intervals of T or in different subsets of TSs. For example, compare the vessel speeds and load behaviors between spring and winter routes.

- *Behavioral combinations.* Observe the A+ behaviors of different TS subsets that occur simultaneously in different areas of S or in different intervals of T; for example, in an area where a landslide or mudflow has occurred, the trajectories of all vehicles might show simultaneous speed reductions and directional changes.

To derive trajectories from enormous amounts of geospatial data and analyze the behavioral characteristics of 3D trajectories, a series of trajectory points collected by the data acquisition system are extracted to form segments, which are then used to create trajectory paths (Chen *et al.*, 2005). The trajectories obtained by this method share a simplistic data structure. Due to either subjective or objective reasons, however, there is a difference between the quality of the acquired data and the quality required for conducting research (Vrotsou *et al.*, 2014). In practical applications, data preprocessing is necessary, which includes coordinate conversion, data filtering, outlier removal, data interpolation, length adjustment, resampling, smoothing, and other operations (Gosink *et al.*, 2013). Figure 1 shows the data processing pipeline that illustrates the procedures of preprocessing and clustering raw datasets into several similar TSs.

Cluster analysis involves dividing data by resemblance, maximizing inter-cluster distances and minimizing intra-cluster distances (Guha *et al.*, 2000). Trajectory clustering is an extension of cluster analysis on space-time trajectories; its purpose is to classify time-space objects with similar behaviors into clusters according to their spatial or temporal similarities (Han *et al.*, 2011). We propose a trajectory spatial similarity metric to simplify the trajectory data overall while considering both timeliness and accuracy. The models for trajectory points, trajectory segments, and trajectories are respectively presented as Equatuins 1, 2, and 3:

$$tr = (id, X, Y, Z) \tag{1}$$

$$SubTr = (ID_{Sub}, tr_i, tr_{i+1}) \tag{2}$$

$$TR = (ID, SubTr_1, SubTr_2, \ldots, SubTr_n) \tag{3}$$

where id represents a unique identifier associated with the trajectory points, ID_{sub} indicates an identifier of $SubTr$, and X, Y, and Z denote the spatial geographic coordinates of trajectory points. The SubTr segment is calculated by interpolating two adjacent trajectory points tr_i and tr_{i+1}. Here, TR denotes an entire trajectory composed of a group of SubTr segments. The spatial similarity metrics we adopt consist of both evaluation indicators in Equatuins 4 and 5, where the former is expressed as follows:

$$D(TR_i, TR_j) = \frac{2l_m(TR_i, TR_j)}{l_s(TR_i) + l_s(TR_j)} \tag{4}$$

where D denotes the proportion of the cumulative length of the overlapping segments to the length of the entire trajectory, $l_m(TR_i, TR_j)$ indicates the length of similar segments between trajectory TR_i and TR_j, and $l_s(TR)$ represents the total lengths of trajectory TR.

In Equation 5, M expresses a fluctuational metric between TR_i and TR_j that is fixed by the respective fluctuational differences between TR_i and TR_j (the variance between the peak and the valley of the trajectory curve). For each point along the trajectories, $P(TR_i)$ and $P(TR_j)$ respectively indicate the maximum curve fluctuations of TR_i and TR_j, whereas $V(TR_i)$ and $V(TR_j)$ mark the minimum fluctuational values of TR_i and TR_j. In practical applications, when $D(TR_i, TR_j) = 0$, the two trajectories are completely dissimilar, and when $D(TR_i, TR_j) = 1$, the two trajectories are identical. Furthermore, when D lies in the range of (0,1), the smaller the value of M, the better the similarity is. The requirements that the above mentioned indicators should meet are determined based on the actual data conditions. For example, suppose we have several flight trajectories representing different routes, and we wish to determine their similarity (which is not high). In this case, we could loosen the requirements between adjacent trajectories. For example, if $D > 0.2$ and $M < 0.3$ are satisfied, the similar trajectories will be clustered into a TS. In contrast, if another TS exists that has different time and highly similar paths, we could set more stringent conditions to eliminate the segments with relatively large differences. In this case, we might determine that $D > 0.7$ and $M < 0.2$ are appropriate values.

$$M(TR_i, TR_j) = \frac{[P(SubTr_i) - V(SubTr_i)] - [P(SubTr_j) - V(SubTr_j)]}{l_s(TR_i) + l_s(TR_j)} \tag{5}$$

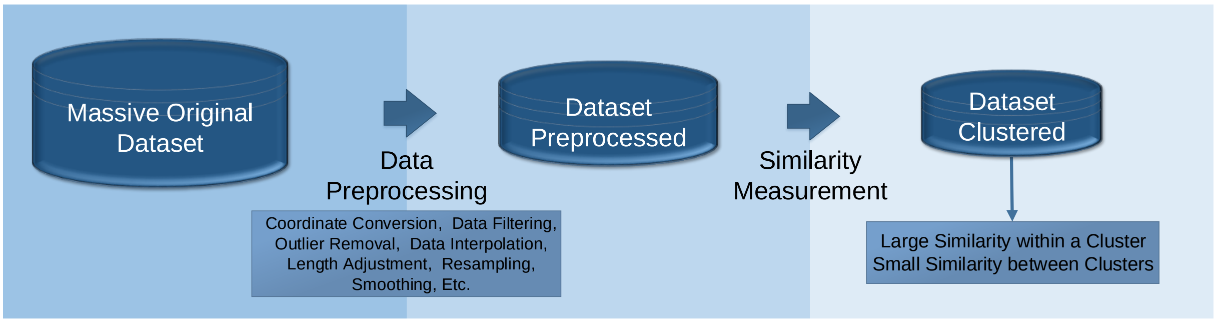

Massive Original Dataset
Data Preprocessing
Coordinate Conversion, Data Filtering, Outlier Removal, Data Interpolation, Length Adjustment, Resampling, Smoothing, Etc.
Dataset Preprocessed
Similarity Measurement
Dataset Clustered
Large Similarity within a Cluster
Small Similarity between Clusters

Fig. 1. Workflow of the processing module for trajectory data.

In the example shown in Figure 2, $D(TR_1, TR_2) = D(TR_1, TR_3)$ and $M(TR_1, TR_2) < M(TR_1, TR_3)$; thus, the similarity between TR_1 and TR_2 is stronger than that between TR_1 and TR_3. Using this approach, spatially similar trajectories are clustered simply and appropriately from massive MOTs.

3. Methods: multi-attribute spatiotemporal data visualization

We consider three-dimensional trajectory data that at least involve one variable as our major research target; their visualization can be divided into spatial data visualization, temporal data visualization and attribute visualization according to their data features, where geospatial data reflect the locations of spatial elements, temporal data describe the dynamic variations of trajectory events, and attribute data represent information other than the spatial locations of research targets over time. The differences in the correspondences between these three data types determine the data structures and storage methods.

3.1 Visualization of data features

In current visualization systems, data analysis is primarily limited to descriptive or exploratory analysis, such as browsing data distributions and detecting anomalous behaviors. Practical problems, such as discovering the causes of vehicle trajectory slowdowns and measuring the adoption of traffic flow planning, require answers with clarity, predictability, and causality. Moreover, trajectory data visual analytics in the context of the big data era involves interesting facts in terms of multiple attributes, especially when numerous trajectories traverse a specific spatial area, when numerous trajectories are collected during a specific time, or when multiple behaviors occur simultaneously. For example, by analyzing trajectories of a certain section of undulating road and visualizing combined attributes such as speed, vehicle model, traffic volume, we can also study traffic-load information, real-time speeding behavior, vehicle emergency responses, the most accident-prone vehicle models and other information. Visualizing multi-attribute data benefits the evaluation of current events and the prevention and optimization of future events.

3.1.1 Visualization of three-dimensional data features

As we process trajectory data with similar geometric paths, placing a 2D map into a virtual 3D space can smoothly address the trajectory-overlapping problem. The 2D map is considered as an interface to trajectory geospatial data, thereby supporting information access and exploratory activities. The MOT on a 2D map is a ground path; however, the movements of moving objects are predominantly dependent on 3D spatial environment, and the MOT in a 3D space-time domain is a space-time path whose spatial location is defined by both the ground position and the elevation (Ferreira *et al.*, 2013). In the geovisualization environment, the method that accurately maps trajectories by calculating and comparing distances between data and corresponding ground locations in 2D maps ignores elevation information. Consequently, this approach does not apply to 3D trajectories that apply to elevation-required ground objects and all flying objects. If elevation information is visualized above the trajectory ribbon, the fluctuation of the ribbon in the Z-axis direction will grow more drastic, the intuitiveness will suffer, and trajectory stacking will hardly be achieved. In this regard, we map the ground position of the 3D trajectory into the $X \times Y$ coordinate system of a 2D map and then perform linear regression

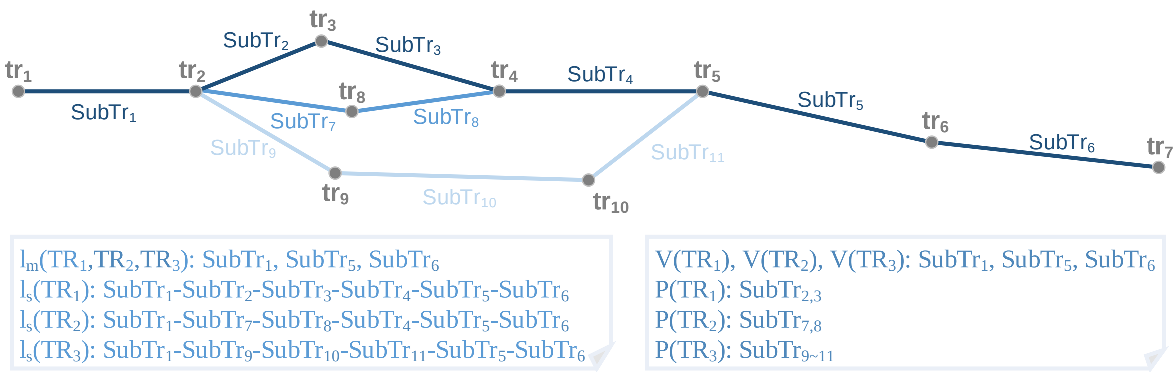

tr_1
tr_2
tr_3
tr_4
tr_5
tr_6
tr_7
tr_8
tr_9
tr_{10}
$SubTr_1$
$SubTr_2$
$SubTr_3$
$SubTr_4$
$SubTr_5$
$SubTr_6$
$SubTr_7$
$SubTr_8$
$SubTr_9$
$SubTr_{10}$
$SubTr_{11}$
$l_m(TR_1,TR_2,TR_3)$: $SubTr_1$, $SubTr_5$, $SubTr_6$
$l_s(TR_1)$: $SubTr_1$-$SubTr_2$-$SubTr_3$-$SubTr_4$-$SubTr_5$-$SubTr_6$
$l_s(TR_2)$: $SubTr_1$-$SubTr_7$-$SubTr_8$-$SubTr_4$-$SubTr_5$-$SubTr_6$
$l_s(TR_3)$: $SubTr_1$-$SubTr_9$-$SubTr_{10}$-$SubTr_{11}$-$SubTr_5$-$SubTr_6$
$V(TR_1)$, $V(TR_2)$, $V(TR_3)$: $SubTr_1$, $SubTr_5$, $SubTr_6$
$P(TR_1)$: $SubTr_{2,3}$
$P(TR_2)$: $SubTr_{7,8}$
$P(TR_3)$: $SubTr_{9\sim 11}$

Fig. 2. Spatial similarity metric for trajectory segments

analysis on all trajectory points to yield a trajectory regression line L. The axis perpendicular to the L direction is defined as an elevation axis or E-axis. Using this approach, the elevation information can be displayed in the trajectory ribbon without increasing the fluctuational range on the Z-axis. Eq. (3) provides a certain type of 3D trajectory denoted as $TR = (ID, SubTr_1, SubTr_2, \ldots, SubTr_n)$. Technically, this trajectory is plotted from a set of trajectory points after data processing. Therefore, the TR point set is denoted as $T = (tr_1, tr_2, \ldots, tr_n)$ (see Figure 3), in which a single trajectory point is indicated as $tr_i = (id_i, x_i, y_i, z_i), i = 1, 2, \ldots, n$. Thus, we have

$$\begin{aligned} tr_1(x_1, y_1, z_1) &\in T(tr_1, tr_2, \ldots, tr_n) \\ &\subset TR \\ &\subset TS(TR, TR_2, \ldots TR_n) \\ &\subset S(X \times Y \times Z) \end{aligned} \tag{6}$$

The projection of the trajectory points onto the $X \times Y$ plane is described as a point set denoted as $P = (P_1, P_2, \ldots, P_n)$, where $P_i = (x_i, y_i)$ is a single projection point. Consequently, we have

$$\begin{aligned} P_1(x_1, y_1) &\in P(P_1, P_2, \ldots, P_n) \\ &\subset S(l_1, l_2, \ldots, l_n) \\ &\subset S(X \times Y \times Z) \end{aligned} \tag{7}$$

If we draw an axis vertical to the regression line L as the E-axis and then create a vector e_i on the E-axis such that $|e_i| = z_i$, we can obtain an elevation point set $E(E_1, E_2, \ldots, E_n)$, where $E_i(x_i, y_i, e_i)$ is a single elevation point:

$$\begin{aligned} E_1(x_1, y_1, e_1) &\in E(E_1, E_2, \ldots, E_n) \\ &\subset S(X \times Y \times Z) \end{aligned} \tag{8}$$

Connecting all of the points in set E using a certain plotting function yields an elevation ribbon. In the same manner, other elevation ribbons can be plotted and stacked. The pseudocode utilized to extract all of the elevation values of T as a set $H = z_i \mid i = 1, 2, \ldots, n$ is shown in Algorithm 1. Since linear regression of P has time complexity $O(p)$, where p is the number of projection points, the computation of unit normal vector N and the coordinates of the elevation ribbons E also have time complexity $O(p)$.

In our elevation view, due to the perspective projection and the 3D rendering in the 3D scene, the height information of the trajectory could be too confusing to perceive, especially if users scan the visualized views with non-3D display tools. To offset this problem, we render the ribbons according to their respective elevations. A gradient color bar is applied to ensure a continuously undulating terrain, thereby yielding a scalariform 3D effect (Juřík *et al.*, 2017).

3.1.2 Visualization of attribute data features

Typically, trajectory data contain complex attributes and thus cannot be effectively represented by traditional visualization methods. It is also necessary to integrate different sources, scales or types of trajectory data to prevent potential misinterpretations caused by an incomplete display of information. We will present the attribute data features in a manner that combines static and dynamic visualizations, where static visualization is based on the visual mapping. Dynamic visualizations are embedded because

Table 1. Pseudocode of elevation ribbon creation

```
Algorithm 1: Elevation Ribbon Creation
def CreateElevationRibbon(Projection points $P(x_p, y_p)$, Elevation $H$):
    $L \leftarrow$ linear regression of $P$
    $N(x_n, y_n) \leftarrow$ compute the unit normal vector of $L$
    $E(x_e, y_e) \leftarrow P + H \times N$
    plot elevation ribbons with $E$
end
```

attribute features are likely to involve time-related visualization issues due to their time-varying characteristics.

- Static visualization

Visual analytics, which is a relatively new term (Andrienko *et al.*, 2010), can transmit different visual perceptions with visual variables. When acquiring, storing, transmitting, comprehending, analyzing, and applying trajectory data, visual variables serve as visual channels for exchanging information so that users can quickly and effectively grasp and control the process (Bezerianos & Isenberg, 2012). These variables can be freely combined, and their combinations often produce synergies. Furthermore, in trajectory visualization, superimposition typically combines data information in an aligned manner based on the same context into an integrated representation in one view, in which the visual information obtainable by users is no longer one-dimensional but multidimensional. We introduce layers to enhance the flexibility of visual variables, thereby providing a solution in which any attribute elements can be processed without damaging others, and the transparencies and sequences can be smoothly configured to alter the aggregation mode of a view (Lulli *et al.*, 2015). We also combine the superimposing technique with layer arrangement to produce an excellent visual effect for analyzing attribute combination (Waser *et al.*, 2010). Figure 4 shows a multi-attribute combination the view that consists of a series of superimposed layers carrying attribute elements. Attribute layers are built from simulation data: Figure 4a colors the segments with a specific rendering scheme, and Figure 4b maps the segment attribute data using different styles of line boxes. After the first two visualizations, Figure 4c superimposes both layers to generate a final combination view of multiple attributes.

By combining several attribute layers into an integrated system, users can build an internal story representation that integrates space-time data and attribute data. The pseudocode of this algorithm is shown in Algorithm 2. Suppose the trajectory set includes n trajectory points, the inner loop (line 3) will execute a total of n times. Since the time complexity of the statements in the inner loop is constant, the total time for this algorithm is $O(n)$. The space complexity of the inner loop is also constant, and thus the total space complexity is also $O(n)$.

In this algorithm, the *trajectory_set* denotes the TS to be analyzed, the *trajectory* is a single trajectory in the TS, and the *trajectory_point* is a single trajectory point along the trajectory. For each trajectory point along each trajectory, a decision is made concerning whether the attribute value of the point and its counterpart in the previous trajectory point is within the same interval; when they are within the same interval, the point is added to the plotting point set, which is eventually plotted by the same visual variable.

- Dynamic visualization

In situations where the visualized information amount of three-dimensional trajectory data becomes excessive and additional attributes are complex and varied, we tend to capture time-varying attribute information of interest in one view, which is geared to the topical evolution of trajectory events simulated by animated map. By configuring the continuity or discreteness of time, users can interactively select the time where one detail changes to another. This operation facilitates the inspection and perception of subtle changes or display dynamics in events. The dynamically interactive representation of attribute information that we designed introduces the principle of key frame interpolation into computer animation to dynamically display attribute information. We

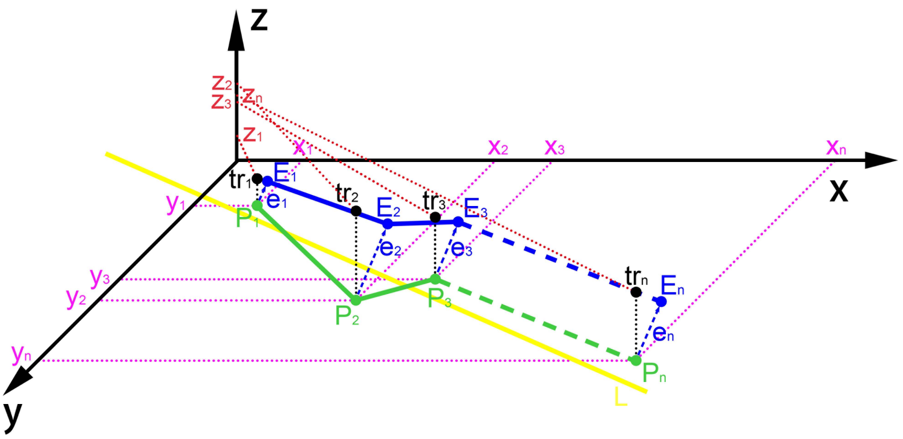


Fig. 3. Conversion of 3D trajectory elevation values. Here, (x_i, y_i, z_i) denotes the spatial location of the original trajectory point, while (x_i, y_i, e_i) maps the geospatial data of the converted trajectory point.

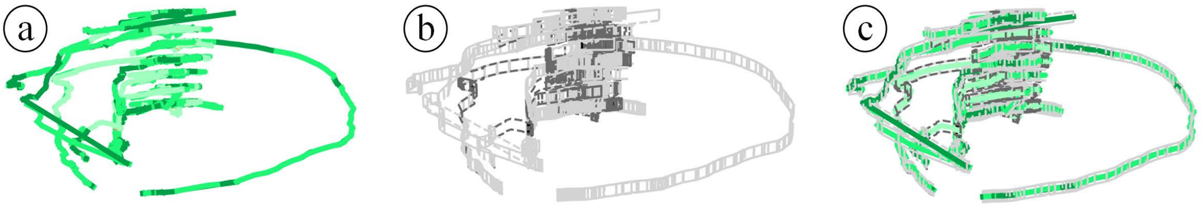


Fig. 4. Superimposing attribute layers. (a) single-attribute visualization with color characterization; (b) single-attribute visualization with line box characterization; (c) multi-attribute visualization by superimposing the color and line boxes from (a) and (b).

consider the spatial and attribute features of moving objects at different times as the key frames and in between frames are interpolated through transforming and mapping process. These key frames and their in between frames are played continuously in chronological order under certain constraint rules, simulating all of the animation information of the geometric primitives in the selected duration and controlling animation progress with time frames of rather short time intervals as time units. Figure 5 illustrates a time-series design model for dynamic-primitive layers. Here, the time difference between the start frame and the end frame (also known as the key frame) indicates the duration of the occurrence, while the temporal relationships between the key frames define the change sequence of the geometric primitives.

The transformation function between key frames requires temporal variations to represent the topical-evolution rate of changing the trajectory events. We designed a time-selection tool, namely speed-variable interactive time legend (SITL), for this task to provide interactive and dynamic support for multi-attribute data visualization. This timing widget adopts a multiscale control method that hierarchically processes the data sources to switch time scales. Moreover, the timeline of animation playback is controlled as a series of frames to facilitate debugging. The backward- / forward-frame button adjacent to the beginning point (B) / ending point (E) plays a single frame backward / forward—i.e., the successive actions of further decomposing the complex trajectory attributes in continuous key frames. SITL also serves as a visual speed control, which freely adjusts the playing speed and time scale to browse the visualization results in any frame (see Figure 6). For example, from a historical animation playback, we might expect to learn whether a particular event occurred, when it happened, and how long it lasted. Suppose it is known that the duration of the entire animation is one hour, and the duration of a specific event is only one minute. Moreover, the number of occurrences of that event is unknown. Playing the animation at a time scale of 1:1seems unacceptable

Table 2. Pseudocode of multi-attribute combination

Algorithm 2: Multi-Attribute Combination

```
def MultiAttributeCombination:
    Initialize plot set(A, B, C)
    for trajectory in trajectorySet :
        for trajectoryPoint in trajectory :
            if current point attributeA and last point attributeA are in the same interval :
                append trajectoryPoint to A
            else:
                // e.g., using lines to indicate attributeA
                plot lines with A
                Initialize plot set(A)
            end if
            if current point attributeB and last point attributeB are in the same interval :
                append trajectoryPoint to B
            else:
                // e.g., using lines to indicate attributeB
                plot lines with B
                Initialize plot set(B)
            end if
            if current point attributeC and last point attributeC are in the same interval :
                append trajectoryPoint to C
            else:
                // e.g., using lines to indicate attributeC
                plot lines with C
                Initialize plot set(C)
            end if
        end for
    end for
end
```

due to its slow progression, that is, playing such historical animations at an excessively slow rate is a waste of time; in contrast, playing at the overly fast rate would make discovering characteristics difficult. Therefore, during the playback process, users typically start browsing at a relatively fast speed; then, after they find a critical point, they prefer a slower speed to view the corresponding time.

With time series involved, trajectory data describes the time-varying feature of spatial data, whose spatial and attribute features of geographic objects could change independently or simultaneously overtime. Our major interest is the attribute-feature variations, especially the shared evolution of multiple attributes. Besides, the strength of animation is in presenting data, verifying, and analyzing results. For example, in terms of enhancing visual scalability, it can effectively prevent the cluttering problem of static visualization caused by insufficient display space or a plethora of data to visualize. Furthermore, the animation is a natural technique to transfer dynamically varying data and a promising application of animation is conveying real-time variations and the spatial and temporal reproduction. Our solution comprises two aspects of dynamically perceptual details:

- Trajectory replay

On a screen with a certain height, the number of visible trajectories is limited. Where the 2D map is out of the visible range, users will lose the spatial reference of the trajectory. Replaying the movement of trajectory ribbons addresses the absence of missing locational information in an arbitrary trajectory in the visible range.

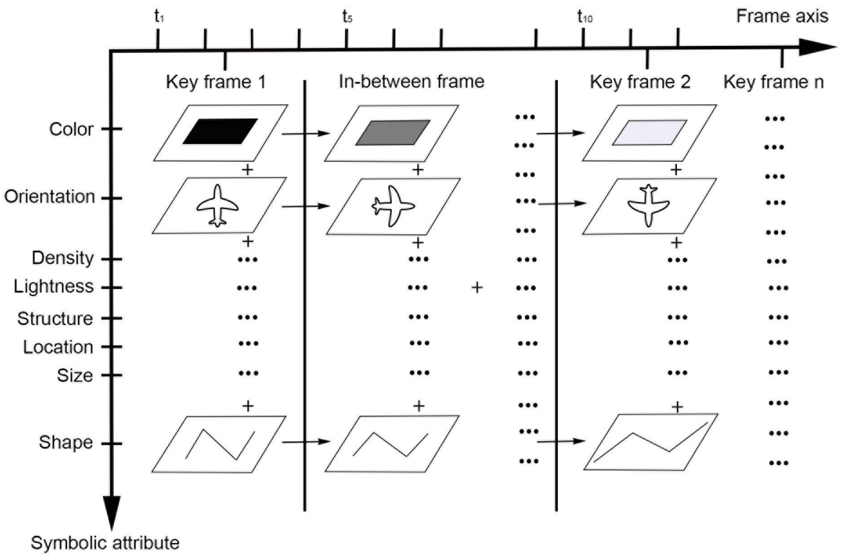
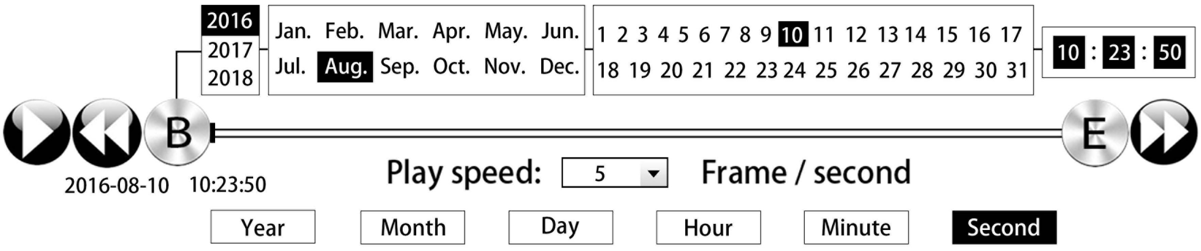


Fig. 5. Logical model of dynamic visual variables.



2016-08-10 10:23:50

Play speed: 5 Frame / second

Fig. 6. The SITL interface.

Figure 7 shows the trajectory-replay behavior with simulated data, where the selected trajectory is highlighted, the trajectory ribbon is encoded by corresponding visual variables, and the converted elevations are plotted. The entire replay operation is conducted by controlling the SITL, narrating the evolution process regarding the movement path and an attribute value of this trajectory.

• Dynamic extraction

We extracted the dynamic change processes of trajectory data in any conditions of users' interest to detect potential anomalies. Here, the extraction conditions can consist of one or more attributes. Moreover, the dynamic change process can be either a process that changes over time or a process that changes based on the geographical location. As shown in Figure 8, through the interactive tools, we set the extraction conditions as a combination of multiple attributes: the speed is $0m/s$–$30m/s$ (the dark-red color), the acceleration is $0m/s^2$–$10m/s^2$ (solid-line box style), and the dynamic change process is changing over time. In this manner, we can extract the evolution of a single attribute over space or time or extract multiple attributes in certain space or time.

3.2 Interactions

An excellent visualization solution should possess a certain level of interactivity to solve the varied requirements of users (Lam, 2008). We have achieved interactive features of overview, zoom filter, recognize, encode, filter, details-on-demand, and adjustments to the visibility and rendering order of different information layers. These interactions enable a comprehensive observation and analysis of the trajectory scene to support research tasks regarding the local and global exploration of space and time. As basic functions, various operations serve to select objects of interest, including trajectories, trajectory segments, moving objects, and trajectory events, in separate or combined forms, thereby reducing the trajectory data complexity.

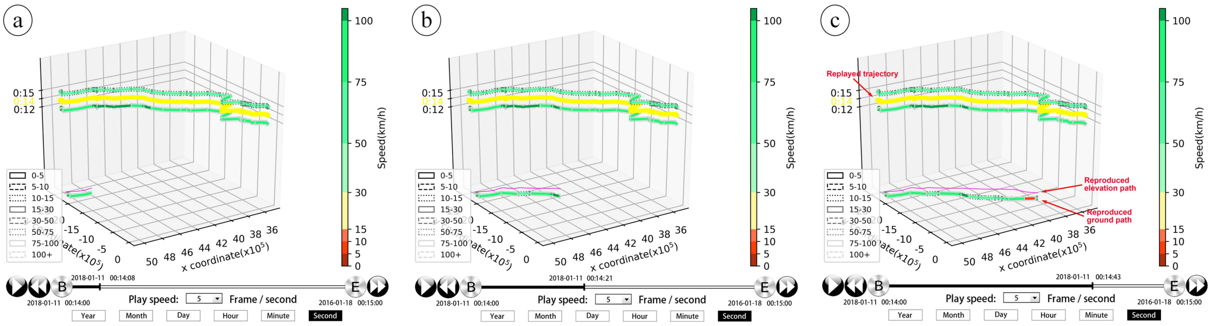


Fig. 7. Three snapshots showing the simulated trajectory replay of a user-selected trajectory at moments of (a) 00:14:08; (b) 00:14:21 and (c) 00:14:43.

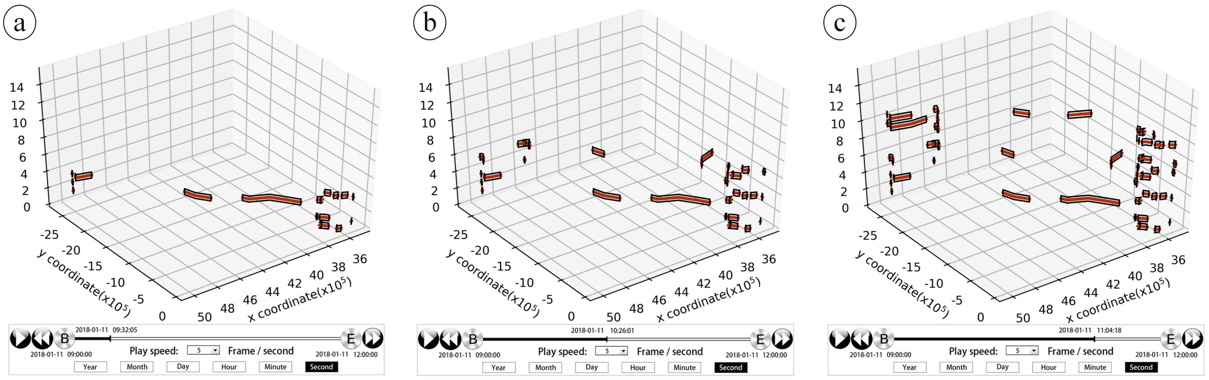


Fig. 8. Multi-attribute dynamic extraction over time of simulated data withspeed in 0–30m/s , acceleration in 0–10m/s^2 and time before (a) 09:32:05, (b) 10:26:01, and (c) 11:04:18.

4. Experiment and discussion

Open-pit coal mining employs giant mining trucks as the primary form of vehicle transportation. The above mentioned multi-attribute analysis and research methods can derive central information of the trucks' movements, the schedule at which inner tubes are changed and determination between empty/heavy loads, and eventually develop a reasonable strategy of achieving safe mine production and reducing production costs.

4.1 Experimental preparation

4.1.1 Data preparation

During the operations of a mining truck, GPS sensors, weather condition sensors, tire temperature sensors and tire pressure sensors connected to a truck-mounted computer continuously collect these attribute values in real time. The result is a composite dataset consisting of multiple attributes. We continuously collected these data from a truck identified as TR207 in an opencast coal mine of Inner Mongolia. In this experiment, we selected the data from June 28, 2016, to August 30, 2016 with a database filter, with a total of 259,200 records. The spatiotemporal data, speed, tire pressure, and tire temperature were selected as the source data for the attribute superimposition investigation. Since the trajectory data are collected by sampling the positions of the truck, there are sampling errors and speed variations. Before the trajectory analysis, trajectory data preprocessing including data cleansing, trajectory compression, trajectory segmentation and trajectory clustering (Section 2.2), and index the clustered trajectories to facilitate subsequently visual analysis. Through the preprocessing, we reject 362 outliers from the original dataset, and the noisy data points are detected and interpolated using smoothing splines (Early & Sykulski, 2020).

4.1.2 Visual variables allocation

Appropriate visual variables should be selected (Dang *et al.*, 2010). Meanwhile, it is necessary to optimize the visual encoding scheme and reasonably adjust the weights and priorities between variables such as colors, textures, and shapes. Speed and tire pressure data should be rendered continuously by geometric primitives due to the large fluctuations in the raw data, while the temperature can be rendered discretely because of its smooth changes.

- Speed and tire pressure mapping

We provide interactive legends to support a range customization so that users can adjust the display style and the corresponding interval. In this case, we customized a color gradient to indicate the speed and used a line box style to denote the tire pressure. To highlight the abnormal distributions of both high tire pressures and low tire pressures, the former is displayed with boxes of thick, solid lines, while the latter is represented as boxes with thick dashed lines. Sequential trajectory paths must be constructed to embed geometric primitives integrated with attribute values into the virtual display space. Consider the rendering of the speed attribute as an example. In Figure 9, it is assumed that the interactive speed legend has n segmented intervals, $[0, a_1), [a_1, a_2)..., [a_{n-1}, a_n)$; each interval consists of a point set $T(tr_1, tr_2, \dots, tr_p)$ for a given trajectory TR in a spatial connection; and $p - 1$ in-between segments are colored. If we were to directly encode these segments with color variables corresponding to the n intervals, the space-speed attribute characteristics of the trajectory could be inadvertently altered. To solve this problem, we divide each segment into several microsegments according to the number of legend intervals they span. The segment coloration corresponds to the rendering scheme of the interactive speed legend. Therefore, the final trajectory segment type that requires continuously rendering depends on the size of the attribute value. Similar logic applies to the tire pressure rendering.

- Tire temperature mapping

For the tire temperature attributes, we apply the color saturation, size, and opacity of points to the discrete rendering. After the trajectory path is constructed, we merge successive attribute points with the same radius, color, and opacity to avoid creating a visual overload. Furthermore, stylized and postprocessing effects are applied to remove visual artifacts, resulting in better visual quality.

4.2 Results and discussion

Combining with the actual requirements of mining truck management, we devise the experimental features as follow:

1) Import of trajectory data, automatic index creation and trajectory clustering.

2) Display of the preprocessed trajectory, with scaling, panning, and multi-linking operations.

3) Trajectory replay and dynamic extraction features, with specified time and trucks.

4) Truck attribute query and multi-attribute interplay analysis.

5) Truck trajectory elevation mapping.

6) Driver behavior detection and Truck operation determination.

4.2.1 Multi-attribute interplay behaviors

For simplicity, we temporarily disabled the stacked elevation ribbons of the trajectory, yielding the snapshot in Figure 10. The trajectories are stacked using days as relative time, and time without a trajectory are displayed as blanks. We can observe from the overview that various attributes, such as the speed, tire pressure, and tire temperature, of truck TR207 from 10:00 to 12:00 change significantly over time and with the spatial location. This is particularly true at a certain area at approximately 11:30 (the centralized dark green area), which is marked as zone M. We speculate that the following conditions occurred:

1) At approximately 11:30 throughout several days, the mining trucks exhibited anomalously behaviors in zone M different from those in other areas. This anomalous behavior typically occurred at a regular time (approximately 11:30) and at a fixed location (zone M).

2) During the anomalous behavior, the trajectory speeds dropped to the interval of $0 - 5m/s$, and the tire pressure decreased suddenly. Therefore, the tire pressure and speed changes did not occur continuously; instead, they were affected by some specific occurrence or behavior. It seems highly probable that the mining truck decelerated to a stop and reduced its load repeatedly at this specific time and location.

3) The tire temperature varied continuously, but it was not particularly consistent with the

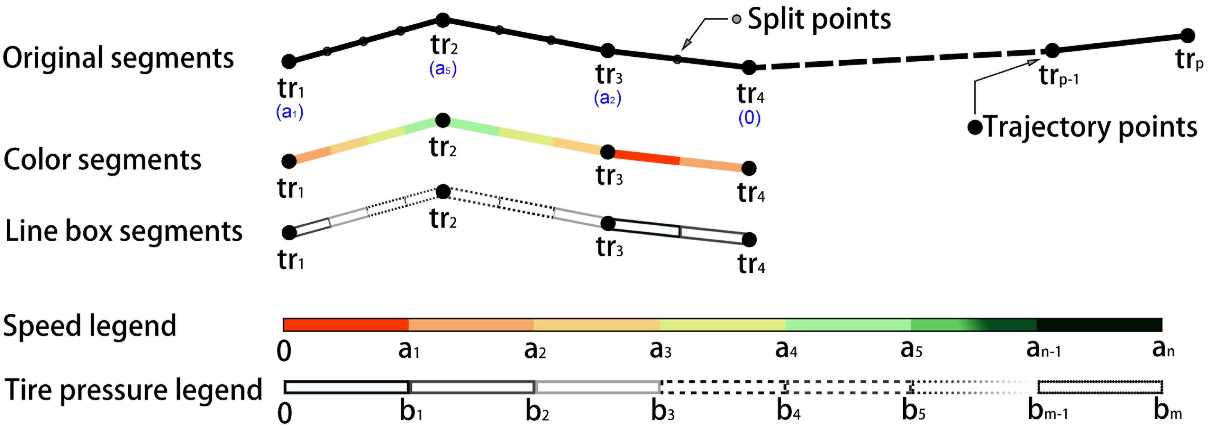


Fig. 9. Plotting color segments and line box segments to construct the trajectory paths.

speed or tire pressure fluctuations. Therefore, the tire temperature may not be strongly related to the behavior, speed, and tire pressure of the truck; rather, it may be more closely related to the time.

4) The truck's speed before 11:30 was generally lower than its speed after 11:30. This result may still be related to the truck's behavior in zone M, and it may also be related to the safety requirements in the mining area or the terrain. However, the time period and spatial locations of occurrences were not fixed, indicating that the speed may be related to unexpected circumstances and also external factors.

5) Two trajectories are missing from the upper part of the chronologically ordered stacked trajectories, demonstrating that truck TR207 was not in the area of the mine during those periods.

6) Below the missing data are 3 consecutive trajectories where the tire pressure remained low for the entire time; this indicates that the truck's tires may have suffered abnormalities during this period.

7) However, in the trajectories above the vacancy, the tire pressure returned to its previous state, indicating that the low-pressure problem with truck TR207 was solved.

4.2.2 Elevation stacking behaviors

In addition to the ground location information, which is recorded as the longitude and latitude, elevation information is also included in the data collected for mining truck TR207. Based on the elevation data changes, we created an interactive legend to customize the trajectory elevation interval and establish an elevation value rendering model of the trajectory to map and render the trajectories (the selected transparency is higher than that of the attribute data). Meanwhile, the stacked elevation ribbons for each trajectory are plotted as well. In this manner, we can transform the original simple 2D trajectories into 3D trajectory ribbons that display elevation information. Each 3D trajectory ribbon contains time information and elevation information in addition to the corresponding ground position information on the 2D map and three types of attribute data (speed, tire temperature, and tire pressure data). We rotate the superimposing view in Figure 10 around the yaw axis by 180° and enable the elevation ribbons. To attain relatively ideal intuitiveness, we then rotate the view around the pitch axis by an appropriate negative angle and lower the opacity of the 2D map, yielding the snapshot in Figure 11. Due to the yawing, on the right side of Figure 11 corresponds to the left side of Figure 10. Figure 11 shows that the elevation differences before the truck reaches zone M are generally greater than those after it reaches zone M; hence, we conjecture that the major cause of the speed difference before and after approaching zone M is related to the terrain and the safe production requirements of the mining area.

4.2.3 Actual situation verification

To confirm our hypothesis of the multi-attribute behavior of the truck, we conducted a site

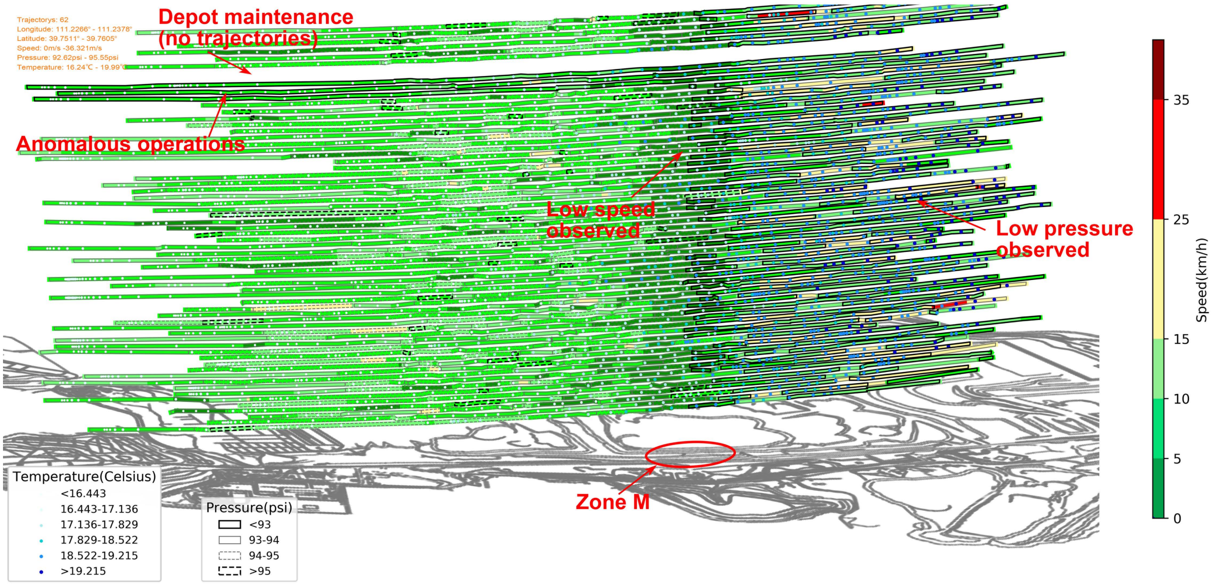


Fig. 10. Multi-attribute superimposition for truck TR207.

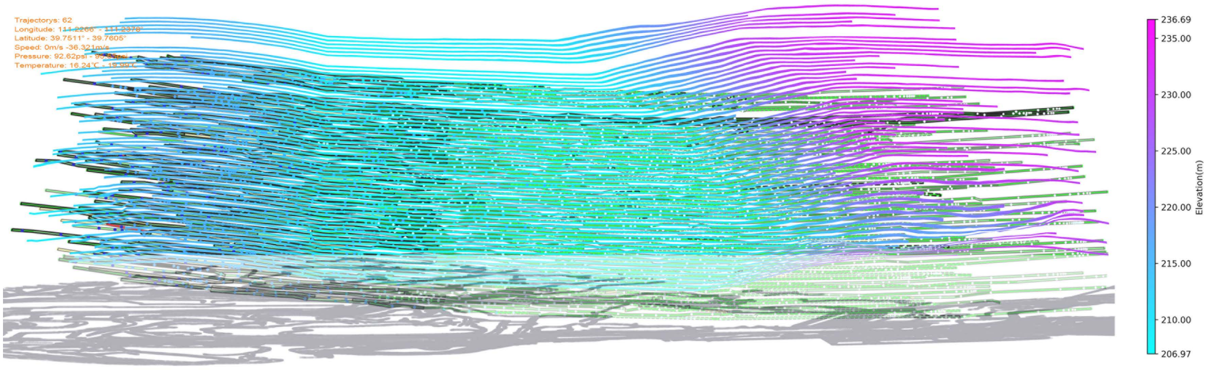


Fig. 11. 3D trajectory stacked elevation ribbons for truck TR207.

investigation in the mining area and verified the following situations:

1) Truck TR207 was driven from the loading area to the dumping area from June 28, 2016, to August 30, 2016. The mining truck decelerated as it approached the dumping area (zone M). When the speed reached zero, the truck began unloading, which completed at approximately 11:30. Subsequently, the load dropped, the tire pressure decreased, the vehicle switched from a full load to an empty load, and then the truck accelerated while leaving the dumping area. Therefore, this unloading behavior affected both the truck's speed and its tire pressure (in this experiment, the phenomenon in which the speed fell to zero and the tire pressure decreased signifies unloading behavior). Moreover, the tire pressure is highly related to the load under certain conditions. Hence, to a certain extent, the method can identify a truck's operational status: empty or loaded.

2) From 10:00 to 12:00, the outdoor temperature gradually increased, and the tire temperature changed linearly with time. Thus, the truck's behavior, speed, and tire pressure had a little effect on the tire temperature, which was primarily determined by the external temperature.

3) The fact that the speed before 11:30 was generally lower than that after 11:30 is related to the safety regulations and the spatial terrain of the mining area. In other words, when a certain load level is reached and the production safety limit is met, elevation differences along the route affects the truck's speed. However, the time and space locations

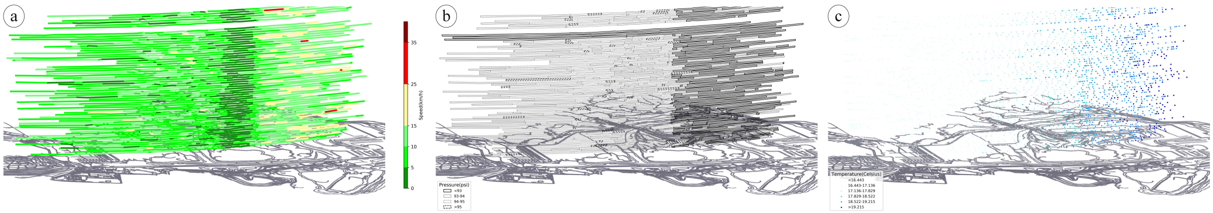


Fig. 12. Individual attribute displays of truck TR207. (a) speed attribute, (b) tire pressure attribute, and (c) tire temperature attribute.

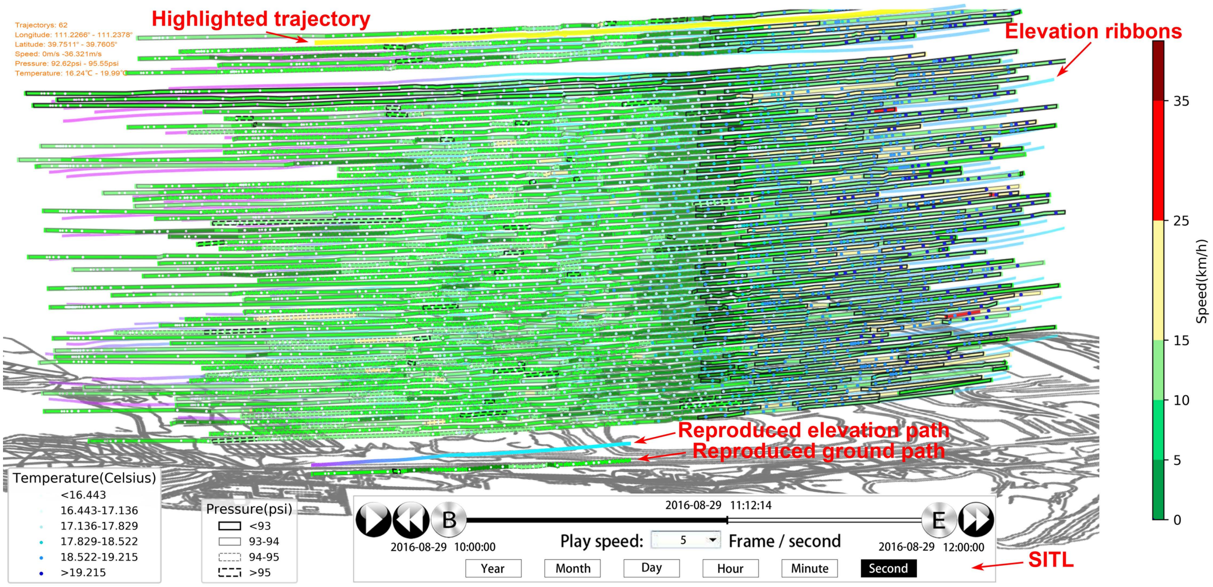


Fig. 13. Trajectory replay of truck TR207 before 11:12:14 on August 29.

of speed changes were not fixed; instead, the speed changes are related to the weather conditions, the occurrence of emergencies, and the driver's on-spot performance.

4) On August 20th, 2016, the tire health monitoring system (THMS) on truck TR207 issued a warning that the tire pressure was too low, making the truck susceptible to undesirable phenomena such as higher fuel consumption and increased tire wear. Consequently, the vehicle entered the workshop on August 23 for inspection and maintenance. This explains the missing visualizations in the upper trajectories in Figure 10 and 11. From August 25 onwards, the truck re-entered the loading area as usual, and the tire pressure returned to normal.

All of the above mentioned observations match the visualization results in Figure 10. Particularly, if there is a study requirement for certain attributes, we can separately display the layer that corresponds to an attribute.

4.2.4 Trajectory replay behaviors

In the trajectories of truck TR207 during the period from June 28th, 2016, to August 30th, 2016, the attribute information of several trajectories at the top of the view is incomplete. Although our visualization view supports interactions of panning / rotating to display complete stacked trajectory, these operations could trigger partial trajectories beyond the display space, therefore losing the overall morphological structure and failing to referring the corresponding 2D-map location of trajectory segments. We enabled the stacked elevation ribbons and selected the trajectory on August 29, 2016, yielding the snapshot shown in Figure 13. The time selection control on the SITL interface then corresponds to the selected period and the spatial sequence is displayed consecutively on the 2D map. The actual movement of the trajectory is presented to reproduce the trajectory of truck TR207 integrated with the speed, tire temperature, tire pressure and elevation.

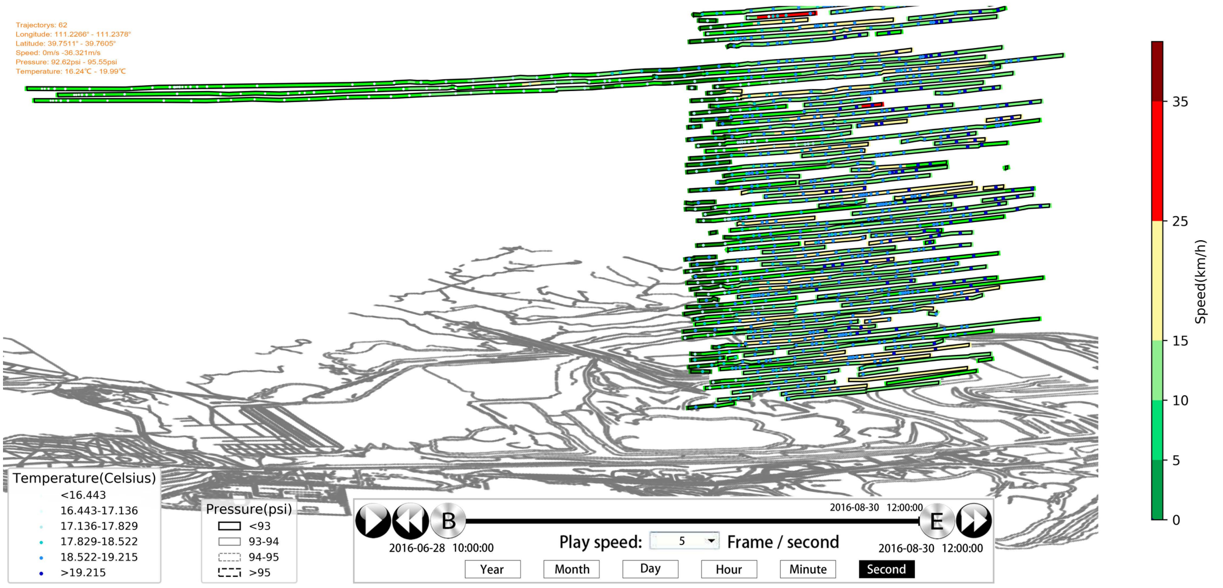


Fig. 14. Attribute extraction of truck TR207 where the tire pressure is less than 93 psi.

4.2.5 Dynamic attribute extraction

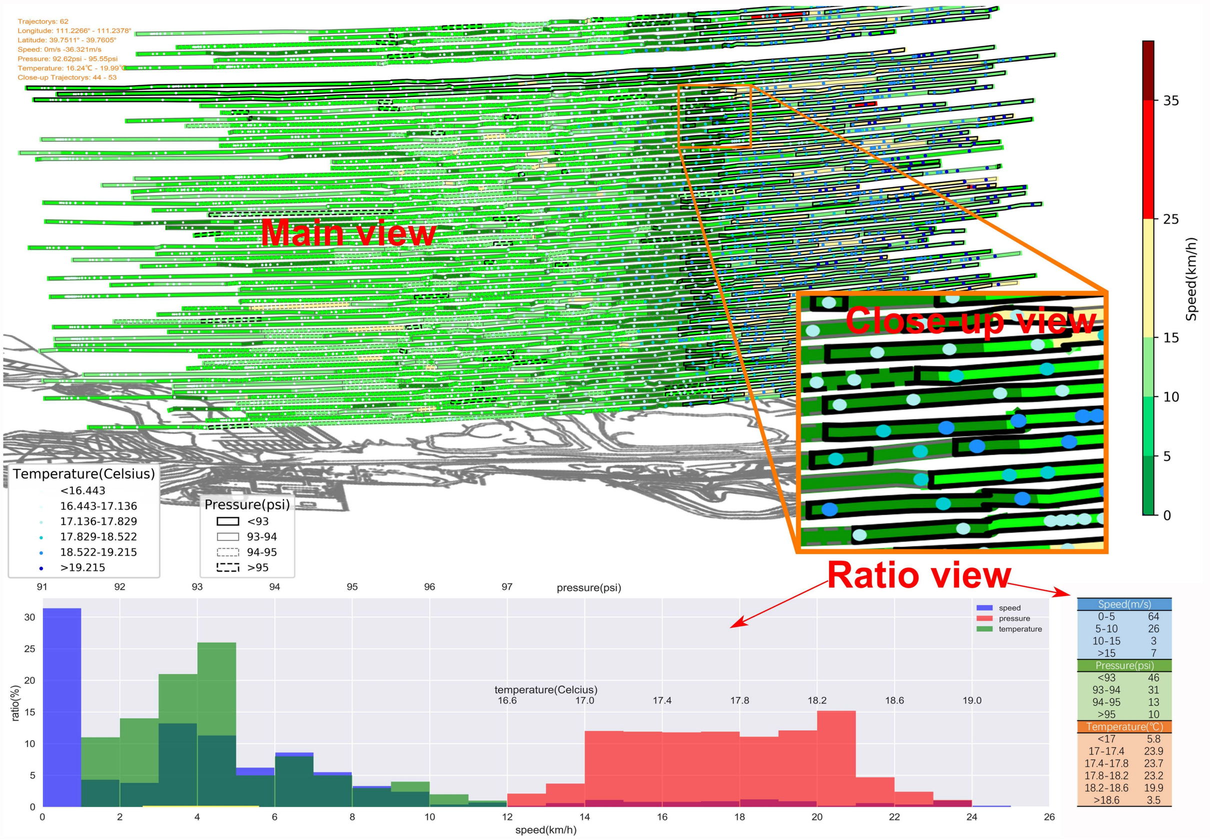
The tires are the only points of contact between the vehicle and road surface: the entire dynamic vehicle performance ultimately derives from the power transmitted from the tires to the road surface. If a tire's pressure is excessively low, the surface contact patch between the tire and ground increases, and the heat generated by rolling is greater than that generated at the standard tire pressure, resulting in additional energy costs. Moreover, the presence of a continuous low pressure deforms the inner ply, causing structural damage and even a flat tire in severe cases. To directly reveal the distribution of overly low tire pressures during the truck's operation, we disabled the elevation ribbons. In the interactive interface, we set the filter condition to a tire pressure of less than 93 psi and selected the dynamic change process as changing over time, yielding the snapshot shown in Figure 14. The entire process of perception can be accomplished through the SITL interface by combining a succession of extracted views into a low-tire-pressure event, which occurs mainly in two periods. First, almost all segments with a relatively low tire pressure occurs after zone M, which is due to the behavior-guided unloading phenomenon. Second, the tire pressure remains exceedingly low over the entire trajectories only from August 20 to August 22, which is undoubtedly a behavioral anomaly. We can also set other types of interactions based on corresponding requirements. For example, by establishing a filter with the speed $> 15km/h$, tire pressure $< 93psi$, elevation difference $< 5m$ (with stacked elevation ribbons enabled) and selecting the dynamic change process as changing with the geographical location, the distributions of high speeds, smooth paths, and low tire pressures can be obtained (most likely representing that the truck is lightly loaded with smooth road conditions).

4.2.6 Other interactions

As the amount of visualized trajectories grows, the details may become blurred due to the limited display space (Ma *et al.*, 2018). The close-up tool solves the problem of coordinating the visualization of both the overall structure and the local details of the view in a limited space. The basic idea is to display a secondary close-up view of a rectangular area of interest selected by the user; meanwhile, the statistical results of various attribute values in the area of interest are visualized in other linked views as texts and charts. Figure 15 displays a close-up analysis performed on certain parts of the trajectories numbered from 44 to 53, yielding multilinked views that consist of the main view, a secondary close-up view, and a ratio view.

5. Conclusion

The paper introduced an information visualization analytics and highly interactive virtual environment. First, we calculated and analyzed the obtained trajectory data, stacked trajectories in subsets with spatial similarity, and displayed



Speed(m/s)	
0-5	64
5-10	26
10-15	3
>15	7
Pressure(psi)	
<93	46
93-94	31
94-95	13
>95	10
Temperature(℃)	
<17	5.8
17-17.4	23.9
17.4-17.8	23.7
17.8-18.2	23.2
18.2-18.6	19.9
>18.6	3.5

Fig. 15. Multilinked views of close-up analysis on an area of interest.

the elevation information of the contained trajectory points in each trajectory. Then, we established a corresponding number of layers for the attribute types to be studied and created geometrical primitives that are reasonably configured based on the selected mapping parameters. While researching the interrelations among the multiple attributes, we superimposed their layers, implementing a combinational association study of those attributes and achieving a real-time dynamic visualization under certain space-time conditions. Finally, we provided a type of interaction to support users' space-time exploration endeavors. Typically, although space-time cube effectively matches temporal information, its utility is often limited by the 3D cognition mechanism of humans. Therefore, this approach seems not adequate for long-term and complex trajectory visualization. Essentially, multi-attribute spatiotemporal trajectory stacking visualization is an enhanced space-time cube that supports exploratory spatiotemporal analysis and trajectory data mining. Compared with previous visualization methods, our solution exhibits an excellent information integration display. When exploring various complexities of dynamic multi-attribute spatiotemporal trajectories, the derived results also meet human cognition and comprehension. Our visualization tool is suitable for those trajectory sets whose data can be clustered into similar geometries under given real-time constraints; however, when the paths of moving objects are completely disordered, our tool is no longer applicable. Besides, the approach depends on collected trajectory data for post visualization analysis. In dealing with real-time trajectory data streams, our approach cannot perform a real-time dynamic trajectory view.

ACKNOWLEDGMENTS

This research was funded by China Postdoctoral Science Foundation, grant number 2020M670267, National Office for Philosophy and Social Sciences, grant number 19ZDA329, and Tsinghua University, grant number 2019GQG1017.

References

Al-Dohuki, S., Wu, Y., Kamw, F., Yang, J., Li, X., Zhao, Y., Ye, X., Chen, W., Ma, C. & **Wang, F.** (**2017**) SemanticTraj: A

New Approach to Interacting with Massive Taxi Trajectories. IEEE Transactions on Visualization and Computer Graphics **23**(1):11–20. doi: 10.1109/TVCG.2016.2598416.

Andrienko, G., **Andrienko, N.**, **Demsar, U.**, **Dransch, D.**, **Dykes, J.**, **Fabrikant, S.I.**, **Jern, M.**, **Kraak, M.J.**, **Schumann, H.** & **Tominski, C.** (**2010**) Space, time and visual analytics. International journal of geographical information science **24**(10):1577–1600.

Bezerianos, A. & **Isenberg, P.** (**2012**) Perception of visual variables on tiled wall-sized displays for information visualization applications. IEEE Transactions on Visualization and Computer Graphics **18**(12):2516–2525.

Chen, L., **Özsu, M.T.** & **Oria, V.** (**2005**) Robust and fast similarity search for moving object trajectories. In Proceedings of the 2005 ACM SIGMOD international conference on Management of data. pages 491–502.

Dang, T.N., **Wilkinson, L.** & **Anand, A.** (**2010**) Stacking graphic elements to avoid over-plotting. IEEE Transactions on Visualization and Computer Graphics **16**(6):1044–1052.

Early, J.J. & **Sykulski, A.M.** (**2020**) Smoothing and Interpolating Noisy GPS Data with Smoothing Splines. Journal of Atmospheric and Oceanic Technology **37**(3):449–465. doi:10.1175/JTECH-D-19-0087.1.

Ferreira, N., **Klosowski, J.T.**, **Scheidegger, C.E.** & **Silva, C.T.** (**2013**) Vector field k-means: Clustering trajectories by fitting multiple vector fields. In Computer Graphics Forum, volume 32. Wiley Online Library, pages 201–210.

Filho, J.A.W., **Stuerzlinger, W.** & **Nedel, L.** (**2020**) Evaluating an Immersive Space-Time Cube Geovisualization for Intuitive Trajectory Data Exploration. IEEE Transactions on Visualization and Computer Graphics **26**(1):514–524. doi: 10.1109/TVCG.2019.2934415.

Gosink, L., **Bensema, K.**, **Pulsipher, T.**, **Obermaier, H.**, **Henry, M.**, **Childs, H.** & **Joy, K.I.** (**2013**) Characterizing and visualizing predictive uncertainty in numerical ensembles through Bayesian model averaging. IEEE transactions on visualization and computer graphics **19**(12):2703–2712.

Guha, S., **Rastogi, R.** & **Shim, K.** (**2000**) ROCK: A robust clustering algorithm for categorical attributes. Information systems **25**(5):345–366.

Han, J., **Pei, J.** & **Kamber, M.** (**2011**) Data mining: concepts and techniques. Elsevier.

Huang, X., **Zhao, Y.**, **Ma, C.**, **Yang, J.**, **Ye, X.** & **Zhang, C.** (**2016**) TrajGraph: A Graph-Based Visual Analytics Approach to Studying Urban Network Centralities Using Taxi Trajectory Data. IEEE Transactions on Visualization and Computer Graphics **22**(1):160–169. doi: 10.1109/TVCG.2015.2467771.

Juřík, V., **Herman, L.**, **Šašinka, Č.**, **Stachoň, Z.** & **Chmelík, J.** (**2017**) When the display matters: A multifaceted perspective on 3D geovisualizations. Open Geosciences **9**(1):89–100.

Lam, H. (**2008**) A framework of interaction costs in information visualization. IEEE transactions on visualization and computer graphics **14**(6):1149–1156.

Lulli, A., **Dazzi, P.**, **Ricci, L.** & **Carlini, E.** (**2015**) A multi-layer framework for graph processing via overlay composition. In European Conference on Parallel Processing. Springer, pages 515–527.

Ma, Y., **Wang, Y.**, **Xu, G.** & **Tai, X.** (**2018**) Multilevel visualization of travelogue trajectory data. ISPRS International Journal of Geo-Information **7**(1):12.

OpenDataCity (**2013**) re:log - Besucherstromanalyse per re:publica W-LAN. https://opendatacitygithubio/relog/ .

Pei, W., **Wu, Y.**, **Wang, S.**, **Xiao, L.**, **Jiang, H.** & **Qayoom, A.** (**2018**) BVis: urban traffic visual analysis based on bus sparse trajectories. Journal of Visualization **21**(5):873–883. doi:10.1007/s12650-018-0489-z.

Tominski, C., **Schumann, H.**, **Andrienko, G.** & **Andrienko, N.** (**2012**) Stacking-based visualization of trajectory attribute data. IEEE Transactions on visualization and Computer Graphics **18**(12):2565–2574.

Von Landesberger, T., **Brodkorb, F.**, **Roskosch, P.**, **Andrienko, N.**, **Andrienko, G.** & **Kerren, A.** (**2015**) Mobilitygraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering. IEEE transactions on visualization and computer graphics **22**(1):11–20.

Vrotsou, K., **Janetzko, H.**, **Navarra, C.**, **Fuchs, G.**, **Spretke, D.**, **Mansmann, F.**, **Andrienko, N.** & **Andrienko, G.** (**2014**) SimpliFly: A methodology for simplification and thematic enhancement of trajectories. IEEE Transactions on Visualization and Computer Graphics **21**(1):107–121.

Waser, J., **Fuchs, R.**, **Ribicic, H.**, **Schindler, B.**, **Bloschl, G.** & **Groller, E.** (**2010**) World lines. IEEE transactions on visualization and computer graphics **16**(6):1458–1467.

Zhang, Y. & **Lin, Y.** (**2019**) An interactive method for identifying the stay points of the trajectory of moving objects. Journal of Visual Communication and Image Representation **59**:387–392.

Zhu, S., **Sun, G.**, **Jiang, Q.**, **Xia, W.** & **Liang, R.** (**2020**) Microscopic Visual Analysis of High-Density Trajectory Data. Journal of Computer-Aided Design and Computer Graphics :1–11.

Submitted: 09/09/2020
Revised: 01/11/2020
Accepted: 21/11/2020
DOI: 10.48129/kjs.v48i4.10548