DOI: 10.1111/cgf.13999
Eurographics Conference on Visualization (EuroVis) 2020
M. Gleicher, T. Landesberger von Antburg, and I. Viola
(Guest Editors)

Volume 39 (2020), Number 3

SEEVis: A Smart Emergency Evacuation Plan Visualization System with Data-Driven Shot Designs

Q. Li[1,3], Y. J. Liu[1], L. Chen[1,2†], X. C. Yang[1], Y. Peng[1], X. R. Yuan[4], and M. L. L. Wijerathne[5]

[1]School of Software, Tsinghua University, Beijing, China
[2]Beijing National Research Center for Information Science and Technology (BNRist), Beijing, China
[3]AI Group, WeBank, Shenzhen, Guangdong, China
[4]Peking University, Beijing, China
[5]The University of Tokyo, Tokyo, Japan

Abstract

Despite the significance of tracking human mobility dynamics in a large-scale earthquake evacuation for an effective first response and disaster relief, the general understanding of evacuation behaviors remains limited. Numerous individual movement trajectories, disaster damages of civil engineering, associated heterogeneous data attributes, as well as complex urban environment all obscure disaster evacuation analysis. Although visualization methods have demonstrated promising performance in emergency evacuation analysis, they cannot effectively identify and deliver the major features like speed or density, as well as the resulting evacuation events like congestion or turn-back. In this study, we propose a shot design approach to generate customized and narrative animations to track different evacuation features with different exploration purposes of users. Particularly, an intuitive scene feature graph that identifies the most dominating evacuation events is first constructed based on user-specific regions or their tracking purposes on a certain feature. An optimal camera route, i.e., a storyboard is then calculated based on the previous user-specific regions or features. For different evacuation events along this route, we employ the corresponding shot design to reveal the underlying feature evolution and its correlation with the environment. Several case studies confirm the efficacy of our system. The feedback from experts and users with different backgrounds suggests that our approach indeed helps them better embrace a comprehensive understanding of the earthquake evacuation.

CCS Concepts

• ***Human-centered computing*** → *Visualization; Visualization design and evaluation methods;*

---

1. Introduction

Tracking human mobility in earthquake evacuation analysis is important for an effective first response and disaster relief. However, the present understanding of the evacuation behaviors remains limited due to the heterogeneous, spatiotemporal and multivariate urban data from different areas of sciences, various data sources or multiple simulations involved in the process [KH12]. For example, an earthquake-tsunami evacuation dataset typically comprises terrains, buildings, damages, roads, and agents, most of which are multivariate and time-varying with thousands of time steps. These data may also originate from multiple sources, e.g., earthquake damages are computed by using a finite element method (FEM)-based structural analysis with multivariate displacement, stress, strain, and build-id attributes, while the agent data are generated by multi-agent simulations that involve millions of points. Given the characteristics of the involved data, extracting and tracking the major features in these data is essential in earthquake evacuation analysis.

Although conventional visualization methods have demonstrated promising performance in emergency evacuation analysis [RWF*13, GESR13, BKLR14], identifying users' exploration purposes and tracking features of interest, and delivering users' exploration results are nontrivial for several reasons. **(1) Massive information.** When users are exploring reasons behind evacuation events like *road blockage* and *congestion*, they often encounter massive information, including the surrounding environment, earthquake damages, and large-scale human movements, that need to be visualized simultaneously. However, due to the limited screen space, only a small part of this information is made available to users. It is thus challenging to enable users to express their exploration purposes in advance and in an appropriate manner. In other words, identifying the most notable features behind such massive information are important for end-users. **(2) Feature dynamics.** Given that multi-source and heterogeneous features may fuse and evolve, i.e., they may experience "merge-split" activities (i.e., multiple features merge into one, and

---

[†] L. Chen is the corresponding author.

then this single feature splits into multiple features again within a specified time interval [OSBM14]), planning a way to track and visualize feature dynamics in a 3D space is also a challenging task for end-users. **(3) Feature correlation.** Conventional visualizations often ignore the correlations between features and other attributes (e.g., the accompanying environment), including but not limited to a comprehensive display of the surroundings and comparison among multiple simulations for a specific event, etc. Comprehensively examining the correlation of features with the surroundings can help users identify the factors that may hinder a smooth evacuation.

Previous studies have leveraged efficient animations to deliver scientific findings [WO90, VFSH03, VFSG06, CO09], controlled the camera parameters [BS05, TFTN05, CO09] and guided the view selection based on visibility and importance [JS06, VFSG06, WYM08]. Although animated visualization has clear advantages, there are two drawbacks when the animation is directly generated from real-world data and applied in our scenario without any interactive editing: (1) When the animation contains rich information in a long time span, the audience may not stay focused. Without considering users' exploration purposes and explicitly identifying the features of interest for exploration and decision-making, viewers can lose patience if they fail to extract meaningful information from it [TMB02]; (2) Real-world data can be irregular and unpredictable. Users may be disappointed if no patterns occur in a long period. Besides, given that users often undergo trial-and-error processes when manually inspecting various features, the resulting evacuation events, and the correlations of features with the environment, different narrative mechanisms must be developed for specific features and events to achieve an effective exploration and storytelling. Therefore, a nontrivial visualization and storytelling method must be developed to fulfill these objectives and to overcome the challenges in delivering the exploration results. What' more, few empirical studies have investigated the extent to which a visualization and storytelling approach can resolve the breakdowns in the experts' understanding of human evacuation mobility dynamics and bridge their knowledge of the factors that influence an earthquake evacuation analysis.

In this paper, we propose a novel shot design approach to directing the narrative storytelling for tracking feature evolution and correlation with the environment in an earthquake evacuation analysis. We first observe the experts' current practices and identify their primary needs and concerns. We then leverage shot designs for earthquake evacuation analysis. Particularly, we first develop various interactions to allow users to specify their tracking purposes, i.e., identifying the features or regions of interest. A scene feature graph that identifies the most dominating evacuation events is then generated. We calculate the optimal camera route (i.e., a storyboard) based on users' tracking purposes. For each evacuation event identified along this route, we apply shot designs to reveal the underlying feature evolution and its correlation with the environment. The primary contributions of this study are summarized as follows:

- We propose an interactive scene feature graph that identifies the most dominating evacuation events based on spatiotemporal trajectory clustering and user-specific tracking purposes.
- We employ different shot types and compositions borrowed from filmmaking to generate customized narrative animations and track different evacuation features with various exploration purposes.

2. Related Work

Literature that overlaps this work can be divided into four categories: feature tracking, emergency visualization, animation, and data storytelling, as well as scene navigation and camera control.

2.1. Feature Tracking

Feature tracking helps reveal regular, periodic, or random patterns [Joh04]. In emergency simulations, the most significant feature locates in human evacuation trajectory data and many techniques are proposed to analyze the trajectory data. For example, Liu et al. [LGL*11] proposed a trajectory visualization method helping users analyze the diversity patterns. Wang et al. [WLY*13] detected traffic jams based on GPS trajectories to derive reasons for urban traffic congestion. Andrienko et al. [AA11] transformed GPS-tracked trajectories into aggregated flows between areas to depict important urban moving patterns. To simplify the trajectory data which is usually very large in quality, they proposed a visual analytics procedure based on event clustering and spatiotemporal aggregation for place-oriented analysis of movement data, and later leveraged interactive filtering tools to attach relevance flags to trajectory elements [AAH*13, AAFG18]. Vrotsou et al. [VJN*15] introduced a systematic stepwise methodology for simplifying, thematically enhancing and analyzing the trajectories. These approaches use multiple views to provide the users with different perspectives on the dataset, while our approach presents a single scene graph to give an overview of the evolution of features. We allow domain experts to focus on the evolution of topology and then automatically generate a realistic animation for them to analyze the correlation between spatial-temporal features and the surrounding urban environment. Sacha et al. [SAMS*17] presented a novel dynamic approach that combines trajectory simplification and clustering techniques to support the interpretation and understanding of movement patterns. Our approach first simplifies the trajectories by spatiotemporal clustering to group the underlying features into different categories. To display and visualize these features, we pioneer a moving camera based on multiple shot designs.

2.2. Emergency Planning and Visualization

Researchers have developed various data analytics applications in the field of emergency planning [MTC*16, IYT*16, TYC*17]. For example, Ribičić et al. [RWF*13] developed a visualization tool for real-time analysis of ensemble-simulation runs. Guest et al. [GESR13] improved traditional evacuation algorithms for near-real-time situational awareness in their tool. Bock et al. [BKLR14] presented a visualization system for urban rescue scenarios to provide incident commanders with inspection and access path planning in post-disaster structures. Doraiswamy et al. [DFD*14] used topological analysis to identify events and support the exploration of large, spatial-temporal urban data. Our approach follows the idea of topological analysis and performs a spatiotemporal aggregation on the trajectories instead of the volumetric, point cloud, or the scene data. For better user experience, Waser et al. [WKS*14] introduced a sketch-based input approach for special simulation expertise to create and investigate multiple scenarios with an intuitive and relevant interface. Different from them, our approach follows the idea of topological analysis and perform spatial-temporal aggregation on the trajectories instead of volumetric data, point cloud data, or scene data. These trajectories are specified by the domain experts as a kind of tracking purpose. We combine different techniques to generate a scene graph and optimize the motion path of the camera, thus supporting users with different purposes to explore the data.

2.3. Animation and Data Storytelling

Data analysts often need to communicate and demonstrate the identified insights to an audience through visualizations. However, a general audience who have no background of the visualizations may find it difficult to understand the insights through exploratory visualization. When data become increasingly complex, the visual design tends to become complicated. To present findings, data analysts need to either demonstrate the visual analytics system in person or display screenshots with further explanations. However, static figures or visual data stories can rarely be created straight out of interactive exploratory tools [GLG*16]. Analysts need to collect artifacts to compose a well-structured data story so that users can understand the data evolution and composition, which is time-consuming.

Segel and Heer [SH10] systematically summarized seven types of narrative visualization. Film, video, and animation form a genre of narrative visualization with which designers can present changes in data through motion changes. Animation is a simpler and more attractive visual form to show temporal development and stories behind data. Yu et al. [YLRC10] presented a digital storytelling approach that generates automatic animations for time-varying data. They analyzed abstract events as an event graph, in which nodes represent data features and links represent event relationships. In this study, we borrow the narrative tactics in cinematography to create engaging data stories. Particularly, we abstract evacuation trajectories into a scene feature graph, with nodes representing events and edges representing evolution. We deliver the evaluation process in animations with diverse shot designs.

2.4. Scene Navigation and Cameral Control

Scene navigation generates a guided tour in a 3D space generally constrained by visiting a set of given landmarks [XYH*18]. Vázquez et al. [VFSH01] leveraged viewpoint entropy to quantify the amount of information that a viewpoint conveys about a specific 3D scenario. Sokolov et al. [SP08] generated a path that interpolates the viewpoints by solving a Travel Salesmen Problem (TSP), in which the cities to traverse are the viewpoints and the cost is a combination of the Euclidean distance between the viewpoints and the visual quality along the path. Serin et al. [SAB12] considered a semantic distance metric between the good views to avoid transitions between unrelated landmarks. Xie et al. [XYH*18] generated a large collection of suitable camera moves around landmarks and designed a global path that selects the best camera to move for each landmark. Mindek et al. [MČV*15] presented a novel method for creating automatized gameplay dramatization of multiplayer video games. Their approach uses the event graph and the flock of cameras to create a video summary for a post-mortem visual analysis of the course of the game. Different from the above work, we take both user-specific interests and feature evolution into consideration to attain the optimal camera route.

Camera control in a virtual 3D space is determined by the specific tasks to perform and has been addressed by many techniques [LC15, LCL*10]. Blinn [Bli88] computed viewpoints with an efficient iterative technique to compute the position and orientation of a camera from the specification of on-screen properties. Visual properties in the image space have also been translated into constraints or evaluation costs, which are applied to the camera freedom and solved through a series of optimization techniques [BTMB00, RU14]. However, almost all the existing work focus on the viewpoint positions, light source, and moving paths. Wang et al. [WCL*16] borrowed time remapping and foreshadowing from cinematography to generate narrative visualizations. In this work, we introduce similar tactics, i.e., shot designs such as *panorama shot*, *dolly shot*, *normal shot*, and *following shot* that are commonly used in film directing for emotion delivering to control camera views and deliver different events along the camera route.

3. Background and Observational Study

3.1. Dataset Description and Term Definition

The dataset in this work simulates the earthquake-tsunami evacuation event in Kochi Prefecture, Japan, including the easily damaged part of the city that has an area of around $51.84\,km^2$. We compute the earthquake damage through a finite element method (FEM)-based structural analysis using GIS and ground motion data, in which those buildings whose maximum inter-story drift angle exceeds the given threshold (0.005) are considered damaged. To calculate tsunami damage, we update the environment based on the inundation data from the tsunami wave propagation simulation, where any cell with a water height exceeding $0.5m$ is considered inundated. We leverage multi-agent simulation [WMH*13] to simulate the evacuation in the selected area. The simulation data are grouped into two categories: **(1) Environment.** The urban environment is simulated by using a hybrid model of a high-resolution 2D grid (terrain) and a topological graph of the open space (road). The grid is automatically generated from GIS data, which represents the detailed model of the spaces (i.e, hills, buildings, and water), and the localized dynamic changes (i.e., the tsunami inundation or the earthquake damage). A node in the topological graph has two states, exit, and no exit. The edge of this graph indicates the passable paths. Other information is also included, such as the width, the number of agents that occupy the road, and the number of agents that have already passed the road. In the environment, only the terrain data are static, while other types of data, including building damages and agents are all time-varying. **(2) Agent.** The residents, visitors, officials, and cars that can see, think and act are all defined as agents with different abilities, roles, and types of information. In this paper, a total of $75,000$ agents are categorized based on the local population distribution, and their behaviors are determined based on their views toward their surroundings. The dynamic movements of these agents are used to construct the evacuation trajectories. Each trajectory includes several fields, such as *location*, *agent id*, *velocity*, *velocity deviation*, *density* and *turn-back* value (i.e., a measure of direction change that is obtained by calculating the difference in the moving direction between adjacent timestamps).

We use **Features** to indicate various implicit attributes in the dataset. We specifically focus on speed, velocity deviation, density,

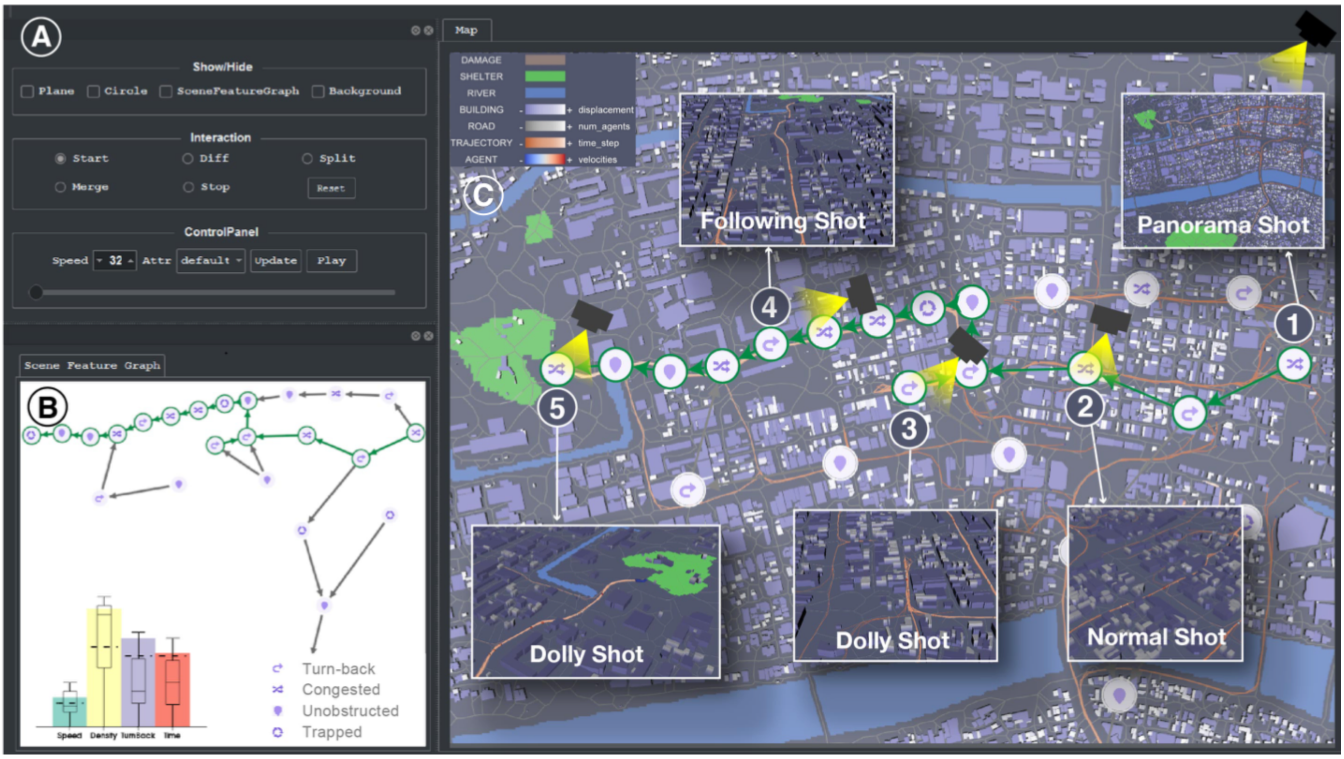


Figure 1: *System overview. (A) A control panel supports users to indicate their tracking purposes in earthquake evacuation analysis, such as selecting regions of interest, and provides displaying options for other views. (B) An interactive scene feature graph summarizes the trajectories and presents the most dominating feature events at different locations. Users can further adjust the threshold of each feature to control the generation of feature events. The green path is the optimal camera route which is automatically generated based on user-specified features or regions. (C) A storyboard view leverages different shot designs (e.g., (1) a panorama shot; (2) a normal shot; (3, 5) dolly shots and (4) a following shot to visualize different feature events along the camera route.*

evacuation time, and its variation across timestamps in earthquake evacuation. We also define a series of **Events** in the earthquake evacuation analysis, e.g., *turn-back* occurs when the road is blocked and people must turn back to find another exit, *congestion* occurs when the road is too narrow to accommodate the agents, *unobstructed* indicates that people can pass quickly and smoothly, *trapped* means that people are trapped in their position and cannot run away.

3.2. Experts' Conventional Practice and Expectations

To understand how the evacuation data are analyzed in practice, we worked with an earthquake researcher (E.1) from Earthquake Research Institute from the University of Tokyo, and an earthquake evacuation official (E.2) and a city planning manager (E.3) from the city of Kochi. E.1 analyzed the evacuation data using general-purpose software such as *ParaView* and *Advanced Visualization System (AVS)* for post-processing and visualization. Although these applications capture some useful information, they do not support the automatic detection of features of interest or sudden feature changes. Therefore, manual operations must be performed to generate useful insights, which usually takes several hours to perform visualization and is very labor-intensive. Moreover, the results generated by these applications are not intuitive enough, especially for users with limited visualization backgrounds. For example, it is hard for E.2 to demonstrate the simulations to the general audience. E.2 also mentioned that the previous visualizations fail to tell him the potential factors that may hinder a smooth evacuation. E.3 desired an interactive mechanism to help him interactively select the areas of interest and observe the best evaluation planning from the selected areas. For example, zooming in and showing the flow of crowded roads or junctions can be great assists to them, especially identifying some zones, in which the agents take a longer time to reach the shelters. Meanwhile, all experts claimed that exploring earthquake evacuation data is a user-dependent process, i.e., the exploration path and the corresponding result from one user may not be adopted by another user according to different tasks and purposes. Therefore, they envisioned a customized visualization for their exploration results.

To ensure that the ontological structure of our approach fits well into the domain tasks, we interviewed E.1-3 in separate sessions to identify their primary concerns in earthquake evacuation analysis. At the end of these interviews, the need for a highly intuitive, interactive, and customized visualization system to ground the results of their earthquake evacuation analysis emerged as a key theme. Despite differences in their expectations for this system, certain requirements were expressed across the board: **R.1 Identifying Notable Features.** According to E.1, most evacuation simulation data are with extremely large size and heterogeneous attributes. Therefore, identifying the most notable features would be of great help

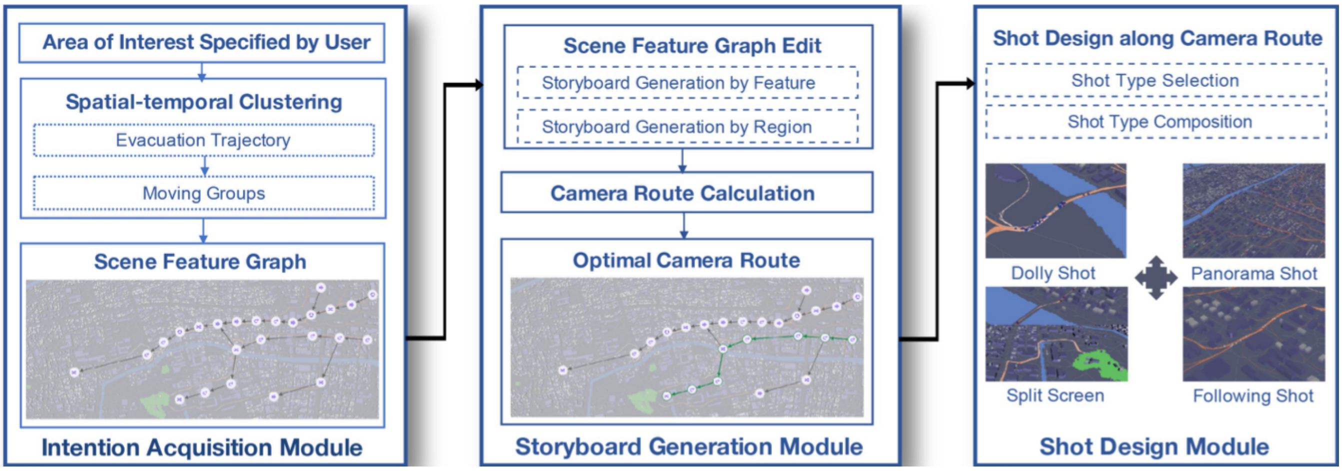


Figure 2: *The overview of our shot design approach. It consists of three modules, namely, the Purpose Acquisition Module, the Storyboard Generation Module, and the Shot Design Module. Firstly, a scene feature graph is generated based on the tracking purpose specified by the user. Secondly, an optimal camera route (indicated by the green path in the scene feature graph) is calculated based on specified feature or region, which serves our storyboard. Finally, different shot designs are employed to generate the narrative animation.*

to evacuation analysis. **R.2 Supporting User Tracking Purposes.** Users should be allowed to define their features or regions of interest for them to indicate their preferred tracking purpose. For example, E.1 wanted to know those factors that may hinder the evacuation that can help him analyze the earthquake and improve the simulation model. E.2 focused on the evacuation situation around some important areas that can help him develop reasonable evaluation plans. E.3 wanted to know the entire evacuation path to the shelters when disasters take place. Identifying users' tracking purpose would be difficult without an efficient method. **R.3 Revealing Feature Dynamics and Correlation.** All experts showed an interest in discovering factors that may hinder the evacuation. For example, *congestion*, *stagnation*, *turn-back* may be driven by certain factors, e.g., dense buildings, narrow roads, and relevant earthquake-tsunami disasters. Therefore, understanding the feature evolution behind these events and their correlation with the environment can help the experts identify evacuation bottlenecks. **R.4 Demanding Intuitive Representations of Events.** During the interview, E.1 mentioned that when using *ParaView* to analyze evacuation data, he could not distinguish features presented by the evacuation groups. In other words, different features and the resulting evacuation events require different treatment for demonstration. Providing customized representations for different events is one of the experts' major concerns.

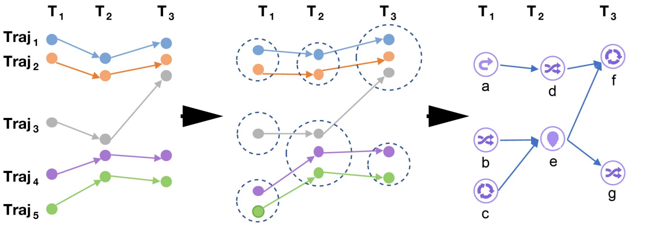
4. Overview of SEEVis

To meet the above requirements in earthquake evacuation analysis, we propose *SEEVis* (Figure 1), a novel shot design approach to tracking different evacuation features with different users' purposes. *SEEVis* contains a control panel that supports users to indicate their tracking purposes in earthquake evacuation analysis, such as selecting regions of interest, and provides displaying options for other views; an interactive scene feature graph that summarizes the trajectories and presents the most dominating feature events at different locations, and a storyboard view leverages different shot designs. Figure 2 provides more details of our shot design approach, which consists of three modules, namely, the *purpose acquisition*, the *storyboard generation*, and the *shot design* modules. The purpose acquisition module generates a scene feature graph which represents the most notable features based on user-specific tracking purpose. Based on this graph, the storyboard generation module optimizes the camera tracking path and generates a storyboard. Different shot designs are then employed to generate narrative animation.

4.1. Purpose Acquisition Module

We capture users' purposes (R.2) when exploring the earthquake evacuation data as follows: **(1) Trajectory extraction.** The users first indicate their area of interest by clicking an area of interest on the map, and the trajectories relevant to this area are then taken into consideration. The underlying data features, including *speed*, *density*, and *feature variation*, are pre-calculated simultaneously. **(2) Trajectory clustering.** We perform a spatiotemporal clustering on the aforementioned trajectories to generate a scene feature graph. Specifically, we develop a two-step trajectory clustering algorithm that considers both the spatial and temporal attributes of these trajectories. First, for each timestamp t, we take the spatial positions of all agents as an input and we apply a density-based clustering algorithm called Mean Shift [FH75] to cluster these agents. In this way, we can obtain the clustering result for every timestamp. Second, we extract the relationships of clusters between two consecutive timestamps based on the agents' movements. **(3) Events generation.** To represent the evacuation groups with intuitive events for a better understanding, we leverage a feature event mapping method that obtains the most dominating feature for each evacuation group among all as $icon(v) = argmax_i \frac{feature_i(v)-threshold_i}{threshold_i}$, where v represents the group circle, $feature_i(v)$ indicates the value of the i_{th} feature for group circle v, and $threshold_i$ denotes the threshold value of the i_{th} feature. That is, we use the icon of the most dominating feature to represent the event for group circle v. If all features do not meet the corresponding thresholds, we just ignore this group circle and do not assign any event for this group.

Figure 3 presents a scene feature graph, which is a directed



1.Trajectory Extraction 2.Trajectory Clustering 3.Event Generation

Figure 3: *Trajectory processing to generate different feature events.*

acyclic graph (DAG) that contains two parts: **(1) Vertex.** A vertex $\boldsymbol{v} = (\boldsymbol{p_v}, \boldsymbol{t_v}, \boldsymbol{n_v}, \boldsymbol{f}_v)$ denotes a cluster with position $\boldsymbol{p_v}$, timestamp $\boldsymbol{t_v}$, number of agents $\boldsymbol{n_v}$, and user-specific features $\boldsymbol{f_v}$ including *congestion* or *turn-back*. The most dominating feature is represented by the event icon on the vertex. **(2) Edge.** An edge $\boldsymbol{e} = (\boldsymbol{v_s}, \boldsymbol{v_d}, \boldsymbol{n_e}, \boldsymbol{f_e})$ denotes an agent flow from cluster $\boldsymbol{v_s}$ to the cluster $\boldsymbol{v_d}$ with the number of agents n_e and the feature variation f_e. A path p in the scene feature graph connects two vertices with an edge, such as $p_1 : a \to b_1$ and $p_2 : a \to b_2$, which corresponds to a feature evolution.

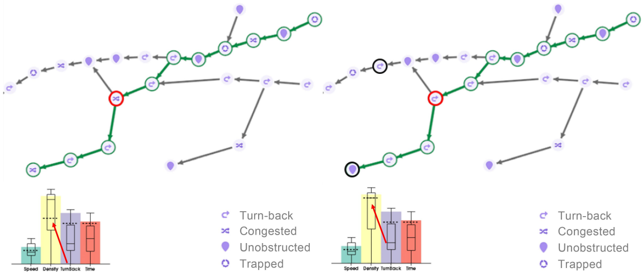


Figure 4: *The box plots shows the feature value distributions of the highlighted red circle. Users can adjust the threshold (indicated by red arrows) to control generating the most dominating feature for each circle. The most dominating feature changes after adjusting the threshold (indicated by the highlighted black circles).*

Figure 1Ⓑ shows a scene feature graph generated from the trajectories in Figure 1Ⓒ. Users initially select the trajectories of interest on the control panel (Figure 1Ⓐ). The selected trajectories are clustered, and a scene feature graph is generated, in which the circles represent the groups of agents and the edges indicate the movement of these groups. The icon in each circle denotes the resulting event of the most dominating feature in each group. Each icon represents an event (i.e., *turn-back*, *congestion*, *unobstructed*, and *trapped*). To support a fine-grained interaction with the scene feature graph, we design a box plot-embedded bar chart that shows the feature distribution of the selected node. The box plot and height of the bar indicate the value distribution (e.g., median and quartiles) and the average value of the corresponding feature of all agents in the selected circle. The dashed line in each bar represents the corresponding threshold value for each feature. Users can dynamically adjust the threshold, which subsequently projects to the whole trajectory graph and generates different dominating features. As shown in Figure 4, when users increase the threshold value of density (i.e., the yellow bar), the icon of relevant nodes would change accordingly. We provide two ways for users to specify their tracking purposes (R.2). First, users can specify areas of interest by clicking an area on the map to generate the trajectories relevant to these areas. Second, users can adjust the feature threshold to generate different dominating features. Our experimental results show that in practice, regarding a single group node in a fixed region, its corresponding feature is not very sensitive to the threshold adjustment since the feature is always dominating others in most cases. We support both representations of scene feature graph on the separate canvas and on the map, since sometimes experts prefer to adjust the scene feature graph first before observing the evacuation motion on the map while in other cases they need to witness the context. Users can switch the options through the control panel (i.e., *Show/Hide: Scene Feature Graph*).

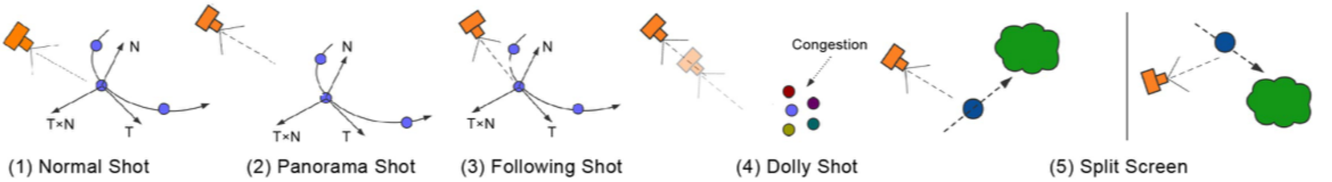
Figure 5: *Design alternatives for the scene feature graph.*

We present one alternative design (Figure 5) that has been evaluated by our experts. A bar-embedded circle presents the feature value distribution for a single group of agents (represented by a circle). Each bar represents different features, and the bar height indicates the value of the corresponding feature. However, the experts reported that this design requires a manual comparison of the features inside one circle or between different circles, which is difficult. Therefore, we identify the most dominating feature and directly represent this feature as an intuitive icon.

4.2. Storyboard Generation Module

The storyboard generation module recommends an optimal camera route according to users' exploration interest, taking both user-specific interests and feature evolution into consideration (R.2). Specifically, this module takes the scene feature graph generated from the previous purpose acquisition module as an input. Afterward, the users specify their preferred features or regions, and an optimal path in the scene feature graph is automatically computed. We then use this optimal path as the camera route of our storyboard.

We provide two interaction mechanisms for users to specify their exploration interests in storyboard generation (R.1). First, users can directly indicate their preferred features. For example, E.3 is interested in evacuation time. Given that the time for an agent to reach a shelter is mainly affected by the distance between his/her location and the shelter, we use the distance function as a feature. We follow the idea of the Dijkstra algorithm in finding the path that contains this feature and then generate the recommended path that includes the optimal camera's route. More specifically, the system automatically calculates the path with the largest cumulative value of the selected feature in the scene feature map, in which the path is called the optimal path. The associated events along this path constitute the camera's storyboard. The algorithm for the feature-based generation of the storyboard is shown in Algorithm 1. Second, users can directly select their areas of interest and observe the corresponding evacuation, and the generated storyboard must include these user-specified areas. In other words, the system calculates a path that passes through the specified area. Users can specify their regions of interest on the map in two ways. First, they can select



(1) Normal Shot (2) Panorama Shot (3) Following Shot (4) Dolly Shot (5) Split Screen

Figure 6: *Camera motion for different shot designs.*

the starting region, and the storyboard must present the most unobstructed evacuation path that begins from this region. To this end, we use Algorithm 1 and use "time" as the feature. Second, these users can select an observing region. We find a vertex u that is located nearest to the user-specified region, generate a path through u by using the depth-first-search algorithm, and find a group that passes through this region and take its path as our storyboard.

```
Data: G: scene feature graph; F: the user-specific feature
Result: P: optimal path of storyboard
S: set of vertices of G with no in-degree;
d[u]: the accumulated feature value of the path from a vertex
with no in-degree to vertex u;
pre[u]: the previous vertex of u when d[u] is updated;
F[u]: feature value in vertex u which is calculated after
clustering;
while S is not empty do
    u ← S.pop();
    while v in u's neighboring nodes do
        if d[v] < d[u] + F[v] then
            d[v] ← d[u] + F[v];
            S.push(v);
            pre[v] ← u;
        end
        e = $argmax_{e \in E}$d[e], where E is the set of vertices with
        no out-degree;
        P.push(e);
    end
end
while pre[e] is not NULL do
    P.push(pre[e]);
    e ← pre[e];
end
```

Algorithm 1: Storyboard Generation by Feature

4.3. Shot Design Module

One major concern about the use of visualization software is that they cannot distinguish various features and their correlations with the surroundings comprehensively. In this case, users of such software cannot achieve a compelling and intuitive understanding. To address this issue, we borrow the shot designs in filmmaking to earthquake evacuation visualization. In filmmaking, the expression of emotions and ideas can be enhanced by employing appropriate shot designs, such as angle transitions and cuts. In our case, we employ different shot designs to depict different features and their correlation with the environment in a narrative animation.

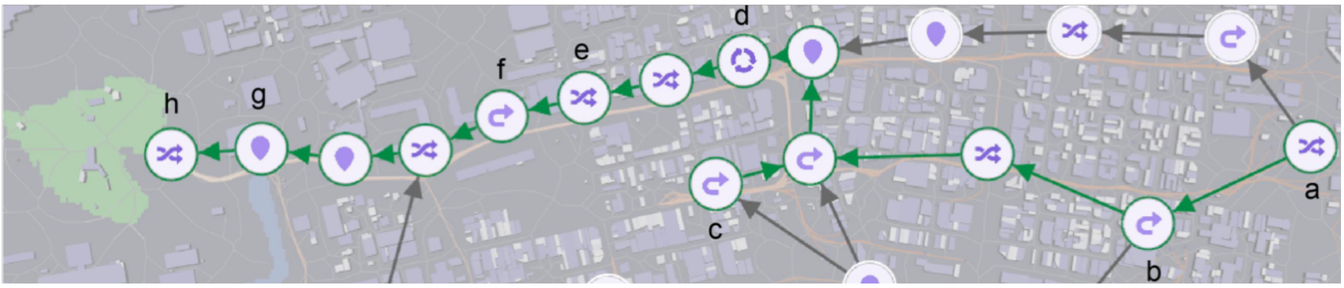
We borrow the following concepts in filmmaking: **(1) normal shot**, which tracks the objects from the side and gives audiences a strong sense of reality by using certain parameters; **(2) panorama shot**, which slowly moves the camera over a landscape with a wide-open space. We mainly use this filmmaking technique to switch the users' view from the surroundings to the starting point of the evacuation; **(3) following shot**, in which the subject being filmed is pursued by the camera. This technique is generally applied to track moving objects from the rear and is particularly useful for following the evacuees; **(4) dolly shot**, which involves moving the camera toward or away from a subject while filming. We observe the trajectory feature changes by sliding and panning the camera; **(5) split-screen**, which divides the screen in half and shows several images simultaneously to present a seamless view of reality. We use this technique to film and watch two or more groups at the same time; **(6) long take**, which is an unbroken and complete filming process that lasts much longer than the conventional editing pace.

We compute the shot designs for each vertex in this way. We determine whether the shot design is a **panorama shot** or a transition based on the topology in the constructed scene feature graph. Otherwise, we change to other shot designs, i.e., **normal shot**, **following shot**, **dolly shot**, and **split-screen** based on events. To be specific, for each vertex v in the scene graph, we define the shot designs according to the topology and events as follows:

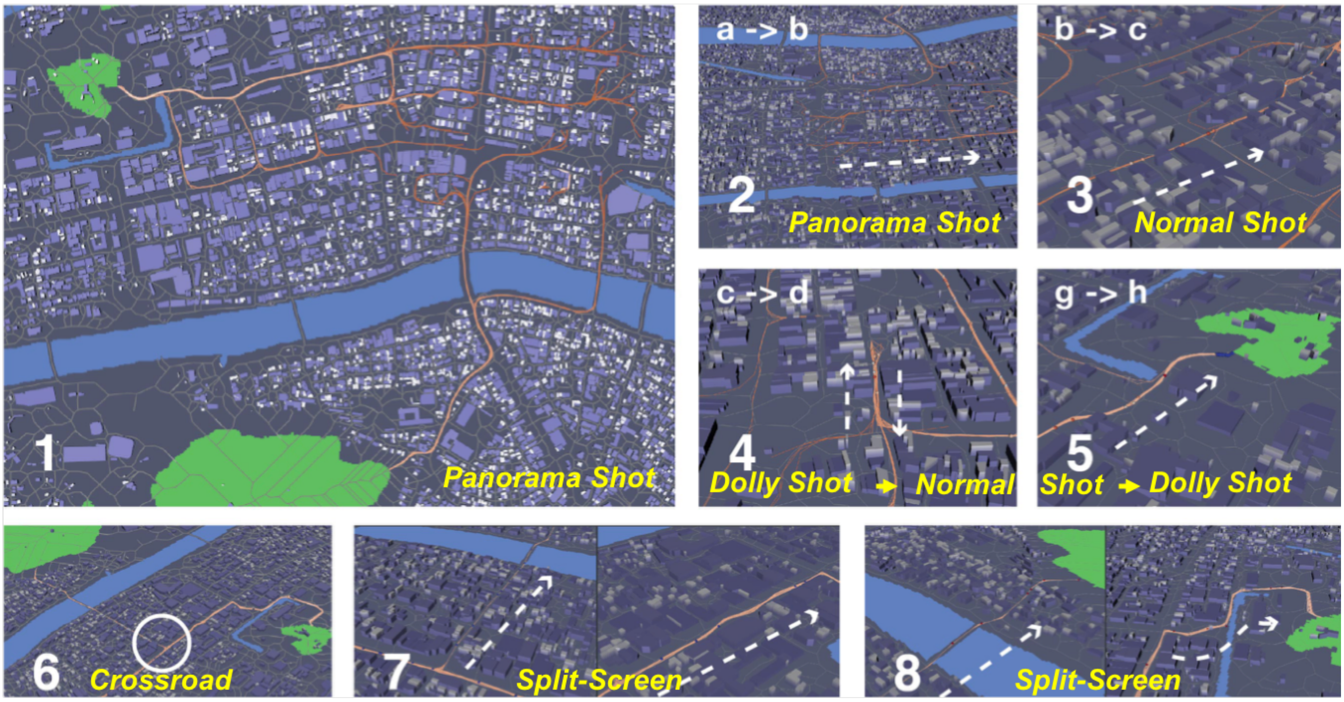
$$s = \begin{cases} \text{Transition} & n_i > 2 \wedge n_o = 1 \\ \text{Panorama} & n_i = 0 \vee n_o = 0 \\ \text{Others} & \text{otherwise} \end{cases} \qquad (1)$$

where n_i (or n_o) denotes the number of incoming (or outgoing) edges. That is, the vertex with more than two incoming edges corresponds to a transition shot. Thus, a merging process can be clearly visualized through a new perspective. The endpoints on a path correspond to a panorama shot for visual continuity. Therefore, both the surroundings and the feature evolution can be shown simultaneously. After the transition, we then control camera motions to simulate each shot design and map each shot design to a certain event. That is, when an event occurs, a type of shot design is automatically applied based on the preset parameters.

As the most basic filmmaking technique, we use the **normal shot** to track the agents. As shown in Figure 6①, we leverage a third-person view as the camera direction in the three-point perspective and the cluster center of the group as the focal point of the camera. The camera orientation d_v of the normal shot is defined by the FrenetSerret frame as $d_v = \frac{1}{\sqrt{3}}d(T + N - T \times N)$, where the constant d denotes the distance between the camera and the focal point, while T and N denote the tangent unit vector and normal unit vector of the trajectory, respectively. To gain a full perspective of the surroundings and feature evolution, we apply the **panorama shot** at the beginning to achieve a transition from the global view to the beginning of the evacuation. We calculate the camera distance, making sure it is high enough to cover the areas for a panoramic view (Figure 6②).



(a) Constructed scene feature graph



(b) Shot composition result

Figure 7: *(a) The scene feature graph. The green path is the optimal route of the storyboard about a turn-back story. (b) Subfigure 1-5 are the shot design results. Subfigure 1 and 2 are panorama shots for view transition. Subfigure 3 is a normal shot tracking the agents from the side. Subfigure 4 shows the turn-back event where the camera slows down using a dolly shot. Subfigure 5 begins with a normal shot and then switches to a dolly shot which helps shoot the evacuation situation around the shelter by pulling the camera away. Subfigure 6-8 are the shot design results of a "split" story. Subfigure 6 shows that the agents are divided into two groups at the road crossing. In subfigure 7-8, two cameras shoot the evacuation process of the two groups simultaneously by using the "split-screen" technology.*

In the **following shot**, the camera tracks the subject from the rear as (Figure 6③). Some events, such as *congestion* and *turn-back*, may occur during the evacuation. Therefore, we leverage the **dolly shot** by pushing or pulling away the camera to better observe and understand the reactions of evacuees to these events (Figure 6④). In other cases where users can specify important areas or certain areas of interest, we use the **dolly shot** to track the trajectories and the surroundings of the selected area. When tracking the evacuation in some special areas such as road intersections, we use the **split screen** technique because either road may lead to the shelter place and the evacuees may split into two groups. With this technique, we can track these two groups simultaneously and monitor their evacuation situations (Figure 6⑤). The entire tracing process is a **long take** story. It applies different rhythms, i.e., slowing down to capture important events or speeding up to simulate a sense of evacuation tension, and gives users a comprehensive understanding of the evacuation process.

We apply the rule of **Simplicity** to conduct shot design composition for a complete tracking process. Particularly, composition refers to the organization of pictorial elements in a frame [Ber04]. Many of the best photographs focus on few basic elements. "*Avoid cluttered backgrounds; by changing angles or perspectives and getting up close to the subject,*" one can often produce a visually stunning photograph without introducing distracting or irrelevant elements that reduce the impact of the composition [Ber04]. In our scenario, we employ a monochromatic color scheme for all backgrounds to avoid a busy or congested feeling and use a diverging color scheme in encoding different values of the attribute of the central object. As shown in Figure 1Ⓒ, the color of buildings ranges from purple to white based on its displacement value, while the color of roads changes from gray to white based on the number of the passing agents. The color of trajectories and agents also changes based on the timestamps and the velocities, respectively. We also hide the labels to avoid visual clutter since our end-users are quite familiar with the urban environment.

5. Evaluation

5.1. Case 1: Exploring Evacuation by Using Shot Designs

The first case shows how E.1 obtained a comprehensive picture of the evacuation process.

Specifying Features of Interest in Evacuation. After loading the earthquake-tsunami evacuation simulation data into the system, E.1 obtained a trajectory overview (Figure 7①) in the urban environment and he then specified an area of interest. Afterward, the system automatically computed the scene feature graph of the trajectories from the central areas to the shelter in the west as shown in 7(a) (R.1). He observed that the agents in the central area had to pass through some building-intensive areas to reach the shelter in the west. The collapsed buildings may hinder the evacuation progress as E.1 witnessed *turn-back* events in the scene feature graph. E.1 was quite interested in this event and set it as his preferred feature. As a result, the group with the maximum *turn-back* value was selected, and the optimal path was automatically generated. In 7(a), the green path is the generated optimal camera route, which indicates the expert's evacuation path of interest. The entire evacuation process of this group is used as the camera's storyboard.

Exploring the Correlation between Features and Environment. After obtaining the camera's storyboard route, the shot design module determines the shot designs of the key locations along the route (R.2). As shown in 7(b)①, the tracking began with a panorama shot that shifted the view from the global map to the target group. At the end of this panorama shot (7(b)②), the camera started tracking the group. From this shot, E.1 observed that the group was passing through narrow roads between buildings and that the camera automatically switches to a normal shot to present the evacuation around the crowded buildings (7(b)③). E.1 also observed that the agents turned back and selected another evacuation route because the buildings along the way were collapsed and the roads were blocked. As shown in 7(b)④, the system reduced the camera movement speed when detecting a *turn-back* event, thereby allowing users to observe the changes in their surroundings and the responses of evacuees. After the *turn-back* event, the agents finally found a way to reach the shelter and our camera continued to use the normal shot to track this group until they reached their destination. We continued filming the situation around the shelter by using the dolly shot as shown in 7(b)⑤. By pulling the camera away, E.1 saw how people evacuated from densely populated areas.

Comparing Different Evacuation Scenarios. In some cases, evacuees may split into several groups during the evacuation. Instead of using a single camera, we employ "split-screen" to present a comprehensive picture of the evacuation (R.2). In 7(b)⑥, shelters are located in the upper left and bottom right corners. Upon arriving at the road intersection (indicated by the circle), some evacuees rushed to the shelter in the upper left corner while some fled to the shelter in the bottom right corner. Our system automatically splits the screen to simultaneously track these two groups (7(b)⑦). E.1 found that the group on the left screen had to cross the river while the other group needed to cross the crowded buildings that may collapse at any time. The "split-screen" keeps tracking the two groups until they reached the shelters (7(b)⑧).

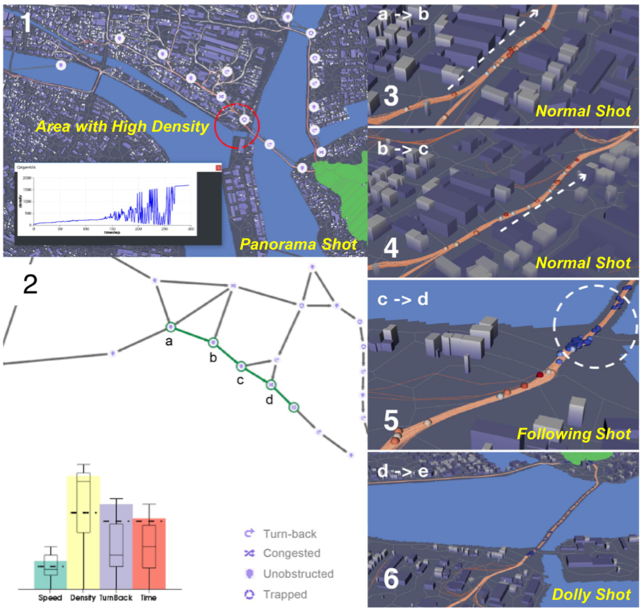


Figure 8: *Discovering factors that cause evacuation obstruction. (1) A density heat map of the trajectories is shown and the expert identified that in front of a bridge there is an area with high density (indicated by the line chart of the density over time), and a panorama shot is applied. (2) The scene feature graph. (3-4) Normal shots are applied for tracking the group. (5) A following shot is applied tracking from the behind until detecting congestion. (6) A dolly shot helps users discover that the congestion is due to the flooded bridge.*

5.2. Case 2: Discovering Evacuation Obstructive Factors

Events such as *congestion* due to blocked or flooded roads and damaged buildings or roads can significantly affect earthquake evacuation by slowing down the evacuation velocity or making people turn back. In this subsection, we show how the official (E.2) identified those factors that hinder a smooth evacuation.

Discovering Areas of Congestion. To observe a *congestion* event and its potential causes, E.2 adjusted the threshold and set *agent density* as his feature of interest. By using a 2D heat map of trajectory density and a line chart of density over time, E.2 identified a high-density area in front of a bridge (Figure 8①) as the system presented an overview of this area's density across time (R.1). From the generated scene feature graph (Figure 8②) E.2 observed a *congestion* event in front of the bridge. He then selected that area, and our storyboard algorithm automatically calculated a green path of the camera route. This path was then taken as the evacuation path of the group of evacuees with the largest density, and the entire evacuation of this group was used as the camera storyboard.

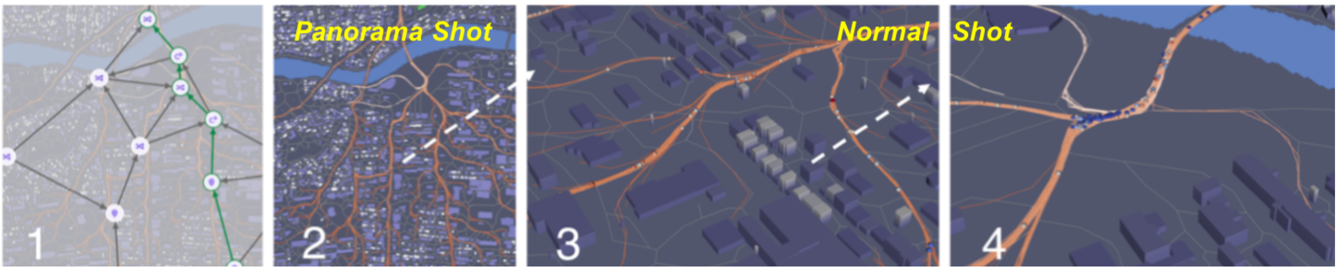


Figure 9: *(1-2) A panorama shot is applied at the beginning of shooting. (3-4) Normal shots show the agent movement.*

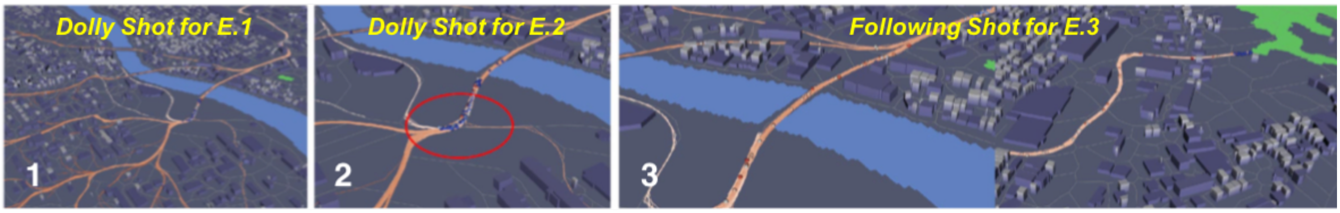


Figure 10: *Different shot designs for different exploration purposes in a same scenario. (1) A dolly shot helps E.1 discover the factors of congestion. (2) Another dolly shot for E.2 shoots the evacuation situation around the circled area. (3) A following shot for E.3 shows that the agents were crossing the river and a subsequent dolly shot shows the evacuation situation around the shelter.*

Revealing Factors that Cause Congestion. *SEEVis* applied a panorama shot at the beginning of the tracking (Figure 8①). When the agents started to evacuate, the camera switched to a normal shot. As shown in Figure 8③, the center of this group is always at the focal point of the camera. E.2 observed that these agents traveled on the same road and congestion started to occur as a result of the increasing agent density. The camera speed was reduced to closely observe the reactions of these agents (R.3). Figure 8④ shows that after a few seconds, people began to disperse, and the road became smooth again. During this process, our camera kept tracking the group by using a normal shot. When the evacuees arrived at the riverbank, E.2 observed that most agents were unable to cross as can be deduced from the reduction in their evacuation speed using a following shot. From Figure 8⑤⑥, E.2 found that upon reaching the bridge, the evacuation speed was 0 as the agents were unable to cross the bridge that was flooded by the tsunami. Therefore, E.2 observed a congestion occurring in this area by using a dolly shot.

5.3. Case 3: Satisfying Different Exploration Purposes

This case demonstrates how *SEEVis* supports different exploration purposes by using different shot designs (R.2, R.4). To be specific, for the same trajectories, the system generates different storyboards for different end-users and users different combinations of shots to capture the content that they are interested in.

Compositing Different Shots. In the northern part of the city, the agents needed to cross the river to arrive at the shelter. In this part of the evacuation trajectories, E.1 was concerned whether certain events, such as *congestion* or *turn-back*, would occur and was curious about those factors that lead to such events. Therefore, he selected certain features, such as *speed* or *density*, for further processing (R.2). Specifically, he selected *speed* as his tracking purpose, and the scene feature graph in Figure 9① presents the clustering result of the trajectories shown in Figure 9②. The green path in the scene feature graph represents the evacuation path of the group with the largest deviation in *speed* (i.e., a high value of speed deviation indicates that the evacuees may be trapped on the river bank). Meanwhile, E.2 focused on the evacuation situation around the river bank given that the bridge may create an evacuation bottleneck if flooded. Therefore, he selected the river bank as his area of interest (R.2). The generated storyboard for E.2 consisted of two parts: 1) the evacuation of this group, and 2) the evacuation situation around his selected area. Figure 9① shows that in the beginning, the camera focused on the evacuation group via a panorama shot. After switching from the global view to the target group, the camera switched to a normal shot by following the movement of the group as shown in Figure 9③. Figure 9④ shows that the group is trapped on the river bank. We then provide different shot compositions for E.1 and E.2. For E.1, we used the dolly shot by pulling the camera away. From Figure 9①, E.1 observed *congestion* resulting from the flooding of the bridge. For E.2, instead of using the dolly shot, we continued shooting this area (Figure 10②). In this way, he could check whether this area presented an evacuation bottleneck and evaluate how the damages affect the agents' evacuation (R.2-3).

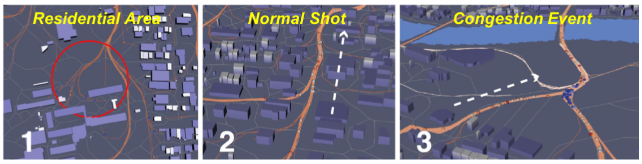


Figure 11: *The shot design result of E.3. (1) The circled area is the place where residents live. (2-3) Normal shots are applied and a congestion event occurred in subfigure 3.*

Exploring a Evacuation Path. Finding a smooth evacuation path is an appealing request for city planning. To meet this demand, an interactive selection mechanism was provided. E.3 selected a resident place (Figure 11①), and the trajectories starting from this place were clustered into several groups (R.2). In this case, evacuation time is considered the top priority. Accordingly, E.3 selected *evacuation time* as the tracking feature, and the system automatically generated the path of the group who escaped to the shelter with the minimum evacuation time. A minimum evacuation time corresponds to an unobstructed path with the evacuation time as short as possible between the shelter and the living place. As usual, a panorama shot was used to track the evacuees and then a normal shot was employed with a third-person view (Figure 11①②). We kept tracking the group until they arrived at the river bank as shown in Figure 11③. When the agents were crossing the bridge, the following shot was used until they reached the shelter (Figure 10③).

5.4. Discussion

We conducted a half-hour semi-structured interview with the experts (E.1-3) to evaluate our system and check whether our shot designs are helpful to earthquake evacuation analysis.

System and Visual Designs. All experts appreciated the ability of our system to support interactive exploration of events, trajectory dynamics, and correlation with the urban environment. "*It only takes several minutes to explore the data and obtain a concise summary of the evacuation evolution,*" said E.1-2. According to E.1, "*the customized shot designs allow us to focus on the most important features,*" while E.2 shared, "*I can now see those features more clearly with the help of the automated adjusted camera angles and view fields*." The experts were satisfied with the visual designs and interactions. We drew inspiration for our design from their responses in the interviews, and we deliberately selected familiar visual metaphors (e.g., icons, box plots, heat maps) to help experts quickly familiarize themselves with our system. After introducing them to the basic views and functions of our system, the experts developed a customized path using this system for exploration purposes. E.2 shared that this system allows him to freely select any areas of interest and adjust the features that helped him identify the potential bottlenecks in the evacuation. "*We used to make decisions based on static charts or data tables, which are quite tedious and not intuitive,*" said E.2. "*Now the delicate shot designs made the evacuation situation around different areas more detailed and vivid.*" E.3 commented that "*this system allows me to explore data without having much background*", and "*the evacuation presented by shot designs is realistic and immersive.*" E.3 also shared that "*by observing how the agents evacuate from their places of residence, I can observe a smooth and safe way when a disaster occurs*."

Feedback and Takeaway. E.1 shared that "*I can explore and compare different evacuation scenarios and understand how the environment affects evacuations.* According to E.1-2, one major objective of their simulations is to find strategies to accelerate the evacuation process, which requires a series of simulations. Traditionally, they had to do more simulations to "*ensure that their updated plan eliminates the observed congestion*". However, since their target area is large (e.g., more than $100km^2$) and the number of agents involved can exceed $100,000$, "*it was rather difficult to analyze the results to identify where and when congestion occurs, as well as cases with statistical significance.*" Our visualizations can help easily detect where congestion happens. Furthermore, by observing the flow of people in the animations, E.1 can "*easily understand what leads to congestion, and easily find a solution to reduce congestion.*"

Generalizability. Although *SEEVis* mainly leverages agents' trajectories to specify users' purposes, other features like iso-surfaces and point topology can be also easily taken into consideration. As a replacement to the current feature extraction algorithms, our approach can fuse different features and define the tracking purposes on the storyboard. Moreover, given that our approach can be generalized to reveal features that come from multi-source heterogeneous datasets, shot designs can provide intuitive storytelling that allows users to comprehensively understand the data (e.g., taxi trajectories).

Algorithm Complexity. *SEEVis* consists of (1) trajectory processing, (2) storyboard generation, and (3) shot design. The first part performs a hybrid search with time complexity of $O(n)$. The storyboard generation presents a linear time solution to the shortest path problem because the generated scene feature graph is directed acyclic, while the final shot design takes the constant time for a fixed number of frames. The overall complexity for these two parts is $O(n)$, which supports real-time interactions.

Limitation. First, we can enrich *SEEVis*'s representations by showing labels of main streets and visual indication of evacuation way and surroundings such as the building displacement and river flooding instead of only using colors. Besides, currently we extract major features and then support interactions to define interesting regions and group for tracking. It is reasonable to show major events which can be outside of the selected group. Second, we can summarize the results of several simulations. Stochastic approaches such as Monte-Carlo simulations, can be applied to deal with potential uncertainties. Analyzing the results of thousands of simulations and categorizing the features based on their probability of occurrence is among the key challenges that we are currently facing. Third, we are unable to provide a concise analysis summary that can help evacuation planners in making decisions. "*Adding an analysis summary similar to the proposed scene feature graph would make the system indispensable in future evacuation studies,*" said E.1. Fourth, we only worked with a few experts in the evaluation and are therefore unable to provide a quantitative assessment of the system. The choice of shot designs can be further justified by a formal evaluation.

6. Conclusion and Future Work

We propose a shot design approach to tracking evacuation feature changes and correlations in earthquake evacuation. We first build a scene feature graph that identifies the most dominating evacuation event, and then calculate the optimal camera route based on user-specific regions or features. For different evacuation events along this route, we apply the corresponding shot design to reveal the underlying feature evolution and correlation with the environment. Several cases confirm the efficacy of our system. In future work, we plan to apply our approach to other applications and examine the various uncertainties involved in multi-agent simulations to obtain more interesting results for evacuation study. We also plan to show major events which can be outside of the selected group, and extend the shot design process to a formal and comprehensive design study.

Acknowledgements

We are grateful for the valuable feedback and comments provided by the anonymous reviewers. This research is partially supported by the National Natural Science Foundation of China (Grant Nos. 61972221, 61572274, 61672055).

References

[AA11] Adrienko N., Adrienko G.: Spatial generalization and aggregation of massive movement data. *IEEE Transactions on visualization and computer graphics 17*, 2 (2011), 205–219. 2

[AAFG18] Andrienko G., Andrienko N., Fuchs G., Garcia J. M. C.: Clustering trajectories by relevant parts for air traffic analysis. *IEEE transactions on visualization and computer graphics 24*, 1 (2018), 34–44. 2

[AAH*13] Andrienko G., Andrienko N., Hurter C., Rinzivillo S., Wrobel S.: Scalable analysis of movement data for extracting and exploring significant places. *IEEE Transactions on Visualization and Computer Graphics 19*, 7 (July 2013), 1078–1094. URL: `http://dx.doi.org/10.1109/TVCG.2012.311`, `doi:10.1109/TVCG.2012.311`. 2

[Ber04] Berdan R.: Composition and the elements of visual design. *Photo composition articles* (2004). 8

[BKLR14] Bock A., Kleiner A., Lundberg J., Ropinski T.: An interactive visualization system for urban search amp; rescue mission planning. In *Safety, Security, and Rescue Robotics (SSRR), 2014 IEEE International Symposium on* (Oct 2014), pp. 1–7. `doi:10.1109/SSRR.2014.7017652`. 1, 2

[Bli88] Blinn J.: Where am i? what am i looking at?(cinematography). *IEEE Computer Graphics and Applications 8*, 4 (1988), 76–81. 3

[BS05] Bordoloi U. D., Shen H.-W.: View selection for volume rendering. *Visualization Conference, IEEE 0* (2005), 62. `doi:http://doi.ieeecomputersociety.org/10.1109/VIS.2005.110`. 2

[BTMB00] Bares W. H., Thainimit S., McDermott S., Boudreaux C.: A model for constraint-based camera planning. In *Proceedings of AAAI spring symposium on smart graphics* (2000), pp. 84–91. 3

[CO09] Christie M., Olivier P.: Camera control in computer graphics: Models, techniques and applications. In *ACM SIGGRAPH ASIA 2009 Courses* (New York, NY, USA, 2009), SIGGRAPH ASIA '09, ACM, pp. 3:1–3:197. URL: `http://doi.acm.org/10.1145/1665817.1665820`, `doi:10.1145/1665817.1665820`. 2

[DFD*14] Doraiswamy H., Ferreira N., Damoulas T., Freire J., Silva C.: Using topological analysis to support event-guided exploration in urban data. *Visualization and Computer Graphics, IEEE Transactions on 20*, 12 (Dec 2014), 2634–2643. `doi:10.1109/TVCG.2014.2346449`. 2

[FH75] Fukunaga K., Hostetler L.: The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Transactions on information theory 21*, 1 (1975), 32–40. 5

[GESR13] Guest J., Eaglin T., Subramanian K., Ribarsky W.: Visual analysis of situationally aware building evacuations. In *Proc. SPIE 8654* (2013), vol. 8654, pp. 86540G–86540G–14. URL: `http://dx.doi.org/10.1117/12.2001535`, `doi:10.1117/12.2001535`. 1, 2

[GLG*16] Gratzl S., Lex A., Gehlenborg N., Cosgrove N., Streit M.: From visual exploration to storytelling and back again. In *Computer Graphics Forum* (2016), vol. 35, Wiley Online Library, pp. 491–500. 3

[IYT*16] Itoh M., Yokoyama D., Toyoda M., Tomita Y., Kawamura S., Kitsuregawa M.: Visual exploration of changes in passenger flows and tweets on mega-city metro network. *IEEE Transactions on Big Data 2*, 1 (2016), 85–99. 2

[Joh04] Johnson C.: Top scientific visualization research problems. *IEEE Computer Graphics and Applications 24*, 4 (2004), 13–17. 2

[JS06] Ji G., Shen H.-W.: Dynamic view selection for time-varying volumes. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 1109–1116. 2

[KH12] Kehrer J., Hauser H.: Visualization and visual analysis of multi-faceted scientific data: A survey. *Visualization and Computer Graphics, IEEE Transactions on PP*, 99 (2012), 1. `doi:10.1109/TVCG.2012.110`. 1

[LC15] Lino C., Christie M.: Intuitive and efficient camera control with the toric space. *ACM Transactions on Graphics (TOG) 34*, 4 (2015), 82. 3

[LCL*10] Lino C., Christie M., Lamarche F., Schofield G., Olivier P.: A real-time cinematography system for interactive 3d environments. In *Proceedings of the 2010 ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2010), Eurographics Association, pp. 139–148. 3

[LGL*11] Liu H., Gao Y., Lu L., Liu S., Qu H., Ni L.: Visual analysis of route diversity. In *Visual Analytics Science and Technology (VAST), 2011 IEEE Conference on* (Oct 2011), pp. 171–180. `doi:10.1109/VAST.2011.6102455`. 2

[MČV*15] Mindek P., Čmolík L., Viola I., Gröller E., Bruckner S.: Automatized summarization of multiplayer games. In *Proceedings of the 31st Spring Conference on Computer Graphics* (2015), ACM, pp. 73–80. 3

[MTC*16] Ma Y., Tao L., Cao Z., Chen L., Fei W., Wei C.: Mobility viewer: An eulerian approach for studying urban crowd flow. *IEEE Transactions on Intelligent Transportation Systems 17*, 9 (2016), 2627–2636. 2

[OSBM14] Ozer S., Silver D., Bemis K., Martin P.: Activity detection in scientific visualization. *IEEE transactions on visualization and computer graphics 20*, 3 (2014), 377–390. 2

[RU14] Ranon R., Urli T.: Improving the efficiency of viewpoint composition. *IEEE Transactions on Visualization and Computer Graphics 20*, 5 (2014), 795–807. 3

[RWF*13] Ribičić H., Waser J., Fuchs R., Blöschl G., Gröller E.: Visual analysis and steering of flooding simulations. *IEEE Transactions on Visualization and Computer Graphics 19*, 6 (2013), 1062–1075. 1, 2

[SAB12] Serin E., Adali S. H., Balcisoy S.: Automatic path generation for terrain navigation. *Computers & Graphics 36*, 8 (2012), 1013–1024. 3

[SAMS*17] Sacha D., Al-Masoudi F., Stein M., Schreck T., Keim D., Andrienko G., Janetzko H.: Dynamic visual abstraction of soccer movement. In *Computer Graphics Forum* (2017), vol. 36, Wiley Online Library, pp. 305–315. 2

[SH10] Segel E., Heer J.: Narrative visualization: Telling stories with data. *IEEE transactions on visualization and computer graphics 16*, 6 (2010), 1139–1148. 3

[SP08] Sokolov D., Plemenos D.: Virtual world explorations by using topological and semantic knowledge. *The Visual Computer 24*, 3 (2008), 173–185. 3

[TFTN05] Takahashi S., Fujishiro I., Takeshima Y., Nishita T.: A feature-driven approach to locating optimal viewpoints for volume visualization. In *Visualization, 2005. VIS 05. IEEE* (2005), IEEE, pp. 495–502. 2

[TMB02] Tversky B., Morrison J. B., Betrancourt M.: Animation: can it facilitate? *International journal of human-computer studies 57*, 4 (2002), 247–262. 2

[TYC*17] Tao L., Yue H., Chen S. C., Navlakha J., Iyengar S. S., Ning X., Zeng C., Zhou W., Li Z., Jiang Y.: Data-driven techniques in disaster information management. *Acm Computing Surveys 50*, 1 (2017), 1. 2

[VFSG06] Viola I., Feixas M., Sbert M., Groller M. E.: Importance-driven focus of attention. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 933–940. 2

[VFSH01] Vázquez P.-P., Feixas M., Sbert M., Heidrich W.: Viewpoint selection using viewpoint entropy. In *VMV* (2001), vol. 1, pp. 273–280. 3

[VFSH03] Vázquez P.-P., Feixas M., Sbert M., Heidrich W.: Automatic view selection using viewpoint entropy and its application to image-based modelling. *Computer Graphics Forum 22*, 4 (2003), 689–700. URL: `http://dx.doi.org/10.1111/j.1467-8659.2003.00717.x`, `doi:10.1111/j.1467-8659.2003.00717.x`. 2

[VJN*15] Vrotsou K., Janetzko H., Navarra C., Fuchs G., Spretke D., Mansmann F., Andrienko N., Andrienko G.: Simplifly: A methodology for simplification and thematic enhancement of trajectories. *Visualization and Computer Graphics, IEEE Transactions on 21*, 1 (Jan 2015), 107–121. `doi:10.1109/TVCG.2014.2337333`. 2

[WCL*16] Wang Y., Chen Z., Li Q., Ma X., Luo Q., Qu H.: Animated narrative visualization for video clickstream data. In *SIGGRAPH Asia 2016 Symposium on Visualization* (2016), ACM, p. 11. 3

[WKS*14] Waser J., Konev A., Sadransky B., Horváth Z., Ribičić H., Carnecky R., Kluding P., Schindler B.: Many plans: Multidimensional ensembles for visual decision support in flood management. *Computer Graphics Forum 33*, 3 (2014), 281–290. 2

[WLY*13] Wang Z., Lu M., Yuan X., Zhang J., Wetering H. V. D.: Visual traffic jam analysis based on trajectory data. *IEEE Transactions on Visualization and Computer Graphics 19*, 12 (2013), 2159–2168. `doi:http://doi.ieeecomputersociety.org/10.1109/TVCG.2013.228`. 2

[WMH*13] Wijerathne M., Melgar L., Hori M., Ichimura T., Tanaka S.: {HPC} enhanced large urban area evacuation simulations with vision based autonomously navigating multi agents. *Procedia Computer Science 18*, 0 (2013), 1515 – 1524. 2013 International Conference on Computational Science. URL: `http://www.sciencedirect.com/science/article/pii/S1877050913004626`, `doi:http://dx.doi.org/10.1016/j.procs.2013.05.319`. 3

[WO90] Ware C., Osborne S.: Exploration and virtual camera control in virtual three dimensional environments. *SIGGRAPH Comput. Graph. 24*, 2 (Feb. 1990), 175–183. URL: `http://doi.acm.org/10.1145/91394.91442`, `doi:10.1145/91394.91442`. 2

[WYM08] Wang C., Yu H., Ma K.-L.: Importance-driven time-varying data visualization. *IEEE Transactions on Visualization and Computer Graphics 14*, 6 (2008), 1547–1554. 2

[XYH*18] Xie K., Yang H., Huang S., Lischinski D., Christie M., Xu K., Gong M., Cohen-Or D., Huang H.: Creating and chaining camera moves for qadrotor videography. *ACM Transactions on Graphics (TOG) 37*, 4 (2018), 88. 3

[YLRC10] Yu L., Lu A., Ribarsky W., Chen W.: Automatic animation for time-varying data visualization. In *Computer graphics forum* (2010), vol. 29, Wiley Online Library, pp. 2271–2280. 3