ParkAnalyzer: Characterizing the Movement Patterns of Visitors

VAST 2015 Mini-Challenge 1

Jieqiong Zhao
Guizhen Wang
Junghoon Chae
Hanye Xu
Siqaio Chen *
Purdue University

William Hatton†
US Air Force Academy

Sherry Towers‡
Arizona State University

Mahesh Babu Gorantla
Benjamin Ahlbrand
Jiawei Zhang, Abish Malik
Sungahn Ko, David S. Ebert §
Purdue University

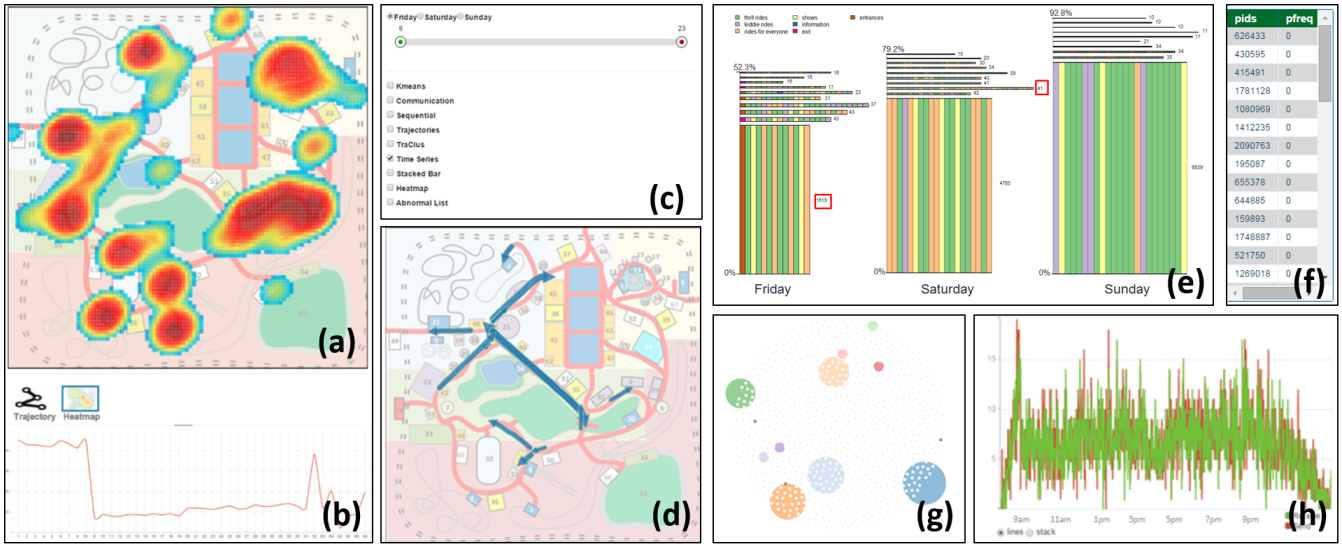

Figure 1: A screenshot of our visual analytics system. Our system comprises of a map view (a), check-in frequency view (b), control panel (c), trajectory view (d), sequence clustering view (e), list view (f), clustering node-link view (g), and communication frequency graph (h).

1 Introduction

The 2015 VAST challenge features movement tracking (Mini-Challenge 1 (MC1)) and communication information (Mini-Challenge 2 (MC2)) datasets of all visitors in an amusement park over a three-day weekend. The data includes around 25 million individual movement records, along with 4 million communication records. Analyzing and exploring such large-scale datasets require intelligent data mining methods that characterize the overall trends and anomalies, as well as interactive visual interfaces to support investigation at different spatiotemporal granularities. The objective of MC1 was to characterize the behavior of different groups of visitors, compare different activity patterns over the three days, and discover anomalies or unusual behavior patterns that relate to the crime that occurred during the weekend. We utilized both movement data provided in MC1 and communication data provided in MC2 to answer the questions asked in MC1.

In order to characterize the data, we created a visual analytics system called ParkAnalyzer that combines advanced clustering algorithms and a multi-view user interface to facilitate the exploration of the challenge data. ParkAnalyzer is a web based system that clusters people based on movement data to accelerate the analytical process of grouping visitors and narrow down the scope of suspects that may relevant to the crime. The system allows users to iteratively filter, link, and compare the different aspects of movement clusters, and drill down to investigate individual behaviors. In order to make the system interactive, we utilize a server-client architecture where the processing and computation are performed on the server back-end, and the client front-end comprises of an interactive visual analytics system. Our system has been designed to support the visual information-seeking mantra where users are provided with an overview of the data and can zoom-in and filter their data as necessary. Brushing and linking interactions are supported between the different views. Since the different clustering methods show different aspects of the data, analysts can explore and discover the relationships between visitors. We employ several visualization techniques in our system, including a node-link diagram to illustrate overview results of the clusters, map view to visualize movement hotspots and trajectories, time series view to show communication patterns for the selected individuals, and list view to show detailed information of the visitors.

2 Visual Analytics Environment

Our interactive visual system, shown in Figure 1, allows users to compare the relationships between the different movement clusters and communication data. Figure 2 provides an overview of our analysis framework. The data are first preprocessed for individual days of the weekend and run through the movement clustering and

*e-mail: zhao413|wang1908|jchae|xu193|chen1722@purdue.edu
†e-mail: C16william.hatton@usafa.edu
‡e-mail: smtowers@asu.edu
§e-mail: mgorantl|bahlbran|zhan1486|amalik|ko|ebertd@purdue.edu

IEEE Conference on Visual Analytics Science and Technology 2015
October 25–30, Chicago, Il, USA
978-1-4673-9783-4/15/$31.00 ©2015 IEEE

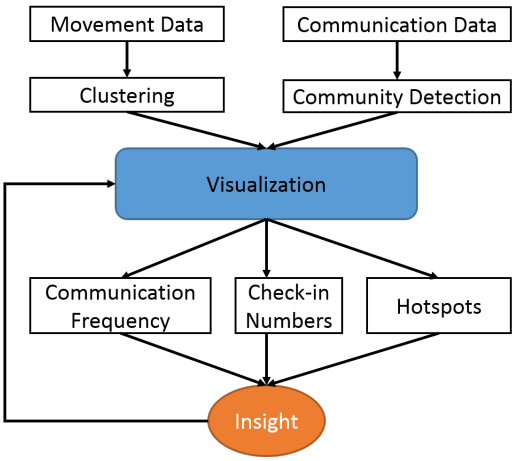

Figure 2: The interactive visual framework of our system.

community detection methods described in Section 3. The clustering results are shown to users as a node-link diagram (Figure 1 (g)) and sequence clustering diagram (Figure 1 (e)). Users can interactively select cluster subsets in order to further examine their characteristics in the other linked views. The system also utilizes density estimated heatmaps (Figure 1 (a)) to quickly identify check-in hotspots. Figure 1 (d) shows the trajectory view where we cluster trajectories into sets of similar sub-trajectories to discover common patterns [2]. Visitors that pass the filters applied by the users are shown in the list view (Figure 1 (f)). The communication frequency time series graph (Figure 1 (h)) shows the communication volumes of receive and send of selected people over customized time interval. Similarly, the check-in frequency time series view (Figure 1 (b)) shows the number of check-ins at the selected park attractions over time. The system also provides an interactive time slider widget (Figure 1 (c)) that allows users to temporally scroll through the data while dynamically updating the other linked windows. After several iterations of exploring the datasets, analysts can find the visitors with suspicious behaviors that may be related to the crime.

3 Characterizing Visitor Behavior Based on Movement Clustering

Due to a large number of visitors and movement records provided, there are a medley of clustering techniques that can be applied to the data. We highlight four major approaches that we use in our system, each of which provides a unique angle to summarize the behavior of the groups.

3.1 Clustering people based on their attraction preferences

In order to discover groups of people who prefer certain attraction types, we utilize the *k*-means clustering technique [5] to cluster visitors based on the time they spend at the attractions. This technique groups people in the same cluster if they have similar attraction preferences (i.e., if they tend to visit the same attractions). Grouping people with similar check-in distributions can not only divide visitors into multiple clusters, but also can assist with the discovery of latent information from the data. For example, people who have a higher preference on Kiddie Land attraction from among the different attractions may be families with children. Using this technique, we find groups of people who have higher attraction towards visiting thrill rides and rides for everyone, and do not visit activities related to Scott. We hypothesize that this group may be youths. We also find people who enjoy visiting rides for everyone and attending Scott's shows.

3.2 Abstracting attraction category visiting sequence

In this approach, we group visitors based on check-in sequences of attraction categories. This technique groups the visitors who visit the attractions in the same sequence. We use the longest common sub-sequence (LCS) [3] to measure the similarity of the sequences of two visitors. Then, we apply a density based clustering algorithm, DBSCAN [4], to group visitors into corresponding clusters. Figure 1 (e) presents the top 10 clusters for the three days of the weekend (ranked based on the number of customers in every cluster). The height of each row encodes the number of people within the cluster. The color of each bin within the cluster shows an attraction category. Every group selects the most frequent sequence as their most representative sequence. This representation allows users to understand and compare the general patterns and trends of people who visit the amusement park over the weekend.

3.3 Grouping people who move together

If people check-in at the same attractions throughout the day in the same order, we assume that they are traveling together in the same group. In order to detect these groups, we cluster the individuals based on similar attraction check-in sequences (i.e., if a group of individuals check-in to the same attractions in the same sequence). As described in the previous approach, this approach clusters people who have the *same* check-in sequences throughout the day (as opposed to similar check-in sequences). This method enables us to detect people who travel together in groups (e.g., friends, family, school field trips). The size of groups detected using this technique ranges from 2 to 40 people, with an average group size between 4-6 people.

3.4 Correlation between communication groups and groups who travel together

Finally, we utilize the communication data (MC2) to group people who frequently communicate among each other. This is accomplished by clustering people together using the community detection algorithm [1]. The groups detected using this method, when combined with the clusters obtained from the trajectory data (MC1), also yield insights in the characteristics of the groups. For example, we find certain groups of individuals that communicate among each other throughout the day, but do not travel together.

Acknowledgements

This work was partially funded by the U.S. Department of Homeland Security's VACCINE Center under Award Number 2009-ST-061-CI0006.

References

[1] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[2] J. Chae, Y. Cui, Y. Jang, G. Wang, A. Malik, and D. S. Ebert. Trajectory-based Visual Analytics for Anomalous Human Movement Analysis using Social Media. In *EuroVis Workshop on Visual Analytics (EuroVA)*. The Eurographics Association, 2015.

[3] C. H. Elzinga. Sequence analysis: Metric representations of categorical time series. *Sociological methods and research*, 2006.

[4] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, volume 96, pages 226–231, 1996.

[5] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 28(1):pp. 100–108, 1979.