Erschienen in: 2019 IEEE Conference on Visual Analytics Science and Technology (VAST) : Proceedings / Chang, Remco; Keim, Daniel A.; Maciejewski, Ross (Hrsg.). - Piscataway, NJ : IEEE, 2019. - S. 140-141. - ISBN 978-1-72812-284-7

cpmViz: A web-based visualization tool for uncertain spatiotemporal data

Fabian Nagel[*] Giuliano Castiglia[†] Gemza Ademaj[‡] Juri Buchmüller[§] Udo Schlegel[¶]
Prof. Dr. Daniel A. Keim[‖]

Data Analysis and Visualization Group, University of Konstanz, Germany

Abstract

The goal of the VAST challenge 2019 Mini Challenge 2 was to visualize radioactive contaminations measured by mobile and static sensors and their changes over time, allowing city officials to determine the severity of the leakage at the city's nuclear power plant. We propose cpmViz, a web-based tool that allows for interactive data exploration of the sensor readings in both of the spatial and temporal dimensions. The tool consists out of three views that are connected via linking and scrolling. We visualize static sensor uncertainty by introducing Voronoi cells to illustrate how much space is covered by an individual measurement unit. For mobile sensors, we showcase their activity periods and introduce the concept of sensor streaks as periods of uninterrupted recordings as a temporal uncertainty measure. As for spatial uncertainty, we color individual districts based on the amount of data that was recorded inside the user's selected time window. Using our system, we were able to easily spot major events like the city's initial earthquake in the sensor readings. Certain southern districts are clearly visible as areas of concern that we consider in need of more static sensors. Furthermore, we were also able to identify static as well as moving contaminations.

Index Terms: Human-centered computing—Visualization—Visualization application domains—Visual Analytics;

1 Introduction

Mini Challenge 2 provided us with time-series data from static and mobile sensors. While the first have a known, fixed position, mobile sensors are attached to cars and thus, move around the city. Position data is provided as GPS locations in a (longitude, latitude) format. In both cases, there is a 5-second sampling interval, supported by an absolute timestamp field; however, mobile sensors suffer from reception losses and may not provide a recorded tuple every 5 seconds. In total, the measurement data reaches over the whole simulation length of 5 days. Individual measurement tuples contain a sensor ID that allow for easy identification inside the specific sensor group (static/mobile).

2 Data Preprocessing and Data Handling

The data was separated into three different datasets, of which two describe static sensors using their actual sensor readings as well as their fixed positions. With the mobile sensor readings as another dataset, our first step was to fuse the data together in order to make the initial exploratory analysis easier. For this, we introduced each static sensor's longitude and latitude as a fixed value into the two columns that were already given by the altering mobile sensor locations. Next, each sensor ID was made unique by prefixing the sensor type (*"static"* or *"mobile"*), so that sensors like *"mobile-1"* and *"static-1"* became clearly separable. During inspection, we identified two negative measurements which we removed as erroneous values as well as a single outlier that was off the second-highest value by a factor of 10. To gain more insight about non-continiuous measurement periods of mobile sensors, we computed a temporal delta to the respective previous measurement. For a perfectly-recording sensor, this value would always refer to the sampling rate of 5 (in seconds). While all static sensors fulfill this criterion, only mobile-35 and mobile-26 out of all mobile sensors record without any hiccups. To use this feature as an uncertainty measure and also to allow more in-depth analysis of individual recording periods, we introduced the concepts of sensor streaks. A streak is essentially a grouping of 5-second temporal deltas of the same sensor, representing a period of time in which the sensor gathered data every 5 seconds. This fact qualifies the concept of streaks an uncertainty measure, since a higher streaks count (per sensor, per neighborhood, ...) describes a higher partitioning of the data, resulting from reception losses. Since we do not know the effects of reception losses on mobile sensors, data from highly-fragmented regions at least need to be inspected with caution.

For every streak and all its belonging measurement tuples, we assigned a numeric ID that is unique for an individual sensor. All of the steps outlined above were implemented in a single KNIME workflow which allows for comprehensible and easily repeatable data transformations. At this point, the data was in a shape good enough for data exploration with standardized tools like R and Tableau in order to gain initial insights.

3 Detecting Contaminations

One of the main tasks of Mini Challenge 2 was to detect possible contaminations which might be fixed or dynamic in terms of their location. In our initial exploratory analysis, we observed a spike-like behavior in plots of measured contamination over time for individual sensors. Our team suspected that this behavior is due to the combined factors of highly-local radioactive effects, car-like velocities and the already described 5-second sampling rate. It was concluded that a contaminated location is in general not very likely to be measured at all, since a car might pass it at a velocity up to 50 km/h (inside the city). At this speed, a sampling interval of 5 seconds corresponds to a distance of 70 meters that was covered. While the effect range of radioactivity may vary, we suspected that the perfect spatial alignment between sensor and contamination right at recording time is rather unlikely.

With that in mind, we decided to treat the task as an outlier detection problem by applying DBSCAN clustering ($\varepsilon = 1$, MinPts = 3) in order to find datapoints classified as noise. The algorithm was integrated into the already existing KNIME workflow in which we also introduced an additional boolean field *"isOutlier"*. The outlier detection was performed separately on each sensor so that possible discrepancies in sensor responses and possible biases are considered.

From the resulting bimodal distribution of outliers, additional thresholding was performed in order to remove numerically invalid outliers. This includes measurements of 0, that often occurs as an initial value at the start of the simulation. In this context, we applied

[*] e-mail: fabian.2.nagel@uni-konstanz.de
[†] e-mail: giuliano-andrea.castiglia@uni-konstanz.de
[‡] e-mail: gemza.ademaj@uni-konstanz.de
[§] e-mail: juri.buchmueller@uni-konstanz.de
[¶] e-mail: schlegel@dbvis.inf.uni-konstanz.de
[‖] e-mail: keim@uni-konstanz.de

automatic thresholding algorithms like Otzu's method and Jenks natural breaks optimization. However, these approaches seemed to act too aggressive and resulted in losing a lot of outliers that we considered valuable. For this reason, we switched to a manual threshold at 70 cpm after careful visual inspection of the outlier distribution.

At this point, a clear trend of increasing outliers hours after the initial earthquake became visible in the data.

4 Application and Implementation

cpmViz uses PostgreSQL/PostGIS for data storage and a node backend to query the database. The data is then transformed into an appropriate data structure and sent to the frontend via a standard REST interface. In the browser, we use a combination of TypeScript and d3.js for the actual visualization.

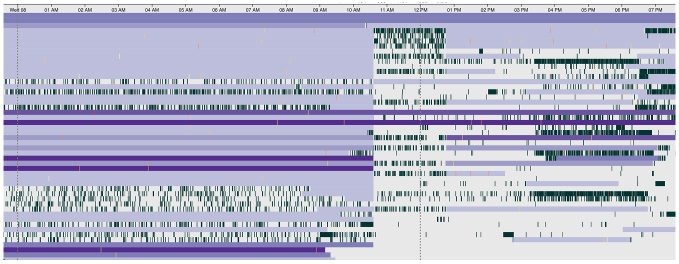
In order to tackle the entanglement of the spatiotemporal data, our team built custom UI components for the respective dimension. This results in three different views:

1) The timeline, which visualizes mobile sensor activity periods. Each sensor is assigned a fraction of the available vertical space, where the time axis is in horizontal orientation. The vertical placement of mobile sensors refers to the amount of data that was recorded throughout the whole simulation length; The sensors at the top refer to the biggest amount of data and are thus the most trustworthy. The width of the browser window always shows an entire day. Horizontal scrolling changes the temporal scope of the whole application and automatically updates other components.

A sensor streak is visible as a single rectangle, colored by the streak's standard deviation w.r.t. the sensor's baseline mean. Additionally, outliers (and thus, potential contaminations) are visible as color-coded vertical bars inside the streaks. Major events like the initial earthquake are clearly visible since they cause nearly all mobile sensors to become unstable in terms of their recording behavior (see 1).

It is worth noting that additional data transformations were performed within PostgreSQL in order to achieve the performance that is necessary for an interactive user experience. At application level, we only store streak metadata (start and end time, number of measurements as streak length, standard deviation, ...), resulting in a much more tolerable data volume. Only streak outliers are stored as-is, since these represent the dataset's most interesting entities. Using this performance-optimized datamodel, the frontend is able to only query for streaks long enough to actually be visible on a pixel level, which further reduces the amount of data that needs to be handled client-side.

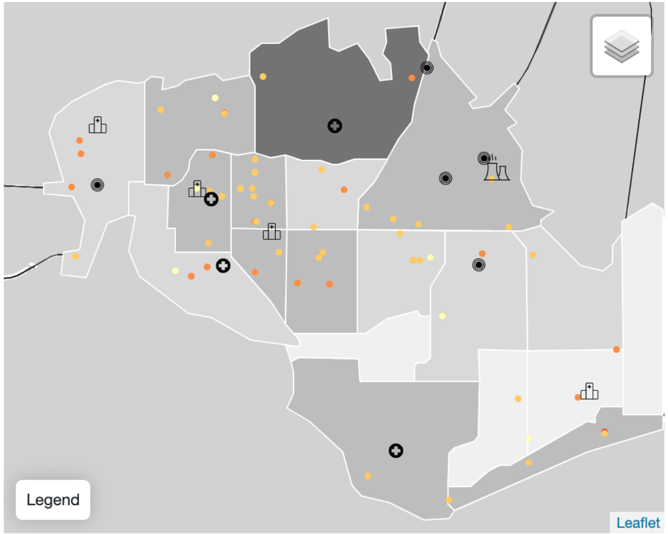
Figure 1: Timeline view



2) A map that illustrates the city of St. Himark with additional background information like points of interest. As a spatial uncertainty measure, we color districts based on the number of available measurements w.r.t. the currently visible timeframe. Futhermore, Voronoi cells of static sensors are visible as an additional map layer. These cells are supposed to show the ground that each static sensor needs to cover; bigger cells indicate an elevated measurement uncertainty.

To prevent overplotting, we only show interesting measurements (outliers) on the map. Color-coding is applied to show the amount of radiation that was measured. Hovering a possible contamination reveals the actual value as a tooltip and also highlights the corresponding vertical outlier bar in the timeline.

Figure 2: Map view



3) The detail view that visualizes the average streak length per district as a secondary spatial uncertainty measure. The higher the average streak length, the lower the uncertainty. Clicking on a district on the map also highlights the corresponding bar in the detail view.

5 Conclusion

cpmViz allows to easily explore Mini Challenge 2's complex dataset in a performant, interactive way. By being able to drill down in both the spatial and temporal dimension of the data separately, we were able to find insights relevant to the challenge's tasks. We identified static contaminations that are measured by multiple sensors, as well as a heavy contamination in the city's south east that appears to be a car originating from the Always Safe nuclear power plant. Using the uncertainty measures for spatial and temporal aspects of the data, we were able to identify multiple southern districts of St. Himark as relatively uncertain and in severe need of more static sensors. These measures allow the application user to judge possible findings with respect to their trustworthiness.