Visualizing Traffic Accident Hotspots Based on Spatial-Temporal Network Kernel Density Estimation (Demo Paper)

Benjamin Romano
The University of Alabama
Tuscaloosa, Alabama
bromano@crimson.ua.edu

Zhe Jiang
The University of Alabama
Tuscaloosa, Alabama
zjiang@cs.ua.edu

ABSTRACT

Understanding where traffic accidents occur is crucial for improving road safety and proper traffic enforcement allocation. One of the most common methods of analyzing traffic accidents is spatial hotspot detection. Existing hotspot detection methods, e.g., spatial scan statistics, spatial and spatiotemporal kernel density estimation, mostly focus on Euclidean space. These methods ignore an important aspect of traffic accident hotspots, i.e., traffic accident locations are constrained to road networks. Several techniques have been proposed to detect spatial hotspot on the network space, including network kernel density-estimation, and significant linear route detection, but the time dimension and temporal dynamics of hotspots are not incorporated. To address the limitations of existing methods, we demonstrated a new method called Spatial-Temporal Network Kernel Density Estimation (STNKDE) that integrates both of these features. We also developed a prototype system and visualized the dynamics of traffic accident hotspots in New York City 2017.

CCS CONCEPTS

• **Information systems** → **Geographic information systems**; **Data mining**;

KEYWORDS

Spatio-temporal hotspot, spatio-temporal network kernel density, traffic accident, law enforcement

ACM Reference format:
Benjamin Romano and Zhe Jiang. 2017. Visualizing Traffic Accident Hotspots Based on Spatial-Temporal Network Kernel Density Estimation (Demo Paper). In *Proceedings of SIGSPATIAL'17, Los Angeles Area, CA, USA, November 7–10, 2017,* 4 pages.
https://doi.org/10.1145/3139958.3139981

1 INTRODUCTION

With knowledge of where and when accidents are most likely to occur, traffic engineers and city officials can modify roads and signage to improve road safety. Additionally, traffic enforcement agents can use this information to better allocate resources to traffic accident hotspots. In Albuquerque, New Mexico, the majority of high traffic accident locations were also determined to be high-crime areas [14]. Higher traffic enforcement in Albuquerque led to reductions in crash injuries, DUIs and homicides. It has been shown that traffic accidents have spatial and temporal patterns [2]. If both the spatial and temporal aspects of the data are not considered together, traffic enforcement agents may over-allocate resources at spatial hotspots during non-peak hours. Better policing can not only have an impact on traffic accidents but also reductions in crime.

Kernel density estimation (KDE) has been mostly used as a visualization tool for hotspot visualization [15]. It has been used with success for analyzing potential fixes of traffic accident hotspots [4]. Additionally, in [4] repeatability analysis was compared against KDE for hotspot detection. Both methods produced similar results. For traffic accident analysis, it is recommended to apply Kernel Density Estimation (KDE) on a large scale to locate hotspots then reapply KDE to each individual hotspot found [11]. This is due to traffic accident distributions primarily being affected by factors at micro scale. Hotspots can also be detected by augmenting KDE for spatial-temporal analysis. Spatial-Temporal KDE has been used for crime hotspot analysis in [7]. The results of these approaches can be visualized using map animations, isosurfaces and comaps [1].

However, it has been shown that kernel density estimation method in the Euclidean space can lead to over-estimation of densities [10]. A network-based approach, Network KDE (NKDE), has been proposed to alleviate this issue [15]. Distance between two points is computed using network distance. Figure 1 demonstrates how Euclidean distance and network distance calculations can produce different results. Both methods use the same search bandwidth parameter in Figure 1; however, KDE finds two additional points.

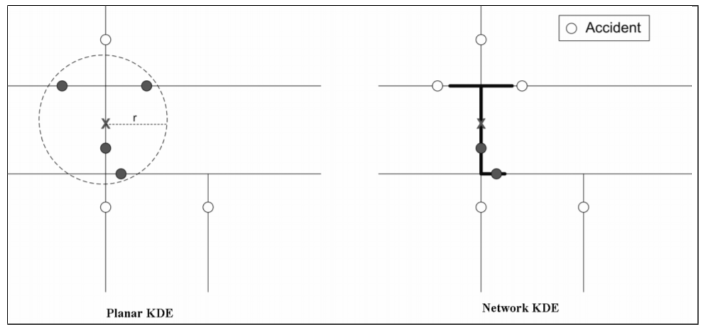

Figure 1: Planar KDE vs Network KDE (adapted from [15])

ACM ISBN 978-1-4503-5490-5/17/11.
https://doi.org/10.1145/3139958.3139981

An example of where this difference can lead to over-estimation is divided highways. Both sides of a divided highway are close to each other by Euclidean distance, but operate for the most part independently. It is expected that crashes on one side should not affect the density values on the other side. Using NKDE instead of KDE can ensure that this behavior is respected.

NKDE has recently been used to analyze spatial-temporal trends using the snapshot model [6]. NKDE was first applied for spatial hotspot detection. Then using the snapshot model, significant temporal changes at a hotspot were detected using the chi-square test. However, the density is estimated within each single snapshot without considering the time interval before and after the snapshot. In contrast, we propose a spatio-temporal network kernel density estimation method, whereby the density is based on both spatial and temporal neighborhoods in the network space.

2 OUR APPROACH

The software used in this paper includes PostgreSQL, PostGIS, pgrouting, and QGIS. PostgreSQL was the database used for storing the crash data, as well as the data describing the road network of NYC. PostGIS was used for spatial functions and creating a graph representation of the road network. Pgrouting was used for calculating the network distance between two points on the road network, that is, the shortest distance between two points while remaining on a segment of the network. QGIS is a free geographic information system application that was used for rendering the visualizations for the project. In QGIS, we used the QGIS2ThreeJS and TimeManager plugins to generate our visualizations.

2.1 Spatial-Temporal Network Kernel Density Estimation (STNKDE)

We created Spatial-Temporal Network Kernel Density Estimation (STNKDE), as an extension of Network Kernel Density Estimation (NKDE), that includes the temporal aspect of event data. In NKDE, edges are split into equal length segments called lixels (linear pixels). Events are bucketed into the nearest lixel. The density is computed for each lixel using the standard KDE equation shown in Equation 1. s_{dis} is the distance between two lixels, r_s is the spatial search bandwidth and k_s is a kernel function used for smoothing. Common kernel functions include Gaussian and Quartic [9]. According to Xie and Yan [15], the choice of kernel function does not have a big impact on the overall density pattern.

$$\lambda(s) = \frac{1}{r_s} \sum_{i=1}^{n} k_s(\frac{s_{dis}}{r_s}) \tag{1}$$

STNKDE extends the concept of lixel to include a temporal aspect. We call this extension an arixel (aerial lixel). Arixels are 2-D cells that are stacked on top of lixels like a wall. The time component of the arixels are expressed using the z-coordinate as in the space-time cube model [5]. The height of the arixels corresponds to the time range it covers. For example, an arixel could cover the time range 7am to 8am. In STNKDE, events are aggregated on arixels instead of lixels. Events can be temporally bucketed by different time types, as specified by the user, such as year, month, time of day, etc. The temporal bucketing used is dependent on the dataset and what the researcher wants to learn from the data. To compute the density at an arixel, we use Equation 2.

$$\lambda(s,t) = \frac{1}{r_s r_t} \sum_{i=1}^{n} k_s(\frac{s_{dis}}{r_s}) k_t(\frac{t_{dis}}{r_t}) \tag{2}$$

This adjusted formula includes several additional terms: a temporal search bandwidth (r_t), temporal distance (d_{is}), and an additional kernel function for time (k_t). These terms allow the formula to include both the spatial and temporal aspects of events.

2.2 A Prototype of STNKDE System

Many of the intermediate outputs of STNKDE can be re-used for different parameter configurations and many operations can be parallelized. To take advantage of these properties of STNKDE, we created a set of open-source python scripts called STNKDE Tools (source codes available at [12]). It can be used to compute either NKDE or STNKDE. Our toolset consists of six steps.

(1) Data Sanitization
(2) Load Data
(3) Create Lixels
(4) Compute Distances
(5) Compute Lixel Densities
(6) Compute Arixel Densities

To perform STNKDE, an event shapefile and a network shapefile are required. In the first step, Data Sanitization, both shapefiles must be converted to the same coordinate reference system and cropped to the relevant area of study. Next, the second step, Load Data, inserts the shapefiles into PostgreSQL. The road network is converted into a network topology using the PostGIS extension and the events are inserted into a table. Creating the network topology is the most computationally expensive operation in the STNKDE process. It is recommended to minimize the network size to speed up this operation.

Once the data is loaded, the network needs to be split into lixels. The Create Lixels operation operates on a copy of the network topology. Therefore, it is not necessary to recreate the network topology to experiment with different lixel lengths. Splitting edges into lixels is run in parallel allowing speedups for multicore machines.

In the next step, Compute Distances, the network distance between midpoints of lixels is pre-computed in parallel for use in both the NKDE and STNKDE algorithms. A spatial search bandwidth must be specified to reduce the number of comparisons. A larger spatial search bandwidth will smooth out the visualization and show global trends. A smaller spatial search bandwidth will highlight local trends and additionally is quicker to compute.

The Compute Lixel Densities step performs NKDE on the dataset for the specified lixel length and spatial search bandwidth. Computing lixel densities is also run in parallel. The Quartic function is used as the spatial kernel function. This step assumes the lixels and distances for the specified parameters have already been pre-computed. The results are stored into a PostgreSQL table. This table can be loaded into GIS software to visualize the densities. It is recommended to compute NKDE before STNKDE to get a feel for where hotspots over the entire time period lie.

The Compute Arixel Densities step performs STNKDE in parallel given a specified lixel length, spatial search bandwidth, temporal search bandwidth and time type. The Quartic function is used for

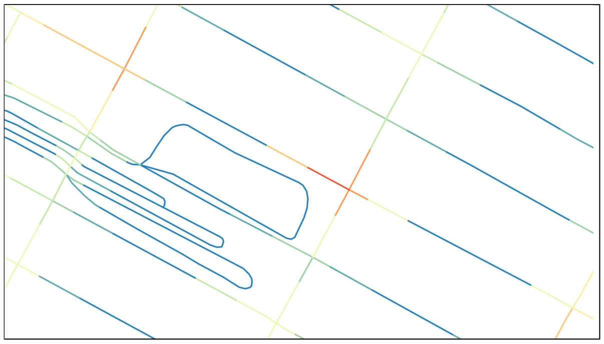

Figure 2: Results of NKDE

both the spatial and temporal kernel functions. This step again assumes the lixels and distances for the specified parameters have already been pre-computed. The time type specifies how to temporally bucket events. This can be days of week, hours of day, weeks, months, seasons and years. Most time types are considered cyclical. For example, the month December is one unit away from January. However in the case of years, in a dataset consisting of events from 2012 to 2017, the year 2017 is five units away from 2012.

3 CASE STUDY

The specific dataset used for the project comes from the public records of the New York City Department of Public Safety [8]. This dataset contains data pertaining to all traffic accidents in NYC between 2012 to 2017, and is updated regularly. The fields included in the dataset are attributes like latitude, longitude, date, time, street name, cross street name, and information about the vehicles involved. The fields that we care about are the latitude, longitude, date, and time, since these are the spatial and temporal attributes. This dataset contains approximately 800,000 records. We used a New York State road shapefile provided by the NYS GIS Clearinghouse [3]. Using QGIS, we cropped the shapefiles to only include events and roads within the Manhattan area. We also saved the cropped shapefiles with the same coordinate reference system so our data was consistent. This reduced our dataset to about 195,000 traffic accidents and about 900 roads from approximately 230,000 roads. This reduction drastically speeds up the computation for STNKDE and NKDE. To evaluate our STNKDE approach, we initially computed the NKDE and KDE over the dataset as a baseline. For NKDE we used a lixel length of 50 meters and a search bandwidth of 100 meters as recommended by Xie and Yan in [15]. These values provided a high level of detail without requiring a large amount of computation. We can see the output of the NKDE in Figure 2.

As expected, NKDE produces a clear visualization of hot spots. NKDE accurately displays which side of an intersection is the most prone to traffic accidents. Figure 3 shows a comparison of NKDE to the standard KDE approach. In Figure 3, we can see that KDE over-estimates hot spots and does not provide any useful insight onto which roads are most dangerous.

NKDE does a great job of displaying hot spots in the data overall; however, it does not capture the temporal aspect of the data. We

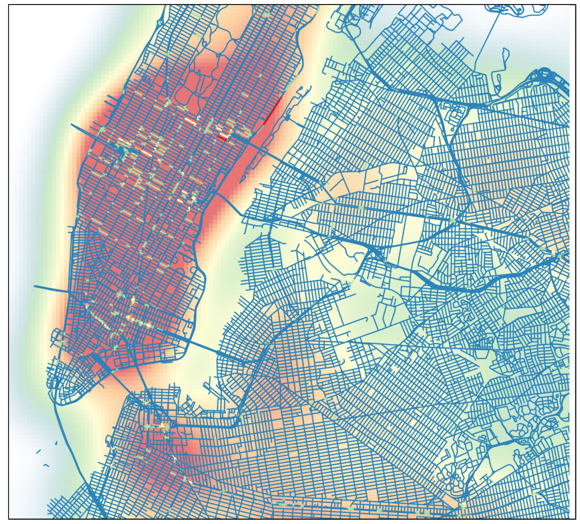

Figure 3: NKDE overlaid on top of KDE

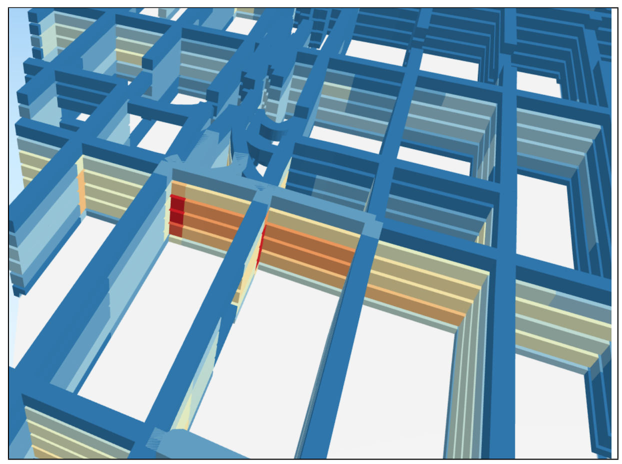

Figure 4: Results of STNKDE by year using space-time cube model

next ran STNKDE for lixel length of 50 meters, spatial search bandwidth 100 meters, temporal search bandwidth of 1 and time type of year. This visualization gave us valuable insight on when traffic accidents are most likely to occur in the Manhattan borough. A 3-D visualization of the results using the QGIS2ThreeJS QGIS plugin is shown in Figure 4. In this figure, we are looking at the beginning of the Queensboro Bridge from the Manhattan side. The top layer is the year 2017. At the time of publication, the NYC traffic accident dataset only extends up to April 2017, which results in the top layer having a lower density.

In the red area of Figure 4, we can see that traffic accidents are equally distributed over the years. This makes it abundantly clear that action needs to be taken for this section of road. Two roads to the right, we can see that the traffic accident density is decreasing year over year, which is ideal. This type of analysis was not possible with the standard NKDE approach.

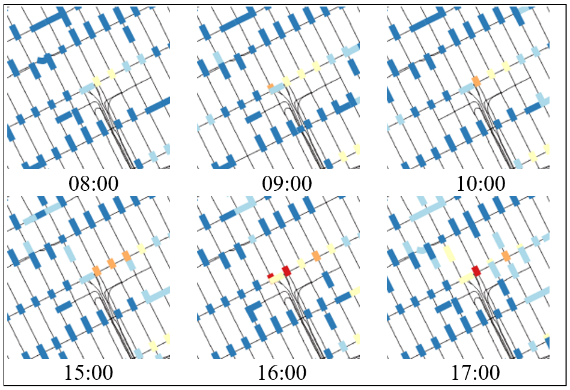

Figure 5: Results of STNKDE by hour using snapshot model

In tight grid-based road networks it can be difficult to view the arixel densities for a specific road segment in the space-time cube model. The QGIS2ThreeJS visualization solves this problem by letting the user arbitrarily reposition and rotate the view in real-time.

We next applied STNKDE with lixel length of 50 meters, spatial search bandwidth of 100 meters, temporal search bandwidth of 2 and time type of hour of day. This temporal search bandwidth was picked arbitrarily.

Figure 5 shows the results over the same stretch of road as in Figure 4 using the snapshot model. We generated this figure using the TimeManager plugin in QGIS. For time types with a large number of arixels per road segment, the snapshot model is preferable. For example, the time type weeks has 52 arixels per road segment resulting in very high walls. In Figure 5, we can notice an interesting temporal trend. During the evening rush hour the area around the Queensboro bridge is more dangerous than during the morning rush hour. This interesting spatial-temporal information was not found with the NKDE approach. One of the downsides of the snapshot model is that it is slightly more difficult to view the temporal change at hotspot compared to the space-time cube model. Combining the frames in Figure 5 into an animated GIF can help make it slightly easier to view the changes between frames.

4 DISCUSSION

One issue of the spatio-temporal network kernel density estimation method is the choice of distance threshold parameters. Similar to other existing kernel density estimation method, selecting appropriate distance threshold (kernel bandwidth) depends on the specific application scenario and the spatio-temporal scale of analysis. For example, the spatial distance threshold should be larger for analysis of crashes in the entire national highway networks, but smaller for analysis in a city. There are also other spatio-temporal hotspot detection methods [13] such as clustering based and spatiotemporal scan statistics based methods that we do not discussion in details due to less relevance to our method.

5 CONCLUSION

The resulting STNKDE approach provides a new way to analyze network-based spatial temporal data. Reseachers can now observe temporal trends in crash data that were previously hidden. With our open-source STNKDE tools, researchers should be able to easily apply our STNKDE algorithm or NKDE to their own datasets. We applied two visualization approaches, an interactive space-time cube model and a snapshot model to visualize STNKDE results. Both methods accurately display the spatial-temporal trends in slightly different ways. The approach to use is largely dependent on the structure of the data. In our use case, we find the snapshot model to be slightly better because the road network edges are close together. For our approach, future work needs to be done to determine optimal temporal search bandwidths for different time types. Additionally, work needs to be done to determine whether or not a hotspots found by STNKDE are statistically significant. Overall, our proposed STNKDE algorithm has produced very promising results.

REFERENCES

[1] Ali Asgary, Alireza Ghaffari, and Jason Levy. 2010. Spatial and temporal analyses of structural fire incidents and their causes: A case of Toronto, Canada. *Fire Safety Journal* 45, 1 (2010), 44 – 57. https://doi.org/10.1016/j.firesaf.2009.10.002
[2] William R Black. 1991. Highway accidents: a spatial and temporal analysis. *Transportation Research Record* 1318 (1991), 75–82.
[3] NYS GIS Clearinghouse. 2017. NYS GIS Clearinghouse - NYC GIS Program Office - NYS Streets. (2017). http://gis.ny.gov/gisdata/inventories/details.cfm?DSID=932
[4] Saffet Erdogan, Ibrahim Yilmaz, Tamer Baybura, and Mevlut Gullu. 2008. Geographical information systems aided traffic accident analysis system case study: city of Afyonkarahisar. *Accident Analysis & Prevention* 40, 1 (2008), 174–181.
[5] Torsten Hägerstraand. 1970. What about people in regional science? *Papers in regional science* 24, 1 (1970), 7–24.
[6] Omur Kaygisiz, Sebnem Duzgun, Ahmet Yildiz, and Metin Senbil. 2015. Spatio-temporal accident analysis for accident prevention in relation to behavioral factors in driving: The case of South Anatolian Motorway. *Transportation Research Part F: Traffic Psychology and Behaviour* 33 (2015), 128 – 140.
[7] Tomoki Nakaya and Keiji Yano. 2010. Visualising Crime Clusters in a Space-time Cube: An Exploratory Data-analysis Approach Using Space-time Kernel Density Estimation and Scan Statistics. *Transactions in GIS* 14, 3 (2010), 223–239. https://doi.org/10.1111/j.1467-9671.2010.01194.x
[8] NYPD. 2017. NYPD Motor Vehicle Collisions. (2017). https://data.cityofnewyork.us/Public-Safety/NYPD-Motor-Vehicle-Collisions/h9gi-nx95/data
[9] C.A. Gotway O. Schabenberger. 2005. *Statistical methods for spatial data analysis.* Chapman & Hall/CRC, Boca Raton, Florida.
[10] R. Pelot and L. Plummer. 2008. Spatial analysis of traffic and risks in the coastal zone. *Journal of Coastal Conservation* 11, 4 (2008), 201–207. https://doi.org/10.1007/s11852-008-0026-7 cited By 8.
[11] Charlotte Plug, Jianhong (Cecilia) Xia, and Craig Caulfield. 2011. Spatial and temporal visualisation techniques for crash analysis. *Accident Analysis & Prevention* 43, 6 (2011), 1937 – 1946.
[12] B. Romano. 2017. An open source tool for spatio-temporal network kernel density estimation. https://github.com/benjaminRomano/stnkde-tools/. (2017).
[13] Shashi Shekhar, Zhe Jiang, Reem Y Ali, Emre Eftelioglu, Xun Tang, Venkata Gunturi, and Xun Zhou. 2015. Spatiotemporal data mining: a computational perspective. *ISPRS International Journal of Geo-Information* 4, 4 (2015), 2306–2338.
[14] Jack Stuster. 2001. *Albuquerque Police Department's Safe Streets Program.* Technical Report.
[15] Zhixiao Xie and Jun Yan. 2008. Kernel density estimation of traffic accidents in a network space. *Computers, Environment and Urban Systems* 32, 5 (2008), 396–406.