Kernel Density Visualization for Big Geospatial Data: Algorithms and Applications

Tsz Nam Chan*, Leong Hou U†, Byron Choi*, Jianliang Xu*, Reynold Cheng‡§
*Department of Computer Science, Hong Kong Baptist University
{edisonchan, bchoi, xujl}@comp.hkbu.edu.hk
†Department of Computer and Information Science, University of Macau
ryanlhu@um.edu.mo
‡Department of Computer Science, The University of Hong Kong
ckcheng@cs.hku.hk
§Guangdong–Hong Kong-Macau Joint Laboratory

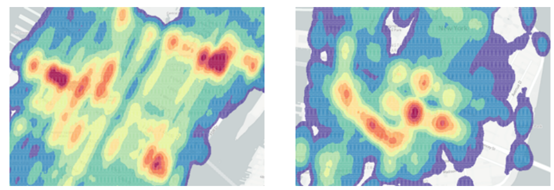
***Abstract*—The use of Kernel Density Visualization (KDV) has become widespread in a number of disciplines, including geography, crime science, transportation science, and ecology, for analyzing geospatial data. However, the growing scale of massive geospatial data has rendered many commonly used software tools unable of generating high-resolution KDVs, leading to concerns about the inefficiency of KDV. This 90-minute tutorial aims to raise awareness among database researchers about this important, emerging, database-related, and interdisciplinary topic. It is structured into four parts: a thorough discussion of the background of KDV, a review of state-of-the-art methods for generating KDVs, a discussion of key variants of KDV, including network kernel density visualization (NKDV) and spatiotemporal kernel density visualization (STKDV), and an outline of future directions for this topic.**

I. INTRODUCTION

Kernel Density Visualization (KDV) [20], [13] is widely used in various applications for analyzing geospatial data. Some representative examples include crime hotspot detection [12], [46], [32], [35], traffic accident hotspot detection [36], [53], [57], [55], [54], disease outbreak detection [31], [26], [25], and ecological modeling [56], [52]. Using Figure 1 (obtained from [20]) as an example, domain experts can adopt KDV to generate hotspot maps [12], [33] for a location dataset in different geographical regions, using various exploratory operations (e.g., zoom-in, zoom-out, and panning), so as to identify the hotspots.

Due to its wide range of applications, many geographical software tools, e.g., QGIS [7] and ArcGIS [1], scientific software tools, e.g., Scikit-learn [41] and Scipy [8], and visualization software tools, e.g., Deck.gl [3] and Seaborn [9], can support this operation. However, with the growing size of big geospatial data, such as the Chicago crime dataset [2] with 7.74 million location data points and the New York traffic accident dataset [6] with 1.97 million location data points, off-the-shelf software tools that use simple algorithms are infeasible for generating multiple KDVs for these large datasets. For this reason, developing efficient solutions for generating KDV is **an important, emerging, database-related, and interdisciplinary topic.** In this tutorial, we aim to bring attention to this important topic among database researchers and practitioners by highlighting the challenges, state-of-the-art methods, and opportunities for further research and development in KDV.

This work was supported by the NSFC grant 62202401, the Science and Technology Development Fund Macau SAR 0015/2019/AKP, 0031/2022/A, SKL-IOTSC-2021-2023, the Research Grant of University of Macau MYRG2022-00252-FST, Wuyi University Hong Kong and Macau joint Research Fund 2021WGALH14, HKRGC RIF R2002-20F, HKRGC C2004-21GF, GDNSF 2019B1515130001, the University of Hong Kong (Projects 104005858 and 10400599), the Guangdong–Hong Kong-Macau Joint Laboratory Program 2020 (Project No: 2020B1212030009), and the Hong Kong Jockey Club Charities Trust 260920140.



(a) Upper Manhattan (b) Lower Manhattan

Fig. 1: Using KDV to generate the hotspot maps for the New York traffic accident dataset [6] in two regions, where each pixel with red color denotes the hotspot location.

Target audience of this tutorial: We mainly target the MDM attendees who are interested in (1) conducting research for geospatial visual analytics, (2) adopting visualization tools for analyzing location data, or (3) incorporating the latest visualization technologies into software packages. The audience needs to understand some basic database concepts, e.g., indexing. However, this tutorial is self-contained, which does not require prior knowledge of geospatial visualization.

Comparisons with other related tutorials: Although many tutorials that are related to spatial/spatiotemporal databases and visual analytics have been conducted in the database community [34], [47], [50], [58], [27], [24], none of them has focused on using KDV to support visual analysis tasks. As a remark, the authors will provide another tutorial [21] in SIGMOD 2023. Compared with [21], this tutorial will deeply focus on visual analytics rather than a general introduction to geospatial analytics.

Related work from authors: We have extensively worked on improving the efficiency of solving KDV-related problems in recent years [21], [20], [15], [14], [18], [13], [19], [22] and have successfully built one system prototype [16] and two python libraries [17], [23] for supporting KDV and its variants. Moreover, we have jointly developed two widely used COVID-19 hotspot maps [4], [5], for Hong Kong and Macau citizens to visualize COVID-19 hotspots.

II. Tutorial Outline

This tutorial lasts for **1.5 hours**, which consists of four parts. First, we will review the background of KDV, including the motivation, the problem definition, the comparison with other traditional visualization methods (e.g., scatter plot [39], [38] and histogram [37], [51], [49]), and the software development for KDV **(30 mins)**. Then, we provide a comprehensive review for the state-of-the-art methods for generating KDVs **(20 mins)**. After that, we discuss other variants of KDV, including network kernel density visualization (NKDV) [18], [26], [55] and spatiotemporal kernel density visualization (STKDV) [15], [32], [25] **(20 mins)**. Lastly, we outline the open problems for future opportunities **(20 mins)**.

A. Background of KDV

In the first part of the tutorial, we will discuss how KDV is used to support different types of applications, including crime hotspot detection, traffic accident hotspot detection, disease outbreak detection, and ecological modeling, in detail. In addition, we will discuss the formal definition of the KDV problem (cf. Definition 1).

Definition 1: (KDV [20]) Given a location dataset $P = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n\}$ with n spatial data points and a geographical region with $X \times Y$ pixels, we need to color each pixel $\mathbf{q}$ based on the kernel density value $\mathcal{F}_P(\mathbf{q})$ (cf. Equation 1).

$$\mathcal{F}_P(\mathbf{q}) = \sum_{\mathbf{p} \in P} w \cdot K(\mathbf{q}, \mathbf{p}) \tag{1}$$

where w and $K(\mathbf{q}, \mathbf{p})$ denote the normalization constant and kernel function, respectively. Some representative kernel functions are shown in Table I.

TABLE I: Representative kernel functions.

Kernel	$K(\mathbf{q}, \mathbf{p})$	Used in
Epanechnikov	$\begin{cases} 1 - \frac{1}{b^2} dist(\mathbf{q}, \mathbf{p})^2 & \text{if } dist(\mathbf{q}, \mathbf{p}) \le b \\ 0 & \text{otherwise} \end{cases}$	[32], [11]
Quartic	$\begin{cases} (1 - \frac{1}{b^2} dist(\mathbf{q}, \mathbf{p})^2)^2 & \text{if } dist(\mathbf{q}, \mathbf{p}) \le b \\ 0 & \text{otherwise} \end{cases}$	[53], [35]
Gaussian	$\exp(-\frac{1}{b^2} dist(\mathbf{q}, \mathbf{p})^2)$	[36], [52]

Furthermore, since there are other types of visualization tools, e.g., scatter plot and histogram, we will provide some examples to show that KDV can achieve better visual quality compared with those visualization tools. In addition, we will survey different software tools for generating KDVs (e.g., ArcGIS, QGIS, Scipy, and Scikit-learn) and discuss their advantages and disadvantages.

B. State-of-the-art Methods for Generating KDVs

In the second part of the tutorial, we will review three types of methods for improving the efficiency of generating KDVs, including (1) function approximation methods, (2) data sampling methods, and (3) computational sharing methods. In addition, we will discuss the pros and cons of these methods.

Function approximation methods: In the first type of research studies, researchers [29], [28], [22], [13], [19] first develop the efficient lower and upper bound functions, $LB(\mathbf{q})$ and $UB(\mathbf{q})$, respectively, for the kernel density function $\mathcal{F}_P(\mathbf{q})$ (cf. Equation 1), i.e., $LB(\mathbf{q}) \le \mathcal{F}_P(\mathbf{q}) \le UB(\mathbf{q})$. Then, they incorporate these bound functions into an index structure (e.g., kd-tree [10] and ball-tree [40]) to progressively tighten $LB(\mathbf{q})$ and $UB(\mathbf{q})$ (by traversing the index structure) so that these bound values can achieve the approximation guarantee ε for computing the approximate kernel density value $R(\mathbf{q})$, where:

$$\frac{UB(\mathbf{q})}{LB(\mathbf{q})} \le 1 + \varepsilon \rightarrow (1 - \varepsilon)\mathcal{F}_P(\mathbf{q}) \le R(\mathbf{q}) \le (1 + \varepsilon)\mathcal{F}_P(\mathbf{q}) \tag{2}$$

Data sampling methods: In the second type of research studies, researchers [43], [44], [60], [59], [42] propose to obtain the subset S of the dataset P. Then, they can compute the modified kernel density function $\mathcal{F}_S^{(M)}(\mathbf{q})$ for this subset S, where:

$$\mathcal{F}_S^{(M)}(\mathbf{q}) = \sum_{\mathbf{p}_i \in S} w_i \cdot K(\mathbf{q}, \mathbf{p}_i) \tag{3}$$

They show that $\mathcal{F}_S^{(M)}(\mathbf{q})$ is theoretically close to the original kernel density value $\mathcal{F}_P(\mathbf{q})$ with the probabilistic guarantee. Since they can also provide the non-trivial upper bound for the subset size, computing $\mathcal{F}_S^{(M)}(\mathbf{q})$ can be significantly faster than $\mathcal{F}_P(\mathbf{q})$.

Computational sharing methods: In the third type of research studies, some researchers [20], [14], [30] exploit some sharing properties in order to improve the efficiency for computing a single KDV or multiple KDVs. Some of these research studies (e.g., [20], [14]) can further reduce the time complexity for generating KDVs with non-trivial accuracy guarantees.

C. Other Variants of KDV

In the third part of the tutorial, we will discuss two important variants of KDV, namely network kernel density visualization (NKDV) and spatiotemporal kernel density visualization (STKDV).

NKDV: Since some categories of geographical events, including traffic accidents and crime events, mainly occur in a road network, using the Euclidean distance $dist(\mathbf{q}, \mathbf{p})$ in the kernel function $K(\mathbf{q}, \mathbf{p})$ (cf. Table I) can overestimate the density value of each position (cf. Figure 2, modified from [18]). Therefore, geographical researchers [55], [54] propose to replace $dist(\mathbf{q}, \mathbf{p})$ in $K(\mathbf{q}, \mathbf{p})$ by the shortest path distance $dist_G(\mathbf{q}, \mathbf{p})$. In this tutorial, we will also review different methods for generating NKDV (e.g., [18], [45], [54]).

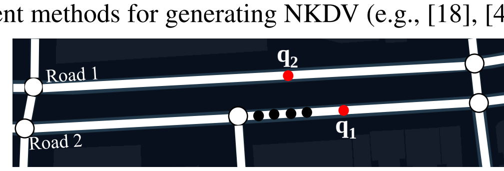
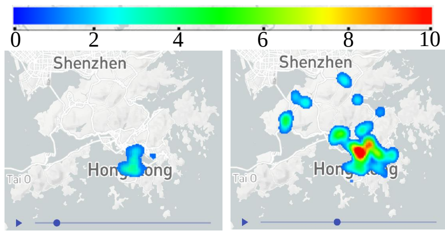


Fig. 2: Since $\mathbf{q}_2$ is far away from the black points in terms of the shortest path distance, we should assign a smaller density value for $\mathbf{q}_2$ compared with $\mathbf{q}_1$.

STKDV: In practice, some geographical phenomena, e.g., the distribution of COVID-19 cases, significantly depend on the event time. Using the COVID-19 cases in Hong Kong (cf.

Figure 3, obtained from [15]) as an example, observe that the third wave is more serious than the second wave in Hong Kong. As such, geographical researchers propose to adopt STKDV [36], [32], [25]. In this tutorial, we will discuss different methods [15], [48] for generating STKDV.



(a) Second wave (b) Third wave

Fig. 3: The seriousness and distribution of COVID-19 cases in Hong Kong, generated by STKDV, depend on the wave/time.

D. Future Opportunities

In the fourth part of the tutorial, we will discuss the future opportunities for both researchers and practitioners. In the following, we will highlight some of the promising directions.

Optimal solutions for solving KDV, NKDV, and STKDV: Although many advanced algorithms have been proposed to improve the efficiency for different variants of KDV (e.g., [20] for KDV, [18] for NKDV, and [15] for STKDV), these algorithms have not been proven to be optimal. We use KDV (cf. Definition 1) as an example. Recall that generating KDV needs to compute the kernel density function $\mathcal{F}_P(\mathbf{q})$ (cf. Equation 1) for each pixel $\mathbf{q}$. Therefore, every algorithm needs to at least access all (i.e., n) data points in P and all (i.e., $X \times Y$) pixels, which takes $\Omega(XY + n)$ time. However, the state-of-the-art algorithm [20] takes $O(Y(X+n))$ time, which still has a significant gap from the lower bound time complexity. As such, finding the optimal solutions for these problems is the promising direction.

Efficient algorithms for other kernel functions: In the state-of-the-art research studies [20], [18], [15], they mainly focus on the limited set of kernel functions (e.g., Epanechnikov and quartic kernels). However, these methods cannot be extended to support other important kernel functions (e.g., Gaussian kernel, cosine kernel, and exponential kernel) that can be supported by the famous software tools (e.g., Scikit-learn [41]). Therefore, finding an efficient solution for supporting other kernel functions is also the important direction.

Efficient algorithms for bandwidth tuning: In Table I, the bandwidth parameter b can significantly affect the visual quality of hotspot maps. Therefore, many geographical researchers (e.g., [57], [35]) need to generate multiple KDVs by varying this parameter, which can further deteriorate the inefficiency issue. However, there are only a few research studies that focus on this issue [14], [30]. In addition, these studies are restricted to handle KDV. As such, developing efficient algorithms for supporting both KDV, NKDV, and STKDV is another important direction.

Software development with efficient algorithms: Although many commonly-used software tools have been developed to generate KDV, most of these tools only adopt the basic algorithms, which are not feasible to support high-resolution KDV with large-scale datasets. Furthermore, only a few software tools can generate NKDV and STKDV. Based on these reasons, developing a new and an efficient software package to support different variants of KDV is also the promising direction.

III. Biographies

Tsz Nam Chan is a Research Assistant Professor in the Hong Kong Baptist University. He received his PhD degree and BEng degree from the Hong Kong Polytechnic University in 2019 and 2014, respectively. His research interests include large-scale data visualization and spatiotemporal databases.

Leong Hou U is an Associate Professor with University of Macau. He received the PhD degree from the University of Hong Kong in 2010. His research interests include large-scale query processing, scalable graph databases, graph learning, and reinforcement learning.

Byron Choi obtained the PhD and MSE degrees in Computer and Information Science from the University of Pennsylvania. He received a BEng degree in Computer Engineering from HKUST. He is currently the Associate Head and a Professor at the Department of Computer Science, Hong Kong Baptist University (HKBU). His research interests include graph-structured databases, database usability, database security, and time series analysis. He was awarded a distinguished program committee (PC) member from ACM SIGMOD 2021 and a best reviewer award from ACM CIKM 2021. He received the distinguished reviewer award from PVLDB 2019.

Jianliang Xu received the BEng degree in computer science and engineering from Zhejiang University in 1998 and the PhD degree in computer science from the Hong Kong University of Science and Technology in 2002. He is currently a Professor in the Department of Computer Science, Hong Kong Baptist University. His research interests include database, blockchain, and trusted computing. He has published 200+ papers in top-tier conferences and journals. He received the best paper awards of WISE2019 and MUST2021, and the best paper award runner-up of CIKM2020. He has served as the associate editor of TKDE and PVLDB, and the program committee member of SIGMOD, VLDB, and ICDE.

Reynold Cheng is a Professor of the Department of Computer Science in the University of Hong Kong (HKU). His research interests are in data science, big graph analytics, and uncertain data management. He received his BEng (Computer Engineering) in 1998, and MPhil (Computer Science and Information Systems) in 2000 from HKU. He then obtained his MSc and PhD degrees from the Department of Computer Science of Purdue University in 2003 and 2005, respectively. He received the SIGMOD Research Highlights Award 2020. He is a member of IEEE and ACM, was a PC co-chair of IEEE ICDE 2021, and has been serving on the program committees and review panels for leading database conferences and journals like SIGMOD, VLDB, ICDE, KDD, and TODS.

References

[1] ArcGIS. http://pro.arcgis.com/en/pro-app/tool-reference/spatial-analyst/how-kernel-density-works.htm.
[2] Chicago data portal. https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-Present/ijzp-q8t2.

[3] Deck.gl. https://deck.gl/docs/api-reference/aggregation-layers/heatmap-layer.

[4] Hong Kong COVID-19 hotspot map. https://covid19.comp.hkbu.edu.hk/.

[5] Macau COVID-19 hotspot map. http://degroup.cis.um.edu.mo/covid-19/.

[6] NYC open data. https://data.cityofnewyork.us/Public-Safety/Motor-Vehicle-Collisions-Crashes/h9gi-nx95.

[7] QGIS. https://docs.qgis.org/2.18/en/docs/user_manual/plugins/plugins_heatmap.html.

[8] Scipy. https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.gaussian_kde.html.

[9] Seaborn. https://seaborn.pydata.org/.

[10] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.

[11] M. Bíl, R. Andrášik, and Z. Janoška. Identification of hazardous road locations of traffic accidents by means of kernel density estimation and cluster significance evaluation. *Accident Analysis & Prevention*, 55:265–273, 2013.

[12] Y. A. Castle and J. M. Kovacs. Identifying seasonal spatial patterns of crime in a small northern city. *Crime Science*, 10(1):1–20, 2021.

[13] T. N. Chan, R. Cheng, and M. L. Yiu. QUAD: Quadratic-bound-based kernel density visualization. In *SIGMOD*, pages 35–50, 2020.

[14] T. N. Chan, P. L. Ip, L. H. U, B. Choi, and J. Xu. SAFE: A share-and-aggregate bandwidth exploration framework for kernel density visualization. *Proc. VLDB Endow.*, 15(3):513–526, 2021.

[15] T. N. Chan, P. L. Ip, L. H. U, B. Choi, and J. Xu. SWS: A complexity-optimized solution for spatial-temporal kernel density visualization. *Proc. VLDB Endow.*, 15(4):814–827, 2021.

[16] T. N. Chan, P. L. Ip, L. H. U, W. H. Tong, S. Mittal, Y. Li, and R. Cheng. KDV-Explorer: A near real-time kernel density visualization system for spatial analysis. *Proc. VLDB Endow.*, 14(12):2655–2658, 2021.

[17] T. N. Chan, P. L. Ip, K. Zhao, L. H. U, B. Choi, and J. Xu. LIBKDV: A versatile kernel density visualization library for geospatial analytics. *Proc. VLDB Endow.*, 15(12):3606–3609, 2022.

[18] T. N. Chan, Z. Li, L. H. U, J. Xu, and R. Cheng. Fast augmentation algorithms for network kernel density visualization. *Proc. VLDB Endow.*, 14(9):1503–1516, 2021.

[19] T. N. Chan, L. H. U, R. Cheng, M. L. Yiu, and S. Mittal. Efficient algorithms for kernel aggregation queries. *IEEE Trans. Knowl. Data Eng.*, 34(6):2726–2739, 2022.

[20] T. N. Chan, L. H. U, B. Choi, and J. Xu. SLAM: Efficient sweep line algorithms for kernel density visualization. In *SIGMOD*, pages 2120–2134, 2022.

[21] T. N. Chan, L. H. U, B. Choi, J. Xu, and R. Cheng. Large-scale geospatial analytics: Problems, challenges, and opportunities. In *SIGMOD*, 2023 (To appear).

[22] T. N. Chan, M. L. Yiu, and L. H. U. KARL: Fast kernel aggregation queries. In *ICDE*, pages 542–553, 2019.

[23] T. N. Chan, R. Zang, P. L. Ip, L. H. U, and J. Xu. PyNKDV: An efficient network kernel density visualization library for geospatial analytic systems. In *SIGMOD*, 2023 (To appear).

[24] G. Cong and C. S. Jensen. Querying geo-textual data: Spatial keyword queries and beyond. In *SIGMOD*, pages 2207–2212, 2016.

[25] E. Delmelle, C. Dony, I. Casas, M. Jia, and W. Tang. Visualizing the impact of space-time uncertainties on dengue fever patterns. *Int. J. Geogr. Inf. Sci.*, 28(5):1107–1127, 2014.

[26] M. Deng, X. Yang, Y. Shi, J. Gong, Y. Liu, and H. Liu. A density-based approach for detecting network-constrained clusters in spatial point events. *Int. J. Geogr. Inf. Sci.*, 33(3):466–488, 2019.

[27] A. Eldawy and M. F. Mokbel. The era of big spatial data. *Proc. VLDB Endow.*, 10(12):1992–1995, 2017.

[28] E. Gan and P. Bailis. Scalable kernel density classification via threshold-based pruning. In *SIGMOD*, pages 945–959, 2017.

[29] A. G. Gray and A. W. Moore. Nonparametric density estimation: Toward computational tractability. In *SDM*, pages 203–211, 2003.

[30] A. G. Gray and A. W. Moore. Rapid evaluation of multiple density models. In *AISTATS*, 2003.

[31] X. Han, J. Wang, M. Zhang, and X. Wang. Using social media to mine and analyze public opinion related to covid-19 in china. *International Journal of Environmental Research and Public Health*, 17(8), 2020.

[32] Y. Hu, F. Wang, C. Guin, and H. Zhu. A spatio-temporal kernel density estimation framework for predictive crime hotspot mapping and evaluation. *Applied Geography*, 99:89 – 97, 2018.

[33] S. Kim, S. Jeong, I. Woo, Y. Jang, R. Maciejewski, and D. S. Ebert. Data flow analysis and visualization for spatiotemporal statistical data without trajectory information. *IEEE Trans. Vis. Comput. Graph.*, 24(3):1287–1300, 2018.

[34] M. Krommyda and V. Kantere. Visualization systems for linked datasets. In *ICDE*, pages 1790–1793, 2020.

[35] P.-F. Kuo, D. Lord, and T. D. Walden. Using geographical information systems to organize police patrol routes effectively by grouping hotspots of crash and crime data. *Journal of Transport Geography*, 30:138–148, 2013.

[36] Y. Li, M. Abdel-Aty, J. Yuan, Z. Cheng, and J. Lu. Analyzing traffic violation behavior at urban intersections: A spatio-temporal kernel density estimation approach using automated enforcement system data. *Accident Analysis & Prevention*, 141:105509, 2020.

[37] Q. Liu, Y. Shen, and L. Chen. Lhist: Towards learning multi-dimensional histogram for massive spatial data. In *ICDE*, pages 1188–1199, 2021.

[38] A. Mayorga and M. Gleicher. Splatterplots: Overcoming overdraw in scatter plots. *IEEE Trans. Vis. Comput. Graph.*, 19(9):1526–1538, 2013.

[39] L. Micallef, G. Palmas, A. Oulasvirta, and T. Weinkauf. Towards perceptual optimization of the visual design of scatterplots. *IEEE Trans. Vis. Comput. Graph.*, 23(6):1588–1599, 2017.

[40] A. W. Moore. The anchors hierarchy: Using the triangle inequality to survive high dimensional data. In *UAI*, pages 397–405, 2000.

[41] F. Pedregosa, G. Varoquaux, A. Gramfort, and et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[42] J. M. Phillips. ϵ-samples for kernels. In *SODA*, pages 1622–1632, 2013.

[43] J. M. Phillips and W. M. Tai. Improved coresets for kernel density estimates. In *SODA*, pages 2718–2727, 2018.

[44] J. M. Phillips and W. M. Tai. Near-optimal coresets of kernel density estimates. In *SOCG*, pages 66:1–66:13, 2018.

[45] S. Rakshit, A. Baddeley, and G. Nair. Efficient code for second order analysis of events on a linear network. *Journal of Statistical Software, Articles*, 90(1):1–37, 2019.

[46] A. Ristea, M. A. Boni, B. Resch, M. S. Gerber, and M. Leitner. Spatial crime distribution and prediction for sporting events using social media. *Int. J. Geogr. Inf. Sci.*, 34(9):1708–1739, 2020.

[47] I. Sabek and M. F. Mokbel. Machine learning meets big spatial data. *Proc. VLDB Endow.*, 12(12):1982–1985, 2019.

[48] E. Saule, D. Panchananam, A. Hohl, W. Tang, and E. Delmelle. Parallel space-time kernel density estimation. In *ICPP*, pages 483–492, 2017.

[49] D. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, 1992.

[50] N. Tang, E. Wu, and G. Li. Towards democratizing relational data visualization. In *SIGMOD*, pages 2025–2030, 2019.

[51] C. Wang, H. Yu, and K. Ma. Importance-driven time-varying data visualization. *IEEE Trans. Vis. Comput. Graph.*, 14(6):1547–1554, 2008.

[52] Z. Wang, C. Ginzler, and L. T. Waser. Assessing structural changes at the forest edge using kernel density estimation. *Forest Ecology and Management*, 456:117639, 2020.

[53] K. Xie, K. Ozbay, A. Kurkcu, and H. Yang. Analysis of traffic crashes involving pedestrians using big data: Investigation of contributing factors and identification of hotspots. *Risk Analysis*, 37(8):1459–1476, 2017.

[54] Z. Xie and J. Yan. Kernel density estimation of traffic accidents in a network space. *Computers, Environment and Urban Systems*, 32(5):396 – 406, 2008.

[55] Z. Xie and J. Yan. Detecting traffic accident clusters with network kernel density estimation and local spatial statistics: an integrated approach. *Journal of Transport Geography*, 31:64 – 71, 2013.

[56] L. Xu, S. Zhao, S. S. Chen, C. Yu, and B. Lei. Analysis of arable land distribution around human settlements in the riparian area of Lake Tanganyika in Africa. *Applied Geography*, 125:102344, 2020.

[57] H. Yu, P. Liu, J. Chen, and H. Wang. Comparative analysis of the spatial analysis methods for hotspot identification. *Accident Analysis & Prevention*, 66:80 – 88, 2014.

[58] J. Yu and M. Sarwat. Geospatial data management in Apache Spark: A tutorial. In *ICDE*, pages 2060–2063, 2019.

[59] Y. Zheng, J. Jestes, J. M. Phillips, and F. Li. Quality and efficiency for kernel density estimates in large data. In *SIGMOD*, pages 433–444, 2013.

[60] Y. Zheng and J. M. Phillips. $L\infty$ error and bandwidth selection for kernel density estimates of large data. In *SIGKDD*, pages 1533–1542, 2015.